



Miguel Ribera Autocares S.L

Santiago Canet Ribera

Proyecto Desarrollo de Aplicaciones Web (DAW) 2024



Datos del proyecto

Datos del alumno:

Nombre y apellidos	Santiago Canet Ribera
NIF/NIE	20942121T
Curso y CF	2º DAW

Datos del proyecto

Título del proyecto	Web Autocares Miguel Ribera
Nombre del tutor individual	Josep Magraner i Ramón
Nombre del tutor del grupo	José Fidel Oltra Landete
Resumen	Web Autocares Miguel Ribera es una aplicación que permitirá a los usuarios encontrar la parada más cercana y poder comprar los billetes, además de proporcionar información como la historia de la empresa o el horario de los autobuses y la posibilidad de alquilarlos.
Abstract	Web Autocares Miguel Ribera is an application that will allow users to find the nearest stop and purchase tickets, as well as provide information such as the company's history, bus schedules, and the possibility of renting them.
Módulos Implicados	<ul style="list-style-type: none"> ● Desarrollo WEB en entorno Servidor ● Desarrollo WEB en entorno Cliente ● Desarrollo de Interfaces Web ● Despliegue de aplicaciones WEB
Fecha de presentación	3 Junio 2024

Índice

Índice

1. Introducción

- 1.1 Descripción del proyecto
 - Captura de la Web antes de su edición
- 1.2 Objetivos
- 1.3 Tecnologías

1. Análisis

- 2.1 El sistema actual
 - Web del dominio
 - Web de Wix

¿Qué necesita la web?

3. Diseño de la solución

Diseño:

4. Desarrollo

Frontend

El mapa:

Configuración de los Markers de las estaciones del mapa

El QR:

Configuración del QR

texts.json

language.service

Cookies

Backend

Resultado

Inicio:

Horario:

Servicio Privado:

Sobre Nosotros:

Iniciar Sesión:

Crear Cuenta:

Perfil:

Pago:

5. Cambios a futuro



Cambios que no he podido solucionar:

6. Conclusión

1. Introducción

1.1 Descripción del proyecto

Este proyecto surge como propuesta por parte de mi familia, ya que hasta ahora las páginas que teníamos eran muy básicas y poco útiles, ya que apenas ayudaban al usuario, por ello decidí que sería una buena idea aprovechar la ocasión para hacer por mi mismo la web y así contribuir al negocio familiar.



Captura de la Web antes de su edición

1.2 Objetivos

El objetivo principal es proporcionar al usuario una cómoda experiencia que facilite tanto la adquisición de billetes como el alquiler de un autobús entre otras cosas, los requisitos eran los siguientes:

- Un mapa en el que poder localizar tanto las paradas como la posición del autobús.
- Un formulario para poder comprar los billetes.
- Una sesión para poder guardarlos y gestionarlos.
- La posibilidad de contactar con la empresa para alquilar los servicios privados.

1.3 Tecnologías

Este proyecto ha sido desarrollado usando las siguientes tecnologías:



Visual Studio Code: He elegido VSCode como mi IDE, ya que llevo usándolo desde hace 4 años, por lo que estoy bastante familiarizado con él y lo tengo configurado a mi gusto.



Angular: El framework que he usado para el Frontend es Angular, lo he elegido ya que es el que he usado durante las prácticas y me ha parecido fácil y cómodo de usar.



Node JS: He usado NodeJS para el Backend a pesar de que en un inicio tenía pensado usar PHP junto con Symphony.



Firebase: es la plataforma que he usado para gestionar las bases de datos, las cuales funcionan por colecciones JSON muy similares a MongoDB y el hosting de imágenes y PDFs.



Tailwind: A pesar de que estuviera acostumbrado a usar Bootstrap, decidí usar Tailwind como framework CSS debido a que ofrece más libertad a la hora del diseño.



Github: He usado GitHub para el control de versiones, debido a que es una herramienta a la que estoy acostumbrado, además de ofrecer muchas ventajas como la rapidez y la fiabilidad



Bootstrap-Icons: A la hora de implementar iconos con, me he decidido por Bootstrap Icons por ser el mejor gestor de Iconos gratuito que he probado, en el he podido encontrar todos los que necesitaba

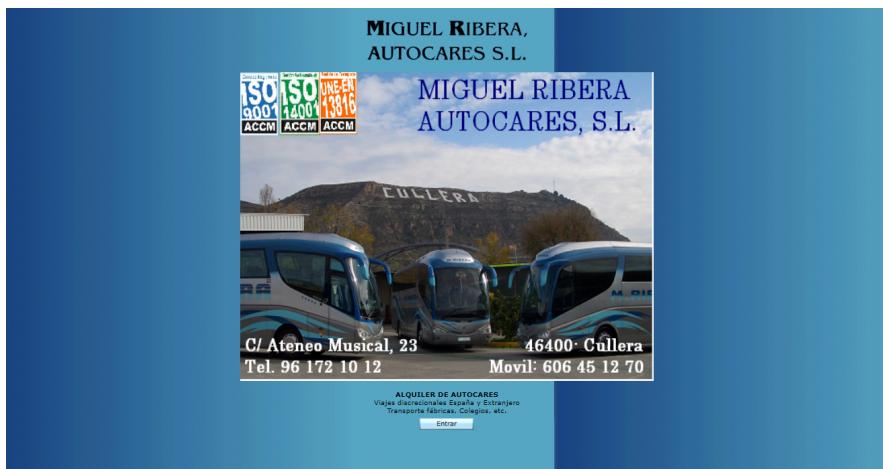


Vercel: He usado Vercel para desplegar la Web ya que es muy simple y ya lo tenía configurado además de que el paquete gratuito es bastante generoso.

1. Análisis

2.1 El sistema actual

Actualmente tenemos 2 webs, una hecha mediante HTML y CSS, pero es completamente estática por lo que es poco útil, además de estar desactualizada, esta tiene el dominio que usa mi web



Web del dominio



Web Autocares Miguel Ribera. Santiago Canet Ribera

La otra, es una página con Wix, la cual sí que es algo más interactiva, pero no tan útil, además de estar desactualizada por lo que muchos servicios no funcionan correctamente

Esta página web se diseñó con la plataforma **WIX**.com. Crea tu página web hoy. [Comienza ya](#)

Inicio Empresa Servicios Horarios Contacto Anúnciese Términos de Uso

Autocares Miguel Ribera S.L. Viajes discrecionales España y extranjero, transporte fábricas, colegios, etc.

Contacte con nosotros

Gestión Medioambiental
ISO 14001
ACCM

<https://autocaresribera.wixsite.com/autocaresribera/contactanos>

Web de Wix

Con mi proyecto, se cubren todas las necesidades de la empresa, además de estar completamente actualizado a las últimas tecnologías con una interfaz más amigable para el usuario.

¿Qué necesita la web?

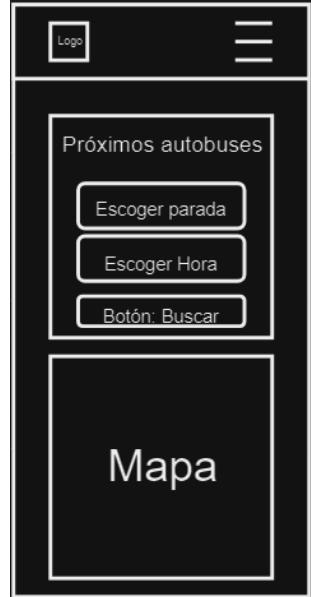
La web necesita implementar los siguientes servicios

- Un mapa en el que poder localizar tanto las paradas como la posición del autobús.
- Un formulario para poder comprar los billetes.
- Una sesión para poder guardarlos y gestionarlos.
- La posibilidad de contactar con la empresa para alquilar los servicios privados.

Lo único que no he cumplido ha sido el gestionar la posición de los autobuses, ya que por ahora no tenemos los recursos para obtenerla.

3. Diseño de la solución

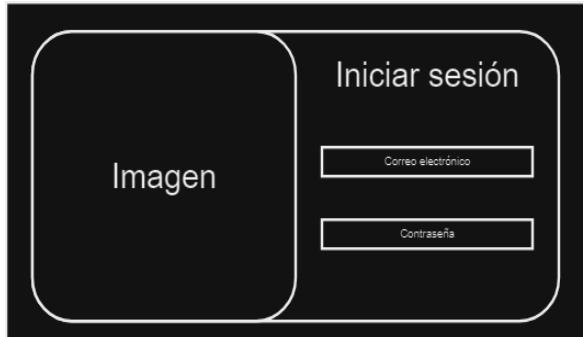
Diseño:

	Descripción	PC	Móvil
Inicio	<p>Esta es la página de Inicio, donde accede el usuario, en ella podrá ver tanto las paradas como los autobuses además de buscar qué autobús le conviene y acceder a la compra de billetes</p>		



Web Autocares Miguel Ribera. Santiago Canet Ribera

Horario	<p>Página simple en la que consultar los horarios de los autobuses, puede ser una foto o una tabla</p>		
Servicio Privado	<p>Pequeño formulario para el alquiler de autobuses el cual se almacena en la BBDD</p>		

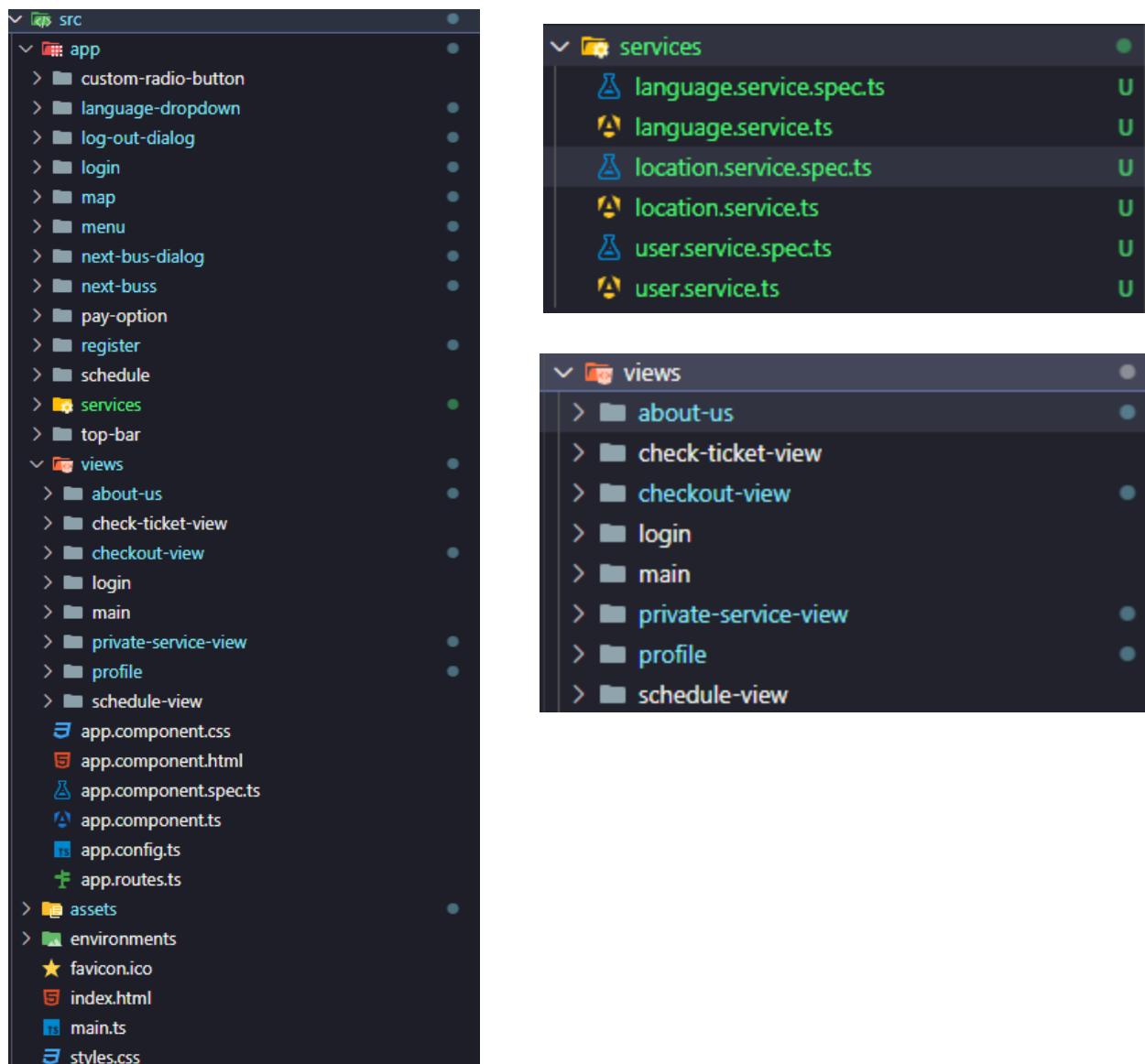
Sobre Nosotros	Texto con la historia de la empresa así como unas cuantas imágenes con fines estéticos		
Iniciar Sesión	Página de Inicio de sesión la cual usa Correo y contraseña para el acceso, mandandolos cifrados gracias a cryptoJS que contacta con el backend mediante la API donde se desencripta la contraseña y se comprueba si coincide		

Crear Cuenta	<p>Página de registro donde el usuario podrá crear una cuenta rellenando el formulario, se comprueba que siga los requisitos (Email bien escrito, DNI correcto y la contraseña tiene al menos 8 dígitos Mayúsculas minúsculas y un número y coinciden)</p>		
Perfil	<p>Pequeña ventana donde el usuario puede comprobar sus viajes restantes además de usarla tener el QR para poder usar un billete (El DNI servirá para comprobar que ese es el usuario legítimo)</p>		

4. Desarrollo

Frontend

El Frontend se basa en la estructura básica de Angular añadiendo una carpeta Views donde se encuentran todas las vistas y Service donde se encuentran los servicios encargados de mantener y compartir los datos entre los componentes.



El mapa:

He usado TomTom ya que es una alternativa gratuita de Google Maps para el mapa, la configuración ha sido bastante simple, aunque me ha dado algunos problemas, el principal ha sido a la hora de customizar los “Markers” de las estaciones y el usuario, para solucionarlo opte por sustituir los que vienen por defecto por unos iconos customizados.

```
setStationsMarkers() {
  fetch('https://miguel-ribera-backend-firebase.onrender.com/stations') ×3 ↵ 200 https://miguel-ribera-backend-
  .then((res) => res.json()) ×3 ↵ 200 https://miguel-ribera-backend-firebase.onrender.com/stations | @ vi-tr.
  .then((res) => {
    res.forEach((station: any) => {
      this.stations.push(station);

      var stationPopup = new tt.Popup().setHTML(
        `<div class="flex flex-col justify-center">
          <div class="flex justify-center items-center rounded w-full overflow-hidden">
            <img class='rounded-3xl w-20 h-20' src='${station.image}' alt='' />
            <h3 class='text-xl p-5'>${station.address}</h3>
          </div>
        </div>
      );
      stationPopup.on('open', () => {
        this.locationService.changeStation(station.address);
      });

      var stationMarker = document.createElement('div');
      stationMarker.className = 'custom-marker';
      stationMarker.style.backgroundImage =
        "url(' ../../assets/station-location.png')";
      new tt.Marker({ element: stationMarker })
        .setLngLat([parseFloat(station.longi), parseFloat(station.lat)])
        .addTo(this.map)
        .setPopup(stationPopup);
    });
  });
}
```

Configuración de los Markers de las estaciones del mapa

Como se puede ver en la imagen de arriba, lo que hago es generar el pop con HTML e insertar los datos, una vez hecho solo tengo que añadirlo al “marker”, lo siguiente es crear el marker, para lo cual creó uno div y le pongo como foto el icono que quiero usar, esto ya que al intentar hacerlo como aparece en la guia no funcionaba. Integrar el mapa fue algo bastante simple, ya que lo único que hay que hacer es una vez instalada la dependencia es añadir la configuración del mapa para que se ejecute al iniciar

```
this.map = tt.map({ x2 200 blob:ht
key: environment.tomtom.key,
style: MAP_STYLE,
container: 'map',
center: CULLERA,
zoom: 15,
minZoom: 12,
});
```

Configuración de mapa TomTom

Para mostrarlo solo hay que añadir una línea a nuestro HTML junto con la configuración previamente descrita.

```
Go to component
<div id="map" class="w-full overflow-hidden rounded-lg" style="height: 75vh;"></div>
```



El QR:

Para usar el QR estuve buscando una librería que pudiera simplificar el trabajo, estuve probando con ngx-qrcode 2, pero estaba obsoleto y @techiediaries/ngx-qrcode el cual no funcionaba correctamente, finalmente encontré ngx-qrcode-styling el cual simplemente lo tenía que configurar

```
this.config = {  
    width: 300,  
    height: 300,  
    data: `https://${environment.url}/comprobar-tiquet?user=${this.user}`,  
    image: '../..../assets/logo.png',  
    margin: 5,  
    dotsOptions: {  
        color: '#1977f3',  
        type: 'dots',  
    },  
    backgroundOptions: {  
        color: '#ffffff',  
    },  
    imageOptions: {  
        crossOrigin: 'anonymous',  
        margin: 0,  
    },  
};
```

Configuración del QR

El QR funciona de forma muy simple, al escanearlo redirige a la página de canje del billete pasando como parámetro el email del usuario y se implementa en el HTML como un componente normal pasándole como input el config

```
<ngx-qrcode-styling [config]="config"></ngx-qrcode-styling>
```

Algo que al principio pensé que sería complicado era el hecho de añadir varios idiomas, lo primero fue buscar una forma de añadir fácilmente las banderas, de forma escalable, encontré un paquete que permite incluirlas como clases, de forma similar a los iconos de BootsTrap, para los textos cree un JSON con todos los textos en los distintos idiomas y un Service encargado de obtenerlos

```
{
  "topBar": {
    "english": {
      "home": "Home",
      "schedule": "Schedule",
      "privateService": "Private Service",
      "aboutUs": "About Us",
      "login": "Login",
      "language": "Language"
    },
    "spanish": {
      "home": "Inicio",
      "schedule": "Horario",
      "privateService": "Servicio privado",
      "aboutUs": "Sobre Nosotros",
      "login": "Iniciar Sesión",
      "language": "Idioma"
    },
    "catalan": {
      "home": "Inici",
      "schedule": "Horari",
      "privateService": "Servei privat",
      "aboutUs": "Sobre a Nosaltres",
      "login": "Iniciar Sessió",
      "language": "Idioma"
    }
  },
  "nextBus": {
    "english": {
      "title": "Next Bus",
      "selectStation": "Choose your station:",
      "selectTime": "Time at which you want to get on the bus:",
      "selectAnOption": "Select an option",
      "searchButton": "Search"
    },
    "spanish": {
      "title": "Próximos autobuses",
      "selectStation": "Escoge en que parada buscas subir:",
      "selectTime": "A que hora quieres subir al autobus:",
      "selectAnOption": "Selecciona una opción",
      "searchButton": "Buscar"
    },
    "catalan": {
      "title": "Pròxims autobusos",
      "selectStation": "Escull en quina parada vols pujar:",
      "selectTime": "A quina hora vols pujar a l'autobús:",
      "selectAnOption": "Selecciona una opció",
      "searchButton": "Buscar"
    }
  }
}
```

texts.json



```
language.service.ts U src/app/services/language.service.ts...
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { CookieService } from 'ngx-cookie-service';
4 import { BehaviorSubject, Observable } from 'rxjs';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class LanguageService {
10   constructor(private http: HttpClient, private cookies: CookieService) {}
11
12   private jsonUrl = '../assets/texts.json';
13
14   private language = new BehaviorSubject<string>(this.cookies.check('language') ? this.cookies.get('language') : 'spanish');
15   currentLanguage = this.language.asObservable();
16
17   changeLanguage(data: string) {
18     this.language.next(data);
19     this.cookies.set('language', data)
20   }
21
22
23  getJSON(): Observable<any> {
24     return this.http.get<any>(this.jsonUrl);
25   }
26
27 }
```

language.service

Este Service se encarga de crear observables de los textos así como de en qué idioma está actualmente la web.

A la hora de implementarlo es tan simple como lo que se ve a continuación:

```
setSubscriptions() {
  this.languageService.getJSON().subscribe((data) => {
    this.data = data.profile;
    this.texts = this.data[this.language];
  });

  this.languageService.currentLanguage.subscribe((value) => {
    this.language = value;

    this.texts = this.data[this.language];
  });
}
```

Uso de Services

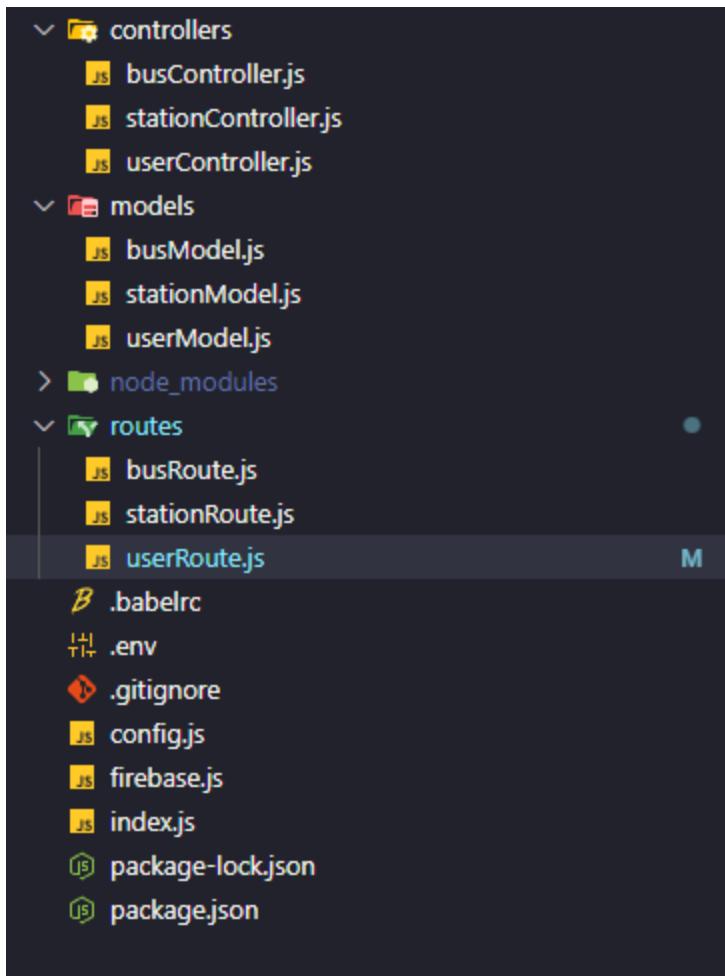
Primero se crea un observable para obtener el JSON y se asignan los textos, luego hago lo mismo con el lenguaje de tal forma que si uno cambia se ajustará

Cookies

Para guardar la información del usuario y que no que iniciar sesión cada vez he implementado un sistema de cookies que se encarga de gestionar los datos del usuario así como aquello que ha de mantenerse

Backend

El backend inicialmente iba a ser basado en SQL y PHP, pero por problemas con el servidor decidí usar firebase como base de datos y un backend con JS, si bien podría haber conectado directamente FireBase al frontend decidí hacerlo con un Backend para poder cambiarlo a futuro sin problema cambiando el backend, ya que mientras este esté bien hecho en el frontend solo deberé cambiar el enlace.



The screenshot shows a file explorer window with the following directory structure:

- controllers
 - busController.js
 - stationController.js
 - userController.js
- models
 - busModel.js
 - stationModel.js
 - userModel.js
- node_modules
- routes
 - busRoute.js
 - stationRoute.js
 - userRoute.js
- .babelrc
- .env
- .gitignore
- config.js
- firebase.js
- index.js
- package-lock.json
- package.json

El sistema del backend es bastante simple, he implementado un sistema de rutas, cada archivo hace las acciones CRUD pero la mayoría las he bloqueado ya que no las uso y pueden ser una vulnerabilidad, Users sí que tiene listo todo el CRUD. He de recalcar que temporalmente la API es completamente abierta (Es decir cualquiera puede hacer uso de ella mediante el link), pero una vez hecha la presentación cambiare su configuración por una más segura

```
js index.js M index.js\...
1 import express from 'express';
2 import cors from 'cors';
3
4 import config from './config.js';
5 import stationRoute from './routes/stationRoute.js';
6 import userRoute from './routes/userRoute.js';
7
8 const app = express();
9
10 app.use(cors());
11 app.use(express.json());
12
13 //routes
14 app.use('/stations', stationRoute);
15 app.use('/users', userRoute);
16
17 app.listen(config.port, () =>
18   console.log(`Server is live @ ${config.hostUrl}`),
19 );
```

La Base De Datos

Firebase usa una base de datos basada en colecciones, de forma similar a MongoDB, por lo que la forma de trabajar con ella es mediante JSON, mi BBDD es muy simple, se basa en 2 tablas, una de Usuarios y una de Estaciones, Estas siguen la siguiente estructura:

- Users:
 - id: string unico autogenerado
 - email: string unico
 - dni: string unico
 - password: string
 - rides: number

<p>+ Agregar documento</p> <pre>JgcFgPEGlwYkT0XmWpOA awzD0YzEooqLHnSoreL8 ></pre>	<p>+ Iniciar colección</p> <p>+ Agregar campo</p> <pre>dni: "00000000A" email: "Testuser@gmail.com" password: "Santi2003" rides: "0"</pre>
--	--

- Stations:
 - id: string unico autogenerado
 - address: string
 - image: string
 - lat: string
 - longi: string

<p> stations</p> <p>+ Agregar documento</p> <pre>0BX793kS6DbzeudTyNsX 51IKOA4Z6LySGEmaG0ib 8eLpwzI6Vq088Du9Cx7a KVes3oyTVIM05WBmPb6V MLeu0mTHG9c5iX68xZU3 RRx6ux0tpTnP9NsL3oYP S7s08i4WqzEhK5vVwHbt ></pre>	<p> 0BX793kS6DbzeudTyNsX</p> <p>+ Iniciar colección</p> <p>+ Agregar campo</p> <pre>address: "C/Sueca, 49" image: "https://streetviewpixels-pa.googleapis.com/v1/thumbnail? panoid=7pScyVXw63dILt5ZzR1InA&cb_client=search.gws- prod.gps&w=408&h=240&yaw=305.7695&pitch=0&thumbfov=100" lat: "39.1740076" (cadena) <input style="width: 20px; height: 20px;" type="button" value="editar"/> longi: "-0.2650113" <input style="width: 20px; height: 20px;" type="button" value="eliminar"/></pre>
---	---



Resultado

Inicio:

The screenshot shows the homepage of the web application. At the top, there is a navigation bar with links for 'Inicio', 'Horario', 'Servicio privado', 'Sobre Nosotros', and an email address 'santicr3@gmail.com'. There is also a language selection dropdown set to 'Idioma' and a 'Mapa' button. The main content area has a title 'Próximos autobuses' and two input fields: 'Escoge en que parada buscas subir:' with a dropdown menu and a location icon, and 'A que hora quieres subir al autobus:' with a date-time picker. A large blue 'Buscar' button is at the bottom. To the right is a map of a town with several bus stops marked by orange rectangles labeled 'CV-401', 'CV-402', 'CV-403', and 'CV-404'. The map includes street names like 'Calle de Riu', 'Calle del General Sáenz', 'Calle de la Unión', and 'Avenida de la Constitución'.

The screenshot shows a mobile version of the application. It features a header with the 'MR' logo and a menu icon. The main content area is identical to the desktop version, with 'Próximos autobuses', bus stop selection, time selection, and a large blue 'Buscar' button. To the right is a map of the same town, showing the same bus stops and street layout as the desktop version.



Web Autocares Miguel Ribera. Santiago Canet Ribera

[Inicio](#) [Horario](#) [Servicio privado](#) [Sobre Nosotros](#) santici3@gmail.com

Idioma

Próximos autobuses

Escoge en que parada buscas subir:

Selecciona una opción Mapa

A que hora quieres subir al autobus:

...

💡 Siguiente autobús en: X
 15 minuto/s
[Comprar billete](#)

[Buscar](#)

Escoge en que parada buscas subir:

Selecciona una opción Mapa

A que hora quieres subir al autobus:

...

💡 Siguiente autobús en: X
 15 minuto/s
[Comprar billete](#)

Horario:

[Inicio](#) [Horario](#) [Servicio privado](#) [Sobre Nosotros](#) santici3@gmail.com

Idioma

MIGUEL RIBERA AUTOCARES S.L.
 HORARIOS AUTOBÚS URBANO
 DE LUNES A VIERNES, EXCEPTO FESTIVOS
 A PARTIR DEL 2/11/23

PLAYA ESTACIÓN	ESTACIÓN PLAYA	ESTACIÓN FARO	MONGRELL FARO	FARO ESTACIÓN
9.30 12.10	18.45 6.47	13.17 19.17	6.00/13.17	13.17 7.57 14.27 6.00 13.40
4.30 12.10	18.45 6.47	13.17 19.17	6.00/13.17	13.17 7.57 14.27 6.00 13.40
6.40 13.20	19.40 7.47	14.17 20.17	7.47 14.17	9.06 15.40 8.00 15.50
6.40 13.20	19.40 7.47	14.17 20.17	7.47 14.17	9.06 15.40 8.00 15.50
7.00 13.50	20.20 8.17	14.47 20.42	8.17 14.47	9.27 16.57 8.32 15.40/2
7.00 13.50	20.20 8.17	14.47 20.42	8.17 14.47	9.27 16.57 8.32 15.40/2
7.50 14.20	20.45 8.56	15.09 21.23	8.56 15.30	9.57 17.27 9.00 17.30
8.00 14.20	20.45 8.56	15.09 21.23	8.56 15.30	9.57 17.27 9.00 17.30
8.40 15.20	21.50 9.47	15.47 22.17	9.47 17.17	10.10 18.30
9.20 15.40	22.20 10.17	16.17 22.47	10.17 17.47	11.27 19.57 10.40 19.30
9.45 16.20	10.47 16.47	23.07	16.47 18.47	11.57 21.21 11.00 20.10
10.00 16.20	10.47 16.47	23.07	16.47 18.47	11.57 21.21 11.00 20.10
10.50 17.20	11.47 17.47	11.47	17.47 21.11	12.57 19.00
11.00 17.50	12.17 18.17	12.17	18.17 22.17	13.27 12.95
11.50 18.20	12.51 18.47	12.51	18.47 23.31	13.57 13.10

1. El servicio de los 15.40/2 se realiza por las Calles, por tanto, debe cogerse el autobús dirección Estación de Autobuses, no la 15.30 y 15.40.
 2. El servicio de los 15.40 Faro Estación, irá por los Calles, por lo que debe cogerse el autobús en sentido Faro Estación o a la Estación.

SABADOS, DOMINGOS Y FESTIVOS
 A PARTIR DEL 28/10/23

PLAYA ESTACIÓN	ESTACIÓN PLAYA	FARO ESTACIÓN	MONGRELL FARO	FARO ESTACIÓN
8.30 14.20	17.17 18.17	17.17 18.17	17.17 18.17	17.17 18.17
7.50 16.45	8.17 17.17	8.17 17.17	8.17 17.17	8.17 17.17
8.45 17.50	9.17 18.17	9.17 19.17	9.17 19.17	9.17 19.17
9.50 18.50	10.17 19.17	10.17 20.17	10.17 20.17	10.17 20.17
11.50 20.50	11.17 21.41	11.17 22.41	11.17 22.41	11.17 22.41
12.50 22.00	13.17 22.17	13.17	13.17	13.17
13.50 22.50	14.17 23.17	14.17	14.17	14.17

1. Los horarios indicados son de salida desde la primera parada, dependiendo el horario de paso por los períodos intermedios del tráfico y otros circunstancias agrega. **El autobús no esperará al tren ni al avión con retraso.**
 2. Entre autobuses no tienen conexión oficial con los trenes de cercanías, ni entre medios de transporte. Por favor, tenga en cuenta que para moverse entre el aeropuerto y la estación de trenes, debe hacer uso del servicio de taxi o de la línea de autobús número 10.

PVP 1.49 (doble servicios especiales)

INVIERNO 2023

MIQUEL RIBERA AUTOCARES S.L.
 HORARIOS AUTOBÚS URBANO
 DE LUNES A VIERNES, EXCEPTO FESTIVOS
 A PARTIR DEL 2/11/23

PLAYA ESTACIÓN	ESTACIÓN PLAYA	FARO ESTACIÓN	MONGRELL FARO	FARO ESTACIÓN
9.30 12.10	18.45 6.47	13.17 19.17	6.00/13.17	13.17 7.57 14.27 6.17 13.40
4.30 12.10	18.45 6.47	13.17 19.17	6.00/13.17	13.17 7.57 14.27 6.17 13.40
6.40 13.20	19.40 7.47	14.17 20.17	7.47 14.17	9.06 15.40 8.00 15.50
6.40 13.20	19.40 7.47	14.17 20.17	7.47 14.17	9.06 15.40 8.00 15.50
7.00 13.50	20.20 8.17	14.47 20.42	8.17 14.47	9.27 17.27 9.30 17.40
7.00 13.50	20.20 8.17	14.47 20.42	8.17 14.47	9.27 17.27 9.30 17.40
7.50 14.20	20.45 8.56	15.09 21.23	8.56 15.30	9.57 17.27 9.30 17.40
8.00 14.20	20.45 8.56	15.09 21.23	8.56 15.30	9.57 17.27 9.30 17.40
8.40 15.20	21.50 9.47	15.47 22.17	9.47 17.17	10.10 18.30
9.20 15.40	22.20 10.17	16.17 22.47	10.17 17.47	11.27 19.57 10.40 19.30
9.45 16.20	10.47 16.47	23.07	16.47 18.47	11.57 21.21 11.00 20.10
10.00 16.20	10.47 16.47	23.07	16.47 18.47	11.57 21.21 11.00 20.10
10.50 17.20	11.47 17.47	11.47	17.47 21.11	12.57 19.00
11.00 17.50	12.17 18.17	12.17	18.17 22.17	13.27 12.95
11.50 18.20	12.51 18.47	12.51	18.47 23.31	13.57 13.10

1. Los horarios indicados son de salida desde la primera parada, dependiendo el horario de paso por los períodos intermedios del tráfico y otros circunstancias agrega. **El autobús no esperará al tren ni al avión con retraso.**
 2. Entre autobuses no tienen conexión oficial con los trenes de cercanías, ni entre medios de transporte. Por favor, tenga en cuenta que para moverse entre el aeropuerto y la estación de trenes, debe hacer uso del servicio de taxi o de la línea de autobús número 10.



Web Autocares Miguel Ribera. Santiago Canet Ribera

Servicio Privado:

The screenshot shows a dark-themed contact form. At the top, there are links for Inicio, Horario, Servicio privado, Sobre Nosotros, and an email address. A language selector for Spanish (Idioma) is also present. The form fields include: Nombre Completo (Name), Correo Electrónico (Email), Número de Teléfono (Phone number), and a question about service needs. There is a large text area for describing the service desired and a 'Enviar' (Send) button.

This screenshot shows a similar dark-themed contact form on the right side of the website. It includes fields for Name, Email, Phone number, and a question about service needs, followed by a text area for service description and a 'Enviar' button.

Sobre Nosotros:

The screenshot shows the 'Sobre Nosotros' (About Us) page. At the top, there are links for Inicio, Horario, Servicio privado, Sobre Nosotros, and an email address. A language selector for Spanish (Idioma) is also present. The main content features a heading '¡Bienvenido a Miguel Ribera Autocares!', a brief history of the company, a photo of two buses, and a list of services. The company logo is visible at the bottom right.

This screenshot shows the same 'Sobre Nosotros' page on the right side of the website. It includes the same heading, history, photo, service list, and company logo. The right side of the page has a sidebar with a welcome message and a detailed description of the company's history and services.



Web Autocares Miguel Ribera. Santiago Canet Ribera

Iniciar Sesión:

Crear Cuenta:



Web Autocares Miguel Ribera. Santiago Canet Ribera

Perfil:

The screenshot shows the user profile page with the following details:

Información del usuario

Correo Electrónico: sanctir3@gmail.com

DNI: 20942121T

Viajes restantes: 39

QR Usuario: A QR code linking to the user's profile.

Información del usuario

Correo Electrónico: sanctir3@gmail.com

DNI: 20942121T

Viajes restantes: 39

QR Usuario: A QR code linking to the user's profile.

Cerrar Sesión **Borrar Cuenta**

The screenshot shows the user profile page with the following details:

Información del usuario

Correo Electrónico: sanctir3@gmail.com

DNI: 20942121T

Viajes restantes: 39

QR Usuario: A QR code linking to the user's profile.

A confirmation dialog box is displayed in the center:

¿Estás seguro de que quieres cerrar sesión?

Sí **No**

Cerrar Sesión **Borrar Cuenta**

Información del usuario

Correo Electrónico: sanctir3@gmail.com

DNI: 20942121T

Viajes restantes: 39

QR Usuario: A QR code linking to the user's profile.

¿Estás seguro de que quieres cerrar sesión?

Sí **No**

Cerrar Sesión **Borrar Cuenta**

Pago:



Web Autocares Miguel Ribera. Santiago Canet Ribera

The screenshot displays two identical ticket purchase forms side-by-side, likely representing a responsive design or a split-screen view.

Top Navigation:

- Inicio
- Horario
- Servicio privado
- Sobre Nosotros
- santicr3@gmail.com
- Idioma (with a flag icon)
- Menu icon (three horizontal lines)

Ticket Purchase Form (Left):

Header: Compra tu billete

Buttons: Billete Individual (blue), Bono (white)

Text: Precio: 1.50€

Section: Método de pago

Buttons: Paypal (blue), Visa (orange), Mastercard (white)

Text Fields: Número de Tarjeta, Titular de la tarjeta, Fecha de expiración de la tarjeta, Código de seguridad (CVV)

Buttons: Comprar (blue)

Ticket Purchase Form (Right):

Header: Compra tu billete

Buttons: Billete Individual (blue), Bono (white)

Text: Precio: 1.50€

Section: Método de pago

Buttons: Paypal (blue), Visa (orange), Mastercard (white)

Text Fields: Número de Tarjeta, Titular de la tarjeta, Fecha de expiración de la tarjeta, Código de seguridad (CVV)

Buttons: Comprar (blue)



Despliegue

Frontend

Finalmente he usado Vercel para desplegar mi App con un proceso muy simple, dándole acceso a mi repositorio.

The screenshot shows the Vercel dashboard for the project "miguel-ribera-web". At the top, there are tabs for Repository, Usage, Domains, and Visit. Below that, the "Production Deployment" section is shown with a preview of the app's interface, which includes a search bar and location markers. The deployment status is listed as "Ready" with a timestamp of "18h ago by santicr3". The source code is shown as "main" with a commit hash "291bde9 Fixed checkTickets". A note at the bottom says "To update your Production Deployment, push to the 'main' branch." There are also "Build Logs", "Runtime Logs", and "Instant Rollback" buttons.

Backend

Para el Backend he usado render, al igual que Vercel está orientado a el ofrecer un deploy simple al conectarse de igual forma a el repositorio.

The screenshot shows the Render dashboard for the project "Miguel-Ribera-Backend-Firebase". The top navigation bar includes links for Dashboard, Blueprints, Env Groups, Docs, Community, Help, New +, and a user profile for "Santi Canet". The main area shows a "WEB SERVICE" named "Miguel-Ribera-Backend-Firebase" running on Node.js with a free plan. It provides a URL "https://miguel-ribera-backend-firebase.onrender.com". There are buttons for "Connect" and "Manual Deploy". On the left, a sidebar lists "Events", "Logs", "Disks", "Environment", "Shell", and "Previews". The "Events" section shows two entries: a recent deployment on May 30, 2024, at 6:44 PM, and another entry from the same date at 6:42 PM.

5. Cambios a futuro

A lo largo de este proyecto han surgido muchos problemas e inconvenientes, si bien he podido solucionar la mayoría de estos, han habido algunos que no he podido por ejemplo:

Cambios que no he podido solucionar:

- Añadir los autobuses al mapa:

Ya que de momento no tenemos forma de localizarlos, no he podido implementar un sistema de seguimiento para estos, pero es algo que planeamos implementar en un futuro próximo.

- Implementar un sistema real de cobros (Debido a la falta de tiempo)

Algunos cambios a futuro serían:

- Mejorar la seguridad de la Web
- Mejor el sistema del escaneo de billetes

6. Conclusión

Durante este proyecto he podido aprender mucho tanto de Angular como del uso de distintas librerías y el consumo de API's, por lo que considero que es ha sido un gran proyecto, además de haber cumplido casi todas las cosas que me proponía con este, espero poder seguir mejorandolo y solucionar e implementar aquello que por la falta de tiempo no he podido, aún así, estoy muy satisfecho con el resultado. Espero que la web pueda resultar de ayuda a todo aquel que quiera usar los autobuses en un futuro así como facilitar las cosas para la empresa.



**MIGUEL RIBERA
AUTOCARES, S.L.**

