

Visión general

Flight Landers es una plataforma de búsqueda y reserva de vuelos totalmente imparcial y sin comisiones pensada para proporcionar al usuario una selección amable e intuitiva de las opciones que mejor se ajusten a sus necesidades.

Objetivos

1. Flight Landers buscará las mejores opciones en la API de Amadeus, una de las empresas de software más importantes del mundo que trabaja con aerolíneas, aeropuertos y los motores de búsqueda de vuelos más importantes. Amadeus tiene una amplia experiencia en este mercado, ya que ha gestionado más de 645 millones de reservas de vuelos durante el año 2019 y su API es una de las más completas y utilizadas.
2. El buscador de vuelos de Flight Landers posibilitará que, a través de sus filtros, el usuario encuentre el vuelo que mejor encaja con él, desde una reserva rápida sin criterios concretos a escoger hasta un vuelo con escalas, en varios días distintos y en primera clase.
3. Cualquier usuario tendrá la posibilidad de interactuar con nuestro buscador y comprobar sus funcionalidades, pero sólo los usuarios registrados podrán realizar reservas de vuelos.
4. La aplicación será compatible con cualquier navegador moderno haciéndola accesible en cualquier dispositivo móvil sin que la funcionalidad se vea comprometida.

Especificaciones

Los recursos gráficos estarán principalmente en formato SVG.

La paleta de colores será amigable y reconocible (Convenciones del sector, azules, blancos, amarillos... incluyendo variaciones de distinción personal).

La tipografía por defecto seleccionada para el proyecto será Atkinson-Hyperlegible, proporcionada por el "Braille Institute" y que facilita la legibilidad a personas con problemas de visión.

Se pondrá especial atención a que la información legible, justa y necesaria para una comprensión correcta.

Partiremos de la filosofía Mobile First

Estructura de Contenidos:

El proyecto se presentará inicialmente con varias páginas: Buscador de vuelos, Respuesta de búsqueda, Registro de usuario, Login, Gestión del perfil de usuario, Histórico de reservas de usuario y About.

Páginas:

I. Buscador de Vuelos

Por defecto el buscador estará configurado para buscar billetes de ida y vuelta en el mismo día para un solo adulto, sin limitación de precio y ordenados por precio de manera ascendente, pero el usuario podrá modificar todos estos ajustes de manera sencilla en la misma interfaz de búsqueda.

El usuario deberá de escoger el aeropuerto de origen y de destino.

La base de datos de la aplicación pone a disposición del usuario la lista de aeropuertos. El usuario rellenará el formulario con el nombre de la ciudad, aeropuerto o código IATA indistintamente.

No entra dentro del alcance inicial del proyecto que el aeropuerto de origen se auto rellene a partir de los sensores de ubicación del dispositivo empleado. Aunque se valora añadir esa funcionalidad en futuras revisiones del proyecto.

La consulta se enviará presionando un botón.

II. Respuesta de la Consulta

Los resultados se presentarán ordenados por defecto según el precio de menor a mayor. Aunque desde la misma interfaz de respuesta, mediante unos selectores siempre visibles en la cabecera, se podrá invertir el orden de filtrado, ordenar por paradas, y por duración; así como modificar rápidamente la consulta sin necesidad de volver al buscador inicial.

El usuario podrá entrar en cada uno de los vuelos de respuesta para ampliar la información y decidir si procede a la reserva pulsando un sobre el corazón

Se creará una tabla en la base de datos para llevar el registro de reservas por usuario y un servicio para reservar el vuelo a nombre del usuario.

Datos consumidos:

Dato	Descripción	Tipo almacenado
source	Tipo de compañía, GDS o LCC	CHAR(3)
lastTicketingDate	Fecha de disponibilidad	DATE
duration	Duración del vuelo (ISO 8601)	VARCHAR(10)
segment	nº de paradas. Se obtiene de segment.length	TINYINT
departure	IATA aeropuerto origen. segment[0].departure.iataCode	CHAR(3)
arrival	IATA destino. segment[segment.length-1].arrival.iataCode	CHAR(3)
departure-time	ISO 8601 date "YYYY-MM-DDTHH:MM:SS". Se obtiene de: segment[segment.length-1].arrival.at	CHAR(19)
carrier	Compañía aérea	CHAR(2)
price	Precio del billete	DECIMAL(7,2)
currency	Divisa	VARCHAR(3)

Confirmación de reserva:

Una vez gestionada la reserva, se enviará un email al usuario con la confirmación de su reserva y una copia de dicha reserva en formato pdf o wallet.

III. Registro de usuario

Para acceder al formulario de registro se habilitará un botón de navegación en el header de la aplicación que nos redirigirá a un formulario.

Del mismo modo habrá un enlace al formulario en el panel de inicio de sesión para usuarios no registrados.

El usuario deberá cubrir los datos especificados y enviar los datos con un botón.

Conforme al principio de minimización de datos del RGPD estos serán los indispensables para la prestación del servicio. Entiéndase: Email, Contraseña.

Si el usuario lo desea podrá subir un avatar de perfil así como escribir una pequeña bio, aunque no es obligatorio.

Cada email será único en el sistema y la contraseña se guardará encriptada en una sola dirección.

Se pondrá a disposición del usuario el EULA pertinente y un checkbox de aceptación de los mismos.

A su vez se establecerá un campo de Status y permisos en la base. El usuario validará su cuenta con un email de verificación para adquirir permisos de reserva.

IV. Inicio de sesión.

En la vista de inicio de sesión el usuario podrá acceder al sistema introduciendo su email y contraseña.

También dispondrá de un enlace para navegar al formulario de registro.

Para acceder al formulario de login se habilitará un botón de navegación en el header de la aplicación. Asimismo, se redirigirá al formulario de login cuando un usuario no registrado intente hacer una reserva sin haber iniciado sesión.

El usuario podrá cerrar sesión desde el menú principal.

V. Gestión del perfil de usuario.

Sólo se podrá acceder con la sesión iniciada. Para acceder a la gestión del perfil se habilitará un botón de navegación sustituyendo al que servía para hacer Login. Se creará una vista del perfil con posibilidades de CRUD a la base.

El usuario podrá decidir si guardar o descartar los cambios mediante dos botones. En el caso de querer guardar aparecerá un pop-up de confirmación.

Para cerrar la sesión, la sesión debe de estar iniciada previamente. El botón que servía para registrarse será el de cerrar sesión. Como añadido, la sesión podrá expirar si se permanece inactivo durante un tiempo determinado, teniendo que volver a hacer Login.

El usuario podrá desactivar o eliminar su cuenta.

Si la desactiva podrá realizar búsquedas pero no reservas.

Si la elimina, en la BBDD se le cambiará el estado a eliminado, pero no se borrará.

Dicho email quedará bloqueado para siempre y no se podrá reactivar.

Esta información se le será transmitida al usuario tanto en el EULA como en el momento en que indique su deseo de eliminar la cuenta mediante el formulario.

VI. Histórico de reservas de usuario.

Sólo se podrá acceder con la sesión iniciada. Se accede desde el menú principal. En esta vista se mostrarán todas las reservas efectuadas por el usuario. Si hay una reserva activa, y está dentro del plazo, se podrá cancelar presionando el botón adecuado.

VII. About

Esta página será accesible desde el menú principal de la aplicación.

En ella se mostrará información sobre la aplicación:

Versión. Año, Desarrolladores, Motivación, Créditos, Agradecimientos, Contacto...

Datos a Manejar:

Se podrá encontrar una relación detallada de todos los datos manejados por la aplicación en el documento adjunto: YYMMDD_Data_Breakdown.

Paradigmas:

I. Mobile first.

El diseño será responsive partiendo de una resolución mínima de 320x568px. Partimos de HTML y CSS puros con vistas a implementar un Framework de React. Una vez gestionada la reserva, se enviará un email al usuario con la confirmación de su reserva y una copia.

II. Privacy by design.

Los datos almacenados en la base serán los mínimos necesarios para la prestación del servicio. Los datos serán alojados en una base de datos MySQL. Dado que es un sistema de reservas y no de compras no es necesario el almacenamiento de ningún dato de carácter personal que pudiera identificar al usuario en la vida real. La reserva está vinculada a un código identificador de usuario, no a un dato personal.

Metodología:

Simularemos un entorno de desarrollo Agile en su modalidad Scrum. Entendemos que HAB es el cliente. No podemos pedirle al director de proyecto (Berto) que realice las tareas de un Product Owner así que nos auto regularemos mediante un PO imaginario al que enriquecemos con los feedbacks proporcionados por Berto.

El Scrum Team solo tiene dos integrantes, así que como estamos en un entorno formativo creemos que puede resultar muy enriquecedor ir alternando la figura del Scrum Master entre Sprints.

La duración inicial de cada Sprint es de dos semanas. La idea es implantar este modelo a partir de la reunión con Berto el 2020/12/10. Adoptando Pablo Viña el rol de SM en este primer sprint.

Entre las herramientas de apoyo para el desarrollo contamos con una serie de tableros de preproducción en Trello. Por recomendación de nuestro PO nos servimos de [Basecamp](#) para el diseño y seguimiento de las épicas, tareas, incidencias...

Emplearemos la metodología GitFlow a la hora de desarrollar el software. Existirá una rama principal en la que se depositará cada incremento. Existirá una rama de desarrollo de la que surgirán subramas para las diferentes funcionalidades. Cada tarea pasa por estos cuatro estados. ToDo, Development. Testing, Done. Una tarea sólo se considerará hecha cuando el otro integrante del Scrum Team haya revisado y validado el código.