

PROYECTO INTERMODULAR DAW.

AEROELITE & CO.

Por Jose Luis Ruiz Romero

- ÍNDICE

1) Introducción.

- a) Descripción general del proyecto.
- b) Detalle del problema planteado.
- c) Áreas generales de enfoque del proyecto.

2) Usuarios y requisitos.

- a) Perfiles de usuario.
- b) Descripción de los perfiles de usuario.

3) Diseño y modelado de datos.

- a) Diagrama Entidad/Relación.
 - i) Diseño del diagrama.
 - ii) Origen de los atributos.
 - iii) Justificación de las claves primarias.

- b) Paso a tablas.

4) Implantación de la Base de Datos.

- a) Código SQL para la creación de tablas.

5) Diseño web.

- a) Estudio de la competencia.
- b) Diagramación.
 - i) Mapa web.
 - ii) Menús de navegación.
- c) Prototipos de bajo nivel.
 - i) Wireframes.
- d) Referentes estéticos y diseño gráfico.
 - i) Colores.
 - ii) Tipografía.
 - iii) Imágenes (logotipo, iconos, botones, formularios).
 - iv) Diseño gráfico de la interfaz.

6) Desarrollo web.

- a) Programación de funcionalidades.
 - i) Funcionalidades con PHP.
 - ii) Funcionalidades con AJAX.

- b) APIs utilizadas.

7) Documentación.

- a) Manuales de usuario.

1) Introducción.

a) Descripción general del proyecto.

El presente proyecto, tiene como objetivo desarrollar una aplicación web diseñada para gestionar eficientemente tanto el alquiler como la planificación de viajes mediante jets privados. Esta solución busca abordar la falta de aplicaciones claras, intuitivas y a su vez transparentes, brindando las herramientas necesarias tanto para los usuarios como para los administradores del sistema.

La aplicación proporcionará:

- Facilidad de uso.
- Accesibilidad y eficiencia.
- Claridad.

Siguiendo la línea de las necesidades detectadas, el proyecto abarcará los siguientes puntos clave:

- **Análisis de perfiles de usuarios**, para proporcionar la facilidad de uso prometida y que las funcionalidades respondan a las necesidades reales de estos.
- **Diseño de una base de datos robusta**, que permita la organización eficiente de la información tanto de la web, como de los usuarios de cara a una fácil accesibilidad.
- **Desarrollo de interfaces gráficas** y funcionalidades específicas para cada tipo de usuario, además de ser amigables y con total transparencia.
- **Implantación de tecnologías actuales**, que aseguren tanto la eficiencia como escalabilidad y mantenimiento del proyecto.

b) Detalle del problema planteado.

En la actualidad, el mercado de alquiler y planificación de viajes mediante jets privados presenta varios desafíos que dificultan una experiencia óptima de cara a dar el paso a realizar un viaje de estas características o a mantener un uso continuado de estos. Entre los principales problemas que surgen encontramos:

- **Falta de plataformas integrales**, las cuales se encuentran fragmentadas con servicios que no llegan a abarcar todo el

proceso desde el alquiler hasta la planificación del mismo vuelo.

- **Experiencias de usuario poco intuitivas**, que dificultan el proceso para el usuario que suele buscar una solución rápida y sencilla.
- **Carencia de transparencia**. La falta de claridad en los costos, disponibilidad de aeronaves y términos de alquiler provoca desconfianza, lo que al usuario le puede dificultar a la hora de apostar por esta alternativa de transporte.
- **Limitada accesibilidad tecnológica**, ya que no todas las plataformas existentes permiten un acceso fluido a través de dispositivos modernos o móviles, lo cual se traduce en la exclusión de un segmento de usuarios que busca una solución desde cualquier lugar.

c) Áreas generales de desarrollo del proyecto.

Hemos definido las siguientes áreas de enfoque para este proyecto:

- **Mejorar la experiencia del usuario**, mediante una aplicación intuitiva que haga más sencillo el proceso.
- **Optimizar la gestión operativa**, de cara a las herramientas necesarias para los administradores.
- **Transparencia en la información**, tanto el costo como servicios y demás.
- **Accesibilidad multiplataforma**, para que siempre puedas acceder desde cualquier lugar y en cualquier situación.
- **Integración de funcionalidades avanzadas**, de cara a planear los vuelos de manera automática, etc.

2) Usuarios y requisitos.

a) Perfiles de usuario

El sistema contará con 3 tipos diferentes de usuarios los cuales tendrán diferentes funciones disponibles y otras que no dispondrán:

- Usuario sin registrar.
- Usuario registrado.
- Administrador.

b) Descripción de los perfiles de usuario

Los distintos usuarios tendrán las siguientes funcionalidades disponibles y no disponibles:

- **Usuario sin registrar:**

- Lo único que podrá hacer es entrar a la web y consultar los precios y planes de vuelo.

- **Usuario registrado:**

- Puede entrar a la web y consultar los precios y planes de vuelo.
 - Puede realizar reservas personalizadas.
 - Puede visualizar su historial de reservas y el estado actual de las mismas.

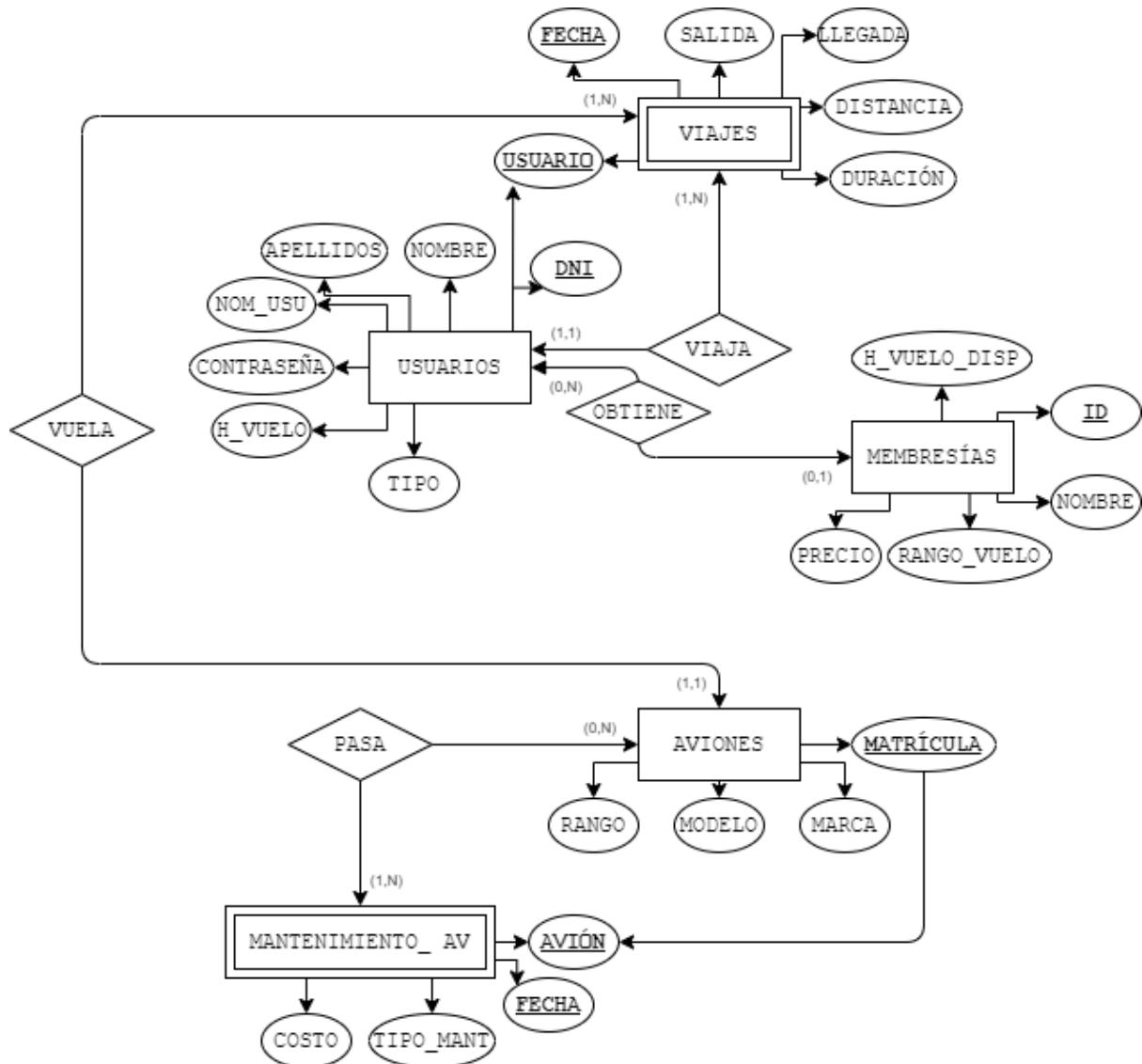
- **Administrador:**

- Puede gestionar reservas que haya en curso y por ende, cancelarlas o modificarlas levemente.
 - Puede ver estadísticas sobre la página web y las reservas.

3) Diseño y modelado de datos.

a) Diagrama Entidad/Relación.

i) Diseño del diagrama.



ii) Origen de los atributos.

Vamos a hablar sobre los atributos de cada tabla tratando de explicar los atributos el cual su nombre no sea tan obvio y no sea tan fácil de identificar:

- **Usuarios**

- DNI.
- Nombre.
- Apellidos.

- Nom_usu: Nombre del usuario en nuestra aplicación web.
- Email: Email del usuario.
- Contraseña.
- H_vuelo: Horas de vuelo que lleva el usuario en el último año con nosotros.
- Membresía: Clave externa que proviene del ID de la membresía, el cual indica que tipo de membresía tiene.
- Tipo: Indica si el usuario está registrado o si es administrador.

- **Aviones**

- Matrícula: Sí, los aviones también tienen matrícula.
- Marca: El fabricante que ha manufacturado el avión.
- Modelo: Modelo específico del avión.
- Rango: Rango de vuelo en kilómetros.

- **Mantenimiento_av**

- Avión: Clave externa que proviene del ID del avión, el cual indica el avión que ha pasado el mantenimiento.
- Fecha: Fecha en la que el avión pasa el mantenimiento.
- Tipo_mant: Indica el tipo del mantenimiento, es decir, si es rutinario, preventivo o correctivo.
- Costo: Precio del mantenimiento.

- **Viajes**

- Usuario: Clave externa que proviene del DNI del usuario e indica qué usuario va a viajar.
- Fecha: Fecha de salida del viaje.
- Salida: Aeropuerto de salida.
- Llegada: Aeropuerto de llegada.
- Distancia: Distancia entre la salida y la llegada en kilómetros.
- Duración: Duración en horas del vuelo.

- **Membresías**

- Id
- Nombre

- H_vuelo_disp: Horas de vuelo disponibles anualmente.
- Rango_vuelo: Indica si es nacional, continental o intercontinental.
- Precio

iii) Justificación de las claves primarias.

En cuanto a las claves primarias, tenemos 5 entidades distintas y vamos a explicar una a una por qué ha sido elegida cada una de las claves primarias.

Antes de nada vamos a definir lo que es una clave primaria. Una clave primaria es un identificador único para cada fila que tendríamos de dicha entidad, es decir, si tuviésemos 10.000 usuarios cada uno debería estar identificado de manera diferente y única, dicho esto, vamos a proceder con cada una de las entidades y sus claves primarias:

- **Usuarios:** Dentro de usuarios hemos seleccionado DNI como clave primaria ya que es un número de identificación único para cada ciudadano, no solo de España, en todo el mundo, ya que cada país tiene su homólogo.
- **Viajes:** Cuando alguien viaja con nuestra empresa debe ser usuario de nuestra aplicación, pero un usuario puede viajar varias contratando nuestros servicios por eso combinamos la fecha ya que los usuarios no pueden hacer dos viajes en el mismo día.
- **Aviones:** Los aviones aunque no lo puedan parecer, también tienen matrículas, y estas obviamente son únicas en todo el mundo.
- **Mantenimiento_av:** En el mantenimiento que pasan los aviones hemos escogido tener una clave primaria compuesta por dos atributos ya que un avión puede pasar varios mantenimientos a lo largo de su vida útil. En cambio si combinamos el avión junto con la fecha en que pasa el mantenimiento estos no pueden duplicarse nunca ya que es imposible hacerle varios mantenimientos en el mismo día a un avión.

- **Membresía:** Hemos decidido dar un ID único a cada tipo de membresía para que sean fácilmente identificables y únicas.

b) Paso a tablas

Vamos a realizar el paso a tablas y acto seguido explicaremos qué es lo que ha cambiado y por qué:

Usuarios(DNI, Nombre, Apellidos, Email, Nom_usu, Contraseña, H_vuelo, Membresía, Tipo)

Aviones(Matrícula, Marca, Modelo, Rango)

Membresías(Id, Nombre, H_vuelo_disp, Rango_vuelo, Precio)

Mantenimiento_av(Avión, Fecha, Tipo_mant, Costo)

Viajes(Usuario, Avión, Fecha, Salida, Llegada, Distancia, Duración)

Hemos añadido a Usuarios el atributo membresía para saber que tipo de membresía tiene cada usuario. También hemos añadido Avión a viajes para poder saber qué aviones viajan a dónde y qué días

Por último tenemos las claves primarias de las entidades débiles que heredan las claves primarias de sus entidades “padres”.

4) Implantación de la Base de Datos.

a) Código SQL para la creación de tablas.

Creación de las tablas:

```
--CREACIÓN DE TABLAS
CREATE TABLE `aviones` (
  `matricula` varchar(8) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `marca` varchar(50) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `modelo` varchar(50) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `rango_max` int(5) NOT NULL,
  `rango_min` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

CREATE TABLE `mantenimiento_av` (
  `avion` varchar(8) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `fecha` date NOT NULL,
  `tipo_mant` enum('Rutinario','Preventivo','Correctivo') CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `costo` int(7) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

CREATE TABLE `membresias` (
  `id` bigint(20) UNSIGNED NOT NULL,
  `nombre` varchar(25) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `h_vuelo_disp` int(5) NOT NULL,
  `rango_vuelo` enum('Nacional','Continental','Intercontinental','') CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `precio` int(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

CREATE TABLE `usuarios` (
  `dni` varchar(9) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `nombre` varchar(100) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `apellidos` varchar(100) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `nom_usu` varchar(50) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `password` varchar(20) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `h_vuelo` int(6) NOT NULL,
  `membresia` bigint(20) UNSIGNED NOT NULL,
  `tipo` enum('R','A') CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

CREATE TABLE `viajes` (
  `usuario` varchar(9) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `avion` varchar(8) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `fecha` date NOT NULL,
  `salida` varchar(150) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `llegada` varchar(150) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `distancia` int(5) NOT NULL,
  `duracion` time NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Creamos las cinco tablas mediante **create table**, especificando el tipo de cada columna y sus restricciones.

Set de claves primarias y foreign keys:

```
--SETEANDO CLAVES PRIMARIAS
ALTER TABLE `aviones`
ADD PRIMARY KEY (`matricula`);

ALTER TABLE `mantenimiento_av`
ADD PRIMARY KEY (`avion`, `fecha`);

ALTER TABLE `membresias`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `id` (`id`);

ALTER TABLE `usuarios`
ADD PRIMARY KEY (`dni`),
ADD KEY `fk_usu_mem` (`membresia`);

ALTER TABLE `viajes`
ADD PRIMARY KEY (`usuario`, `avion`, `fecha`),
ADD KEY `fk_via_avi` (`avion`);

--MODIFICANDO EL ID PARA QUE COINCIDA CON SU FOREIGN KEY
ALTER TABLE `membresias`
MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT;

--SETEANDO LAS FOREIGN KEY DE LAS TABLAS
ALTER TABLE `mantenimiento_av`
ADD CONSTRAINT `fk_man_avi` FOREIGN KEY (`avion`) REFERENCES `aviones` (`matricula`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `usuarios`
ADD CONSTRAINT `fk_usu_mem` FOREIGN KEY (`membresia`) REFERENCES `membresias` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `viajes`
ADD CONSTRAINT `fk_via_avi` FOREIGN KEY (`avion`) REFERENCES `aviones` (`matricula`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `fk_via_usu` FOREIGN KEY (`usuario`) REFERENCES `usuarios` (`dni`) ON DELETE CASCADE ON UPDATE CASCADE;

COMMIT;
```

Seteamos las claves primarias de cada tabla alterando la columna correspondiente mediante **alter table** y **add** con la que también añadimos las foreign key mediante **constraint**.

5) Diseño web

a) Estudio de la competencia.

Vamos a analizar 3 empresas de nuestro mismo sector, aviación, que operan tanto a nivel nacional como continental e intercontinental.

Empezamos por definir las competencias de cada empresa:

- **Gestair**

Es un operador y gestor de jets con servicios integrales de aviación. Cuenta con una flota propia diversa además de ofrecer servicios adicionales de mantenimiento, logística, gestión de jets entre otras.

- **VistaJet**

Es propietario y operador de una flota uniforme que ofrece programas de suscripción y soluciones personalizadas, lo que no permite obtener un servicio puntual.

- **LunaJets**

Es un corredor que conecta clientes con operadores y propietarios, por tanto no cuenta con una flota propia, pero le hace contar con una red muy extensa de jets a través de asociados. Además, ofrece vuelos para grupos y logística de carga.

Flota y operaciones:

Empresa	Tipo de Flota	Propiedad	Alcance
Gestair	Jets privados diversos	Propia y gestión de terceros	Europa y Norte de África
VistaJet	Flota homogénea	Propia	Global
LunaJets	Acceso a más de 4.800 jets	No posee aviones	Global

Servicios ofrecidos:

Servicio	Gestair	VistaJet	LunaJets
Vuelos chárter privados	✓	✓	✓
Gestión de aeronaves	✓	✗	✗
Mantenimiento y logística	✓	✗	✗
Vuelos por suscripción	✗	✓	✗
Empty legs (descuentos en vuelos vacíos)	✗	✗	✓
Venta y adquisición de jets	✗	✗	✓
Vuelos para grupos grandes	✗	✗	✓
Fletes de carga	✗	✗	✓

Precios y accesibilidad:

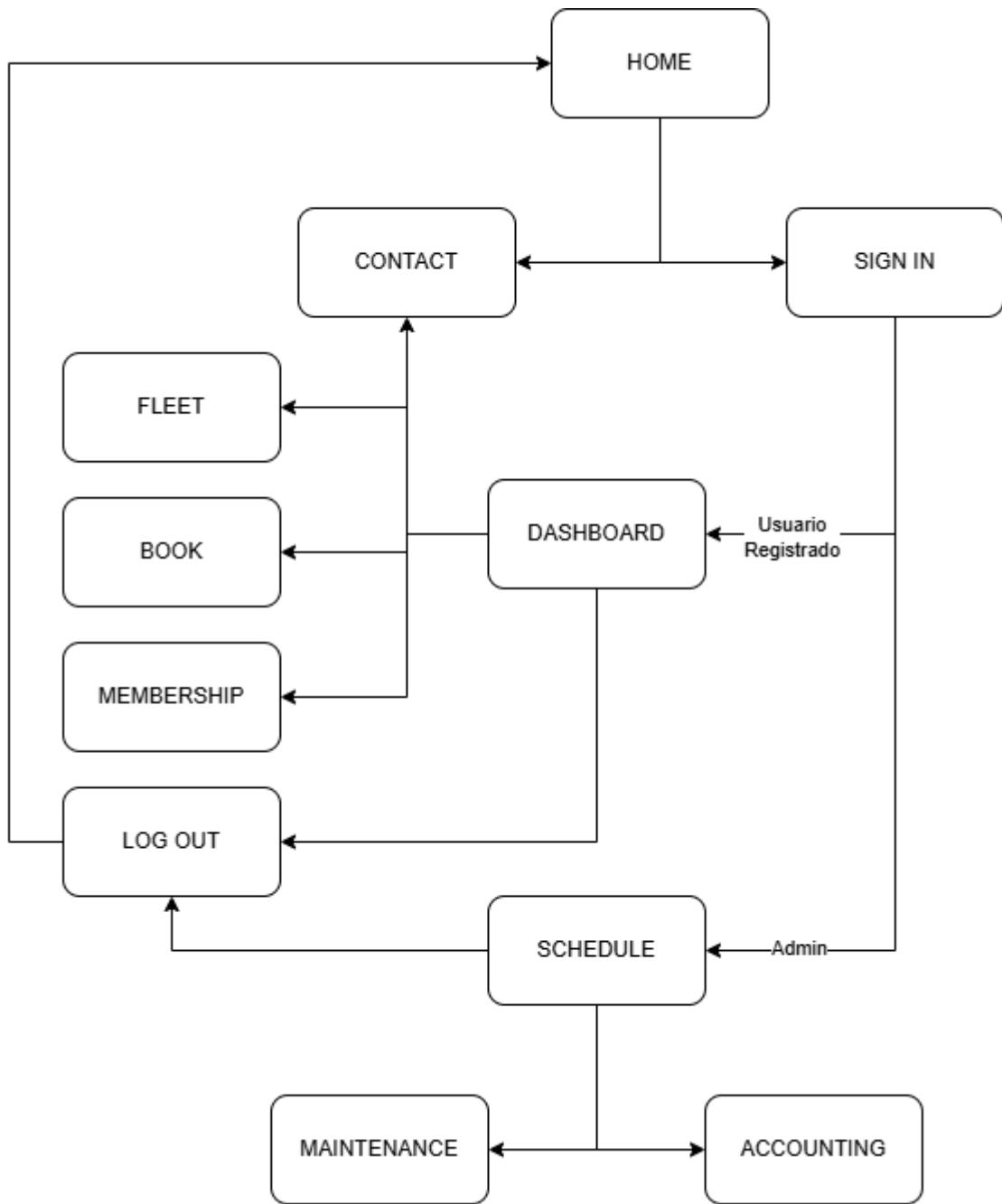
Empresa	Precio promedio	Costos extras	Flexibilidad
Gestair	Medio-alto	Personalización y mantenimiento	Alta (Gestión personalizada)
VistaJet	Alto (Suscripción)	Compromiso a largo plazo	Media (Depende del plan contratado)
LunaJets	Variable (Depende del operador)	No hay costos fijos	Máxima (Opciones con descuentos)

C clientes objetivo:

Empresa	Tipo de cliente
Gestair	Directivos, empresarios, grandes patrimonios
VistaJet	Empresas multinacionales, altos ejecutivos, gobiernos
Luna Jets	Particulares, viajeros ocasionales, grupos y empresas con presupuesto ajustado

b) Diagramación.

i) Mapa web.



A través de la página Home podemos entrar tanto a Contacto como a la página de Inicio de Sesión, por la cual, al iniciar sesión dependiendo del tipo de usuario que seamos nos enviará a la página Dashboard, en caso de que seamos usuarios básicos, o a la página Schedule, en caso de ser administradores.

Una vez dentro de la web podemos navegar en las distintas páginas dentro de nuestras restricciones como usuarios:

- Usuarios básicos (registrados): Tendrán acceso a Dashboard, Fleet, Book, Membership y Contact.
- Administradores: Tendrán acceso a Schedule, Maintenance y Accounting, además de poder entrar a las demás páginas mediante la indexación en la Url, por si se diese el caso de que estas necesiten mantenimiento.

En caso de ser un usuario sin registrar tan solo tendrá acceso a las páginas Home, Contact y Sign In.

ii) Menús de navegación.

- Usuarios no registrados:



- Usuarios registrados:



- Administradores:

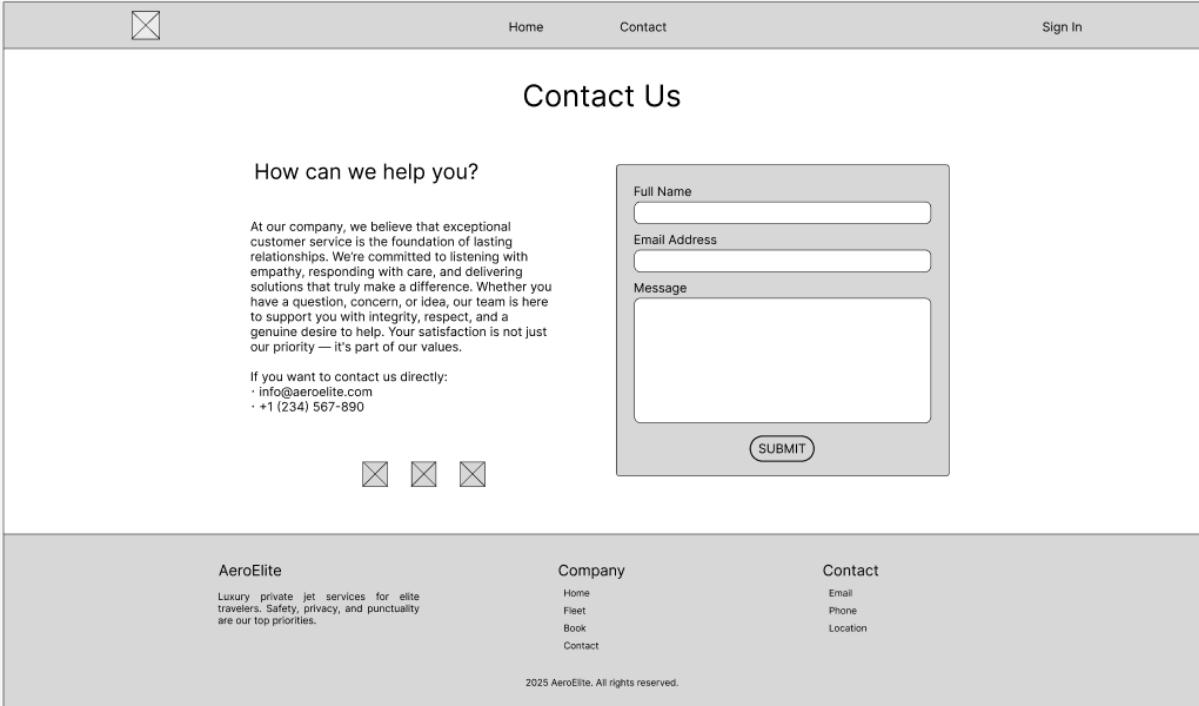


c) Prototipos de bajo nivel.

i) Wireframes.



“Home”: Landing principal donde los usuarios nuevos o sin registrar llegan. Contiene información general sobre la empresa.



The wireframe shows a contact form with fields for Full Name, Email Address, and Message, along with a SUBMIT button. Below the form are three social media sharing icons (Facebook, Twitter, LinkedIn). At the bottom, there's a footer with links for AeroElite, Company, and Contact.

Home Contact Sign In

Contact Us

How can we help you?

At our company, we believe that exceptional customer service is the foundation of lasting relationships. We're committed to listening with empathy, responding with care, and delivering solutions that truly make a difference. Whether you have a question, concern, or idea, our team is here to support you with integrity, respect, and a genuine desire to help. Your satisfaction is not just our priority — it's part of our values.

If you want to contact us directly:

- info@aeroelite.com
- +1 (234) 567-890

AeroElite

Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company

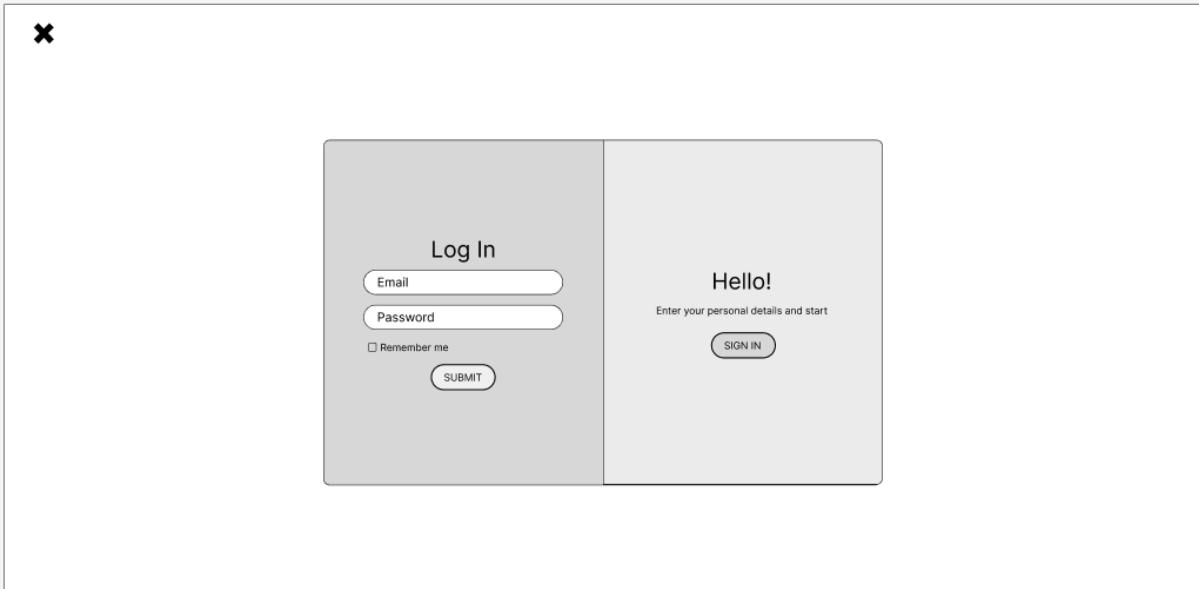
Contact

Home Fleet Book Contact

Email Phone Location

2025 AeroElite. All rights reserved.

“Contact”, contiene información de contacto de la empresa, proporcionando la posibilidad de contactar o ser contactado.



The wireframe shows a login form with fields for Email and Password, a Remember me checkbox, and a SUBMIT button. To the right, there's a "Hello!" greeting and a SIGN IN button.

X

Log In

Email

Password

Remember me

SUBMIT

Hello!

Enter your personal details and start

SIGN IN

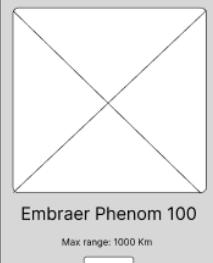
“Sign In”, página en la que si no tienes una cuenta puedes crearla y registrarte en la web o en cambio acceder a tu perfil si ya estás registrado.

“Dashboard”, es el área personal del usuario registrando donde se le muestra su información personal además de sus vuelos programados y su planificación.

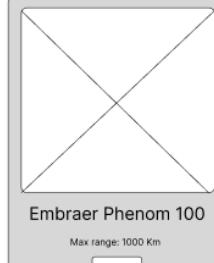
“Memberships”, donde se exponen las membresías disponibles junto con sus capacidades y el usuario puede comprarlas.

Dashboard Fleet Book Membership Contact Log Out

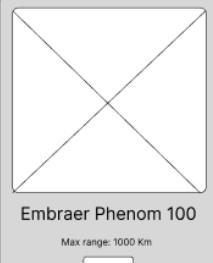
Our Fleet



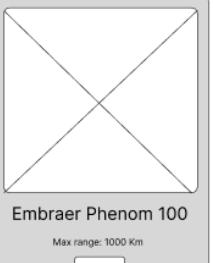
Embraer Phenom 100
Max range: 1000 Km
[Book](#)



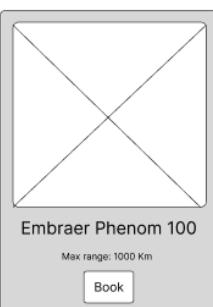
Embraer Phenom 100
Max range: 1000 Km
[Book](#)



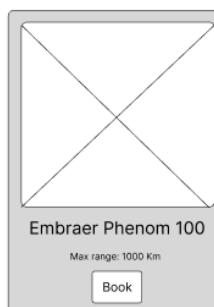
Embraer Phenom 100
Max range: 1000 Km
[Book](#)



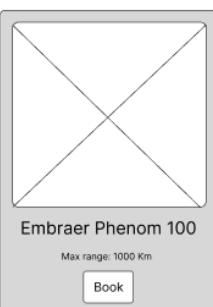
Embraer Phenom 100
Max range: 1000 Km
[Book](#)



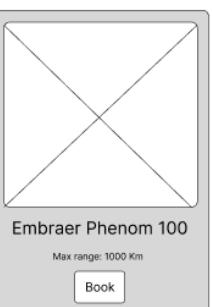
Embraer Phenom 100
Max range: 1000 Km
[Book](#)



Embraer Phenom 100
Max range: 1000 Km
[Book](#)



Embraer Phenom 100
Max range: 1000 Km
[Book](#)



Embraer Phenom 100
Max range: 1000 Km
[Book](#)

AeroElite
Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact
[Email](#)
[Phone](#)
[Location](#)

2025 AeroElite. All rights reserved.

“Fleet”, donde se expone la flota y el usuario puede verlo y seleccionar un avión para reservar.

 Dashboard Fleet Book Membership Contact Log Out

Book Your Private Jet

Choose your Jet:

Choose the departure airport. Choose the arrival airport.

When are you going to travel?

AeroElite
Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact
[Email](#)
[Phone](#)
[Location](#)

2025 AeroElite. All rights reserved.

“Book”, en esta página es en la que el usuario puede hacer la reserva de su vuelo.

 Schedule Maintenance Accounting Log Out

Flight Schedule

Name: Date:

User	Model	Date	Departure	Arrival	Distance	Duration	
Pepe	Boeing 737	Today	Federico Garcia Lorca Airport (GRX)	Federico Garcia Lorca Airport (GRX)	0 Km	0 Horas	<input type="button" value="Cancel"/>
Pepe	Boeing 737	Today	Federico Garcia Lorca Airport (GRX)	Federico Garcia Lorca Airport (GRX)	0 Km	0 Horas	<input type="button" value="Cancel"/>
Pepe	Boeing 737	Today	Federico Garcia Lorca Airport (GRX)	Federico Garcia Lorca Airport (GRX)	0 Km	0 Horas	<input type="button" value="Cancel"/>
Pepe	Boeing 737	Today	Federico Garcia Lorca Airport (GRX)	Federico Garcia Lorca Airport (GRX)	0 Km	0 Horas	<input type="button" value="Cancel"/>

AeroElite
Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact
[Email](#)
[Phone](#)
[Location](#)

2025 AeroElite. All rights reserved.

“Schedule”, la página principal del administrador, en la que puede ver y cancelar cualquier vuelo reservado.

Jet's Maintenance Panel

Model: Type: Submit

Last 5 Maintenances

Model	Date	Departure
Boeing 737	Today	Preventivo

Preventivo
Rutinario
Corrective
Sin Mantener

AeroElite
Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
Home
Fleet
Book
Contact

Contact
Email
Phone
Location

2025 AeroElite. All rights reserved.

“Maintenance”, es la página en la que el administrador puede dar de baja los aviones por mantenimiento.

Accounting

Total Income
1000000€

Viajes
Membresías
Donaciones

Total Income
1000000€

Viajes
Membresías
Donaciones

Total Income
1000000€

Viajes
Membresías
Donaciones

AeroElite
Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
Home
Fleet
Book
Contact

Contact
Email
Phone
Location

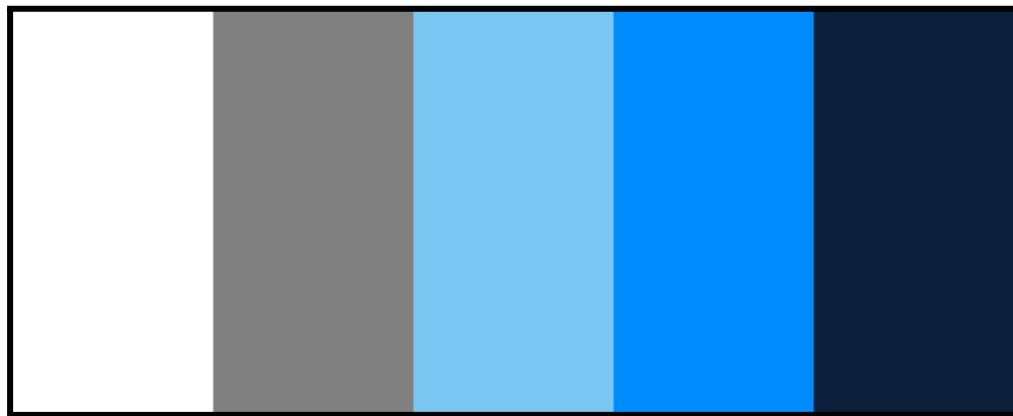
2025 AeroElite. All rights reserved.

“Accounting”, página plenamente informativa sobre el rendimiento económico básico de la empresa.

[Wireframes FIGMA.](#)

d) Referentes estéticos y diseño gráfico.

i) Colores.



ii) Tipografía.

Se está utilizando la tipografía DM Sans Serif.

DM Sans

iii) Imágenes (logotipo, iconos, botones).

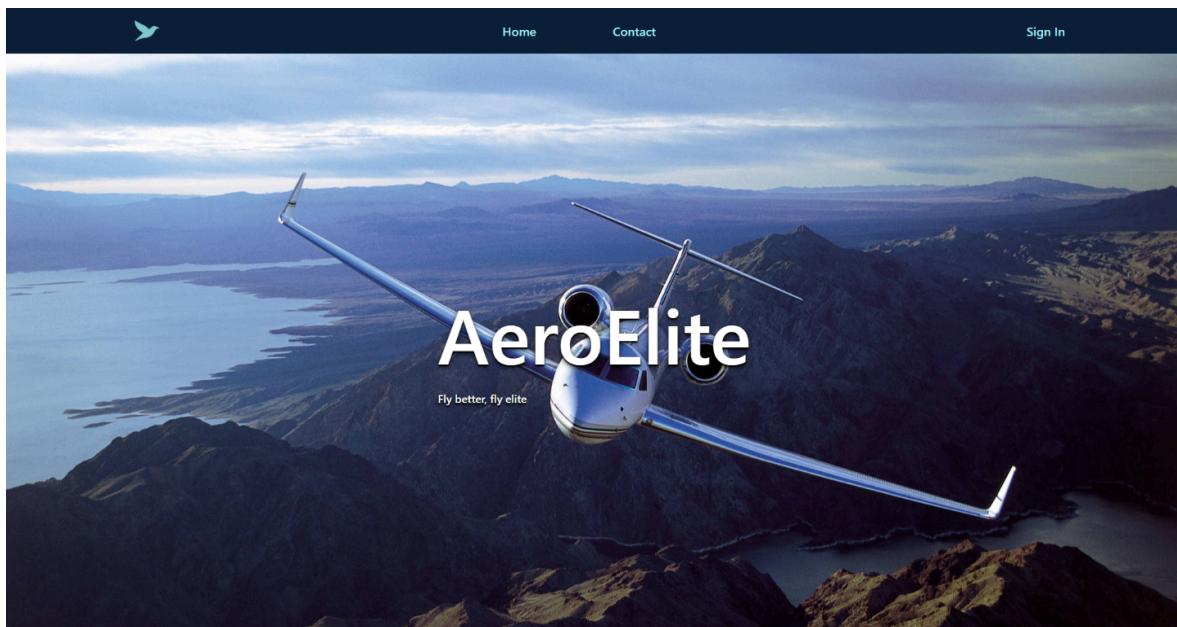
El logotipo de la empresa es el siguiente:



Y algunos de los iconos utilizados son los siguientes:



iv) Diseño gráfico de la interfaz.



✓ Total privacy

✓ Transparent prices

✓ Fully customizable

Our fleet



[See More](#)

Our Clients

"Exceptional service from start to finish. The jet was immaculate, and the crew was incredibly professional."
- James

"Flying privately has never felt this smooth and personalized."
- Tod

"AeroElite offers top-tier private jet services with exceptional comfort, punctuality, and personalized attention—ideal for business or luxury travel."
- Jeff

Fly wherever you want, whenever you want.

[Book your jet](#)

AeroElite

Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company

[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact

Email: info@aeroelite.com
Phone: +1 (234) 567-890
Location: Miami, FL, USA

© 2025 AeroElite. All rights reserved.



Contact Us

How can we help you?

At our company, we believe that exceptional customer service is the foundation of lasting relationships. We're committed to listening with empathy, responding with care, and delivering solutions that truly make a difference. Whether you have a question, concern, or idea, our team is here to support you with integrity, respect, and a genuine desire to help. Your satisfaction is not just our priority—it's part of our values.

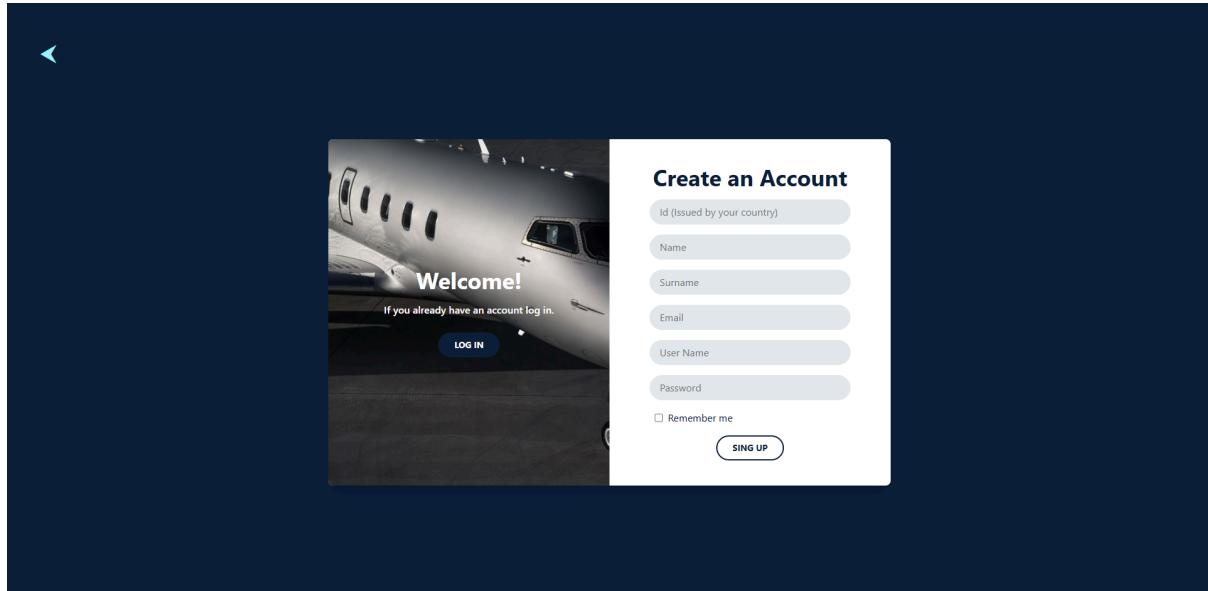
If you want to contact us directly:

- info@aeroelite.com
- +1 (234) 567-890

Full Name

Email Address

Message




Dashboard
Fleet
Book
Membership
Contact
Log Out

Dear AGCH,

Your schedule:

2025-06-08 → Berlin Brandenburg Airport (BER)
 2025-06-17 → Gary Chicago International Airport (GYY)
 Departure from Madrid Barajas International Airport (MAD)
 You will be flying on a **Boeing BBJ 737-800** during **7.48 hours**, traveling **6728 Km.**
 The price is **40000€**. It doesn't match your membership



12345678X
 Alejandro Aguayo
 AGCH
 ale@gmail.com
 Flight hours, 0

AeroElite
 Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact
 Email: info@aeroelite.com
 Phone: +1 (234) 567-890
 Location: Miami, FL, USA

© 2025 AeroElite. All rights reserved.


Dashboard
Fleet
Book
Membership
Contact
Log Out

Our Fleet



Embraer Phenom 100
Max range: 1900 Km
[Book](#)



Cessna Citation M2
Max range: 1800 Km
[Book](#)



Honda Jet HA-420
Max range: 1900 Km
[Book](#)



Eclipse 500
Max range: 1800 Km
[Book](#)









[Dashboard](#)
[Fleet](#)
[Book](#)
[Membership](#)
[Contact](#)
[Log Out](#)

Book Your Private Jet

Choose your Jet:

-- Select --

Choose the departure airport:

Choose the arrival airport:

When are you going to travel?

mm/dd/yyyy

Your book does not match with your membership

[BOOK](#)

AeroElite
 Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact
 Email: info@aeroelite.com
 Phone: +1 (234) 567-890
 Location: Miami, FL, USA

© 2025 AeroElite. All rights reserved.



Memberships



FlyLocal

Enabled flight hours: 40
Enabled range: Nacional

Perfect for those who need to travel quickly and exclusively within the country. Access light jets for business or personal trips without the hassle of commercial flight delays.

[BUY](#)



AeroPlus

Enabled flight hours: 100
Enabled range: Continental

Designed for entrepreneurs or frequent regional travelers. Fly between countries within the same continent in mid-size jets that balance efficiency, comfort, and flexibility.

[BUY](#)



GlobalReach

Enabled flight hours: 200
Enabled range: Intercontinental

For those crossing oceans. Gain access to long-range jets, ideal for intercontinental flights with premium amenities. With this membership, the world is truly within your reach.

[BUY](#)

AeroElite
Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact
Email: info@aeroelite.com
Phone: +1 (234) 567-890
Location: Miami, FL, USA

© 2025 AeroElite. All rights reserved.



Flight Schedule

Name: Date: mm/dd/yyyy

User	Model	Date	Departure	Arrival	Distance	Duration	
AGCH	Gulfstream G650ER	2025-06-08	Federico Garcia Lorca Airport (GRX)	Berlin Brandenburg Airport (BER)	2158 Km	2.4 Hours	CANCEL
AGCH	Boeing BBJ 737-800	2025-06-17	Madrid Barajas International Airport (MAD)	Gary Chicago International Airport (GYY)	6728 Km	7.48 Hours	CANCEL
MESSI	Gulfstream G650ER	2025-05-09	Barcelona International Airport (BCN)	London Gatwick Airport (LGW)	1109 Km	1.23 Hours	CANCEL
MESSI	Bombardier Global 7500	2025-07-09	Federico Garcia Lorca Airport (GRX)	Los Angeles International Airport (LAX)	9611 Km	10.68 Hours	CANCEL

AeroElite
Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company
[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact
Email: info@aeroelite.com
Phone: +1 (234) 567-890
Location: Miami, FL, USA

© 2025 AeroElite. All rights reserved.

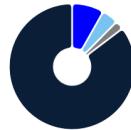


Jet's Maintenance Panel

Model: Type:

Last 5 Maintenances

Modelo	Fecha	Tipo
Gulfstream G150	2025-06-08	Preventivo
Cessna Citation Mustang	2025-06-08	Correctivo
Cessna Citation M2	2025-06-08	Rutinario
Bombardier Learjet 70	2025-06-08	Preventivo
Gulfstream G280	2025-06-08	Rutinario



- Routine
- Preventive
- Corrective
- Unmaintained

AeroElite

Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company

[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

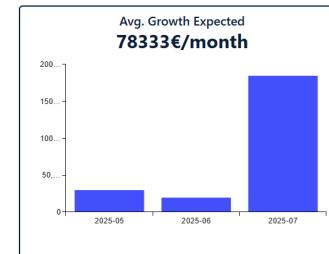
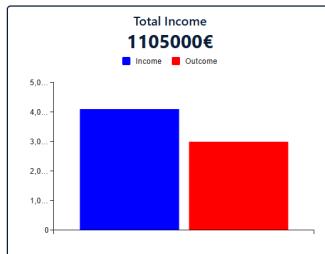
Contact

Email: info@aeroelite.com
Phone: +1 (234) 567-890
Location: Miami, FL, USA

© 2025 AeroElite. All rights reserved.



Accounting



AeroElite

Luxury private jet services for elite travelers. Safety, privacy, and punctuality are our top priorities.

Company

[Home](#)
[Fleet](#)
[Book](#)
[Contact](#)

Contact

Email: info@aeroelite.com
Phone: +1 (234) 567-890
Location: Miami, FL, USA

© 2025 AeroElite. All rights reserved.

6) Desarrollo Web

a) Programación de funcionalidades.

i) Funcionalidades con PHP.

```
function submitContactData(){  
    require_once('../vendor/autoload.php');  
  
    $mail = new PHPMailer(true);  
  
    try{  
        $mail->isSMTP();  
        $mail->Host      = 'smtp.aeroelite.com';  
        $mail->SMTPAuth  = true;  
        $mail->Username   = 'info@aeroelite.com';  
        $mail->Password   = '12345';  
        $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;  
        $mail->Port       = 587;  
  
        // Remitente y destinatario  
        $mail->setFrom('info@aeroelite.com', 'AeroElite S.L.');//  
        $mail->addAddress($_POST['email']);  
  
        // Contenido  
        $mail->isHTML(true);  
        $mail->Subject = 'Confirmación de Contacto';  
        $mail->Body    = '<h1>Gracias por ponerte en contacto con nosotros</h1><p>Nos pondremos en contacto contigo a mayor brevedad posible</p>';  
        $mail->AltBody = 'Gracias por ponerte en contacto con nosotros. Nos pondremos en contacto contigo a mayor brevedad posible';  
  
        $mail->send();  
    }catch(Exception $e){  
        echo json_encode(['err' => $mail->ErrorInfo]);  
    }  
}
```

Función submitContactData()

Esta función en PHP se encarga de enviar un correo de confirmación a los usuarios que rellenan el formulario de contacto. Para ello utiliza la librería PHPMailer, una solución robusta y ampliamente utilizada para el envío de correos mediante SMTP.

1. Carga de PHPMailer:

Se incluye automáticamente mediante Composer con [autoload.php](#).

2. Configuración del servidor SMTP:

Se establece la conexión con un servidor SMTP (en este caso, [smtp.aeroelite.com](#)) usando autenticación, cifrado STARTTLS y puerto 587.

3. Remitente y destinatario:

El correo se envía desde `info@aeroelite.com` al email introducido por el usuario (`$_POST['email']`).

4. Contenido del mensaje:

Se define un asunto, cuerpo en HTML y versión alternativa en texto plano. Es un mensaje de agradecimiento tras el contacto.

5. Envío y control de errores:

Si el envío falla, se captura el error y se devuelve como JSON.

```

function signInNewUser() {
    require_once('../MODELOS/class.user.php');

    $user = new User();

    $validate = $user->signInNewUser($_POST['id'], $_POST['name'], $_POST['surname'], $_POST['email'], $_POST['nom_usu'], $_POST['pss']);

    if(!$validate) {
        echo json_encode(['err' => 'That user name is already used.']);
    }else{
        $_SESSION['nom_usu'] = $_POST['nom_usu'];

        $info = $user->getUser($_SESSION['nom_usu']);
        $_SESSION['type'] = $info[0][7] ?? 'G';

        echo json_encode([
            'nom' => $_SESSION['nom_usu'],
            'type' => $_SESSION['type']
        ]);

        if($_POST['check'] === 1){
            setcookie('email', $_POST['email'], time() + (30*24*3600));

            echo json_encode(['cookies' => true]);
        }
    }
}

```

```

public function signInNewUser($id, $nom, $ape, $email, $nom_u, $pass) {
    $sentencia = 'select count(nom_usu) from usuarios where nom_usu = ?';

    $consulta = $this->con->prepare($sentencia);
    $consulta->bind_param('s', $nom_u);
    $consulta->bind_result($cont);
    $consulta->execute();

    if($consulta->fetch() && $cont > 0) {
        $consulta->close();

        return false;
    }else{
        $consulta->close();

        $sentencia = 'insert into usuarios values (?,?,?,?,?,?, 0, 1, "R")';

        $consulta = $this->con->prepare($sentencia);
        $consulta->bind_param('ssssss', $id, $nom, $ape, $email, $nom_u, $pass);
        $consulta->execute();

        if($consulta->affected_rows > 0) {
            $consulta->close();

            return true;
        }else{
            $consulta->close();

            return false;
        }
    }
}

```

Función `signInNewUser()`

Esta función permite registrar un nuevo usuario en la aplicación. Se apoya en la clase `User` para gestionar las operaciones con la base de datos.

1. Carga del modelo de usuario:

Se importa el archivo `class.user.php`, que contiene la clase `User` con acceso a la base de datos.

2. Validación del nombre de usuario:

Se llama al método `signInNewUser()` del modelo, que primero ejecuta una consulta SQL para verificar si el nombre de usuario (`nom_usu`) ya existe en la base de datos.

3. Registro del usuario:

Si el nombre de usuario no existe, se inserta un nuevo registro en la tabla `usuarios`.

4. Los campos insertados incluyen ID, nombre, apellidos, email, nombre de usuario, contraseña y valores por defecto para rol y estado.

5. Inicio de sesión automático:

Si el registro fue exitoso:

- Se guarda el nombre de usuario en la sesión.
- Se consulta su tipo de usuario y se guarda también en la sesión.
- Se devuelve esta información como JSON para su uso en frontend.

6. Opción "Recordarme":

Si el usuario selecciona la opción de recordar su correo (`$_POST['check'] == 1`), se guarda una cookie con su email por 30 días.

```
function obtainTotalIncome(){
    require_once ('../MODELOS/class.maintenance.php');
    require_once ('../MODELOS/class.viaje.php');
    require_once ('../MODELOS/class.user.php');
    require_once ('../MODELOS/class.membership.php');

    $mant = new Maintenance();
    $viaje = new Viaje();
    $user = new User();
    $membership = new Membership();

    $users = $user->getAllUsers();
    $viajes = $viaje->getViajes();
    $members = $membership->getMemberships();
    $mants = $mant->getMantenimientos();

    $mem1 = 0;
    $mem2 = 0;
    $mem3 = 0;

    for ($i = 0; $i < count($users); $i++) {
        if($users[$i]['membresia'] == 2) $mem1++;
        if($users[$i]['membresia'] == 3) $mem2++;
        if($users[$i]['membresia'] == 4) $mem3++;
    }

    $totalMem1 = $mem1 * $members[0]['precio'];
    $totalMem2 = $mem2 * $members[1]['precio'];
    $totalMem3 = $mem3 * $members[2]['precio'];

    $totalMems = $totalMem1 + $totalMem2 + $totalMem3;

    $totalViajes = 0;

    for ($i = 0; $i < count($viajes); $i++) {
        $totalViajes += $viajes[$i]['precio'];
    }

    $totalIngreso = $totalViajes + $totalMems;

    $totalGasto = 0;

    for ($i = 0; $i < count($mants); $i++) {
        $totalGasto += $mants[$i]['costo'];
    }

    echo json_encode([
        'income' => $totalIngreso,
        'outcome' => $totalGasto
    ]);
}
```

Esta función se encarga de calcular los ingresos y gastos totales de la empresa a partir de la información de usuarios, membresías, vuelos y mantenimientos. Es clave para las estadísticas contables del sistema.

1. Carga de modelos

Se incluyen los modelos **Maintenance**, **Viaje**, **User** y **Membership**, necesarios para acceder a la información relacionada.

2. Instanciación de objetos

Se crean objetos de cada clase para poder ejecutar sus métodos.

3. Obtención de datos

- **getAllUsers()** obtiene todos los usuarios registrados.
- **getViajes()** devuelve todos los vuelos.
- **getMemberships()** proporciona la información de precios de las membresías.
- **getMantenimientos()** obtiene los mantenimientos realizados y sus costes.

4. Cálculo de ingresos por membresía

Se cuentan cuántos usuarios tienen cada tipo de membresía (**2**, **3**, **4**), y se multiplica por su precio correspondiente para obtener el ingreso por cada tipo. Luego se suma todo.

5. Cálculo de ingresos por viajes

Se suman los precios de todos los vuelos registrados.

6. Cálculo de gastos

Se suman los costes de todos los mantenimientos realizados.

7. Devolución del resultado

Finalmente, se devuelve un JSON con el ingreso total (**income**) y el gasto total (**outcome**).

ii) Funcionalidades con AJAX.

```
const calcularItinerario = () => {
  const aeropuertoSalida = aeropuertos.find(a => salida.includes(a.iata || a.icao));
  const aeropuertoLlegada = aeropuertos.find(a => llegada.includes(a.iata || a.icao));
  const avion = aviones.find(a => a.matricula === plane);

  if (!aeropuertoSalida || !aeropuertoLlegada || !avion) return;

  if ((aeropuertoSalida.iata || aeropuertoSalida.icao) === (aeropuertoLlegada.iata || aeropuertoLlegada.icao)) {
    alert('Departure and arrival airports cannot be the same.');
    setVisible(false);
    return;
  }

  const dist = calcularDistancia(aeropuertoSalida.lat, aeropuertoSalida.lon, aeropuertoLlegada.lat, aeropuertoLlegada.lon);
  const horas = dist / 900;
  const escalas = Math.floor(dist / avion.rango_max);
  const precioHora = calcularPrecioHora(avion.rango_max);
  const precioEscalas = calcularCostoEscalas(avion.rango_max, escalas);

  let precio = Math.ceil(horas) * precioHora + precioEscalas;

  if (puedeReservar) {
    precio = 0;
  }

  setDistancia(dist.toFixed(2));
  setDuracion(horas.toFixed(2));
  setEscalas(escalas);
  setPrecioTotal(precio);
  setSelectedPlane(avion);
  setVisible(true);
}
```

Activar Windows

Función `calcularItinerario()`

Esta función es responsable de calcular la ruta, duración y precio estimado de un itinerario de vuelo a partir de los aeropuertos seleccionados y el avión elegido.

1. Búsqueda de datos relevantes

Se localizan el aeropuerto de salida, el de llegada y el avión seleccionado, usando códigos IATA/ICAO y matrícula del avión.

2. Validaciones básicas

- Si alguno de los datos no se encuentra, se cancela la ejecución.
- Si los aeropuertos de salida y llegada son el mismo, se muestra una alerta y se cancela

la operación.

3. Cálculos de itinerario

Se realiza una serie de cálculos usando funciones auxiliares:

- Distancia total del trayecto
([calcularDistancia](#))
- Duración del vuelo (dividiendo la distancia entre 900 km/h, velocidad promedio)
- Número de escalas (dividiendo la distancia entre el rango máximo del avión)
- Precio por hora de vuelo
([calcularPrecioHora](#))
- Precio adicional por escalas
([calcularCostoEscalas](#))
- Cálculo final del precio, redondeando a la hora más próxima.

4. Condición especial de precio

Si el usuario tiene permiso para reservar gratis ([puedeReservar](#)), el precio se fuerza a 0.

5. Actualización del estado

Se actualiza el estado del componente con todos los datos calculados, haciendo visible el resumen del itinerario.

Librerías o dependencias:

- React: uso de `useState` para manejar el estado del itinerario.
- Funciones auxiliares personalizadas: `calcularDistancia`, `calcularPrecioHora`, `calcularCostoEscalas`, que encapsulan la lógica de cálculo.

```
useEffect(() => {
  const aeropuertoSalida = aeropuertos.find(a => salida.includes(a.iata || a.icao));
  const aeropuertoLlegada = aeropuertos.find(a => llegada.includes(a.iata || a.icao));
  if (!aeropuertoSalida || !aeropuertoLlegada) return;

  const continent1 = aeropuertoSalida.tz.split('/')[0];
  const continent2 = aeropuertoLlegada.tz.split('/')[0];

  if (membership == 2 && aeropuertoSalida.country === aeropuertoLlegada.country) {
    setPuedeReservar(true);
  } else if (membership == 3 && continent1 === continent2) {
    setPuedeReservar(true);
  } else if (membership == 4) {
    setPuedeReservar(true);
  } else {
    setPuedeReservar(false);
  }
}, [salida, llegada, aeropuertos, membership])
```

```
!puedeReservar ? (
  <p className='text-red-700 text-center font-semibold my-4'>Your book does not match with your membership</p>
) : (
  <p className='text-green-600 text-center font-semibold my-4'>Your book matches your membership</p>
)
```

`useEffect` para Validación de Reserva según Membresía

Esta función determina si un usuario puede realizar una reserva de vuelo en función de su nivel de membresía y la localización geográfica del trayecto.

1. Búsqueda de aeropuertos

A partir de los códigos de salida y llegada, se localizan los objetos completos de los aeropuertos usando:

2. Segmentación por continente y país

Se extrae el continente de cada aeropuerto desde

su zona horaria (`tz.split('/')[0]`) y se compara su país.

3. Condiciones de membresía

La lógica establece si el usuario puede reservar gratuitamente dependiendo del nivel de su membresía (`membership`):

- Nivel 2: sólo vuelos dentro del mismo país.
- Nivel 3: vuelos dentro del mismo continente.
- Nivel 4: vuelos internacionales sin restricciones.
- Si ninguna condición se cumple, la reserva gratuita se deniega.

4. Actualización del estado

Se actualiza la variable de estado `puedeReservar`, que se utiliza para controlar el acceso y la visualización.

Se han elegido las funcionalidades más complejas e interesantes para explicar.

b) APIs utilizadas

Se ha utilizado un JSON con todos los aeropuertos a nivel mundial, el cual incluye información como la latitud, longitud, código IATA entre otras:

```
"LEGR": {  
    "icao": "LEGR",  
    "iata": "GRX",  
    "name": "Federico Garcia Lorca Airport",  
    "city": "Granada",  
    "state": "Andalusia",  
    "country": "ES",  
    "elevation": 1860,  
    "lat": 37.1887016296,  
    "lon": -3.7773599625,  
    "tz": "Europe/Madrid"  
},
```

Obtenido de:

<https://github.com/jbrooksuk/JSON-Airports>

7) Manual de usuario.

1. INICIO DE SESIÓN

En la página principal encontrará la opción:

- "Sign In"

1.1. Inicio de Sesión

Haga clic en "Log In"

Complete el formulario:

- Email: Ingrese su correo electrónico
- Password: Ingrese su contraseña
- Remember me: Active esta casilla si desea que la sesión se mantenga iniciada
- Haga clic en "SUBMIT"

1.2. Creación de Cuenta

Haga clic en "Sign In"

Complete el formulario:

- Id: Su documento de identidad (DNI o similar)
- Name: Nombre propio
- Surname: Apellidos
- Email: Correo electrónico
- Username: Nombre de usuario deseado
- Password: Contraseña segura
- Remember me: (Opcional) Mantener la sesión iniciada

Haga clic en "SUBMIT"

2. USUARIOS REGISTRADOS (ROL R)

2.1. DASHBOARD

2.1.1. Al iniciar sesión, verá una pantalla con un mensaje personalizado como: "Dear [nombre_usuario]", seguido de:

- ID
- Nombre y apellidos
- Nombre de usuario
- Correo electrónico
- Horas de vuelo acumuladas

2.1.2. Programación de Vuelos

En la sección "Your schedule:" verá una lista de vuelos programados con:

- Fecha de salida
- Lugar de llegada
- Información adicional expandible:
 - Lugar de salida
 - Modelo de avión

- Duración estimada
- Distancia
- Precio
- Indicador de si su membresía permite ese vuelo

2.1.3. Perfil de Usuario

En el panel lateral derecho podrá consultar:

- Foto de perfil
- Información personal
- Membresía activa
- Horas de vuelo acumuladas

2.2. GESTIÓN DE VUELO

Funcionalidad disponible para usuarios con vuelos asignados

- Ver lista de vuelos asignados
- Detalles:
 - Descripción del vuelo
 - Fecha límite
 - Estado actual

2.3. COMUNICACIÓN (Página de Contacto)

Los usuarios pueden contactar con el equipo de administración a través de la página de contacto accesible desde el menú.

- Complete el formulario con su correo electrónico
- El sistema enviará un mensaje automático de confirmación
- El equipo de AeroElite responderá a la mayor brevedad

3. ADMINISTRADORES (ROL A)

3.1. PROGRAMACIÓN DE VUELOS

3.1.1. Vista General

Al iniciar sesión, accederá a la sección de programación. Allí encontrará una tabla con todos los vuelos programados:

- Usuario
- Modelo de avión
- Fecha
- Lugar de salida y llegada
- Distancia
- Duración
- Botón para cancelar vuelos

3.1.2. Filtros de Búsqueda

Puede filtrar los vuelos en tiempo real por:

- Nombre de usuario
- Fecha del vuelo

3.1.3. Cancelación de Vuelos

- Haga clic en el botón "CANCEL"
- Confirme la cancelación
- El sistema mostrará un mensaje de confirmación
- La tabla se actualizará automáticamente

4. REPORTES Y ESTADÍSTICAS

4.1. Reportes Generales

- Progreso de vuelos
- Vuelos completados
- Uso de recursos

4.2. Visualizaciones

- Gráficos de barras y pastel

- Métricas clave
- Comparativas entre periodos

5. CONFIGURACIÓN DEL SISTEMA

5.1. Perfil

- Ver y modificar membresía (mediante su compra)

6. PREGUNTAS FRECUENTES (FAQ)

6.1. Problemas Comunes

- ¿No puedo iniciar sesión?
 - Verifique conexión a Internet
 - Use Google Chrome (v90 o superior)
 - Limpie caché y cookies
 - Contacte a soporte si persiste el error
- ¿Cómo cancelar un vuelo?
 - Póngase en contacto a través de la página Contact
- ¿Cómo puedo ver mi programación de vuelos?
 - Inicie sesión como usuario registrado
 - Acceda al Dashboard
 - Consulte la sección "Your schedule"

7. SOPORTE TÉCNICO

Email: info@aeroelite.com

Teléfono: +1 (234) 567-890

Ubicación: Miami, FL, USA

Horario: Lunes a Viernes, 9:00 – 18:00

Por Jose Luis Ruiz Romero.