



MAKINON
bikes

2023-2024

Proyecto FCT 2-DAW

AUTORES:

JOSE ANTONIO HOLGADO BONET
ALBERTO MORENO FERNÁNDEZ

Índice

0 – Introducción	4
0.1 Objetivo.....	4
0.2 Requisitos de la aplicación.....	5
Requisitos funcionales.....	5
Requisitos no funcionales.....	5
0.3 Diseño de la aplicación.....	6
Herramientas del backend	6
Herramientas del frontend	6
Diseño de la base de datos	7
1 – Diseño de Interfaces	8
1.1 Color dominante.....	8
1.2 Paleta de colores	9
1.3 Fuente del prototipo.....	9
1.4 Combinación armónica de fuentes.....	10
2 – Landing	15
2.1 Animaciones y transiciones con el DOM.....	15
2.2 Responsive.....	16
2.3 Uso de SASS y BEM	16
2.4 Elementos multimedia.....	17
2.5 Imagen Vectorial.....	20
2.6 Despliegue.....	20
2.6 Despliegue mediante Codedeploy	23
3 – Angular	27
3.1 Introducción	27
3.2 Estructura de la Aplicación.....	27
3.3 Rol Usuario	27
3.3.1 Flujo de Acceso del Usuario.....	27
3.3.2 Proceso Completo de Solicitud de Cita.....	29
3.3.3 Componente del Calendario.....	30
3.3.4 Componente de Citas Anteriores.....	31
3.3.5 Componente de Dirección del taller	33

3.4 Rol de Administrador	34
3.4.1 Landing Page	34
3.4.2 Componente del Calendario.....	34
3.4.3 Componente de Citas Pendientes	35
3.4.4 Componente de Gráfica de Citas.....	36
3.5 Estructuración y Tecnología del Proyecto.....	38
4 – Laravel	41
4.1 Introducción	41
4.2 Estructura de la aplicación.....	41
4.3 Rol del Usuario	42
4.3.1 Flujo de acceso del usuario	42
4.3.2 Registro	43
4.3.3 Login.....	43
4.3.4 Integración y seguridad.....	44
4.3.5 Proceso de compra	44
4.3.6 Panel del usuario	47
4.4 Rol del Administrador	50
4.4.1 Flujo de acceso del administrador	50
4.4.2 Integración y seguridad.....	50
4.4.3 Panel de administración.....	52
4.4.3.1 Gestión de proveedores.....	53
4.4.3.2 Gestión de Productos	54
4.4.3.3 Gestión de Usuarios.....	56
4.4.3.4 Gestión de Ventas.....	56
4.4.3.5 Gestión de citas del taller.....	57
4.5 Implementaciones	58
4.5.1 Livewire	58
4.5.2 Mail.....	60
4.5.3 Biblioteca DomPDF.....	61
4.6 Despliegue.....	62
4.7 Documentación API Rest.....	65
5 – Bibliografía.....	67

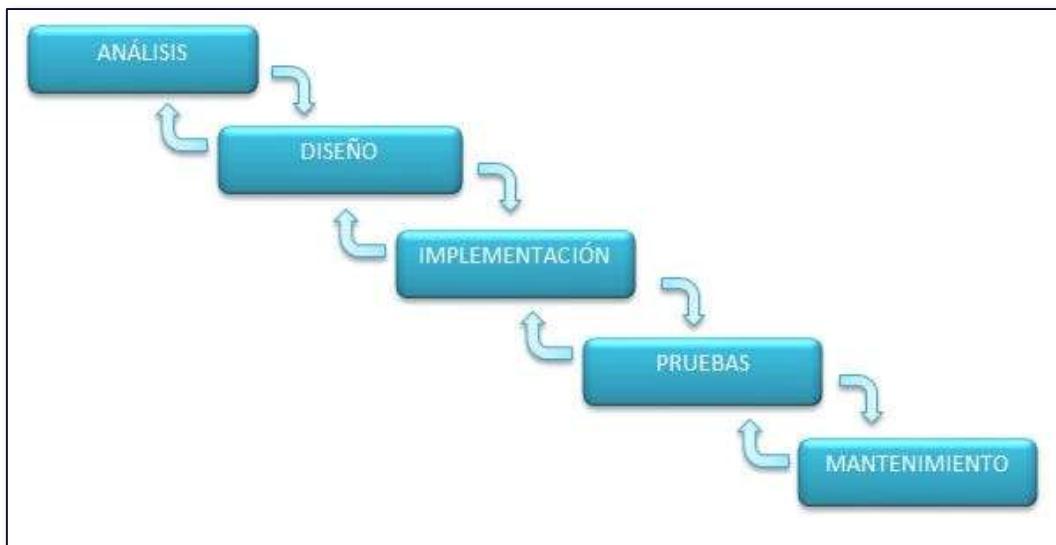
0 – Introducción

0.1 Objetivo

El objetivo principal del proyecto, además de tratar de implementar la gran mayoría de conocimientos adquiridos durante estos dos años, y en especial durante el último año de curso, es el desarrollo de la virtualización de lo que en su momento fue Makinon Bikes, una tienda dedicada a la venta de bicicletas, componentes y todo tipo de artículos relacionados con el mundo del ciclismo, tratando de hacer una web shop moderna, intuitiva, con un diseño poco “recargado” y donde el usuario puede navegar libremente de una sección a otra sin perderse por el camino.

Hemos creado al mismo tiempo un sistema de citas para el taller de reparaciones, de modo que sea el propio usuario el que pueda elegir el día y la hora que mas le convenga para llevar su bicicleta a reparar.

Desde el comienzo se han seguido todas las etapas en el proceso de desarrollo de software: planificación, análisis, diseño, implementación, pruebas, instalación y mantenimiento, usando un modelo de ciclo de vida en cascada. Es un modelo clásico, ideal para proyectos estables y pequeños, donde los requisitos quedan bien documentados al inicio del ciclo.



Cada fase se completa en orden y tras la verificación, se puede volver a la fase anterior en caso necesario. No se realizará la entrega del producto final hasta la validación de la última etapa. Esta secuencia lineal puede tener varios inconvenientes, como los riesgos de modificar los requisitos en una fase más avanzada, por ello en proyectos reales, se opta por seguir otros modelos como el iterativo, incremental, en espiral, etc., pero como habíamos dicho, al ser un proyecto pequeño optamos por el modelo en cascada.

0.2 Requisitos de la aplicación

Requisitos funcionales

- Requisitos de la aplicación
 - La aplicación permite ver el contenido de la web aunque el usuario no este registrado/logeado.
 - La aplicación permitirá completar un proceso de compra o solicitar una cita en el taller sólo a los usuarios que estén registrados y logeados en ese momento.
 - La aplicación cuenta con un módulo para la gestión de los usuarios
 - La aplicación cuenta con un módulo para la gestión de todo lo que es la tienda (productos, proveedores, ventas, citas del taller, etc.)
 - La aplicación tiene un servicio de contacto, bien para recibir consultas o sugerencias, así como para comunicarse automáticamente con el usuario después de su registro, un pedido realizado, una cita de taller concertada o modificada.
 - La aplicación debe permitir ver la cantidad de producto que queda en el almacén y poner un límite en la compra dependiendo de ese número de unidades.
- Requisitos del administrador
 - El administrador tiene acceso a secciones privadas que le permiten el control de la tienda
- Requisitos del usuario
 - Los usuarios pueden o no registrarse para navegar libremente por Makinon Bikes, pero en el momento de realizar la compra o solicitar una cita de taller el usuario debe de estar logeado, por lo que previamente ha debido de registrarse.
 - El usuario tendrá acceso a un panel propio desde donde gestionar su cuenta.
 - En el apartado de citas de taller, el usuario podrá adjuntar, junto con su solicitud de cita, una fotografía de su bicicleta.

Requisitos no funcionales

- Interfaz y usabilidad.
 - La interfaz gráfica será sencilla, moderna, atractiva y fácil de manejar incluso para personas que no tengan demasiados conocimientos sobre internet.
 - La aplicación tendrá al menos dos idiomas en sus contenidos principales, el español y el inglés.
- Recursos
 - La información de todos los usuarios, productos, marcas, proveedores, etc. se almacenará en una base de datos usando el servicio RDS de AWS.
 - La tienda Makinon Bikes se alojará en una instancia EC2 de AWS, mientras que la aplicación de gestión de citas se alojará utilizando el servicio Netlify.

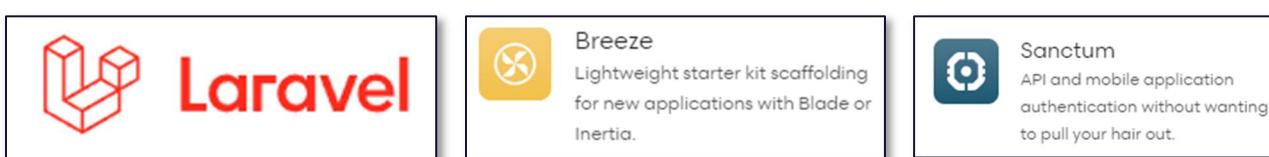
0.3 Diseño de la aplicación

Herramientas del backend

En el lado del backend se ha optado por usar PHP mediante el uso del framework Laravel, el cual crea un entorno de trabajo y nos proporciona las herramientas para ayudarnos en la aplicación.

Con Laravel lo que buscamos es construir aplicaciones sólidas y estables, “fáciles de desarrollar” y con el aprovechamiento de una parte del código que se encuentra preprogramada.

En concreto hemos utilizado los denominados ecosistemas Breeze y Sanctum, este último nos permitirá la autenticación de cara a la aplicación desarrollada en Angular, mientras que el primero ha sido la base de todo el proyecto de la tienda.



Herramientas del frontend.

Para el lado del frontend de la Tienda hemos utilizado Laravel Blade, es decir el motor de plantillas por defecto que tiene Laravel, esto nos permite utilizar variables, bucles, sentencias condicionales y otras características de PHP directamente sobre nuestro código HTML el cual debe su diseño al uso de CSS, Bootstrap y en los casos que lo hemos necesitado, Tailwind y JavaScript



Al mismo tiempo hemos empleado, en algunas de las secciones de nuestro proyecto Livewire, el cual es una biblioteca de Laravel que nos permite crear aplicaciones web reactivas sin la necesidad de emplear Javascript o Ajax.



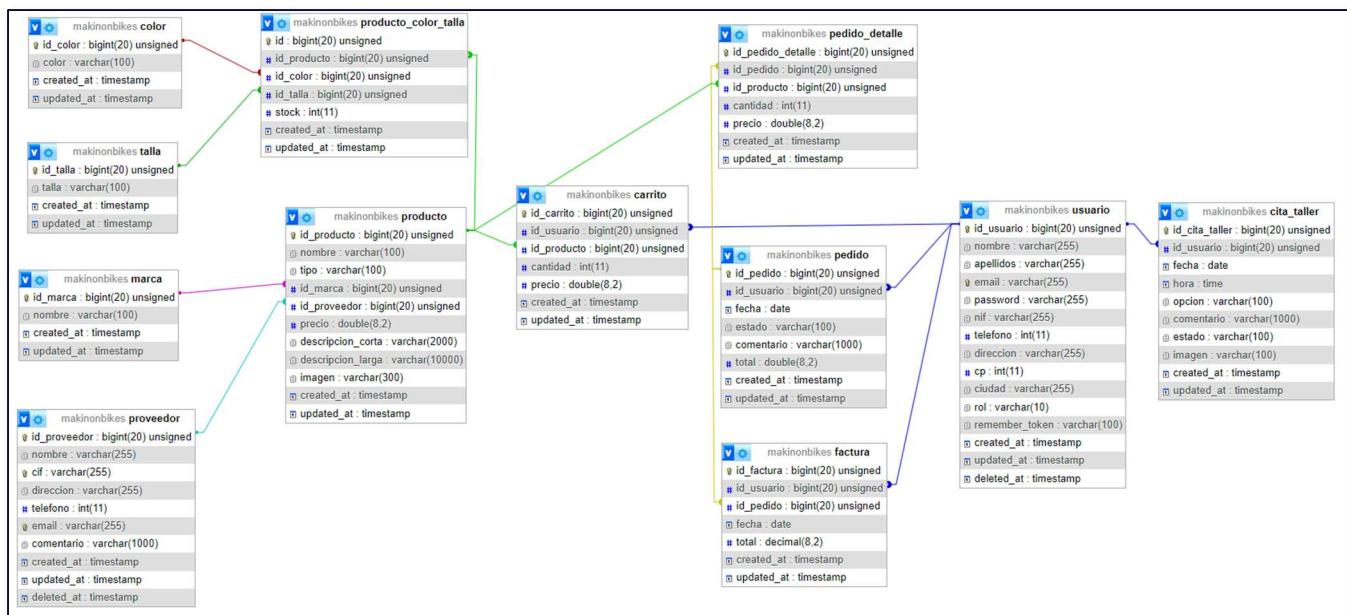
Para el lado del frontend del Gestor de citas de taller hemos utilizado Angular, este es un software de código abierto mantenido por Google, que sirve para desarrollar aplicaciones web de estilo Single Page Application (SPA) y Progressive Web App (PWA). Sirve tanto para versiones móviles como de escritorio.



Diseño de la base de datos

Makinon Bikes usa una base de datos relacional para almacenar la información de todos los usuarios, productos, marcas, colores, talla y stock, pedidos y sus detalles, así como las citas del taller y los tokens que se generan al hacer login y que nos permiten la comunicación con la aplicación de Angular.

El esquema de nuestra base de datos finalmente quedará de la siguiente forma:



1 – Diseño de Interfaces

1.1 Color dominante

Para ello nos hemos apoyado en los colores del logo de la misma, con lo cual tenemos una paleta de colores en la que trabajamos principalmente con el color negro y varios tonos de grises que contrastan perfectamente con el color blanco y dejando nota de color al naranja del logo, que también está presente en algún componente de la web, algún elemento en el proceso de compra y a las fotografías.



Algunos ejemplos de esas notas de color a las que hago referencia serían:

The screenshot displays a product listing for two KTM Bikes. On the left, there's a sidebar with three dropdown menus. The main area shows two bicycles: one black and blue KTM Scarp Prime and one red KTM Prowler. Below each bike is its recommended price: 6.725,00 EUR and 12.499,00 EUR respectively. To the right, there's a summary of the order total (10.049,90 €), including IVA (2.110,48 €). Below this, there are two green checkmarks: 'Artículo en stock' and 'Plazo de entrega en Península de 2 días'. There are three payment buttons: a blue 'PayPal' button with the text 'La forma rápida y segura de pagar', an orange 'amazon' button with the text 'Utiliza tu cuenta Amazon', and an orange 'PASAR POR CAJA' button.

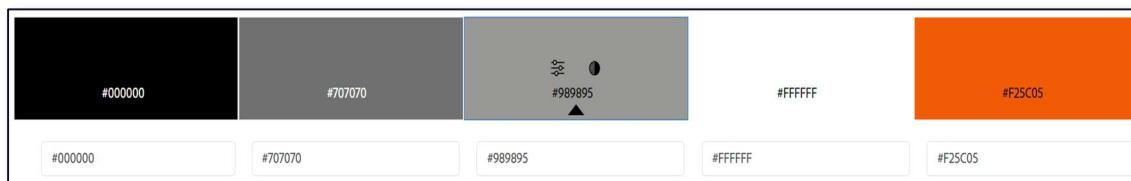
1.2 Paleta de colores

Si recurrimos a una herramienta online como tiene la web Colors y sometemos a análisis el logo de Makinon Bikes obtendríamos este resultado:



Viendo el logo de la tienda teníamos muy claro que había tres colores que debían estar, el blanco, el negro y el color naranja. De este modo decidimos crear una paleta de colores propia en la que haría principalmente uso de los colores negro y blanco, apoyándonos en distintos tonos de grises para algunos de los textos como pueden ser las descripciones de los artículos y por supuesto el color naranja que debía de estar presente.

De este modo la paleta de colores quedaría de la siguiente manera:



1.3 Fuente del prototipo

Para la realización de este proyecto se ha optado principalmente por el uso de una fuente de tipo OpenType como es “PORT CONDENSED”, que es una fuente tipo monotype y sans serif, es decir una fuente con caracteres sin serifas y apariencia monoespaciada, es decir que cada carácter ocupa la misma cantidad de ancho.

Su elección principal es porque tiene un aspecto moderno y minimalista, proporcionando un diseño limpio y contemporáneo, además otros beneficios de este tipo de fuentes son una apariencia ordenada y una simetría en la estructura.

Además, esta fuente, o este tipo de fuentes, facilita la legibilidad en dispositivos con una pantalla pequeña, como los teléfonos móviles, lo cual nos vendrá bien a la hora de hacer nuestra web responsive o adaptada a pequeños viewports

BH LYNX RACE LT 9.9

Rápido siempre. En distancias cortas. En distancias largas. En pistas. En trialeras. Subiendo. Bajando. Es lo que buscas en una doble de XC y bike-maraton. El cuadro Nº1 para las World Cup con David Valero y BH Coloma Team.

La otra fuente que he usado, y que se denomina “Coach”, es un guiño a la marca KTM, de la cual era distribuidor, es otra fuente OpenType que solamente uso para el menú de navegación hacia las categorías.

BICICLETAS COMPONENTES

1.4 Combinación armónica de fuentes

Para los aplicar los pesos de las diferentes jerarquías visuales he recurrido al uso de letras en mayúsculas para dar mas importancia a los títulos, así como cambiar el tamaño de la fuente, e incluso jugar con el uso de un color mas claro, para en algunos casos diferenciar el contenido.

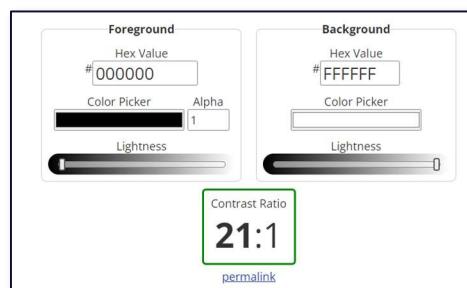
En el siguiente ejemplo la fuente más grande se utiliza para el nombre del sitio web y la fuente más pequeña se utiliza para el menú de navegación. El tamaño de fuente para la descripción del producto y el precio es ligeramente mayor que el menú de navegación, pero más pequeño que el nombre del sitio web, lo cual ayuda a llamar la atención sobre la información importante en la página



Contraste para la fuente y background (WCAG 2)

Si hacemos una prueba de accesibilidad para los colores de los textos, sobre todo donde más conflictos podríamos tener a la hora de una lectura correcta, gris oscuro sobre fondo negro y negro sobre un fondo blanco, los resultados son favorables en ambos casos.

Nuestras bicicletas son más que simples medios de transporte; son la encarnación de la excelencia en ingeniería y diseño. Cada componente está meticulosamente seleccionado y ensamblado para ofrecer un rendimiento inigualable en cada pedaleo. Desde nuestras robustas estructuras hasta nuestros suaves sistemas de cambio, cada detalle ha sido cuidadosamente diseñado para brindarte una experiencia de ciclismo incomparable.



Haciendo uso de la web <https://webaim.org/>, y su herramienta en línea, al igual que en el caso de la página Colors, he analizado el contraste de la fuente principal, a diferentes tamaños, con respecto al background para las dos parejas de colores que mayor conflicto podrían causar para comprobar si la lectura por parte del usuario pudiese verse reducida.

The image contains two side-by-side screenshots of the WebAIM Contrast Ratio tool, demonstrating the impact of different background colors on text readability.

Left Screenshot (Original Colors):

- Foreground Color:** Hex #989895, Lightness slider at ~10%.
- Background Color:** Hex #000000, Lightness slider at ~0%.
- Contrast Ratio:** 7.26:1 (highlighted in green).
- WCAG AA:** Pass (green button)
- WCAG AAA:** Pass (green button)
- Text Examples:** "The five boxing wizards jump quickly." (Normal Text), "The five boxing wizards jump quickly." (Large Text), and a "Text Input" field.

Right Screenshot (Dark Background Variation):

- Foreground Color:** Hex #707070, Lightness slider at ~10%.
- Background Color:** Hex #FFFFFF, Lightness slider at ~100%.
- Contrast Ratio:** 4.95:1 (highlighted in green).
- WCAG AA:** Pass (green button)
- WCAG AAA:** Fail (red button)
- Text Examples:** "The five boxing wizards jump quickly." (Normal Text), "The five boxing wizards jump quickly." (Large Text), and a "Text Input" field.

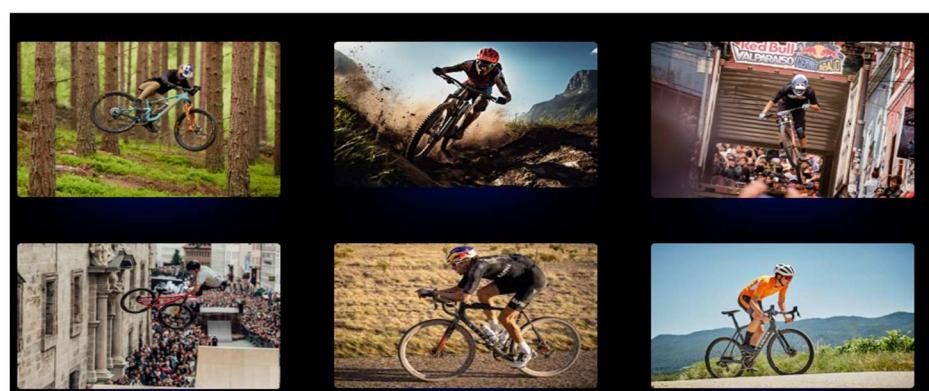
Jerarquía visual y tensión compositiva

En la siguiente captura podemos encontrar seis fotografías de ciclistas en diferentes entornos y situaciones.

Las imágenes en el centro (tanto horizontal como verticalmente) suelen atraer primero la atención del espectador. En este caso, la fotografía del ciclista en el barro (arriba al centro) y la del ciclista en una competición (centro a la derecha) son prominentes debido a su posición central.

Las fotografías con colores más vibrantes y contrastantes, como la imagen del ciclista en el barro (arriba al centro) y la del ciclista en un entorno urbano con público (arriba a la derecha), tienden a captar más la atención debido a la intensidad y variedad de colores.

Las imágenes situadas en las esquinas pueden ser vistas después de las imágenes centrales. En este caso, las fotografías en la esquina inferior izquierda (ciclista saltando en un entorno urbano) y en la esquina inferior derecha (ciclista en carretera) pueden atraer la mirada después de las imágenes centrales.



La combinación de color, posición, y el contenido dinámico de las imágenes ayuda a guiar la mirada del espectador a través de la composición de manera efectiva, asegurando que se note cada elemento importante en el orden deseado.

En cuanto a la tensión compositiva se ha utilizado un descuadre de los cards.

Las imágenes están inclinadas en ángulos inusuales, creando una sensación de dinamismo e inestabilidad. Las líneas diagonales aportan una energía y movimiento que no se lograría con imágenes alineadas horizontalmente. Hay un claro contraste entre las áreas oscuras de las bicicletas y el fondo negro en las tres primeras imágenes, y la imagen más colorida del ciclista en acción al final. Este contraste capta la atención y refuerza la diferenciación entre los elementos.

La alineación horizontal de las imágenes crea un flujo visual, pero las diferentes inclinaciones de cada imagen introducen una tensión que hace que el ojo del espectador se mueva a lo largo de la composición de manera irregular y dinámica.

Cada imagen tiene un punto focal claro: la bicicleta entera, detalles de la bicicleta, y el ciclista en acción. Estos puntos focales destacados en ángulos inusuales añaden una sensación de urgencia y energía.



La composición utiliza líneas diagonales, contraste, espacio negativo, proximidad y dirección implícita para crear una sensación de tensión compositiva. Esto genera dinamismo y capta la atención del espectador, haciendo que la composición sea visualmente intrigante y activa.

Prototipo de diseño

Aunque esto es para la ventana gráfica, se ha utilizado una resolución de 1920 x 1080. Para poder utilizar el desplazamiento únicamente a nivel de diseño, se ha ampliado el formato para poder crear este efecto de desplazamiento, pero siempre procurando que la experiencia siga teniendo una resolución de 1920 x 1080. (seguir vínculo a [Landing Figma PC](#)).

Nuevamente, el proyecto se creó en un sistema de 12 cuadrículas y el contenido se ha coordinado lo más posible con este diseño y dándole estructura. En cambio, en la versión móvil es un sistema de 8 y una resolución 360 x 4062 (seguir vínculo a [Landing Figma Móvil](#)).



2 – Landing

2.1 Animaciones y transiciones con el DOM

Se han implementado las siguientes animaciones y transiciones interactivas utilizando CSS3:

Menú fijo con redirección de secciones:

Menú: Se mantiene fijo, se podrá redireccionar a cada sección de la landing que el usuario desee acceder



Hover para los Cards: Cada card contiene una imagen que al poner el ratón encima cambia a otra imagen de una bicicleta.



Menú Responsive: Al pulsar en el menú responsive, aparece un div que despliega el menú.



Parallax con Scroll: En las secciones 3 y 6, se ha implementado un efecto parallax que se activa al hacer scroll.



Cards con Efecto Vuelta: Otros cards que giran haciendo el efecto vuelta cuando el ratón pasa por encima de ellas, mostrando un video en lugar de la imagen original.



Formulario de Validación: En el formulario, cuando se escribe incorrectamente en el input, los bordes se ponen en rojo y aparece un texto indicando el error. Si el input es correcto, los bordes se ponen en verde. Todo esto se ha implementado con CSS3.

2.2 Responsive

El diseño de la interfaz es responsive, garantizando que la página se vea y funcione correctamente en diferentes tamaños de pantalla. Se han utilizado Media Queries, FlexBox y Grid layout para lograr esta funcionalidad.

2.3 Uso de SASS y BEM

Se ha utilizado el preprocesador SASS para escribir los estilos de manera modular y organizada. Todos los archivos SASS individuales se importan en un archivo main.scss, que luego se compila en un único archivo main.css.

The image shows a file explorer on the left displaying a project structure with files like menu_Fijo.scss, recursos.scss, and various section and style files. In the center, a code editor shows SCSS variables:

```
$color1: #f25c05;
$color2: #000000;
$color3: #707070;
$color4: #ffffff;
```

On the right, another code editor shows a larger SCSS file with imports and a body selector:

```
@use './recursos/fuentes';
@use './menu_Fijo/headerHamburguesa';
@use './menu_Fijo/header';
@use './menu_Fijo/navbar';
@use './menu_Fijo/botonesAudio';
@use './section1/plegadoSvg';
@use './section2/cardsHover';
@use './section2/texto';
@use './section3/cardsFlip';
@use './section4/canvas';
@use './section4/grafica';
@use './section5/threejs';
@use './section6/contacto';
@use './section6/section6';
@use './section6/svg';

@import "./recursos/variables";
body {
    background: $color2;
    font-family: 'Coach', sans-serif;
}
```

La metodología BEM (Block Element Modifier) se utilizará para nombrar los selectores y las variables. Esta metodología ayuda a mantener el código limpio y entendible

2.4 Elementos multimedia

Se han añadido varios elementos multimedia para mejorar la interactividad y la experiencia del usuario:

Música en el Menú Fijo: Se ha añadido música de fondo que se reproduce desde el menú fijo.

Video en la Sección Uno: En la sección uno, hay un video detrás de un SVG que actúa como cortina. Al pulsar un botón, la cortina SVG se quita y revela el video.

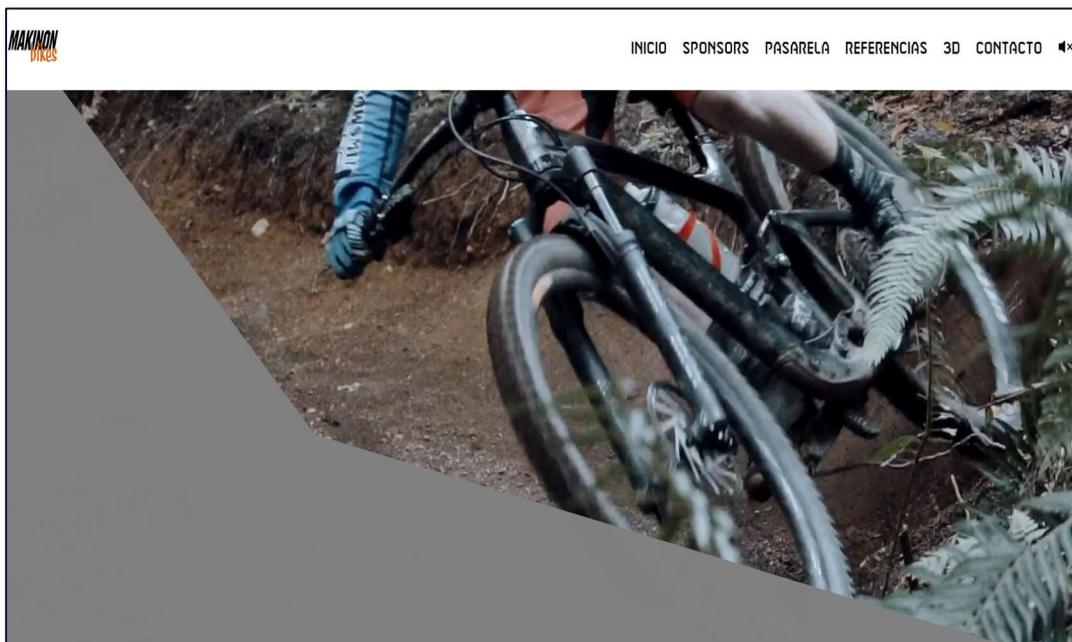
```
<div class="contenedorPrincipal parallax" id="section1">
    <div class="contenedorPrincipal_video">
        <video autoplay loop muted>
            <source src="multimedia/page1.mp4" type="video/mp4">
        </video>
    </div>

    <svg id="plegable" class="contenedorPrincipal_plegable">
        <polygon id="poligono" points="0,0 2520,0 2520,1080 0,1080 0,0" style="fill: gray" />
    </svg>
```

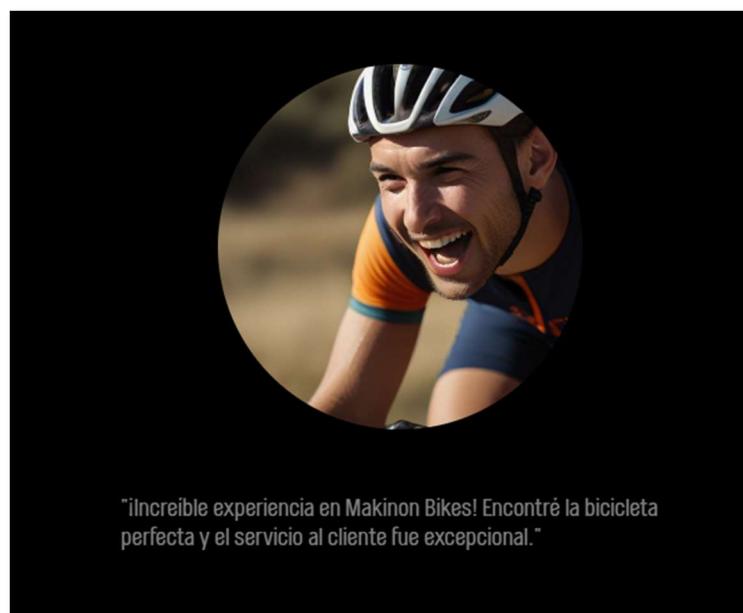
Videos en los Cards: En los cards que giran, se muestra un video cuando el ratón pasa por encima de ellos.

Imágenes Relacionadas: Se han incluido imágenes relacionadas con el mundo de las competiciones de bicicletas.

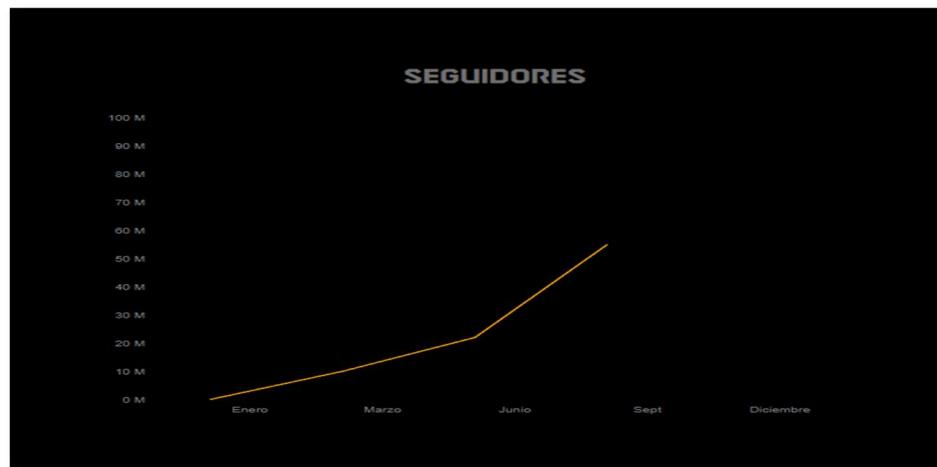
SVG en la Sección Uno: Un SVG que actúa como cortina antes del video y que se quita al pulsar un botón.



Canvas para Clientes Satisfechos: Un canvas que muestra imágenes de clientes satisfechos y el texto de sus comentarios.

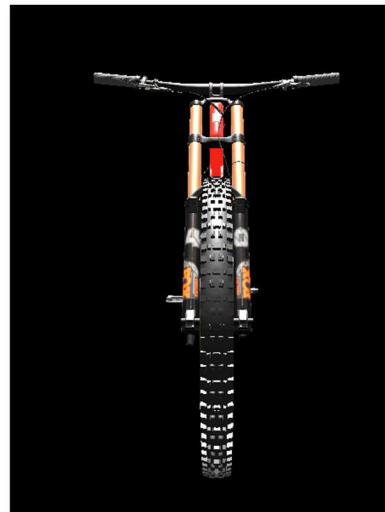


Gráfica Interactiva con D3.js: Una gráfica SVG unida a la librería D3.js que muestra el crecimiento por mes con un efecto interactivo.



Efecto de Biela con Anime.js: Un SVG que muestra el efecto de una biela de bicicleta con texto alrededor utilizando Anime.js para la animación. Este SVG fue creado con el software Inkscape.

Bicicleta 3D con Three.js: Un canvas unido a la librería Three.js que muestra una bicicleta en 3D que se puede ver en distintos ángulos con el scroll.



Iconos SVG: Se han utilizado iconos SVG para el altavoz y el botón de play.



2.5 Imagen Vectorial

Aunque se ha mencionado el uso de la imagen vectorial en el punto anterior cabe añadir que el efecto de la biela se ha hecho con el software Inkscape para tener una diseño propio y apropiado para la temática.



2.6 Despliegue

Al igual que hemos hecho con el backend hemos optado por desplegar nuestra Landing page en una instancia EC2 de AWS, asignarle una IP Elástica y finalmente registrar el dominio el Duck DNS.

En la siguiente imagen podemos ver las características de nuestra instancia.

Resumen de instancia de i-05f0c18e89d38cdb9 (Landing Makinon Bikes) Información		
Se ha actualizado hace 3 minutes		
ID de la instancia i-05f0c18e89d38cdb9 (Landing Makinon Bikes)	Dirección IPv4 pública 34.193.112.56 dirección abierta	Direcciones IPv4 privadas 172.31.48.54
Dirección IPv6 -	Estado de la instancia En ejecución	DNS de IPv4 pública ec2-34-193-112-56.compute-1.amazonaws.com dirección abierta
Tipo de nombre de anfitrión Nombre de IP: ip-172-31-48-54.ec2.internal	Nombre DNS de IP privada (solo IPv4) ip-172-31-48-54.ec2.internal	Direcciones IP elásticas 34.193.112.56 [IP pública]
Responder al nombre DNS de recurso privado IPv4 (A)	Tipo de instancia t2.medium	Hallazgo de AWS Compute Optimizer Suscríbete a AWS Compute Optimizer para recibir recomendaciones. Más información
Dirección IP asignada automáticamente -	ID de VPC vpc-04ec052c81dd52362	Nombre del grupo de Auto Scaling -
Rol de IAM -	ID de subred subnet-012e56504adb927c4	
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-1:527326672295:instance/i-05f0c18e89d38cdb9	

Una vez tenemos acceso a nuestra instancia mediante SSH hemos seguido los siguientes pasos para clonar la carpeta Landing que teníamos alojada en el repositorio de Makinon Bikes.

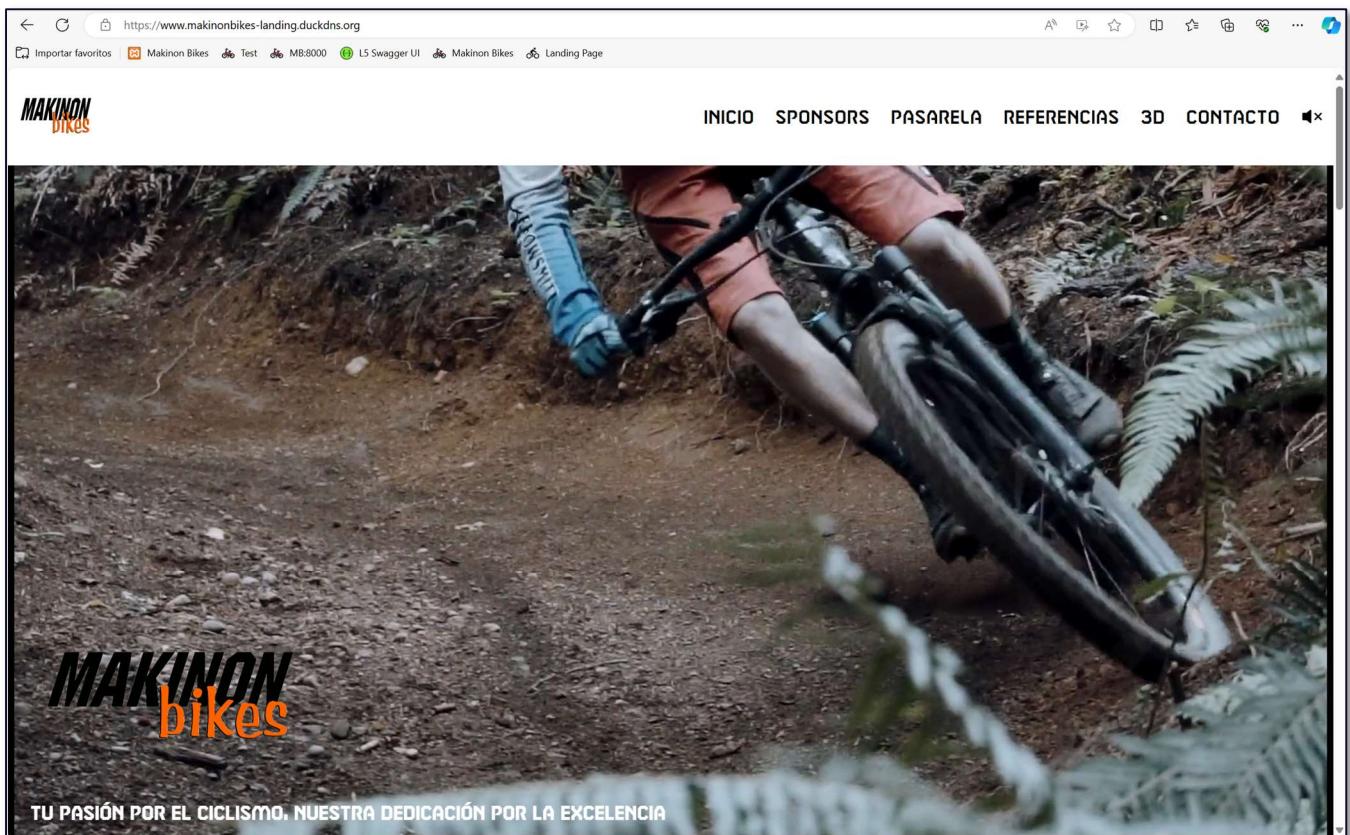
```
● ● ●

1 # Actualizar la instancia
2 sudo apt update && sudo apt upgrade -y
3
4 # Instalar Apache
5 sudo apt install apache2
6
7 # Crear un directorio para el proyecto
8 sudo mkdir /var/www/landing
9
10 # Clonar el repositorio, pero en concreto la carpeta landing, para ello usamos el comando git sparse-checkout y nuestras credenciales de github
11 cd /var/www/landing
12 sudo git init
13 sudo git remote add -f origin https://github.com/username/repo.git
14 tricomonas@hotmail.com
15 (token github) *****
16
17 sudo git config core.sparseCheckout true
18 echo "landing/*" | sudo tee -a .git/info/sparse-checkout
19 sudo git pull origin main
20 tricomonas@hotmail.com
21 cd(token github) *****
22
23 # Dirigirse a la carpeta de sites-available de apache
24 cd /etc/apache2/sites-available
25
26 # Crear un archivo de configuración para el sitio
27 sudo nano landing.conf
28
29 <VirtualHost *:80>
30     # The ServerName directive sets the request scheme, hostname and port that
31     # the server uses to identify itself. This is used when creating
32     # redirection URLs. In the context of virtual hosts, the ServerName
33     # specifies what hostname must appear in the request's Host: header to
34     # match this virtual host. For the default virtual host (this file) this
35     # value is not decisive as it is used as a last resort host regardless.
36     # However, you must set it for any further virtual host explicitly.
37     ServerName www.makinonbikes-landing.duckdns.org
38
39     ServerAdmin webmaster@localhost
40     DocumentRoot /var/www/landing
41
42     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
43     # error, crit, alert, emerg.
44     # It is also possible to configure the loglevel for particular
45     # modules, e.g.
46     #LogLevel info ssl:warn
47
48     ErrorLog ${APACHE_LOG_DIR}/error.log
49     CustomLog ${APACHE_LOG_DIR}/access.log combined
50
51     # For most configuration files from conf-available/, which are
52     # enabled or disabled at a global level, it is possible to
53     # include a line for only one particular virtual host. For example the
54     # following line enables the CGI configuration for this host only
55     # after it has been globally disabled with "a2disconf".
56     #Include conf-available/serve-cgi-bin.conf
57 </VirtualHost>
```

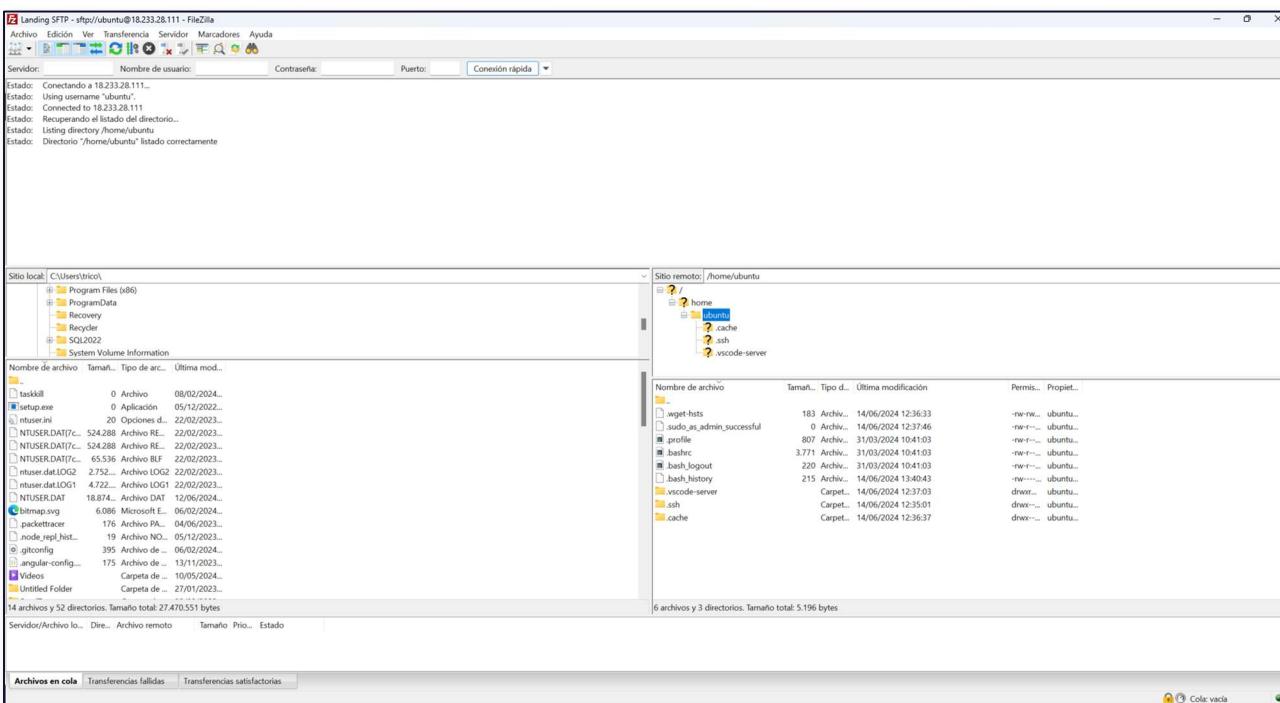
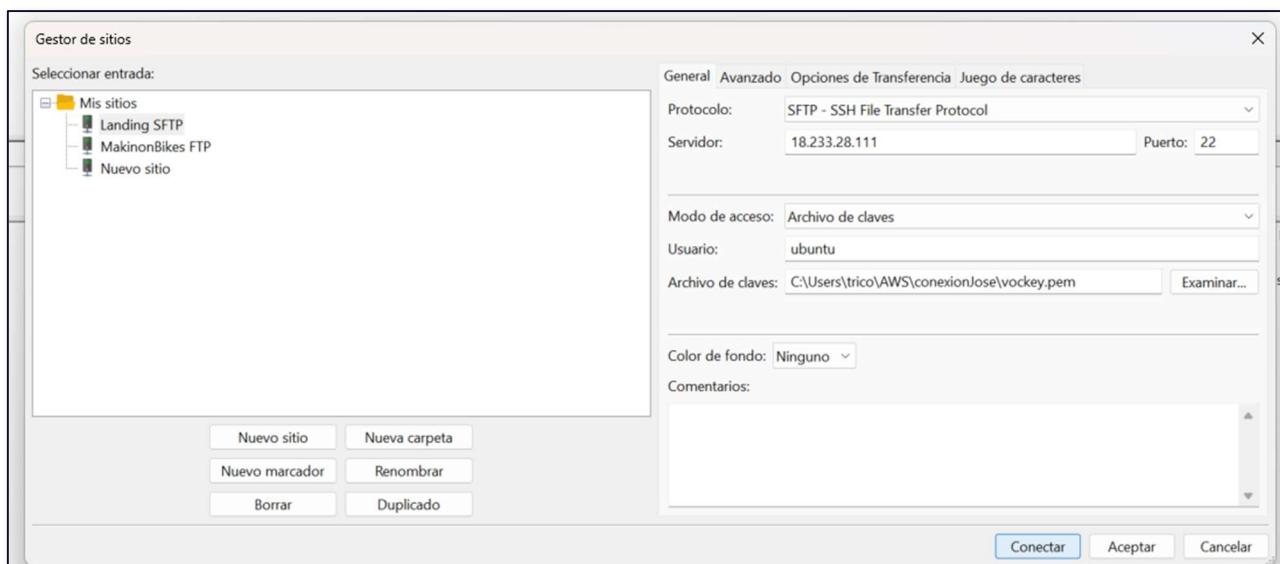


```
1 # Habilitar el sitio y deshabilitar el sitio por defecto para despues reiniciar apache
2 sudo a2ensite landing.conf
3 sudo a2dissite 000-default.conf
4 sudo systemctl restart apache2
5
6 # Instalar Node.js y npm dentro de la carpeta del proyecto, de este modo se creará la carpeta node_modules
7 cd /var/www/landing/
8
9 sudo apt install nodejs
10 sudo apt install npm
11 sudo npm install
```

Finalmente, utilizando la herramienta Certbot, instalaremos un certificado SSL para el dominio y así tendremos nuestra Landing en la red.



Posteriormente al despliegue hemos implementado un protocolo SFTP para a posteriori, y en caso de necesitarlo, poder transferir archivos a la instancia de manera segura, y para ello hemos hecho uso de la vockey.pem para la autentificación.



2.6 Despliegue mediante Codedeploy

Una vez teníamos subida nuestra Landing Page hemos querido probar a hacer un despliegue de la misma mediante codedeploy haciendo uso de una plantilla .yml en *CloudFormation*. Recordemos que AWS CloudFormation es un servicio que nos ayuda a modelar y configurar nuestros recursos de AWS. Tan solo debemos crear una plantilla que describa todos los recursos de AWS que queremos y CloudFormation se encarga de aprovisionar y configurar esos recursos.

Para ello hemos creado una carpeta en nuestro Ubuntu y dentro encontramos tres archivos:

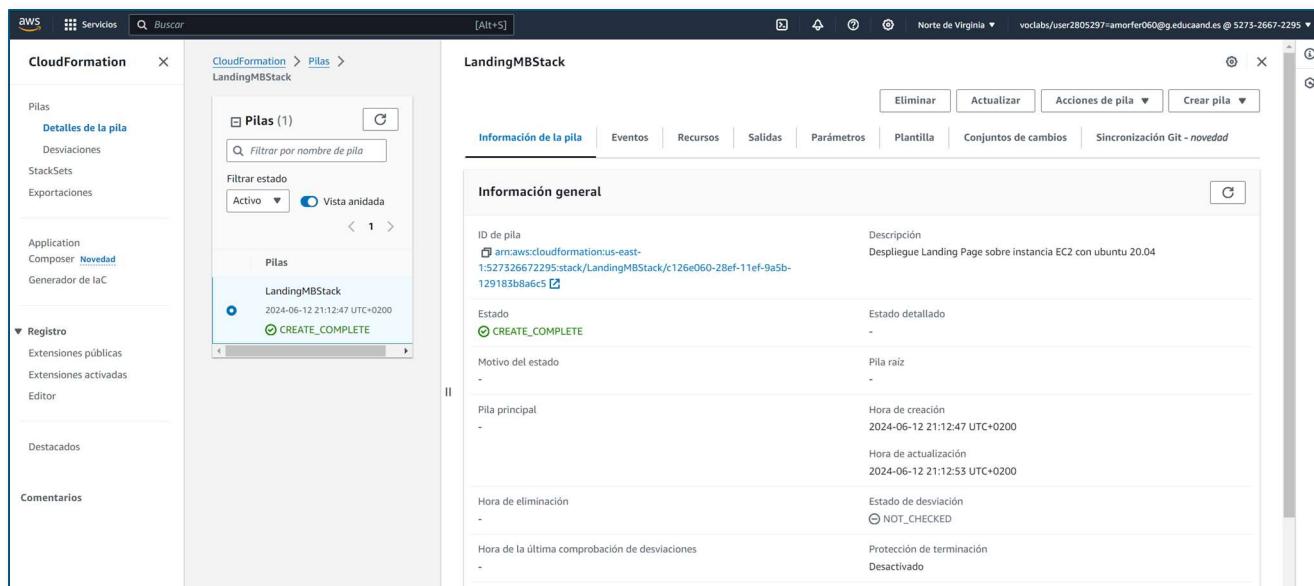
- **desplegar.sh** - Ejecuta el comando `aws cloudformation deploy` para desplegar la pila. Este comando toma el nombre de la pila, el archivo de plantilla de CloudFormation y las capacidades necesarias como argumentos.
- **borrar.sh** – Elimina la pila de AWS
- `exit.sh` - Lista los valores de exportación de una pila de AWS CloudFormation
- **landing.yml** – Es la plantilla donde se definen los parámetros de la pila. Dentro definimos que tipo de instancia EC2 queremos, el grupo de seguridad, etc.

Para hacer el despliegue:

- Nos vamos dentro de la carpeta `DespliegueLanding` que tenemos la home de nuestro usuario en Ubuntu.
- Una vez hemos accedido a dicha carpeta, por consola, ejecutamos `./desplegar.sh`

```
alberto@Alberto:~$ cd DespliegueLanding/
alberto@Alberto:~/DespliegueLanding$ ./desplegar.sh
Desplegando la pila LandingMBStack
La pila LandingMBStack ha sido desplegada correctamente.
```

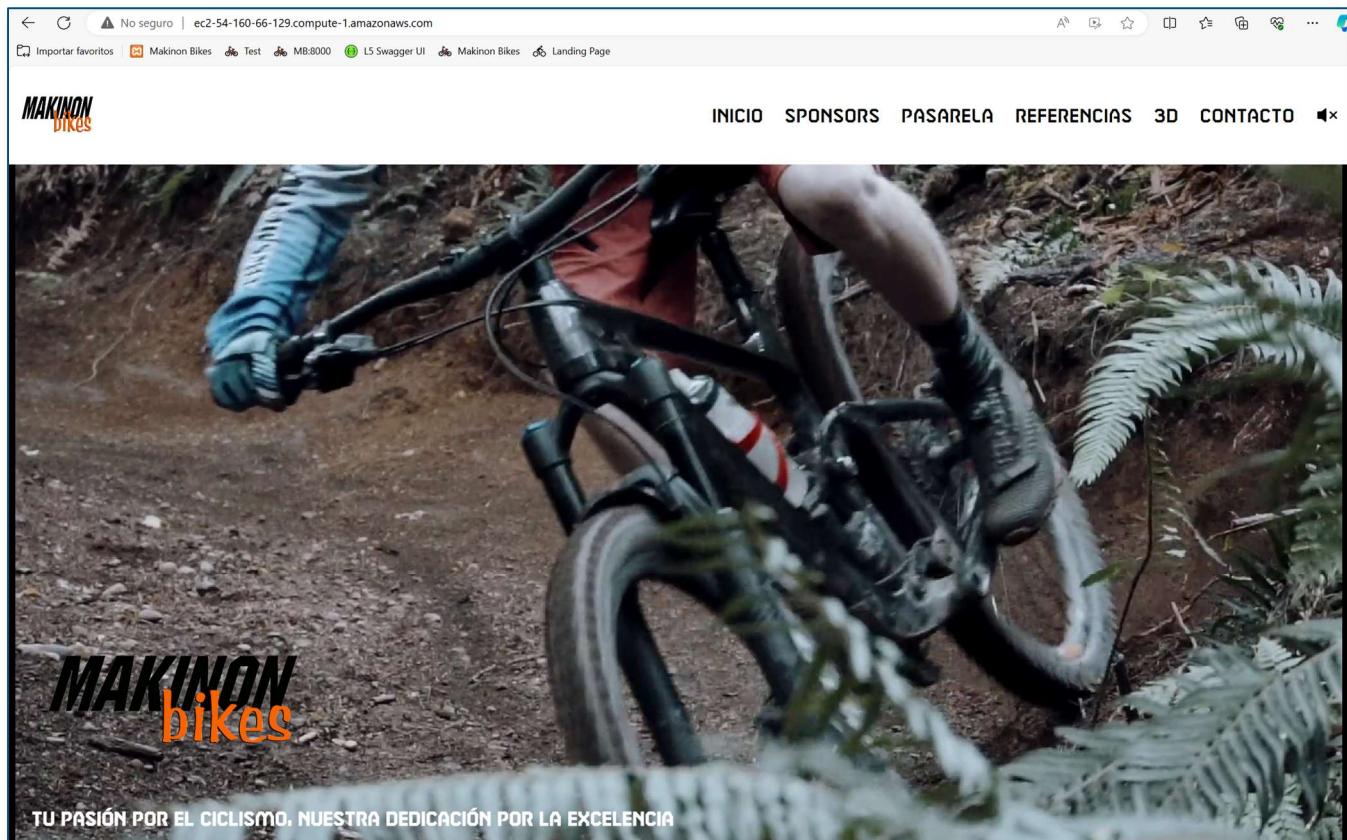
- Nos dirigimos a la sección CloudFormation de AWS y buscamos las Pilas que hay en nuestro laboratorio y podremos observar que se ha creado correctamente.



- Podemos observar los eventos que han ocurrido hasta su despliegue.

Información de la pila					
Eventos		Recursos	Salidas	Parámetros	Plantilla
Eventos (11)		Sincronización Git - novedad			
Buscar eventos					Detectar causa raíz
Marca temporal	ID lógico	Estado	Estado detallado	Motivo del estado	
2024-06-12 21:13:37 UTC+0200	LandingMBStack	CREATE_COMPLETE	-	-	
2024-06-12 21:13:34 UTC+0200	EC2Instance	CREATE_COMPLETE	-	-	
2024-06-12 21:13:13 UTC+0200	LandingMBStack	CREATE_IN_PROGRESS	CONFIGURATION_COMPL ETE	Eventual consistency check initiated	
2024-06-12 21:13:13 UTC+0200	EC2Instance	CREATE_IN_PROGRESS	CONFIGURATION_COMPL ETE	Eventual consistency check initiated	
2024-06-12 21:13:02 UTC+0200	EC2Instance	CREATE_IN_PROGRESS	-	Resource creation Initiated	
2024-06-12 21:13:00 UTC+0200	EC2Instance	CREATE_IN_PROGRESS	-	-	
2024-06-12 21:12:59 UTC+0200	SecurityGroup	CREATE_COMPLETE	-	-	
2024-06-12 21:12:59 UTC+0200	SecurityGroup	CREATE_IN_PROGRESS	-	Resource creation Initiated	
2024-06-12 21:12:55 UTC+0200	SecurityGroup	CREATE_IN_PROGRESS	-	-	
2024-06-12 21:12:53 UTC+0200	LandingMBStack	CREATE_IN_PROGRESS	-	User Initiated	
2024-06-12 21:12:47 UTC+0200	LandingMBStack	REVIEW_IN_PROGRESS	-	User Initiated	

- Pero lo que mas nos interesa es que si nos vamos al apartado de salidas de la Pila y hacemos click sobre el valor de la clave InstancePublicDNS nos redirige a nuestra Landing Page.



A su vez, si nos conectamos mediante ssh a la nueva instancia que se ha creado a través de la pila y accedemos a la carpeta /var/www podremos ver que dentro hay una carpeta llamada *landing*, en la cual, en su interior, se encuentra clonado todo el contenido de nuestro repositorio de GitHub.

```
ubuntu@ip-172-31-56-104:~$ cd /var/www
ubuntu@ip-172-31-56-104:/var/www$ ls
html landing
ubuntu@ip-172-31-56-104:/var/www$ cd landing/
ubuntu@ip-172-31-56-104:/var/www/landing$ ls
css index.html multimedia package-lock.json public webpack.config.js
dist js node_modules package.json sass
ubuntu@ip-172-31-56-104:/var/www/landing$ Connection to ec2-54-160-66-129.compute-1.amazonaws.com closed by remote host.
Connection to ec2-54-160-66-129.compute-1.amazonaws.com closed.
```

3 – Angular

3.1 Introducción

La aplicación es un sistema de gestión de citas para un taller de bicicletas, diseñado con Angular 17. El sistema permite a los usuarios gestionar citas mientras proporciona a los administradores herramientas para supervisar y analizar las actividades del taller.

3.2 Estructura de la Aplicación

Rol Usuario

Componentes y Rutas:

- Navbar: Navegación entre las diferentes secciones.
- Pedir Cita: Componente para solicitar nuevas citas. Utiliza un formulario vinculado a modelos de Angular para recoger la información necesaria.
- Calendario: Muestra todas las citas actuales del taller (con limitaciones) y los huecos disponibles.
- Citas Anteriores: Lista las citas pasadas del usuario, mostrando detalles como fecha, hora, tipo de servicio, texto e imagen.
- Dirección: Proporciona la dirección y otros detalles de contacto del taller.

Rol Administrador

Componentes y Rutas:

- Navbar: Navegación entre las distintas secciones administrativas.
- Calendario: Similar al del usuario, pero sin restricciones, muestra todas las citas del taller.
- Citas Pendientes: Lista todas las citas actuales que aún no se han atendido, permitiendo al administrador gestionar y priorizar las actividades diarias.
- Gráfica de Citas: Representación gráfica de las citas, utilizando Chart.js para visualizar estadísticas como el número de citas por día/semana, tipos de servicios más solicitados y fechas más demandadas.

3.3 Rol Usuario

3.3.1 Flujo de Acceso del Usuario

- Inicio de Sesión en Laravel: Cuando el usuario accede a la aplicación a través de Laravel, y está correctamente autenticado, se redirige automáticamente a la landing page de Angular.
- Componente Landing Page: Este componente es la primera interfaz que ve el usuario tras su autenticación. Funciona como un punto de entrada personalizado que da la bienvenida al usuario.

- Detalles del Componente Landing Page
 - Bienvenida Personalizada:
 - Visualización del Texto de Bienvenida: Al cargar el componente, se muestra un mensaje de bienvenida. Este mensaje podría ser algo simple como "Bienvenido de nuevo, [Nombre del Usuario]".
 - Guardado del Token en LocalStorage:
 - Simultáneamente, el componente activa un servicio que almacena el token de sesión en el LocalStorage del navegador. Esto es crucial para mantener la sesión activa y autenticada durante la navegación del usuario en la aplicación.
 - Obtención de Datos del Usuario:
 - Método de Llamada a la Base de Datos: El componente incluye un método específico que realiza una llamada a la base de datos para obtener datos del usuario, como su nombre.
 - Reflejo de los Datos en la UI:
 - Los datos obtenidos se utilizan para personalizar tanto la barra de navegación (Navbar) como la propia página de landing, asegurando que la experiencia del usuario sea acogedora y personalizada.

BIENVENIDO

user

¡Bienvenidos al Taller de MakinoBikes! Estamos encantados de recibirlos en nuestro espacio dedicado a la pasión por el ciclismo. Aquí en MakinoBikes, ofrecemos servicios de mantenimiento, reparación y personalización para que tu bicicleta siempre esté en óptimas condiciones. Nuestro equipo de expertos está comprometido con la calidad y la atención al detalle para garantizar que cada bicicleta que pase por nuestras manos salga mejor que nunca. No importa si eres un ciclista profesional o un entusiasta del fin de semana, en MakinoBikes estamos aquí para ayudarte a disfrutar de cada pedaleo al máximo. ¡Gracias por confiar en nosotros y bienvenidos a nuestra comunidad ciclista! ¡Vamos a rodar juntos hacia nuevas aventuras! 🚴‍♂️🌟

Integración y Seguridad

- Servicios Angular:
 - Servicio de Autenticación: Encargado de manejar las operaciones relacionadas con la autenticación, incluido el almacenamiento y recuperación del token de LocalStorage.
 - Servicio de Usuario: Se encarga de hacer las peticiones al backend para recuperar los datos del usuario necesarios para personalizar la interfaz.
- Consideraciones de Seguridad:
 - HTTPS: Todas las comunicaciones entre el cliente y el servidor se realizan sobre HTTPS para proteger los datos sensibles del usuario y el token de autenticación.
 - Validación del Token: Cada solicitud al servidor incluye la validación del token para asegurar que las peticiones sean hechas por un usuario autenticado y evitar accesos no autorizados.

3.3.2 Proceso Completo de Solicitud de Cita

Formulario de Solicitud de Cita

- Campos del Formulario: El usuario debe llenar diversos campos requeridos para programar una cita. Esto incluye selecciones de fecha y hora, información personal, y cualquier dato relevante para la cita.
- Componente con Restricciones de Avance: Se ha implementado un sistema de validación paso a paso dentro del formulario, donde el usuario no puede avanzar al siguiente paso hasta que el actual esté correctamente completado. Este enfoque asegura que la información sea recogida de manera estructurada y completa.

The screenshot shows a user interface for booking an appointment. It consists of three horizontal steps: 1. Paso 1: Selección de fecha y hora (Step 1: Select date and time), 2. Paso 2: Detalles de la cita (Step 2: Appointment details), and 3. Paso 3: Subir archivo (Step 3: Upload file). Step 1 is active and visible. It contains fields for selecting a date (Elige una cita, MM/DD/YYYY) and a time (Time, hh:mm tt). A 'Siguiente' (Next) button is at the bottom. The overall design is clean with a light gray background and white cards for each step.

Validación y Envío del Formulario

- Validación Frontend: Un servicio específico se encarga de realizar todas las validaciones necesarias en el frontend. Esto incluye la comprobación de formatos de datos, la integridad de los archivos subidos, y que todos los campos necesarios estén llenos.
 - Interacción con Backend:

- Si la información es incorrecta o incompleta, se muestra un diálogo de error personalizado, pidiendo al usuario que corrija los errores.
- Si la información pasa todas las validaciones frontend, se procede a enviarla al backend para una validación adicional.
- Validación Backend y Almacenamiento de la Cita
 - El backend recibe los datos del formulario y realiza sus propias validaciones para asegurarse de que la cita puede ser programada según los criterios del sistema y la disponibilidad.
 - Verificación de la Cita
 - Si hay un conflicto o si la cita ya existe, el sistema informará al usuario con un mensaje de error, indicando que no se puede programar la cita.
 - Si no existe ningún conflicto y la cita puede ser programada, se procederá a almacenar los detalles de la cita en la base de datos.

Confirmación al Usuario

- Diálogo de Confirmación: Una vez que la cita se haya almacenado exitosamente, se mostrará un mensaje de confirmación al usuario, indicando que la cita ha sido programada correctamente.
- Retroalimentación Visual y Auditiva: El sistema proporcionará tanto confirmaciones visuales como, opcionalmente, auditivas para asegurar que el usuario comprenda claramente el estado de su solicitud.

3.3.3 Componente del Calendario

Descripción del Componente del Calendario

- El componente del Calendario es una interfaz clave dentro de la aplicación que permite a los usuarios visualizar las citas programadas en un formato de calendario claro y accesible, utilizando la librería FullCalendar.io.

- Características y Funcionalidades del Calendario
 - Visualización de Citas:
 - Las citas se muestran en el calendario con indicadores de los días y horas específicas en que están programadas.
 - Cada cita aparece marcada como "ocupado" sin mostrar el nombre o detalles específicos del cliente, asegurando privacidad y simplicidad en la visualización.
 - Integración con Festivos:
 - API de Festivos: La aplicación integra una API que provee información sobre los días festivos en España y, más específicamente, en la comunidad de Andalucía. Esto permite al calendario mostrar estos días como no disponibles para citas.
 - Visualización de Festivos: Los días festivos se destacan en el calendario, diferenciándolos claramente de los días normales para evitar la programación de citas en estas fechas.

- Interacción con la Base de Datos:
 - Servicio de Acceso a la Base de Datos: Un servicio dedicado se encarga de recuperar la información de todas las citas programadas desde la base de datos y las integra en el calendario.
 - Actualización en Tiempo Real: El calendario se actualiza dinámicamente para reflejar cualquier cambio o adición en las citas, asegurando que la información esté siempre actualizada.
 -

The screenshot shows a monthly calendar for May 2024. The days of the week are labeled MON, TUE, WED, THU, and FRI at the top. The dates from 1 to 31 are listed below. Several events are overlaid on the calendar:

- MON 29:** A blue box covers the entire day, labeled "PISTA DEL TRABAJO".
- TUE 1:** A yellow box covers the entire day.
- WED 2:** A blue box covers the entire day.
- THU 3:** A yellow box covers the entire day.
- FRI 4:** A blue box covers the entire day. Below it, two small blue circles indicate "● 12P OCUPADO".
- SAT 5:** A blue box covers the entire day. Below it, three small blue circles indicate "● 6:27P OCUPADO", "● 7:27P OCUPADO", and "● 8:27P OCUPADO".

 The calendar also includes navigation buttons for "TODAY", "PREV", and "NEXT".

Tecnologías y Herramientas Utilizadas

- FullCalendar.io: Se utiliza para la creación y manejo del componente del calendario, aprovechando sus capacidades para manejar eventos y personalizar la visualización.
- APIs de Festivos: Integración con APIs externas para obtener datos actualizados sobre días festivos.

3.3.4 Componente de Citas Anteriores

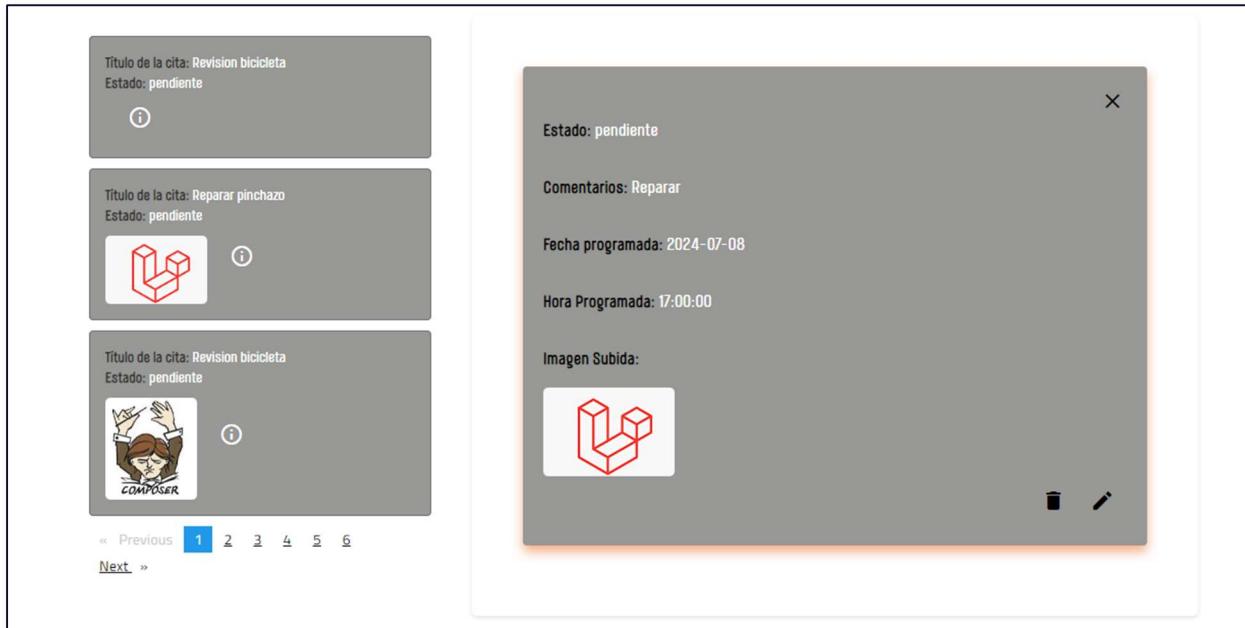
Descripción del Componente

El componente de Citas Anteriores permite a los usuarios visualizar un historial de todas las citas programadas, desde la más reciente hasta la primera, facilitando un acceso rápido y organizado a la información pasada.

Funcionalidades y Características

- Visualización Pagina:
 - Las citas se muestran de manera paginada para mejorar la navegabilidad y el rendimiento, presentando las citas de la más reciente a la más antigua.
 - Se proporcionan detalles breves de cada cita directamente en la lista para una referencia rápida.
- Acceso a Más Información:
 - Icono de Información ('i'): Al hacer clic en este ícono, se abre un componente adicional donde se pueden visualizar todos los detalles de la cita seleccionada.

- Opciones de Gestión de Citas: Dentro de este detalle, el usuario puede ocultar, eliminar o modificar la cita, dependiendo de su estado actual.
- Modificación de Citas: Solo se permite modificar las citas que estén en estado "pendiente". Las citas "canceladas" o "aprobadas" no pueden ser modificadas.



- Ruta de Modificación:
 - Ruta Hija para Modificación: Si el usuario decide modificar una cita pendiente, se redirige a una ruta hija donde puede acceder al formulario de solicitud/modificación de citas.
 - Reutilización de Componentes y Servicios: Para la modificación de citas, se reutilizan muchos de los servicios y componentes previamente mencionados en la aplicación, como los formularios de solicitud de cita. Pero en este caso se le muestra en los campos los datos de la misma cita a modificar.

The image shows a three-step process for modifying a service appointment:

- Paso 1: Selecciona fecha y hora**: A date and time selection screen. It shows a date input field with '6/4/2024' and a calendar icon. Below it is a time input field with 'Time 12:00 PM' and a clear 'X' button. At the bottom is a 'Siguiente' (Next) button.
- Paso 2: Detalles de la cita**: A step where users can enter details about the appointment.
- Paso 3: Subir archivo**: A step where users can upload files related to the appointment.

Tecnologías y Herramientas Utilizadas

- Angular Material: Utilizado para diseñar los elementos de interfaz de usuario, como iconos de información y botones de acción.
- Servicios de Backend: Comunicación con la base de datos para recuperar y gestionar las citas.

3.3.5 Componente de Dirección del taller

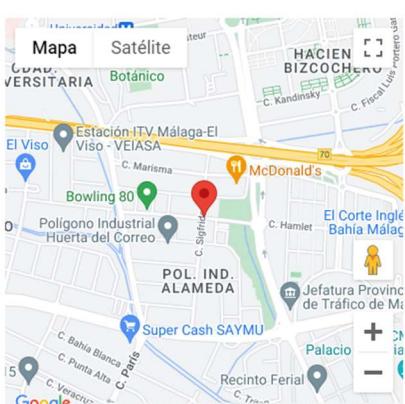
Descripción del Componente

El componente de Dirección ofrece a los usuarios una visualización clara y detallada de la ubicación física del taller de bicicletas, facilitando el acceso y la planificación de visitas.

Características y Funcionalidades

- Visualización de la Dirección:
 - Se muestra la dirección completa del taller, proporcionando a los usuarios la información necesaria para llegar al lugar.
 - Integración de Mapa con Google API:
 - Canvas Interactivo: Utiliza un canvas para mostrar un mapa interactivo que señala la ubicación exacta del taller.
 - Google Maps API: Este componente está conectado a la API de Google Maps, lo que permite una representación precisa y actualizada de la ubicación en un contexto geográfico amplio.
- Interactividad: Los usuarios pueden interactuar con el mapa, por ejemplo, acercando o alejando la vista, y explorando las áreas circundantes para una mejor orientación.

UBICACIÓN



DIRECCIÓN:

Av. de la Constitución 1, 29006 Málaga

HORARIO

Lunes a Viernes: 9:00 - 14:00 y 16:00 - 20:00
Sábado: 9:00 - 14:00

TELÉFONO

+34 954 212121

Tecnologías y Herramientas Utilizadas

- Google Maps API: Proporciona las herramientas necesarias para integrar un mapa interactivo en la aplicación, con capacidades de visualización y navegación detalladas.

3.4 Rol de Administrador

3.4.1 Landing Page

Descripción de la Landing Page

Al igual que en el rol de usuario, cuando un administrador accede a la aplicación, se le redirige a una landing page. Esta página sirve como punto de entrada principal y mantiene el mismo formato y diseño que la utilizada para los usuarios, proporcionando una experiencia coherente y familiar.

Características y Funcionalidades

- Bienvenida al Administrador:
 - La página muestra un mensaje de bienvenida personalizado que puede incluir el nombre del administrador y cualquier otra información relevante para su rol.
- Navegación General:
 - Al igual que en el rol de usuario, la landing page contiene una barra de navegación (Navbar) que facilita el acceso a diferentes secciones específicas del rol de administrador.
 - Aunque el diseño es similar al del usuario, las opciones en la barra de navegación se adaptan a las necesidades y permisos del administrador.

3.4.2 Componente del Calendario

Descripción del Componente del Calendario

El componente del Calendario para el administrador comparte la base y funcionalidades del calendario del usuario, utilizando la misma librería FullCalendar.io para la presentación. Sin embargo, se ha adaptado para satisfacer las necesidades específicas de gestión del administrador.

Características y Funcionalidades

- Visualización Completa de Citas:
 - A diferencia del rol de usuario, donde las citas aparecen marcadas solo como "ocupado", el administrador tiene acceso a detalles más completos de cada cita, incluyendo el título y la información del cliente.
 - Esta funcionalidad permite al administrador tener una visión completa del calendario de citas, facilitando la gestión y el seguimiento de la programación diaria.

3.4.3 Componente de Citas Pendientes

Descripción del Componente

El componente de Citas Pendientes ofrece una interfaz dinámica para la gestión de las citas en diferentes estados: pendientes, canceladas y aceptadas. Utilizando un enfoque visual e interactivo, este componente facilita la organización y la actualización del estado de las citas mediante drag and drop.

Características y Funcionalidades

- Visualización de Citas por Estado:
 - El administrador puede visualizar tres tablas distintas que representan los estados de las citas: pendientes, canceladas y aceptadas.
 - Cada cita se muestra en forma de tarjeta dentro de la tabla correspondiente, con información esencial visible para una gestión rápida.
- Gestión Interactiva con Drag and Drop:
 - Interfaz de Arrastrar y Soltar: Permite al administrador mover las tarjetas de citas entre las tablas para cambiar su estado de forma intuitiva y eficaz.
 - Confirmación de Cambios: Una vez realizadas las modificaciones deseadas, el administrador puede confirmar los cambios con un botón de "Aceptar", lo que desencadena la actualización de la base de datos.
- Integración con la Base de Datos:
 - Servicio de Base de Datos: Un servicio dedicado gestiona la conexión con la base de datos para reflejar los cambios en el estado de las citas, asegurando que la información sea precisa y esté actualizada.



Tecnologías y Herramientas Utilizadas

- Librerías de Drag and Drop de angular material: Utilización de herramientas y librerías especializadas para implementar la funcionalidad de arrastrar y soltar, asegurando una experiencia de usuario fluida y efectiva.

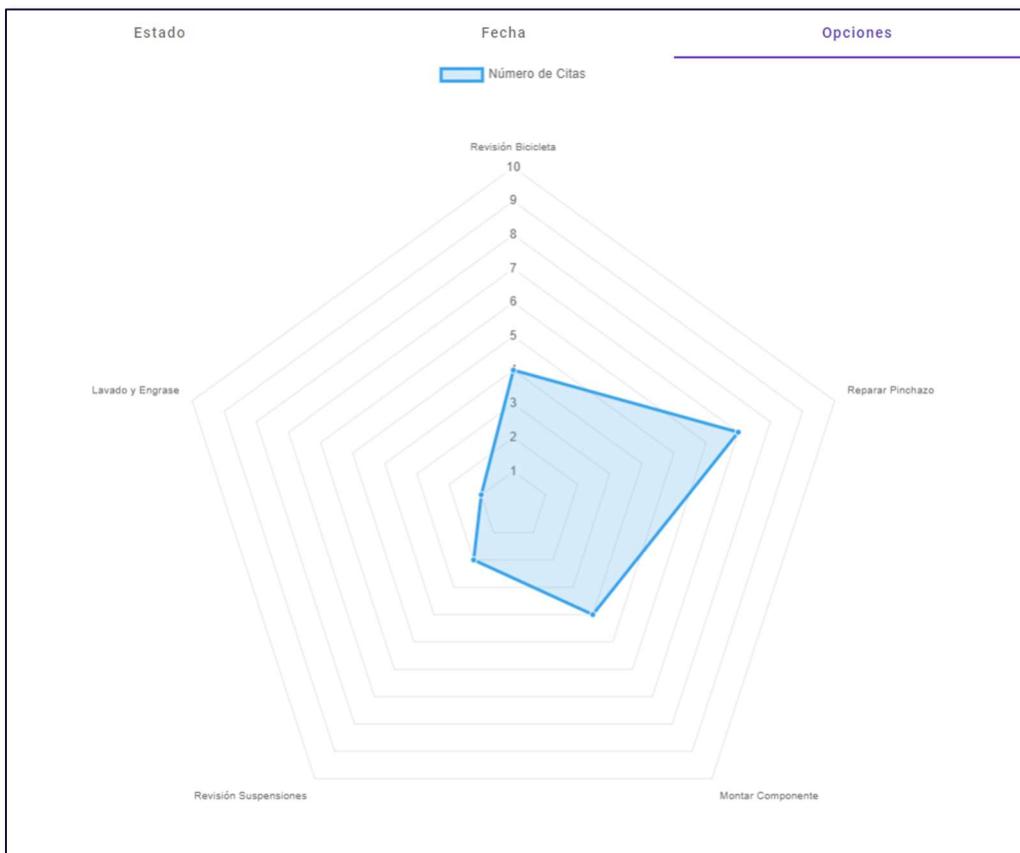
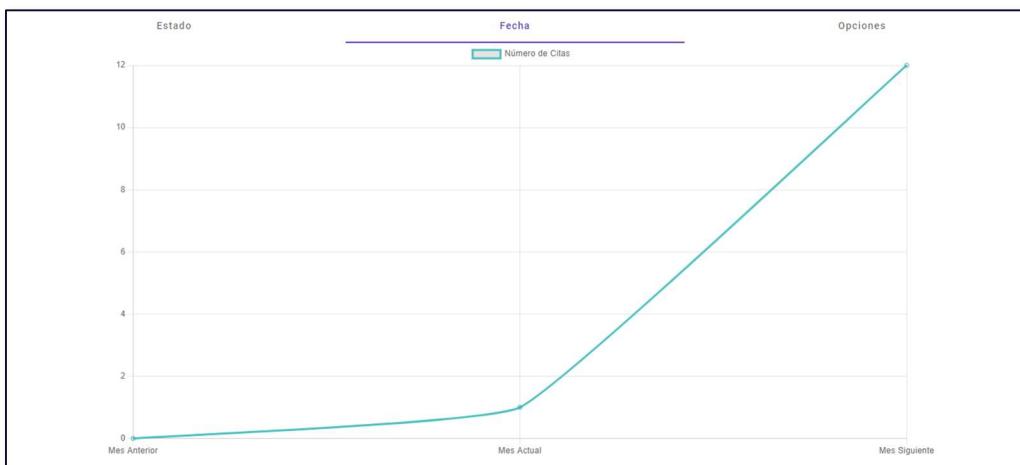
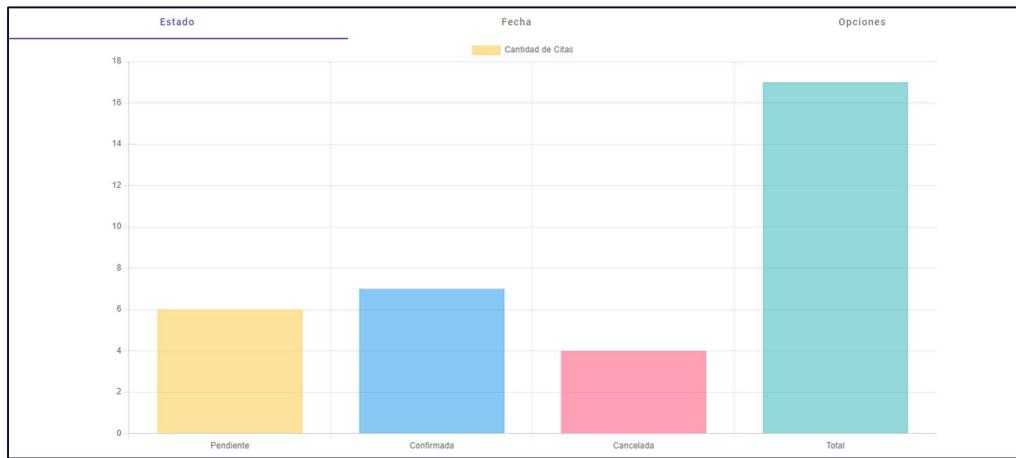
3.4.4 Componente de Gráfica de Citas

Descripción del Componente

El componente de Gráfica de Citas permite al administrador visualizar diferentes aspectos de las citas a través de gráficos dinámicos, facilitando un análisis comprensivo de las tendencias y patrones de citas. Utiliza la librería Chart.js para la representación gráfica de los datos.

Funcionalidades y Características

- Gráfico del Estado de Todas las Citas:
 - Muestra un gráfico que representa los estados de las citas (pendientes, canceladas, aceptadas), permitiendo al administrador obtener una visión rápida de la distribución actual de las citas.
- Gráfico de Citas por Fecha:
 - Presenta un análisis de las citas basado en el mes de solicitud, destacando cuáles meses tienen mayor demanda, lo que puede ayudar en la planificación y asignación de recursos.
- Gráfico de Opciones de Citas:
 - Visualiza las opciones más frecuentes que los usuarios seleccionan al programar una cita, proporcionando insights sobre las preferencias y necesidades de los clientes.
 - Servicio de Conexión a la Base de Datos: Un servicio backend está dedicado a recuperar y procesar los datos necesarios para alimentar los gráficos, asegurando que la información esté actualizada y sea precisa.



Tecnologías y Herramientas Utilizadas

- Chart.js: Se utiliza para la creación de gráficos interactivos y bien diseñados que representan los diversos datos de las citas.

3.5 Estructuración y Tecnología del Proyecto

Descripción General

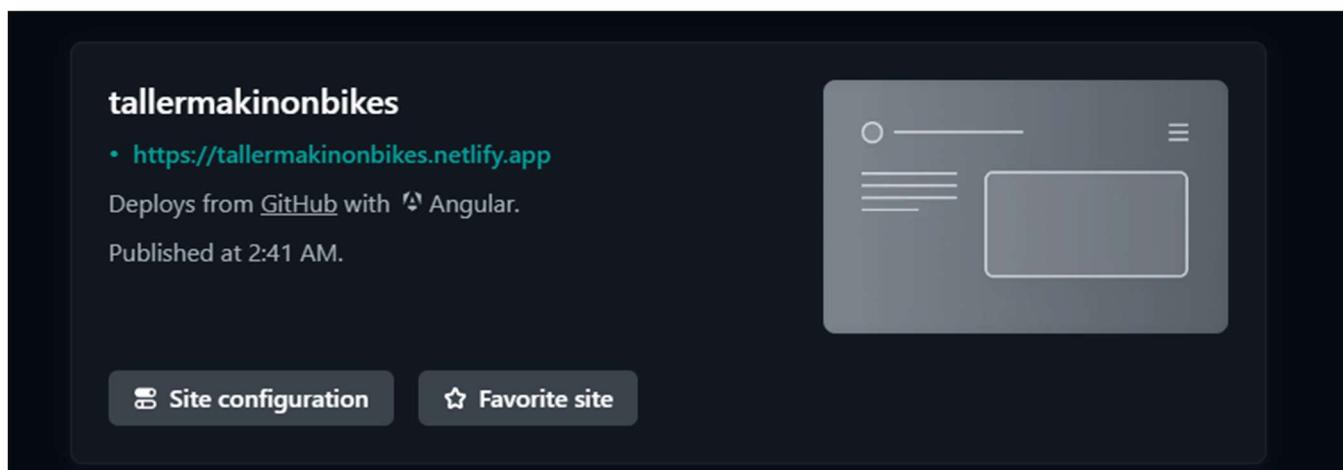
El proyecto ha sido desarrollado con una estructura de código limpia y organizada, utilizando Angular 17 y TypeScript. Este enfoque garantiza la separación de funcionalidades y la modularidad del código, facilitando el mantenimiento y la escalabilidad.

Características y Funcionalidades

- Separación de Código:
 - El proyecto sigue una arquitectura modular, donde cada componente y servicio tiene una responsabilidad única y bien definida, evitando la mezcla de lógicas y facilitando las pruebas unitarias y la depuración.
- Uso de TypeScript:
 - TypeScript proporciona tipado estático y otros recursos modernos que mejoran la legibilidad del código, la detección de errores en tiempo de compilación y el soporte durante el desarrollo.
- Peticiones Asíncronas:
 - Se utilizan las importaciones HTTP de Angular para manejar comunicaciones asíncronas con APIs, permitiendo un manejo eficiente de las respuestas y los estados de la aplicación.
- Persistencia de Datos:
 - La persistencia de datos se maneja a través de LocalStorage, permitiendo almacenar información esencial del usuario y mantener el estado de la sesión.
- API REST:
 - Se integra una API REST para facilitar la comunicación con el backend, permitiendo un intercambio eficiente y estructurado de información.

Despliegue y Producción

- Github y Netlify
 - El código fuente del proyecto está alojado en GitHub, aprovechando las funcionalidades de control de versiones y colaboración.
 - La implementación y el despliegue continuo se gestionan a través de Netlify, facilitando un ciclo de vida de desarrollo ágil y una integración continua sin interrupciones.



BIENVENIDO

¡Bienvenidos al Taller de MakinoBikes! Estamos encantados de recibirlos en nuestro espacio dedicado a la pasión por el ciclismo. Aquí en MakinoBikes, ofrecemos servicios de mantenimiento, reparación y personalización para que tu bicicleta siempre esté en óptimas condiciones. Nuestro equipo de expertos está comprometido con la calidad y la atención al detalle para garantizar que cada bicicleta que pase por nuestras manos salga mejor que nunca. No importa si eres un ciclista profesional o un entusiasta del fin de semana, en MakinoBikes estamos aquí para ayudarte a disfrutar de cada pedaleo al máximo. ¡Gracias por confiar en nosotros y bienvenidos a nuestra comunidad ciclista! ¡Vamos a rodar juntos hacia nuevas aventuras! 🚴‍♂️✨

SOPORTE

- [Contacto](#)
- [Nuestra tienda](#)
- [Mapa del sitio](#)

LO MÁS TOP

- [Promociones especiales](#)
- [Nuevos productos](#)

SÍGUENOS

- [Instagram](#)
- [YouTube](#)
- [Tik Tok](#)

Consideraciones de Seguridad y Privacidad

- Gestión Segura de Datos:
 - Se implementan prácticas robustas para la seguridad de los datos almacenados y transmitidos, particularmente aquellos almacenados en LocalStorage y los intercambiados a través de la API REST.

Consideraciones de Desarrollo y Mantenimiento

- Mantenimiento del Código:
 - El uso de una estructura de código modular y herramientas modernas como TypeScript y Angular facilita el mantenimiento y la actualización del proyecto.
- Pruebas y Depuración:
 - Se enfatiza la importancia de realizar pruebas continuas y depuración para asegurar la funcionalidad y estabilidad de la aplicación.
- Responsive:
 - Todo el proyecto se ha llevado a cabo para ser accesible para cualquier plataforma como smartphone, iPad y ordenadores.

4 – Laravel

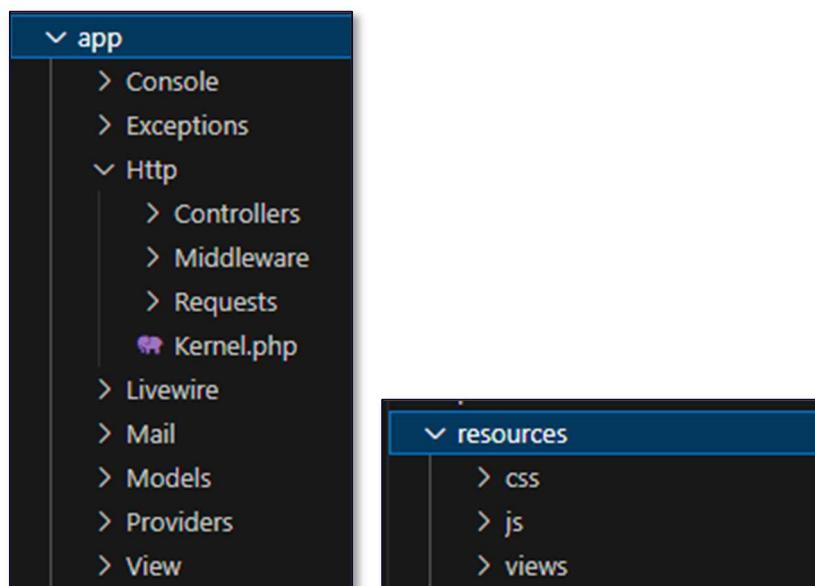
4.1 Introducción

La aplicación Makinon Bikes es la tienda virtual de bicicletas y artículos relacionados con el ciclismo, diseñada con el framework Laravel en su versión 10.48.10. El sistema permite a los usuarios navegar por toda la web entre sus diferentes menús y artículos, buscar si hay algún producto que les interesa, ver su imagen o su descripción con mas detalle y, si así lo quieren, poder comprarlos.

4.2 Estructura de la aplicación

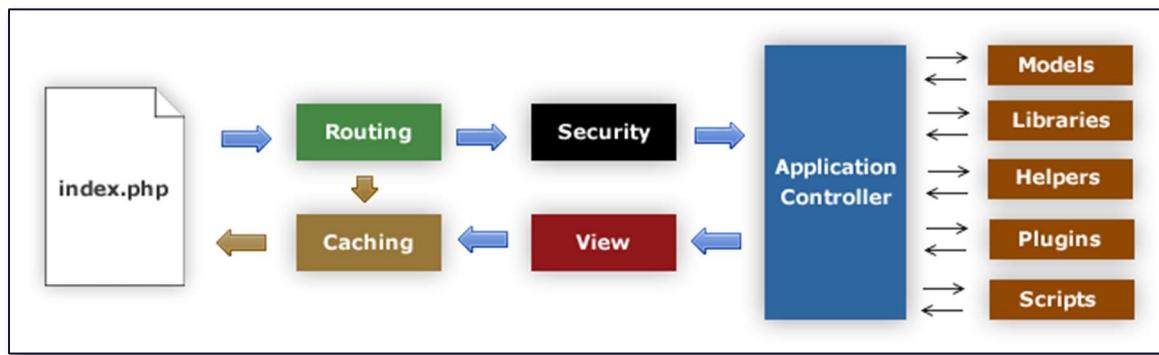
Laravel lo que nos permite es una gran funcionalidad en cada una de las carpetas y archivos en los que está estructurado, siendo dos de ellos los mas utilizados, app y resources.

En app es donde se trabaja el back de la aplicación, es donde podemos encontrar los controladores donde se implementa la lógica del proyecto, middleware para el filtrado de seguridad, modelos



En resources es donde se trabaja el frontend, donde tenemos las vistas HTML, el motor de plantillas, las migraciones de las bases de datos con sus seeders y factorías, el idioma de la aplicación o recursos que no deban ser públicos.

En Laravel existe un único punto de acceso a la aplicación web y cualquier petición del usuario se resolverá según las rutas definidas en *routes.php*, este mismo punto de acceso es el encargado de las autorizaciones de modo que se ofrece un resultado positivo o se devuelve un error.



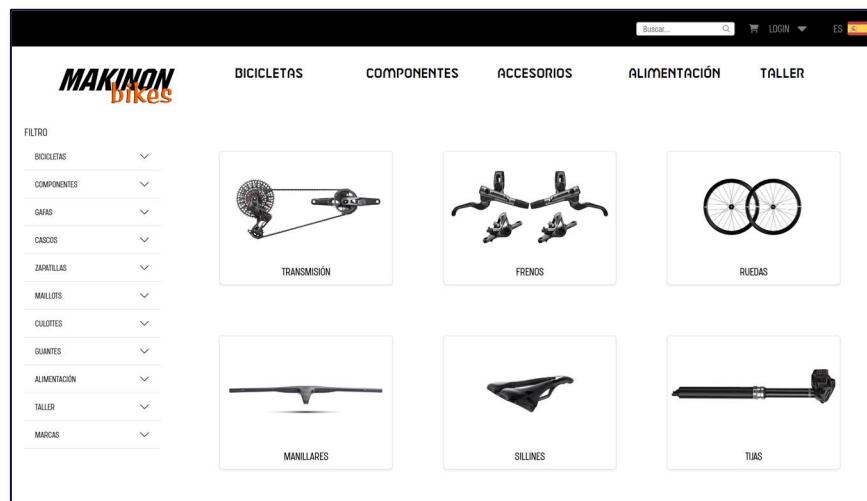
Con esta breve introducción de cómo funciona Laravel, y lo que nos aporta, nos centraremos en los dos roles principales que tenemos en nuestra aplicación.

4.3 Rol del Usuario

4.3.1 Flujo de acceso del usuario

Como habíamos dicho anteriormente el usuario puede navegar de forma libre por la aplicación Makinon Bikes, esto es, no tiene ninguna restricción de acceso a todo el contenido público de productos de la tienda, incluso si lo desea puede crear un carrito de compras que quedará almacenado en la sesión del usuario para posteriormente ser recuperado si decide finalizar la compra.

Pero si bien, desea acceder al Panel de Usuario donde puede ver sus datos, pedidos, cambiar la contraseña o eliminar la cuenta, deberá de estar logeado



4.3.2 Registro

Nuestro proceso de registro es sencillo, sólo pedimos algunos datos básicos, aunque todos ellos son de obligado cumplimiento. El registro posee además un sistema de verificación de datos para que por ejemplo un teléfono no tenga mas de 9 dígitos o contenga alguna letra o que un usuario se registre dos veces con el mismo correo electrónico.

Registro de usuarios

Nombre	NIF
<input type="text"/>	<input type="text"/>
Apellidos	Teléfono
<input type="text"/>	<input type="text"/>
Correo electrónico	Dirección
<input type="text"/>	<input type="text"/>
Contraseña	Código Postal
<input type="text"/>	<input type="text"/>
Confirmar contraseña	Ciudad
<input type="text"/>	<input type="text"/>
¿Ya estás registrado? <input type="button" value="Regístrate"/>	

Una vez registrado el usuario, si ha cumplimentado correctamente el formulario de registro, automáticamente será logeado

4.3.3 Login

Si el usuario ya está registrado y quiere logearse tan sólo ha de clicar sobre la palabra Login que se encuentra en la barra superior de la página y de momento se le redirige a la página de Login donde sólo ha de introducir el correo electrónico con el que se registró y su contraseña.



Bienvenido

Correo electrónico
<input type="text"/>
Contraseña
<input type="text"/>
<input type="checkbox"/> Seguir conectado
Recordar contraseña <input type="button" value="Iniciar sesión"/>
¿No tienes cuenta? Regístrate.

En esta misma vista el usuario puede optar por:

- Marcar Seguir conectado con lo cual se creará una cookie que evitara tener que volver a logearse en caso de que cierre su navegador, cosa que si ocurre si no lo marca, puesto que su autentificación se guarda en la sesión del servidor.
- Recordar contraseña, en ese momento se le redirige a una nueva página donde introducir el correo electrónico con el que se registró en su momento y directamente se le enviará un mail junto con un enlace a través del cual podrá cambiar su contraseña por una nueva.

4.3.4 Integración y seguridad

- Servicios Laravel:
 - Laravel maneja la sesión del usuario automáticamente. Cuando un usuario inicia sesión, Laravel creará una nueva sesión y asociará la sesión con el usuario. Las sesiones se almacenan en el servidor, lo que es más seguro que almacenarlas en el lado del cliente.
 - Laravel también proporciona una forma sencilla de manejar la autorización. Podemos controlar fácilmente qué usuarios tienen permiso para acceder a ciertas áreas de la aplicación.
- Consideraciones de Seguridad:
 - HTTPS: Todas las comunicaciones entre el cliente y el servidor se realizan sobre HTTPS para proteger los datos sensibles del usuario y la sesión de autenticación.
 - Validación del Sesión: Cada solicitud al servidor incluye la validación de la sesión para asegurar que las peticiones sean hechas por un usuario autenticado y evitar accesos no autorizados.

4.3.5 Proceso de compra

Es indispensable que el usuario esté logeado para finalizar el proceso de compra, eso no impide que el usuario pueda consultar los productos y crear un carrito de compra.

Una vez el usuario navega por nuestra tienda si clica sobre un producto es redirigido a la ficha completa de dicho producto, allí podrá seleccionar, en caso de que este disponible la opción, el color y la talla que desea, y con solo hacer clic sobre el botón Añadir al carrito el producto pasará directamente al carrito de compra, como así lo indicará el icono de carrito de la parte superior de la página que pasará de un color gris a un color naranja.

Añadir que en el desplegable de las tallas se le indica al usuario si esa talla en concreto del producto está disponible en cantidad, quedan pocas unidades o si por el contrario el producto está agotado, si el usuario seleccionase esta última opción el botón de agregar al carrito se deshabilita impidiendo que el producto sea añadido para la compra.



BH LYNX RACE 9.9

La Lynx es la bicicleta que te hará llegar al podium de todas las carreras

Colores disponibles

○

Tallas disponibles

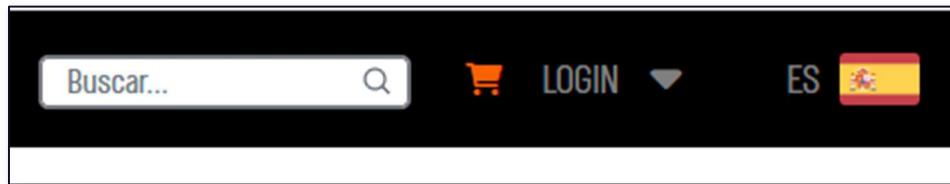
S - Últimas unidades

S - Últimas unidades

M - Últimas unidades

P.V.P. 9999.9 €

Añadir al carrito



Para continuar el proceso de compra el usuario deberá dirigirse al carrito donde se mostrará un resumen del mismo.

TU CESTA DE LA COMPRA				RESUMEN DEL PEDIDO	
PRODUCTO	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL		
	1	9999.9 €	9999.9 €		
BH Lynx Race 9.9 Talla: M Color: White					Valor total de los productos IVA incluido 9.999,90 €
					Gastos de envío 0,00 €
					Total del pedido 9.999,90 €

Seguir comprando

Realizar Pedido

En este momento podrá decidir si realizar la compra o continuar por nuestra web viendo y añadiendo mas productos al carrito.

Si se decide por realizar la compra, pero no está logeado entonces la aplicación lo redirige automáticamente a la página de login, y si está logeado podrá finalizar la compra en tres simples pasos:

1. Resumen de datos del usuario y resumen del importe del pedido.

Mis Datos	Forma de Pago	Confirmar Datos
DATOS PERSONALES		
Email user@user	Teléfono 123456789	TU PEDIDO
Valor total de los productos IVA incluido 9.999,90 € Gastos de envío 0,00 € Total del pedido 9.999,90 €		
Continuar		
DIRECCIÓN DE ENVIO		
Nombre user	Apellidos user	
Dirección Calle del user, 2		
Código Postal 12345	Ciudad Cártama	

2. Selección de la forma de pago, bien mediante tarjeta de crédito, Paypal o transferencia bancaria.

Mis Datos	Forma de Pago	Confirmar Datos
FORMA DE PAGO		
<input checked="" type="radio"/> Tarjeta de crédito Si seleccionas esta opción en el siguiente paso te pediremos los datos de tu tarjeta de crédito, así que tenlos a mano. Continuar		
<input type="radio"/> Paypal		
<input type="radio"/> Transferencia bancaria		

3. Confirmación de datos, donde si ha elegido como forma de pago la tarjeta de crédito deberá llenar los campos correspondientes. Estos campos se verifican mediante el uso de JavaScript, de modo que debe de cumplir que introduzca 16 dígitos en el número de tarjeta, que la fecha de caducidad no sea anterior a la fecha actual y que el CVV sólo contenga 3 dígitos en forma de número.

Cabe añadir que, por temas de seguridad, hemos optado por no añadir la tarjeta de crédito a nuestra base de datos como en un principio pensamos, sino que será la pasarela de pago la que se encargará de la verificación final y el cobro del importe.

Finalmente el usuario deberá aceptar los términos y condiciones de compra para que el botón *Realizar Pedido* esté activo, el hecho de no cumplir la condición de una tarjeta de crédito válida supone que el usuario recibirá un mensaje de error en el proceso de compra.

Como decíamos, hemos añadido además un sistema de validación visual para que el usuario pueda saber en todo momento si los campos que está rellenando son válidos o no

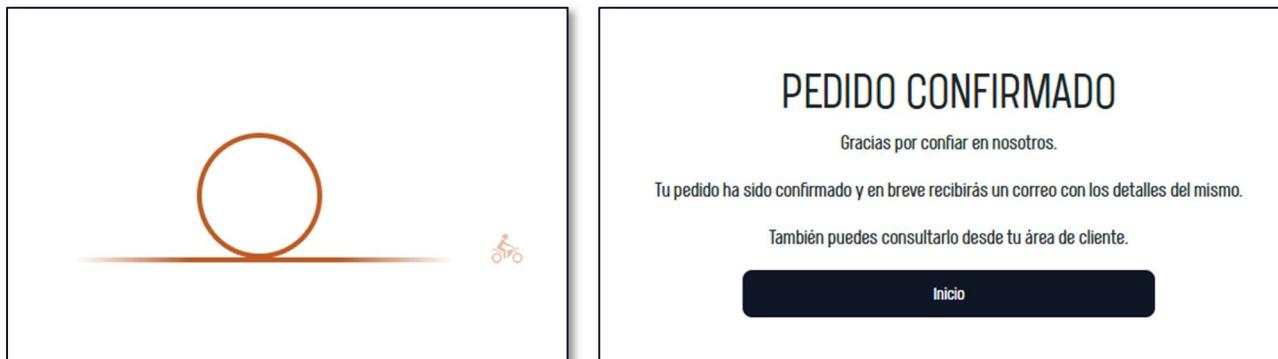
COMPRUEBA TU PEDIDO

FORMA DE PAGO

Ha elegido pagar mediante Tarjeta de crédito

Número de tarjeta	Fecha de caducidad	CVV
123456781234567	12/24	12
Nombre del titular		
al		

Una vez finalizado el proceso de compra el usuario es redirigido a una página de confirmación de pedido donde simulamos el tiempo de espera de la plataforma de pago para finalmente recibir un mensaje de que el pedido ha sido confirmado.



4.3.6 Panel del usuario

En cualquier momento un usuario puede acceder a su perfil a través del icono en forma de V que hay en la cabecera la de página, si pincha sobre él se abre un desplegable donde puede seleccionar entrar a su perfil o hacer log out.

Si atendemos al panel del usuario, que es lo que nos atañe en este punto, tenemos una primera visión rápida en la que el usuario puede ver todos los datos con los que se registró, así como la fecha en la que lo hizo.

Perfil del usuario

Nombre: user
Apellidos: user
Email: user@user
NIF: 12345678b
Teléfono: 123456789
Dirección: Calle del user, 2
Código postal: 12345
Ciudad: Cártama
Fecha de alta:

Perfil del usuario

Nombre: user
Apellidos: user
Email: user@user
NIF: 12345678b
Teléfono: 123456789
Dirección: Calle del user, 2
Código postal: 12345
Ciudad: Cártama
Fecha de alta:

- < />
- 🔒
- 📝
- ✖
- 🔒
- 📄

- ↗ Mis pedidos
- 👤 Modificar perfil
- ⓧ Eliminar cuenta
- 🔒 Cambiar contraseña
- ✖ Salir

Desde este panel podemos:

- Consultar los pedidos que ha realizado, ver el detalle del mismo, ver la factura o descargarla en formato pdf.

Mis Pedidos

Número de pedido	Fecha	Importe	Estado	Comentario	Detalle	Factura	Descargar
11	2024-06-05	9999.9 €	pendiente				

[Volver](#)

MAKINON
bikes

www.makinonbikes.com

FACURA

DATOS DEL CLIENTE

Nº Factura: 9
Pedido: 11
Fecha: 2024-06-05

user user
Calle del user, 2
12345, Cártama
123456789
user@user

PRODUCTO	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Bicicleta montaña BH Lynx Race 9.9	1	8.264,38 €	8.264,38 €

Gastos de envío: 0,00 €
IVA: 1.735,52 €

TOTAL: 9999,90 €

Makinon Bikes S.L.
CIF: B-12345678
C. La Boheme, 8, 29006 Málaga
Teléfono: 952 33 95 58
Email: new.makinonbikes@gmail.com

- Modificar los datos del perfil. Los datos del usuario serán recuperados automáticamente desde la base de datos en el formulario y estarían listos para modificar.

Modificar perfil del usuario

Nombre	Teléfono
<input type="text" value="user"/>	<input type="text" value="123456789"/>
Apellidos	Dirección
<input type="text" value="user"/>	<input type="text" value="Calle del user, 2"/>
Email	Código Postal
<input type="text" value="user@user"/>	<input type="text" value="12345"/>
NIF	Ciudad
<input type="text" value="12345678b"/>	<input type="text" value="Cártama"/>

- Cambiar contraseña. Para ello deberá introducir la contraseña actual la cual se comprobará con la almacenada en la base de datos para la validación y la nueva contraseña que deberá coincidir con la confirmación.

Cambiar contraseña

Contraseña actual
<input type="text"/>
Nueva contraseña
<input type="text"/>
Confirmar la nueva contraseña
<input type="text"/>

- Eliminar la cuenta de usuario. En este caso, aunque el usuario elimine su cuenta, sus pedidos y sus facturas no serán eliminadas de las tablas correspondientes en la base de datos, no perdiéndose así el histórico de los mismos.

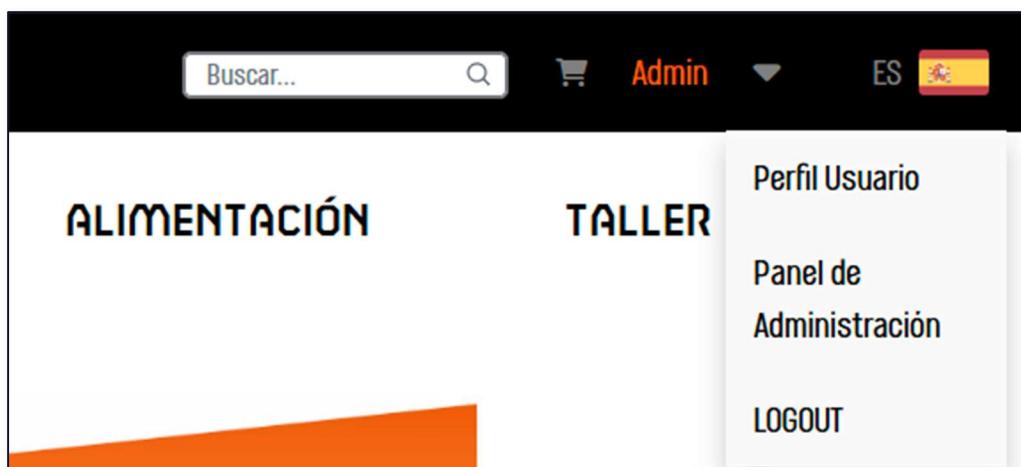
4.4 Rol del Administrador

4.4.1 Flujo de acceso del administrador

Un usuario administrador, al igual que cualquier otro usuario, puede navegar libremente por la aplicación, pero sólo tendrá acceso a las secciones reservadas a los administradores si se logea.

El proceso de login es el mismo que para el usuario, pero se realizará mediante el correo electrónico y la contraseña reservados para tal caso

Una vez logeado el nombre del administrador aparecerá en la barra superior de la página y el dropdown junto al mismo contendrá una nueva sección sólo reservada para los administradores, el Panel de Administración.



4.4.2 Integración y seguridad

- **Servicios Laravel.**

- Laravel maneja la sesión del usuario automáticamente. Cuando un usuario inicia sesión, Laravel creará una nueva sesión y asociará la sesión con el usuario. Las sesiones se almacenan en el servidor, lo que es más seguro que almacenarlas en el lado del cliente.
- Laravel también proporciona una forma sencilla de manejar la autorización. Podemos controlar fácilmente qué usuarios tienen permiso para acceder a ciertas áreas de la aplicación, en este caso hemos aplicado un Middleware denominado *CheckUserRole* y que nos permite comprobar si el usuario logeado es un administrador.

```

  ● ○ ●
1  class CheckUserRole
2  {
3      /**
4      * Middleware que comprueba si el usuario autenticado es admin
5      *
6      * @param \Illuminate\Http\Request $request
7      * @param \Closure $next
8      * @param string $rol
9      */
10     public function handle(Request $request, Closure $next, $rol): Response
11     {
12         Log::info('Middleware CheckUserRole ejecutado');
13
14         if (!Auth::check()) {
15             Log::info('Usuario no autenticado');
16             return redirect('login');
17         }
18
19         $user = Auth::user();
20         Log::info('Usuario autenticado', ['user' => $user]); // Imprime los datos del usuario
21
22         if ($user->rol == $rol) {
23             Log::info('Usuario es admin');
24             return $next($request);
25         }
26
27         // Si el usuario no es admin, redirige a la página principal
28         Log::info('Usuario no es admin');
29         return redirect('/');
30     }
31 }

```

Posteriormente en las rutas usaremos la condición de este middleware para dar acceso, o no, a las secciones de la administración de Makinon Bikes

```

  ● ○ ●
1  /*
2  Rutas para los proveedores
3  */
4  Route::middleware(['auth', 'rol:admin'])->group(function () {
5      Route::get('/listadoProveedores', [ProveedorController::class, 'datosProveedores'])->name('listadoProveedores');
6      Route::get('/fichaProveedor/{id}', [ProveedorController::class, 'fichaProveedor'])->name('fichaProveedor');
7      Route::get('/añadirProveedor', [ProveedorController::class, 'nuevoProveedor'])->name('añadirProveedor');
8      Route::post('/agregarProveedor', [ProveedorController::class, 'agregarProveedor'])->name('agregarProveedor');
9      Route::get('modificarProveedor/{id}', [ProveedorController::class, 'editarProveedor'])->name('modificarProveedor');
10     Route::post('modificarProveedor/{id}', [ProveedorController::class, 'updateProveedor'])->name('updateProveedor');
11     Route::delete('eliminarProveedor/{id}', [ProveedorController::class, 'eliminarProveedor'])->name('eliminarProveedor');
12     Route::get('buscarProveedor', [ProveedorController::class, 'buscarProveedor'])->name('buscarProveedor');
13 });

```

- **Consideraciones de Seguridad**

- HTTPS: Todas las comunicaciones entre el usuario administrador y el servidor se realizan sobre HTTPS para proteger los datos sensibles y la sesión de autenticación.
- Validación del Sesión: Cada solicitud al servidor incluye la validación de la sesión para asegurar que las peticiones sean hechas por un usuario autenticado y evitar accesos no autorizados.

4.4.3 Panel de administración

Una vez correctamente logeado el administrador tendrá acceso al Panel de Administración, desde el cual podrá consultar cualquier tema relacionado con la tienda.

The screenshot shows the 'Panel de Administración' (Admin Panel) interface. It includes a welcome message and a brief description of the panel's purpose. Six main sections are displayed in a grid:

- Proveedores:** Descripción: Desde aquí podrás gestionar los proveedores de la tienda. Botón: Gestión de Proveedores.
- Productos:** Descripción: Desde aquí podrás gestionar los productos de la tienda. Botón: Gestión de Productos.
- Usuarios:** Descripción: Desde aquí podrás gestionar los usuarios de la tienda. Botón: Gestión de Usuarios.
- Ventas:** Descripción: Desde aquí podrás gestionar las ventas de la tienda. Botón: Gestión de Ventas.
- Citas del taller:** Descripción: Desde aquí podrás gestionar las citas del taller. Botón: Gestión de citas.

Desde este panel el administrador podrá:

- Gestionar los proveedores.
- Gestionar los productos.
- Gestionar las marcas.
- Gestionar las ventas de la tienda.
- Gestionar de manera básica las citas del taller.

4.4.3.1 Gestión de proveedores.

Al acceder a la Gestión de Proveedores lo que nos encontramos es un listado con todos los proveedores de la tienda.

Buscar proveedor		Buscar	Mostrar todos					
Nombre	CIF	Dirección	Teléfono	Email	Comentario	Ficha	Editar	Eliminar
Shimano	S12345678	Calle de Shimano, 12	123456789	shimano@bike	Proveedor de componentes de bicicleta			
Team Bike	A12345678	Calle de la Cierva, 12	123456789	team@bike				
Comet	C12345678	Calle de Comet, 12	123456789	comet@bike				
Gobik	G12345678	Calle de Gobik, 12	123456789	gobik@bike	Proveedor de ropa de ciclismo			
KTM Bike	K12345678	Calle de KTM, 12	123456789	ktm@bike				
Focus Bikes	F12345678	Calle de Focus, 12	123456789	focus@bike	Proveedor de bicicletas de alta gama			
BH Bikes	B12345678	Calle de BH, 12	123456789	bh@bike				
Giro	I12345678	Calle de Giro, 12	123456789	giro@bike	Proveedor de cascos de ciclismo			
Castelli	T12345678	Calle de Castelli, 12	123456789	castelli@bike	Proveedor de ropa de ciclismo			
Oakley	O12345678	Calle de Oakley, 12	123456789	okley@glasses	Proveedor de gafas			

Mostrando 1 al 10 de 20 resultados

[Añadir proveedor](#) [Volver](#)

Ficha	Editar	Eliminar

Junto a cada proveedor tenemos tres botones que nos permitirán:

- Ver la ficha con los datos del proveedor, y dentro de ella:
 - Productos suministrados. Nos redirige a una pantalla con todos los productos de dicho proveedor.
 - Modificar datos. Nos redirige a un formulario donde se cargan los datos del proveedor y los cuales podremos modificar.
 - Eliminar proveedor. Eliminaremos todos los datos de ese proveedor.
 - Volver. Nos regresa al listado de proveedores.
- Editar los datos del proveedor.
- Eliminar el Proveedor

Ficha proveedor

Nombre: Shimano

CIF: S12345678

Dirección: Calle de Shimano, 12

Teléfono: 123456789

Email: shimano@bike

Comentario: Proveedor de componentes de bicicleta

[Productos suministrados](#)
[Modificar datos](#)
[Eliminar proveedor](#)
[Volver](#)

Bajo el listado de proveedores encontramos dos nuevos botones:

- Añadir proveedor. Nos lleva a un formulario donde rellenaremos los datos necesarios para añadir un nuevo proveedor.
- Volver. Nos devuelve al Panel de Administración

4.4.3.2 Gestión de Productos.

Al acceder encontramos un listado con todos los artículos de la tienda y sus datos mas importantes.

Listado de artículos

Buscar producto
Buscar
Mostrar todos

Nombre	Tipo	Marca	Proveedor	Stock Total	Precio (€)	Descripción	Richa	Editar	Eliminar
Lynx Race 9.9	Bicicleta de montaña	BH	BH Bikes	1	9999.9	La Lynx es la bicicleta que te hará llegar al podium de todas las carreras...			
Scarp Prime	Bicicleta de montaña	KTM	KTM Bike	0	7899	HACERTE UN LÍDER. El cuadro SCARP EVO II llevó a KTM hasta el podio de la Copa del Mundo....			
IZALCO MAX 8.9	Bicicleta de carretera	Focus	Focus Bikes	0	4799	Las formas refinadas y aprobadas por el túnel de viento NACA (Comité Asesor Nacional de Aeronáutica y Astronáutica)...			
Ridley Noah Fast	Bicicleta de carretera	Ridley	Giro	0	7999	La Noah Fast es la bicicleta de carretera más rápida de Ridley...			
SAM2 6.9	Bicicleta eléctrica	Focus	Focus Bikes	1	5999	La SAM2 6.9 es la bicicleta eléctrica de montaña más polivalente de Focus...			
Ridley Fenix SL	Bicicleta de carretera	Ridley	Ridley	0	1999	La Fenix SL es la bicicleta de carretera más polivalente de Ridley...			
Ridley Helium SLX	Bicicleta de carretera	Ridley	Ridley	2	3999	La Helium SLX es la bicicleta de carretera más ligera de Ridley...			
Ridley Kanzo Fast	Bicicleta de gravel	Ridley	Ridley	2	7999	La Kanzo Fast es la bicicleta de gravel más rápida de Ridley...			
S-Works Epic World Cup	Bicicleta de montaña	Specialized	Comet	1	9999	La S-Works Epic World Cup es la bicicleta de montaña más rápida de Specialized...			
Register MIPS XL	Casco	Abus	Team Bike	5	55	El casco Register™ Mips® XL está diseñado para ciclistas que aprecian el estilo y el ajuste clásico...			

Mostrando 1 al 10 de 124 resultados

< 1 2 3 4 5 6 7 8 9 10 11 12 13 >

Junto a cada producto tenemos tres botones que nos permitirán:

- Ver la ficha con los datos del producto, y dentro de ella:
 - Modificar producto. Nos redirige a una pantalla con todos los datos del producto para modificar el que necesitemos.
 - Eliminar producto. Eliminaremos el producto en concreto.
 - Volver. Nos regresa al listado de productos.
- Editar los datos del producto.
- Eliminar el producto.

Ficha de producto



Nombre: Lynx Race 9.9
Tipo: Bicicleta de montaña
Marca: BH
Proveedor: BH Bikes
Talla:
Precio: 9999.9 €
Descripción: La Lynx es la bicicleta que te hará llegar al podium de todas las carreras
Descripción larga: LIGEREZA Construida como todos los cuadros de alta gama de BH utilizando la tecnología de moldeado de carbono HCIM- Hollow Core Internal Molding que permite reducir al máximo el peso del cuadro y la bieleta. Con la mayor cantidad de fibras de alto módulo, EC Layup, para conseguir...
Ruta imagen: lynx99.webp

[Modificar producto](#) [Eliminar producto](#) [Volver](#)

Y al igual que en el caso de los proveedores, encontraremos, bajo el listado de productos un botón que nos permitirá añadir un nuevo producto en la base de datos.

En el formulario que encontramos para añadir un nuevo producto tendremos que cumplimentar:

- Nombre.
- Tipo de producto.
- Marca, de no existir podemos seleccionar la posibilidad de añadir una nueva.
- Proveedor.
- Stock
- Color, donde también podremos crear uno nuevo en caso de no existir el que corresponde al nuevo producto.

- Talla.
- Precio
- Descripción corta
- Descripción larga
- Subir una imagen del producto desde nuestro dispositivo y que se guardará en la carpeta `public/images`

4.4.3.3 Gestión de Usuarios

Tenemos una vista muy parecida a las anteriores, un listado de todos los usuarios registrados en la web. Las funcionalidades de los botones junto a los nombres de cada usuario también son similares a las anteriores, pudiendo ver la ficha del usuario, modificarlo o eliminarlo. Y también tenemos el botón por si desde la administración queremos registrar un usuario y por ejemplo asignarle un rol de administrador.

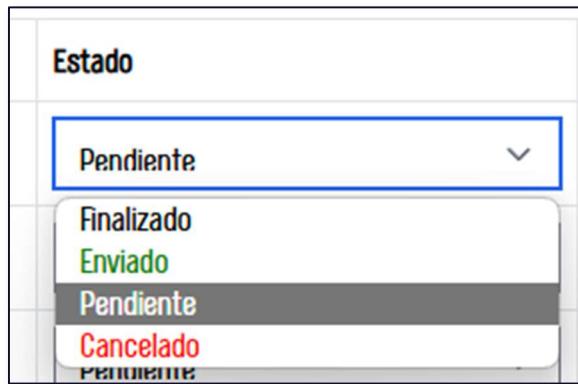
Registro de usuario por administración

Nombre	NIF
<input type="text"/>	<input type="text"/>
Apellidos	Teléfono
<input type="text"/>	<input type="text"/>
Correo electrónico	Dirección
<input type="text"/>	<input type="text"/>
Contraseña	Código Postal
<input type="text"/>	<input type="text"/>
Rol	Ciudad
<input type="text" value="Administrador"/>	<input type="text"/>
<input type="button" value="Añadir usuario"/> <input type="button" value="Volver"/>	

4.4.3.4 Gestión de Ventas

En el listado que nos devuelve esta vista tenemos todos los pedidos de la tienda, junto al nombre del usuario que lo realizó y el importe total del mismo. Pero además tenemos un desplegable en cada línea de pedido que nos permite modificar el estado del pedido:

- *Pendiente*. Que el pedido está pendiente de enviar.
- *Enviado*. El pedido ya ha sido enviado.
- *Finalizado*. El cliente ya ha recibido su pedido.
- *Cancelado*. Si el pedido se ha cancelado por algún motivo.



Si hacemos click sobre el botón *Detalle* nos lleva a una nueva vista donde veremos que productos componen ese pedido y la cantidad de cada uno, y si queremos además podremos ver la factura correspondiente a dicho pedido.

Detalle del pedido #5

Producto	Cantidad	Precio	Subtotal
Scarp Prime	1	7899 €	7899 €
Reverb AXS 31.6	1	885 €	885 €

[Volver](#)
[Ver factura](#)

4.4.3.5 Gestión de citas del taller

Aquí veremos:

- Fecha de la cita
- Hora
- Cliente
- Opción elegida por el cliente para su servicio de taller.
- Comentario sobre lo que quiere hacer en concreto
- Imagen. Si el usuario lo desea, puede adjuntar a la solicitud de cita de taller una imagen de su bicicleta o del problema que tiene, y que nosotros podremos ver gracias a una ventana modal que sea abrirá y mostrará la imagen
- Estado de la cita. Donde nosotros mismos podemos modificarla.

Listado de citas						
Fecha ▲▼	Hora	Cliente	Opción ▲▼	Comentario del cliente	Imagen*	Estado ▲▼
2024-05-28	18:56:00	Usuario no encontrado	Montar componente	manillar		Pendiente ▾
2024-05-31	19:00:00	Usuario no encontrado	Revision bicicleta	REvision completa		Pendiente ▾
2024-05-31	17:00:00	Usuario no encontrado	Montar componente	púños	images/clientes_taller/1716768957.png	Pendiente ▾
2024-05-31	12:00:00	usuarioTaller@usuario usuario	Revision suspensiones	amortiguador		Pendiente ▾

[Ver en calendario](#) [Volver](#)



Y si pulsamos sobre el botón Ver en calendario nos redirige hacia la aplicación Taller Makinon Bikes donde tendremos mas información de las citas y estadísticas de las mismas.



4.5 Implementaciones

Aunque ya hemos hablado de ellas en apartados anteriores, dedicamos este nuevo apartado a tres implementaciones que se han añadido al proyecto con el fin de complementar el mismo. Por un lado tenemos el uso de la biblioteca Livewire, por otro lado el uso de la biblioteca Mail y por último la biblioteca DomPDF para poder descargar la factura en pdf.

4.5.1 Livewire

Livewire es una biblioteca full-stack para Laravel que facilita la creación de componentes de interfaz de usuario dinámicos en aplicaciones web modernas. Su enfoque principal es simplificar el desarrollo de aplicaciones dinámicas, eliminando la necesidad de escribir y mantener una gran cantidad de código JavaScript.

Esto nos permite crear componentes que pueden ser actualizados en tiempo real, pero todo el procesamiento se hace en el servidor. Utiliza una combinación de AJAX y la funcionalidad de

Blade (el motor de plantillas de Laravel) para manejar actualizaciones de componentes y mantener el estado del componente en el servidor

En Makinon Bikes hemos utilizado esta característica en varias vistas de la aplicación, pero sobre todo en:

1. Buscador de productos para el usuario. De modo que sólo es el listado el que se recarga cuando se hace una búsqueda y no la página completa.



2. Buscador en los diferentes listados que aparecen en la sección de administración. Permitiéndonos un menor tiempo de respuesta en las búsquedas y renderización de la vista, ya que sólo cambia el listado



3. Sistema de ordenación del listado de citas de taller. Nos permite ordenar las citas por días, opción de servicio que necesita el cliente o estado de las citas.

Listado de citas						
Fecha ▲▼	Hora	Cliente	Opción ▲▼	Comentario del cliente	Imagen*	Estado ▲▼
2024-05-28	18:56:00	Usuario no encontrado	Montar componente	manillar		Pendiente

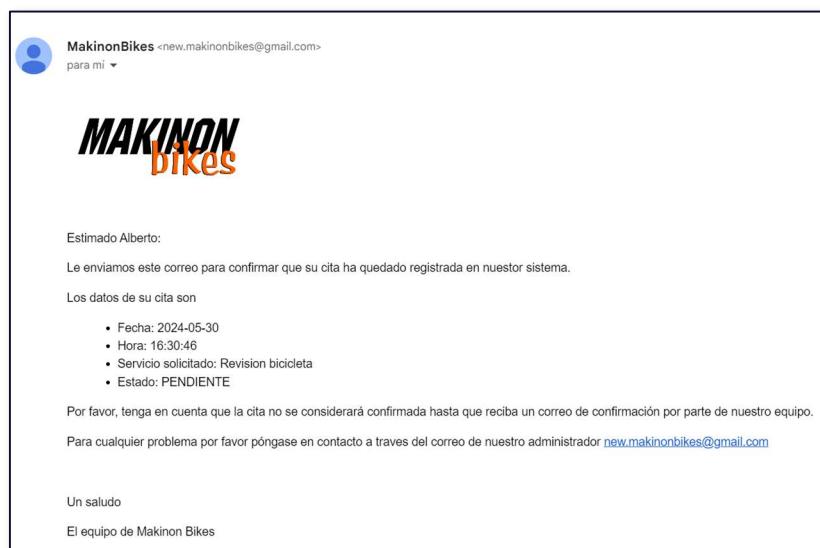
Ventajas de su uso, y por qué hemos querido implementarla, en primer lugar la simplicidad, permite escribir interfaces de usuario dinámicas sin necesidad de dominar JavaScript, en segundo lugar por su integración con Laravel aprovechando características como validaciones, rutas y controladores, y en tercer lugar por su reactividad, puesto que proporciona una experiencia de usuario más fluida y reactiva sin la complejidad de los frameworks front-end como Vue.js o React.

4.5.2 Mail

La biblioteca Mail de Laravel es una poderosa y flexible herramienta que permite enviar correos electrónicos fácilmente desde aplicaciones Laravel.

Está diseñada para manejar todas las tareas comunes relacionadas con el envío de correos electrónicos, como la construcción de mensajes utilizando el motor de plantillas de Blade, el manejo de archivos adjuntos, y la integración con múltiples servicios de envío de correos electrónicos como SMTP, Mailgun, Postmark, Amazon SES, entre otros.

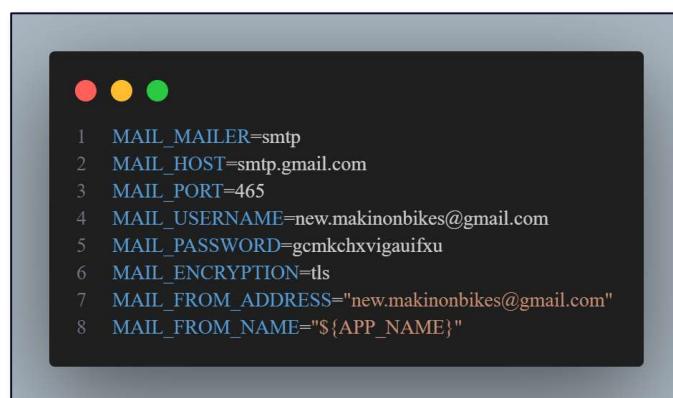
Pero además permite el envío de correos electrónicos de forma asíncrona usando colas de trabajo, de modo que el rendimiento de la aplicación no se vería mermado.



¿Dónde hemos implementado este servicio?

- Correo de bienvenida
 - Recuperación de contraseña
 - Confirmación de pedido realizado
 - Confirmación de cita del taller o modificación de la misma.

Para ello, aparte de importar la biblioteca correspondiente, crear su controlador, sus vistas y demás hemos configurado el archivo `.env` para que haga uso de este servicio mediante `stmp`, de modo que aprovechamos el correo Gmail de la cuenta de la web para enviar los mails.



Otro uso de esta biblioteca es que nos permite recibir, en nuestro correo electrónico, las consultas que hagan los usuarios desde el formulario de contacto de la aplicación.

Contacto

Información de contacto

Si tienes alguna duda o pregunta, no dudes en ponerte en contacto con nosotros. Estaremos encantados de ayudarte.

Si lo prefieres, puedes visitarnos en nuestra tienda física en la siguiente dirección:

Calle La Boheme, 8
29006 Málaga
Teléfono: 952 33 95 58
Horario de 10:00 a 14:00 y de 17:00 a 20:00h de lunes a viernes, sábados de 10:00 a 14:00

Nombre

Email

Si lo deseas déjanos tu teléfono

Mensaje

4.5.3 Biblioteca DomPDF

DomPDF es una biblioteca PHP que convierte HTML y CSS a archivos PDF. Es ampliamente utilizada para generar documentos PDF dinámicos en aplicaciones web y por ello la hemos implementado con el fin de que el usuario desde su panel pueda descargarse en formato PDF la factura correspondiente a un pedido que haya realizado.

Cabe destacar que para generar esta factura descargable hemos tenido que crear un nuevo documento HTML, con su estilo dentro del <head>, más sencillo y basado en el sistema de tablas, puesto que de momento DomPDF presenta muy poca compatibilidad con CSS3 y sus nuevas características, o con Bootstrap.

Aun así, hemos conseguido que, si el cliente de Makinon quiere tener una factura de su pedido en su ordenador o en su dispositivo móvil, pueda tenerla.

Makinon Bikes			
FACTURA		CLIENTE	
Nº Factura: 5 Pedido: 5 Fecha: 2024-06-02			Alberto Moreno NIF: 12345678a Calle del administrador, 1, 12345, Cártama 123456789 alberto@moreno
PRODUCTO			CANTIDAD
Barrita SIS GO Energy sabor plátano y caramelo 40 g	1	1.28 €	1.28 €
Gastos de envío:			16.53 €
21% IVA:			3.74 €
TOTAL:			21.55 €
Makinon Bikes S.L. CIF: B-12345678 C. La Boheme, 8, 29006 Málaga Teléfono: 952 33 95 58 Email: new.makinonbikes@gmail.com			

4.6 Despliegue

La web de Makinon Bikes se aloja en una instancia EC2 de Amazon Web Service, la elección de la misma ha sido gracias a las ventajas que nos propone Amazon en este tipo de servicios como por ejemplo la flexibilidad de seleccionar un sistema operativo, el lenguaje de programación, elegir su tamaño, posibilidad de añadir un balanceador de carga o la misma base de datos.

La secuencia de pasos que hemos llevado a cabo para el despliegue han sido los siguientes:

1.- Acceder al panel EC2

2.- Configurar el grupo de seguridad que regirá las reglas de tráfico entrante para nuestra instancia.

Configurar las reglas de entrada, principalmente para los puertos:

- 22 – Permite la conexión mediante SSH
- 80 – Permite las conexiones entrantes mediante solicitudes HTTP
- 443 – Permite las conexiones entrantes mediante solicitudes HTTPS

Configurar las reglas de salida:

- En este caso no tenemos que configurar nada puesto que vamos a permitir todo el tráfico de salida desde nuestra web

Detalles																																																																																								
Nombre del grupo de seguridad	ID del grupo de seguridad	Descripción	ID de la VPC																																																																																					
MyMakinonbikesSecurity	sg-0018bac9a1f04ae63	Grupo de seguridad de Makinon Bikes	vpc-04ec052c81dd52362																																																																																					
Propietario	527326672295	Número de reglas de entrada	4 Entradas de permiso																																																																																					
Reglas de entrada		Reglas de salida		Etiquetas																																																																																				
Reglas de entrada (4)																																																																																								
<table border="1"> <thead> <tr> <th colspan="10">Reglas de entrada (4)</th> </tr> <tr> <th colspan="10"> <input type="button" value="C"/> Administrar etiquetas <input type="button" value="Editar reglas de entrada"/> < 1 > @ </th> </tr> <tr> <th colspan="10"> <input type="text" value="Buscar"/> </th> </tr> </thead> <tbody> <tr> <th>Name</th> <th>ID de la regla del gr...</th> <th>Versión de IP</th> <th>Tipo</th> <th>Protocolo</th> <th>Intervalo de puertos</th> <th>Origen</th> <th>Descripción</th> <th colspan="2"></th> </tr> <tr> <td>-</td> <td>sgr-0df4a29fd5631d479</td> <td>IPv4</td> <td>TCP personalizado</td> <td>TCP</td> <td>8090</td> <td>0.0.0.0/0</td> <td>-</td> <td colspan="2"></td> </tr> <tr> <td>-</td> <td>sgr-0e67b25de811edf96</td> <td>IPv4</td> <td>SSH</td> <td>TCP</td> <td>22</td> <td>0.0.0.0/0</td> <td>-</td> <td colspan="2"></td> </tr> <tr> <td>-</td> <td>sgr-03a20ae7107f16ac0</td> <td>IPv4</td> <td>HTTPS</td> <td>TCP</td> <td>443</td> <td>0.0.0.0/0</td> <td>-</td> <td colspan="2"></td> </tr> <tr> <td>-</td> <td>sgr-0e15985d85461b...</td> <td>IPv4</td> <td>HTTP</td> <td>TCP</td> <td>80</td> <td>0.0.0.0/0</td> <td>-</td> <td colspan="2"></td> </tr> </tbody> </table>									Reglas de entrada (4)										<input type="button" value="C"/> Administrar etiquetas <input type="button" value="Editar reglas de entrada"/> < 1 > @										<input type="text" value="Buscar"/>										Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción			-	sgr-0df4a29fd5631d479	IPv4	TCP personalizado	TCP	8090	0.0.0.0/0	-			-	sgr-0e67b25de811edf96	IPv4	SSH	TCP	22	0.0.0.0/0	-			-	sgr-03a20ae7107f16ac0	IPv4	HTTPS	TCP	443	0.0.0.0/0	-			-	sgr-0e15985d85461b...	IPv4	HTTP	TCP	80	0.0.0.0/0	-		
Reglas de entrada (4)																																																																																								
<input type="button" value="C"/> Administrar etiquetas <input type="button" value="Editar reglas de entrada"/> < 1 > @																																																																																								
<input type="text" value="Buscar"/>																																																																																								
Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción																																																																																	
-	sgr-0df4a29fd5631d479	IPv4	TCP personalizado	TCP	8090	0.0.0.0/0	-																																																																																	
-	sgr-0e67b25de811edf96	IPv4	SSH	TCP	22	0.0.0.0/0	-																																																																																	
-	sgr-03a20ae7107f16ac0	IPv4	HTTPS	TCP	443	0.0.0.0/0	-																																																																																	
-	sgr-0e15985d85461b...	IPv4	HTTP	TCP	80	0.0.0.0/0	-																																																																																	

3.- Creación de la instancia

Creamos una instancia nueva con una AMI basada en Ubuntu 24.04 con las siguientes características:

- Tipo de instancia t2.medium lo cual nos otorga 4GiB de memoria.
- Par de claves seleccionadas: `vokey.pem`
- Para el grupo de seguridad elegimos el que configuramos en el paso anterior.
- Y para el almacenamiento hemos elegido 15Gb por el tema de las fotos de la web y de los clientes del taller.

Resumen de instancia de i-0b1697720883a8247 (Makinon Bikes)		Información				
Se ha actualizado hace 10 minutes						
ID de la instancia		Dirección IPv4 pública		Direcciones IPv4 privadas		
	i-0b1697720883a8247 (Makinon Bikes)	54.172.113.86 dirección abierta		172.31.53.202		
Dirección IPv6	-	Estado de la instancia		DNS de IPv4 pública		
	-	En ejecución		ec2-54-172-113-86.compute-1.amazonaws.com dirección abierta		
Tipo de nombre de anfitrión		Nombre DNS de IP privada (solo IPv4)		Direcciones IP elásticas		
Nombre de IP: ip-172-31-53-202.ec2.internal		ip-172-31-53-202.ec2.internal		54.172.113.86 [IP pública]		
Responder al nombre DNS de recurso privado		Tipo de instancia		Hallazgo de AWS Compute Optimizer		
IPv4 (A)	-	t2.medium		Suscribirse a AWS Compute Optimizer para recibir recomendaciones.		
Dirección IP asignada automáticamente	-	ID de VPC		Más información		
	-	vpc-04ec052c81dd52362				
Rol de IAM		ID de subred		Nombre del grupo de Auto Scaling		
-		subnet-012e56504adb927c4		-		
IMDSv2		Instance ARN				
Required		arn:aws:rds:us-east-1:527326672295:instance/i-0b1697720883a8247				

4.- Creación de la Base de Datos

Haciendo uso del servicio RDS que nos proporciona AWS creamos una base de datos donde posteriormente volcaremos los datos de los seeders de Laravel.

Para nuestra base de datos hemos hecho uso de un motor MySQL, en cuanto a las credenciales hemos preferido ser nosotros los que creemos nuestra propia contraseña, de tamaño elegimos la t3.micro de 1GiB de RAM, 20 GiB de almacenamiento y hemos permitido la conexión a nuestra instancia Makinon Bikes.

Resumen					
Identificador de base de datos makinonbikes-db	Estado Disponible	Rol Instancia	Motor MySQL Community	Recomendaciones 2 Informativo	
CPU 2.92%	Clase db.t3.micro	Actividad actual 0 Conexiones	Región y AZ us-east-1e		
Conectividad y seguridad					
Conectividad y seguridad					
Punto de enlace y puerto	Redes	Seguridad			
Punto de enlace makinonbikes-db.cmxhk0o3kixd.us-east-1.rds.amazonaws.com	Zona de disponibilidad us-east-1e	Grupos de seguridad de la VPC rds-ec2-1 (sg-036bffff6e27f73f) Activo			
Puerto 3306	VPC vpc-04ec052c81dd52362	Accesible públicamente No			
	Grupo de subredes rds-ec2-db-subnet-group-1	Entidad de certificación rds-ca-rsa2048-g1			
	Subredes subnet-0e477fe7d409acf9 subnet-099125bd4151d535a subnet-0efbd36dd8c5adbc subnet-0234565fd387473e2 subnet-089e10c73c408dec4	Fecha de la entidad de certificación May 26, 2061, 01:34 (UTC+02:00)			
	Tipo de red IPv4	Fecha de expiración del certificado de instancia de base de datos May 27, 2025, 23:44 (UTC+02:00)			

5.- Asignar una IP elástica a nuestra instancia.

6.- Registrar el dominio en Duckdns.org

Utilizaremos la dirección IP elástica que hemos asociado a nuestra instancia para registrar el nombre y el dominio.

7.- Acceso a la instancia mediante SSH

Después de haber creado y configurado el Host de SSH

8.- Instalación de apache2 en la instancia

Así como de PHP y todas las dependencias necesarias

9.- Copiado de la carpeta del proyecto de Laravel

La copia la realizamos en la nueva carpeta que hemos creado en /var/www a la que previamente le hemos dado permisos de escritura

10.- Creación del archivo makinonbikes.conf

Concretamente en la carpeta /etc/apache2/sites-availables

11.- Modificación del archivo dir.conf

De modo que lo primero a lo que apunte sea a un index.php y no al index.html

12.- Habilitamos el archivo makinonbikes.conf

En este punto si accedemos a nuestra dirección http podríamos ver la portada de la web

13.- Configuraremos el archivo .env

De este modo tendremos acceso a la base de datos a través de la instancia EC2

```
APP_NAME=MakinonBikes
APP_ENV=local
APP_KEY=base64:Om8Zb19u5f
APP_DEBUG=true
APP_URL=http://www.makinonbikes.duckdns.org

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=makinonbikes-db.cmxhk0o3kixd.us-east-1.rds.amazonaws.com
DB_PORT=3306
DB_DATABASE=makinonbikes-db
DB_USERNAME=admin
DB_PASSWORD=.
```

14.- Hacer la migración y el poblado de la base de datos creada.**15.- Instalar el certificado SSL**

Utilizando la herramienta Certbot e instalando el certificado podremos acceder a Makinon Bikes mediante el protocolo HTTPS



<https://www.makinonbikes.duckdns.org>

4.7 Documentación API Rest

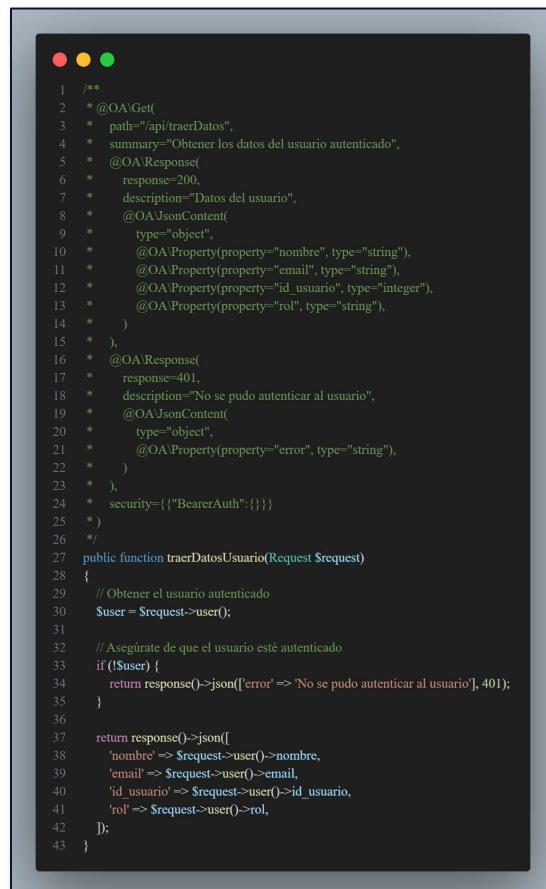
Para documentar nuestra API Rest hemos optado por utilizar Swagger por los siguientes motivos:

Utilizar Swagger para generar la documentación de una API en Laravel ofrece varios beneficios significativos, que se pueden agrupar en varias categorías clave:

- Swagger proporciona un formato estandarizado para la documentación de API.
- Generación Automática de la documentación. Esto también reduce el riesgo de que la documentación esté desactualizada o sea incorrecta.
- Swagger UI nos proporciona una interfaz gráfica interactiva para probar la API directamente desde el navegador mediante ejemplos de solicitudes y respuestas
- El paquete l5-swagger nos permite integrar Swagger fácilmente con Laravel.

Para implementar Swagger con l5-swagger en Laravel hemos seguido los siguientes pasos:

1. Instalamos del paquete mediante el comando `composer require "darkaonline/l5-swagger"`
2. Publicamos la configuración del paquete con `php artisan vendor:publish --provider "L5Swagger\L5SwaggerServiceProvider"`
3. Agregamos las anotaciones a los controladores usando anotaciones Swagger en los comentarios de nuestros métodos para describir los endpoints de nuestra API.



```

1 /**
2  * @OA\Get(
3  *     path="/api/traerDatos",
4  *     summary="Obtener los datos del usuario autenticado",
5  *     @OA\Response(
6  *         response=200,
7  *         description="Datos del usuario",
8  *         @OA\JsonContent(
9  *             type="object",
10 *             @OA\Property(property="nombre", type="string"),
11 *             @OA\Property(property="email", type="string"),
12 *             @OA\Property(property="id_usuario", type="integer"),
13 *             @OA\Property(property="rol", type="string"),
14 *         )
15 *     ),
16 *     @OA\Response(
17 *         response=401,
18 *         description="No se pudo autenticar al usuario",
19 *         @OA\JsonContent(
20 *             type="object",
21 *             @OA\Property(property="error", type="string"),
22 *         )
23 *     ),
24 *     security={"BearerAuth":{}}
25 * )
26 */
27 public function traerDatosUsuario(Request $request)
28 {
29     // Obtener el usuario autenticado
30     $user = $request->user();
31
32     // Asegúrate de que el usuario esté autenticado
33     if (!$user) {
34         return response()->json(['error' => 'No se pudo autenticar al usuario'], 401);
35     }
36
37     return response()->json([
38         'nombre' => $request->user()->nombre,
39         'email' => $request->user()->email,
40         'id_usuario' => $request->user()->id_usuario,
41         'rol' => $request->user()->rol,
42     ]);
43 }

```

4. Generamos la documentación mediante `php artisan l5-swagger:generate`
5. Finalmente accedemos a la documentación que podremos encontrar en <http://localhost:8000/api/documentation> si es de manera local o bien <https://www.makinonbikes.duckdns.org/api/documentation> si es a través de la página desplegada en el servidor. Aquí podremos ver todos nuestros endpoints y los métodos que cada uno utiliza.

API de Makinon Bikes 1.0.0

<http://localhost:8000/docs/api-docs.json>

Esta es la API para gestionar las citas en el taller de Makinon Bikes

[Contact the developer](#)

[Authorize !\[\]\(88562f02c39ac1e45de2a489e457c2ee_img.jpg\)](#)

Filter by tag

default 

<code>GET /api/traerDatos</code>	Obtener los datos del usuario autenticado	 
<code>GET /api/cita_taller</code>	Obtener todas las citas del taller de un cliente	 
<code>POST /api/cita_taller</code>	Crea una nueva cita en el taller	 
<code>DELETE /api/cita_taller</code>	Elimina una cita del taller	 
<code>POST /api/editarCitaUsuario</code>	Editar una cita de la bdd	 
<code>GET /api/obtenerCitaId/{id_cita_taller}</code>	Obtener una cita por su ID	 
<code>PUT /api/citaPendiente</code>	Edita el estado de varias citas del taller desde el panel del administrador de Angular	 
<code>GET /api/calendario</code>	Obtiene todas las citas del taller	 
<code>PUT /api/subir_imagen</code>	Sube una imagen para una cita del taller	 

5 – Bibliografía

Laravel - [Documentación oficial de Laravel](#)

Livewire - [Documentación oficial Livewire](#)

DomPDF Laravel - [Github Barryvdh para la generación de documentos PDF](#)

En Laravel: ¿Cómo enviar correos? –

[Tutorial para enviar correos desde Laravel](#)

W3Schools - [Página oficial W3School](#)

TailWind CSS Component - [Página sobre los componentes de Tailwind](#)

Bootstrap - [Documentación oficial Bootstrap en español](#)

Swagger - [Documentación oficial Swagger](#)

Cómo documentar nuestras APIs con Swagger –

[Tutorial para generar documentación de una API mediante Swagger](#)

Cómo documentar una API en Laravel usando Swagger –

[Tutorial como documentar una API en Laravel usando Swagger](#)

Angular - [Documentación oficial](#)

Rutas- [Aprender rutas en Angular 17](#)

Rutas hijas- [Aprender rutas hijas en Angular 17](#)

Componentes Angular- [Documentación de Angular material](#)

Servicios Angular- [Relación entre componentes a través de servicios con Angular](#)

Sesiones- [Configuración de sesiones en Angular](#)

Netlify- [Documentación para el despliegue en Netlify con Angular](#)

Threejs- [Documentación oficial](#)

CSS cards - [Como crear efecto hover para los cards](#)

CSS menú- [Menú fijo con CSS](#)

CSS menú hamburguesa- [Cómo hacer un menú hamburguesa en responsive](#)

MAKINON
bikes