

CityTour

Web App de exploración turística

The logo of the Universitat Oberta de Catalunya (UOC), consisting of the letters 'UOC' in a bold, blue, sans-serif font.

Juan Camilo Tejada Lozano

CityTour
Aplicaciones interactivas

Nombre Tutor/a de TF

Daniel Julià Lundgren

**Profesor/a responsable de
la asignatura**

Carlos Casado Martínez

Junio de 2023

Universitat Oberta
de Catalunya



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>CityTour</i>
Nombre del autor:	<i>Juan Camilo Tejada Lozano</i>
Nombre del consultor/a:	<i>Daniel Julià Lundgren</i>
Nombre del PRA:	<i>Carlos Casado Martínez</i>
Fecha de entrega (mm/aaaa):	<i>06/2023</i>
Titulación o programa:	Grado en Técnicas de Interacción Digital y Multimedia
Área del Trabajo Final:	<i>Aplicaciones interactivas</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Web App, Turismo, NextJs, React</i>
Resumen del Trabajo	
<p>CityTour consiste en una aplicación web que permita a aquellas personas interesadas en realizar un viaje a una ciudad descubrir lugares nuevos que se ajusten a sus preferencias, a partir de las recomendaciones de los usuarios de la plataforma.</p> <p>El sistema permite a los usuarios publicar los lugares o rutas (agrupaciones de lugares), así como realizar búsquedas sobre el contenido existente y guardar su preferencias de contenido.</p> <p>A nivel técnico, la aplicación utiliza en el front-end NextJS y React, y en el back-end, Supabase. Por otra parte, se emplean librerías de componentes como NextUI o Bootstrap para ofrecer una mejor experiencia de usuario y agilizar el desarrollo, así como el uso de la API de GoogleMaps para consultar información geográfica de cada lugar y mostrar la posición de los lugares a través del mapa interactivo.</p> <p>A través de estas tecnologías, muy utilizadas en la actualidad, se han podido explorar sus posibilidades, aprender sobre el desarrollo de un proyecto digital en este nuevo entorno y ofrecer un prototipo que cumpla con los requisitos planificados.</p>	
Abstract	
CityTour is a Web Application targeted at users who want to visit a city to discover new places that fit their preferences, from the recommended spots uploaded by the platform users.	

The system allows users to publish places or routes (A grouping of places), as well as perform searches on the existing content and saving their user preferences.

In the technical section, the App uses NextJs and React on the front-end and Supabase on the back-end. On one hand, the use of external component libraries like NextUI or Bootstrap allows a better user experience and an improvement in the development time. On the other hand, GoogleMaps API is used to query geographical data of each place and to show the position of each spot through the interactive map.

Through these technologies, fairly popular nowadays, it has been possible to explore its possibilities and learn about the development of a digital project in this new environment as well as offer a prototype that meets the planned requirements.

Índice

1. Introducción.....	3
1.1 Contexto y justificación del Trabajo	3
1.2 Fuentes de información turística	4
1.3 Análisis de mercado	5
1.4.Objetivos del Trabajo.....	5
1.4 Impacto en sostenibilidad, ético-social y de diversidad.....	6
1.5 Enfoque y método seguido	6
1.6 Planificación del Trabajo	7
1.7 Breve resumen de productos obtenidos.....	8
1.8 Breve descripción de los otros capítulos de la memoria	8
2. Requisitos del proyecto	9
3. Materiales y métodos	10
2.1 Casos de uso	11
2.2 Diagrama de flujo	15
2.3 Diseño de base de datos.....	17
2.4 Prototipo	21
4. Resultados	28
3.1 Implementación del proyecto.....	28
3.2 Componentes	31
3.3 Base de datos	32
3.4 Estilos.....	33
3.5 Logo	33
3.6 Funcionamiento	33
3.7 Acceso a sitio web.....	44
5. Conclusiones y trabajos futuros	45
Mejoras futuras.....	45
6. Glosario.....	47
7. Bibliografía	49
8. Créditos.....	51

Lista de ilustraciones

Ilustración 1. Gráfico turismo.....	3
Ilustración 2. Planificación en Microsoft Project	8
Ilustración 3. Casos de uso	12
Ilustración 4. Diagrama de flujo – Crear contenido.....	15
Ilustración 5. Diagrama de flujo – Realizar búsqueda	16
Ilustración 6. Diagrama de flujo – Ver datos de elemento	16
Ilustración 7. Estructura de base de datos	17
Ilustración 8. Estructura Bucket de lugares	18
Ilustración 9. Estructura Bucket de avatares	18
Ilustración 10. Vista SQL de interacciones con lugar	20
Ilustración 11. Políticas RLS en Supabase	20
Ilustración 12. Pantalla de búsqueda. Visualización de filtros	21
Ilustración 13. Pantalla de búsqueda. Resultados de búsqueda de rutas.	22
Ilustración 14. Pantalla de búsqueda. Resultados de búsqueda de lugares. ...	22
Ilustración 15. Visualizador de lugar.....	23
Ilustración 16. Visualizador de ruta	24
Ilustración 17. Perfil de usuario (Listado de rutas)	25
Ilustración 18. Perfil de usuario (Listado de lugares).....	25
Ilustración 19. Edición de ruta	26
Ilustración 20. Edición de lugar	27
Ilustración 21. Barra de navegación sin sesión iniciada	27
Ilustración 22. Estructura de proyecto	28
Ilustración 23. Páginas del sitio web	28
Ilustración 24. Componente MyApp	29
Ilustración 25. Componente Document	30
Ilustración 26. Ruta de API de Google Maps	30
Ilustración 27. Componente MessageWrapper	31
Ilustración 28. Listado de componentes	31
Ilustración 29. Logotipo City Tour.....	33
Ilustración 30. Colores de marca.....	33
Ilustración 31. Favicon de sitio	33
Ilustración 32. Barra de navegación (Sesión iniciada).....	34
Ilustración 33. Barra de navegación (Móvil)	34
Ilustración 34. Barra de navegación (Sin sesión iniciada)	34
Ilustración 35. Pantalla de visor de lugar.....	37
Ilustración 36. Ventana modal de imagen de lugar	37
Ilustración 37. Pantalla de visor de ruta	38
Ilustración 38. Pantalla de visor.....	39
Ilustración 39. Edición de lugar. Selección de ciudad	39
Ilustración 40. Edición de lugar. Selección de coordenadas	40
Ilustración 41. Edición de lugar. Introducción de datos	40
Ilustración 42. Edición de lugar. Mensaje de actualización con éxito	41
Ilustración 43. Pantalla de edición de ruta.....	41
Ilustración 44. Ventana de confirmación de cancelación de ruta.....	42
Ilustración 45. Pantalla de Perfil de usuario	43
Ilustración 46. Perfil de usuario, edición activada.....	43
Ilustración 47. Inicio de sesión	44

1.Introducción

La propuesta que plantea este proyecto consiste en una aplicación Web que permita a los usuarios compartir sus puntos de Interés de una zona geográfica. A través de esta plataforma, los lugares se pueden buscar por categorías y descubrir zonas recomendadas por otras personas, así como guardar las rutas por usuario.

La idea surge de una necesidad detectada cuando se realizan viajes a ciudades que no conocemos y que cuando queremos descubrirlas es posible que nos encontremos con una gran cantidad de recomendaciones turísticas genéricas que puede que no se ajusten totalmente a lo que buscamos, o que dejen de lado sitios que nos podrían interesar.

1.1 Contexto y justificación del Trabajo

Debido a los efectos de la pandemia del COVID-19, el turismo estuvo muy limitado durante el año 2020 hasta la progresiva relajación de restricciones en el año 2022. A partir de este cambio, se prevé que los posibles viajeros opten por hacer igual o más desplazamientos que antes de la pandemia¹. Además, en una encuesta realizada en 2020² los encuestados respondían en un 63% que preferían realizar viajes internacionales frente a nacionales una vez se relajaran las medidas sanitarias.

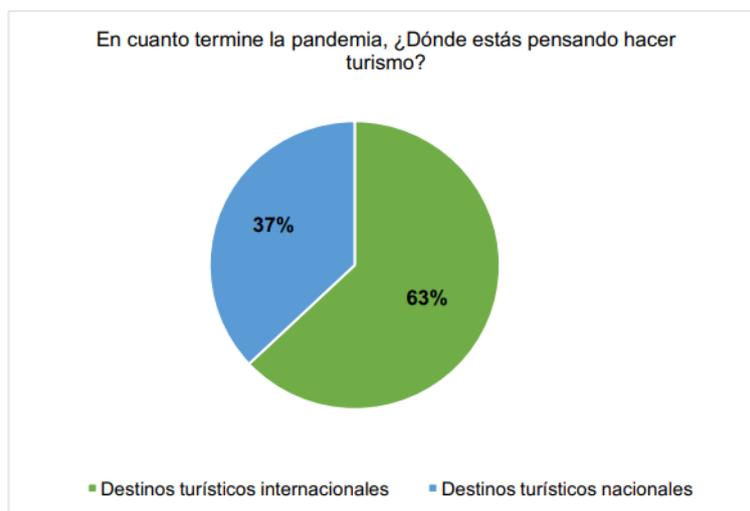


Ilustración 1. Gráfico turismo

Fuente: POST-PANDEMIA Y TURISMO SLOW: ANÁLISIS Y ESTUDIO EMPÍRICO. Nuria Portela Gabian

En otro apartado de esa misma encuesta de 2020², los principales motivos de viaje fueron: 'Vivir experiencias únicas' y 'Conocer nuevas ciudades'. Por ello, el servicio planteado es conveniente para facilitar el descubrimiento de lugares únicos y adaptables a las preferencias del usuario, además de ser de gran utilidad para que las ciudades entiendan mejor las preferencias de sus turistas³. En cuanto a la necesidad de los turistas de informarse sobre su destino, en un estudio realizado en Guayaquil, Ecuador, se detecta que entre las expectativas en un viaje post-COVID, en el apartado de 'Cuidado digital' en los viajes, toman especial importancia los asistentes virtuales de información turística, seguidos de otros servicios digitales de medidas sanitarias.

En el estudio de 2012 realizado por Rosa Marrero y Beatriz Abdul⁷ sobre el turismo en La Laguna, Tenerife, se perfilaron dos tipos de turistas, los intencionales y los accidentales. El primer grupo suele realizar una planificación previa mediante guías de viaje e Internet, mientras que el segundo, prefiere informarse en destino (hotel, publicidad, guías turísticos).

Ante estos dos tipos de usuarios potenciales se estima que el servicio pueda ser útil para ambos, ya sea para servir de punto inicial de búsqueda para los turistas intencionales, o para mostrar lugares de interés en destino de forma sencilla para los accidentales.

1.2 Fuentes de información turística

Vemos que la formas más populares de informarse de los turistas sobre su plan de viaje se centran en páginas web, blogs de viaje y redes sociales⁴. Es especialmente relevante el efecto de las redes sociales, ya que en estudios como el de Sunny Sun en 2017⁵, se descubre que los consumidores se ven muy influenciados por el boca a boca y que aprecian la calidad de la información en las redes sociales, en donde el género femenino tiene mayores expectativas en lo referente a la calidad de los datos.

Además en su estudio de 2018⁸ se reafirma este concepto ya que se demuestra que las redes sociales se perciben como el canal de búsqueda más importante a la hora de buscar información de viaje, mientras que las agencias de viaje se quedan como el canal más importante de búsqueda de información relacionada con el hotel. Por ello, la propuesta se desarrolla en un entorno social en el que los usuarios pueden compartir sus preferencias, permitiendo a los turistas potenciales tener un punto de referencia adicional.

En relación a la pericia en el uso de los motores de búsqueda para buscar información sobre los viajes, los usuarios experimentados son capaces de encontrar lo que buscan con más facilidades gracias a los modificadores u opciones de búsqueda avanzada, mientras que el resto de los usuarios suelen distraerse con información irrelevante⁴. En consecuencia, la Web propuesta

busca ser un opción de búsqueda de planes turísticos que optimice el tiempo de búsqueda que emplean los usuarios.

1.3 Análisis de mercado

En cuanto a las opciones existentes podemos encontrar desde blogs o videoblogs de viaje a aplicaciones dedicadas al tratamiento de puntos de interés a través de mapas.

Entre estas aplicaciones encontramos algunas como AroundMe, la cual permite mostrarnos lugares cercanos a nuestra posición y filtrarlos por unas categorías específicas (bares, hospitales, hoteles, etc.). Su principal debilidad es su limitada selección de categorías de lugares y que no ofrece ningún tipo de interacción con otros usuarios.

Minube es una aplicación Web, que está destinada a descubrir lugares turísticos del mundo. Su principal atractivo es que ofrece una visualización cómoda y agradable de los posibles sitios, además de que permite a los usuarios ofrecer sus experiencias mediante comentarios y publicar nuevos sitios recomendados. En cuanto a su debilidad, vemos que la gran mayoría de publicaciones son sitios de turismo general (playas, castillos, palacios, parques, monumentos...) y no abundan los lugares relacionados con aficiones (Cine, música, ocio...) ya que está orientada a visitas turísticas 'tradicionales'.

TripAdvisor contiene una sección en su web 'Things to do', que permite buscar lugares ya sea recomendados por usuarios o tours con opción de reserva. Su puntos fuertes son que es una página web ya establecida en el mercado que goza de un amplio número de usuarios que ya la utilizan, ofrece una serie de categorías predefinidas que pueden facilitar la búsqueda, tiene opciones de traducción y recomendaciones en base a los lugares consultados. Su debilidad es que limita su búsqueda a las categorías establecidas por el sistema y que no permite mostrar lugares agrupados por ruta de usuario.

1.4. Objetivos del Trabajo

En este TFG se busca realizar un estudio del estado actual del turismo y la elección de lugares de visita en los viajes, así como analizar el mercado de aplicaciones turísticas para incorporar funcionalidades no desarrolladas o exploradas.

El trabajo, plantea una aplicación web que permita a los usuarios consultar lugares y rutas según sus preferencias mediante una plataforma intuitiva y funcional, así como publicar contenido y almacenar sus interacciones con los lugares disponibles.

Para lograr la propuesta planteada, se emplean los conocimientos web obtenidos y se empleará el entorno de trabajo NextJs. La elección de este Framework se debe al interés personal por descubrir uno de los entornos de desarrollo web más

utilizados en la actualidad, por lo que se busca realizar la formación en NextJs a través de un proyecto completo desde cero.

Debido al límite de tiempo de desarrollo, se busca también adquirir la destreza para manejar un proyecto individual y la toma de decisiones ante los problemas que se presenten durante el proceso.

1.4 Impacto en sostenibilidad, ético-social y de diversidad

En el apartado de **sostenibilidad**, el producto en sí no produce un impacto negativo en el medio ambiente, pero sí está englobado en el marco de las tecnologías de la información, que son responsables de 2% de las emisiones de gases contaminantes⁶. Es por ello por lo que el efecto negativo se ve ligado al uso responsable de los dispositivos electrónicos y el uso de internet.

En la dimensión del **comportamiento ético** y **responsabilidad social**, se podría debatir que el proyecto podría poner en riesgo el trabajo de las agencias de viajes. Sin embargo, la página web busca ser un complemento a los planes de viaje, una forma de cubrir las carencias de las agencias cuyo fin es ofrecer servicios turísticos y gestionar las reservas.

Por otra parte, mediante la publicación de rutas, se puede dar visibilidad a establecimientos menos conocidos e impulsar su visibilidad entre usuarios afines.

En cuanto a la perspectiva de la **diversidad, género y derechos humanos**, la aplicación no busca generar un impacto negativo y está determinada a alojar rutas con gran variedad temática, permitiendo expresar la diversidad cultural, ideológica y sexual.

1.5 Enfoque y método seguido

Dado que el objetivo del proyecto es obtener un producto nuevo desde cero y utilizar NextJs como nuevo entorno de trabajo que requiere formación durante el proceso, se ha realizado una planificación que permita ofrecer un margen a los posibles percances.

Para lograrlo, se han utilizado herramientas como los casos de uso, diagramas de flujo y wireframe que permitan agilizar la fase de desarrollo.

Por otra parte, se han enumerado las funciones esenciales que permiten a la aplicación cumplir su objetivo básico, además de funciones secundarias que se podrán ir desarrollando en base al avance del proyecto.

1.6 Planificación del Trabajo

Para llevar a cabo el proyecto, se han planificado los aspectos de diseño y de desarrollo. En el apartado de diseño se han seleccionado tres herramientas: Casos de uso, diagrama de flujo y wireframe. En base al tiempo total disponible, se han dejado fuera otras opciones muy útiles como el uso de Personas, Card Sorting, Experience maps o User journeys.

Fases

1. Definir casos de uso
2. Definir diagrama de flujo
3. Diseñar wireframe
4. Desarrollo
5. Estructura de base de datos y conexión.
6. Desarrollo de pantallas y componentes
 - a. Pantalla Inicio y barra de navegación común
 - b. Pantalla visualizadora de ruta
 - c. Pantalla visualizadora de lugar
 - d. Pantalla de búsqueda
 - e. Pantalla de resultados de búsqueda
 - f. Pantalla de perfil de usuario
 - g. Pantalla de creación de ruta
 - h. Pantalla de creación de lugar
 - i. Pantalla de *login* *
7. Codificación de lógica de negocio
 - a. Consulta, filtrado y visualización de datos
 - b. Lógica de controles y componentes
 - c. Interacción de usuario con mapa
8. Adaptación a diseño responsive
9. Despliegue de sitio web
10. Fase de pruebas y corrección de errores
11. Estructura final de la memoria
12. Preparación de la presentación

En la planificación se han tenido en cuenta los posibles inconvenientes que pueden surgir en el desarrollo, por ello, se han marcado con asterisco (*) los elementos de la aplicación no esenciales para representar la funcionalidad del proyecto.

Estos elementos opcionales son el login de usuarios, que en caso de no disponer del tiempo suficiente, se ofrecerá un único usuario con acceso y una serie de datos de ejemplo almacenados con usuarios de prueba. En segundo lugar, el diseño web responsive es algo que se tendrá en cuenta durante el diseño de pantallas pero las optimizaciones se realizarán como última fase, una vez se disponga de las funciones requeridas del prototipo.

A continuación el calendario planteado, a través de Microsoft Project:

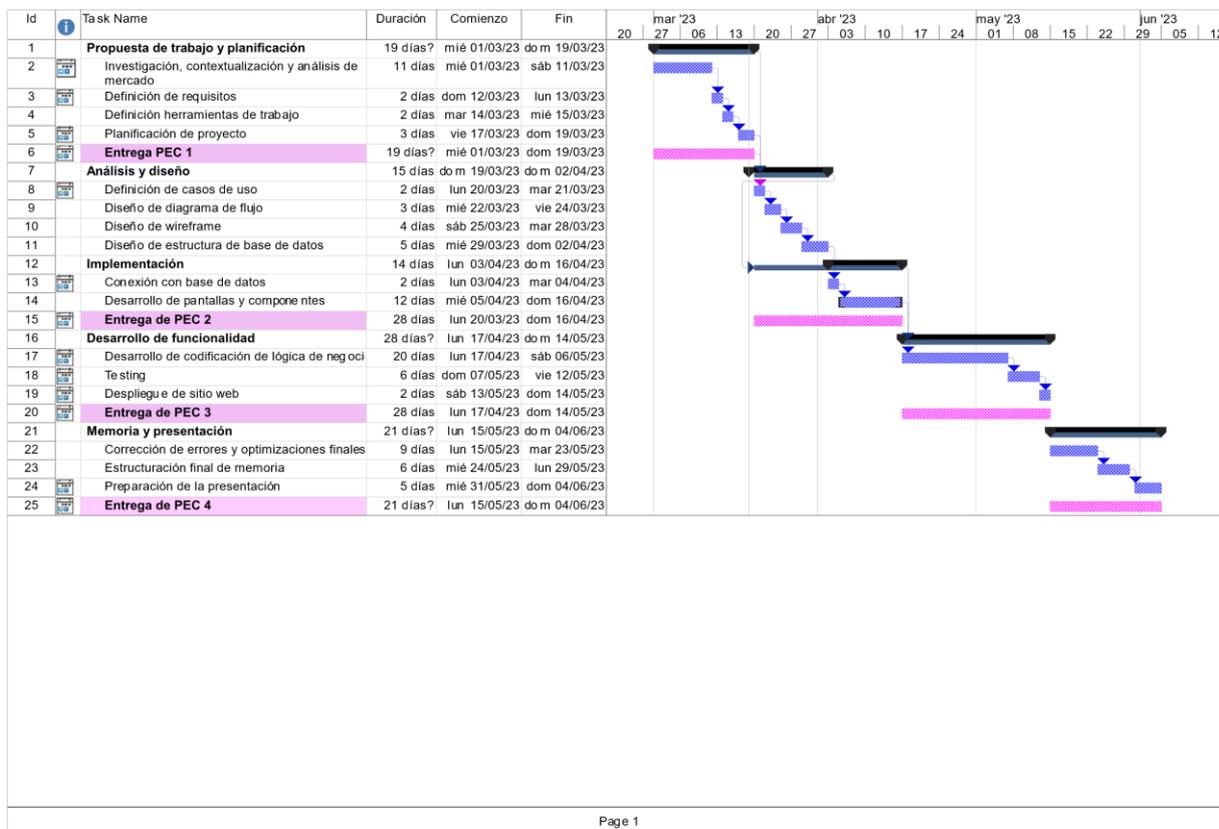


Ilustración 2. Planificación en Microsoft Project

1.7 Breve resumen de productos obtenidos

A través de este proyecto, se ha obtenido un sitio web con diseño *responsive* que cumple las funcionalidades esenciales planteadas de publicación y consulta de lugares con el apoyo de la API de Google Maps, en el cual se ha trabajado tanto en el desarrollado *back-end* con *Supabase*, como el *front-end* con *NextJs* y *React*.

1.8 Breve descripción de los otros capítulos de la memoria

En el apartado de Materiales y métodos se exponen los lenguajes de programación, herramientas de trabajo empleadas en la fase de análisis y diseño.

En la sección de Resultados se presenta el producto final, explicando la estructura de proyecto y su funcionamiento y posteriormente, las funciones específicas de cada página del sitio web.

2. Requisitos del proyecto

Se han desglosado las funciones más específicas que son necesarias para el funcionamiento de la aplicación.

Requisitos principales

- Manejo de datos con Supabase: Búsqueda y filtrado, creación, modificación y eliminación.
- Interacción con mapa mediante API de GoogleMaps:
 - Gestión de marcadores de mapa en el visualizador de ruta y lugar, y en la creación de nuevos lugares.
 - Selección de marcador al clicar en botón dedicado de cada lugar del listado en una ruta.
 - Centrado automático de marcadores
 - Conexión con GooglePlaces para sugerir ciudades y obtener información de lugares.
- Opción en lugares de 'Ver ruta' (o volver a ruta) para regresar a la ruta completa si pertenece a una.
- Gestionar el contenido visible por cada tipo de usuario y sus posibles interacciones (Marcar como favorito, Guardar)
- Gestión de etiquetas de lugares como forma de clasificación
- Diseño responsive de las páginas

Requisitos secundarias

- Gestión de registro e inicio de sesión de usuarios.
- Botón que redireccione a google maps en otra pestaña para trazar ruta.
- Sistema de notificaciones de mensajes del sistema, con fin de avisar al usuario del resultado de sus acciones.

Funciones complementarias

- Paginación de resultados de búsqueda
- Funcionalidad del botón de ajustes del Perfil de usuario (cambiar nombre, avatar...).
- Búsqueda inteligente: Si no hay resultados por texto, mostrar resultados con el resto de los filtros.
- Ver listado de lugares valorados.
- Sección en perfil de sitios en los que ha estado el usuario.
- Realizar búsqueda automática al clicar en etiquetas de lugares con resultados similares.
- Acceso a la consulta de perfiles de otros usuarios
- Creación personalizada de listas de rutas o lugares por usuario

- Creación de *landing page*

3. Materiales y métodos

En cuanto a las herramientas de trabajo, se opta por un entorno de desarrollo HTML mediante NextJS ya que es un framework Web muy utilizado en la actualidad y me permite partir de los conocimientos de HTML, CSS y Javascript que ya conozco.

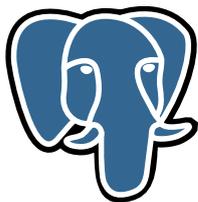


Además se emplean componentes funcionales de React, que permitirán modular la aplicación y optimizar la eficiencia a través de su actualización virtual del DOM.

Por otra parte, se utilizan módulos de componentes de NextUI y React Bootstrap para determinados controles de usuario, con fin de presentar elementos visuales atractivos visualmente que ofrecen la funcionalidad necesaria y optimizan el tiempo de desarrollo.



Por otro lado, en el backend se utiliza **Supabase** ya que permite hacer las funciones requeridas por la aplicación, almacenamiento de datos y archivos, y gestión de autenticación. Además, es una opción de código abierto compatible con React y que agiliza el desarrollo en el backend.



Supabase emplea una base de datos **PostgreSQL** y dispone de una api para realizar las consultas o modificaciones de los datos. Por otra parte, dispone de un módulo de autenticación de usuario, mediante correo y contraseña, además de ofrecer la posibilidad de acceder mediante otros medios (Google, Github, Facebook, Twitter...)



En el apartado de control de cambios, se emplea **GitHub** como repositorio, permitiendo así la revisión de versiones anteriores del proyecto.



Como editor de código se usa **Visual Studio Code**, ya que posee una interfaz intuitiva que facilita el desarrollo y gran variedad de extensiones.



En cuanto al despliegue de la aplicación, se emplea **Vercel**, ya que permite conectar con un proyecto en GitHub y realizar despliegues automáticos con cada modificación realizada. Además ofrece un plan gratuito para proyectos de pequeña escala como este.



En relación a las funciones de mapa, geolocalización e información de lugares y ciudades, se ha decidido emplear Google Maps Platform. A pesar de que existen opciones de código abierto como OpenStreetMap, MapBox o TomTom, la plataforma de Google permite hacer uso de la API Google Places y cuenta con componentes React que agilizarían el desarrollo. Por otra parte, existía el interés personal en utilizar esta herramienta por ser la más completa y reconocida. A pesar de sus limitaciones por sus elevados precios, la tarifa básica permite cumplir los objetivos de este proyecto sin problemas.

2.1 Casos de uso

Existen dos actores principales en la aplicación:

Usuario sin identificar: Es el usuario que accede a la aplicación por primera vez y no tiene asociada una cuenta. Es por ello, que no podrá interactuar con los contenidos (crear, valorar...), pero sí podrá visualizar el contenido existente.

Usuario identificado: Es aquel usuario con una cuenta relacionada, que tiene acceso a todas las funciones de la aplicación. Tiene acceso a su propio perfil desde donde manejar el contenido con el que ha interactuado.

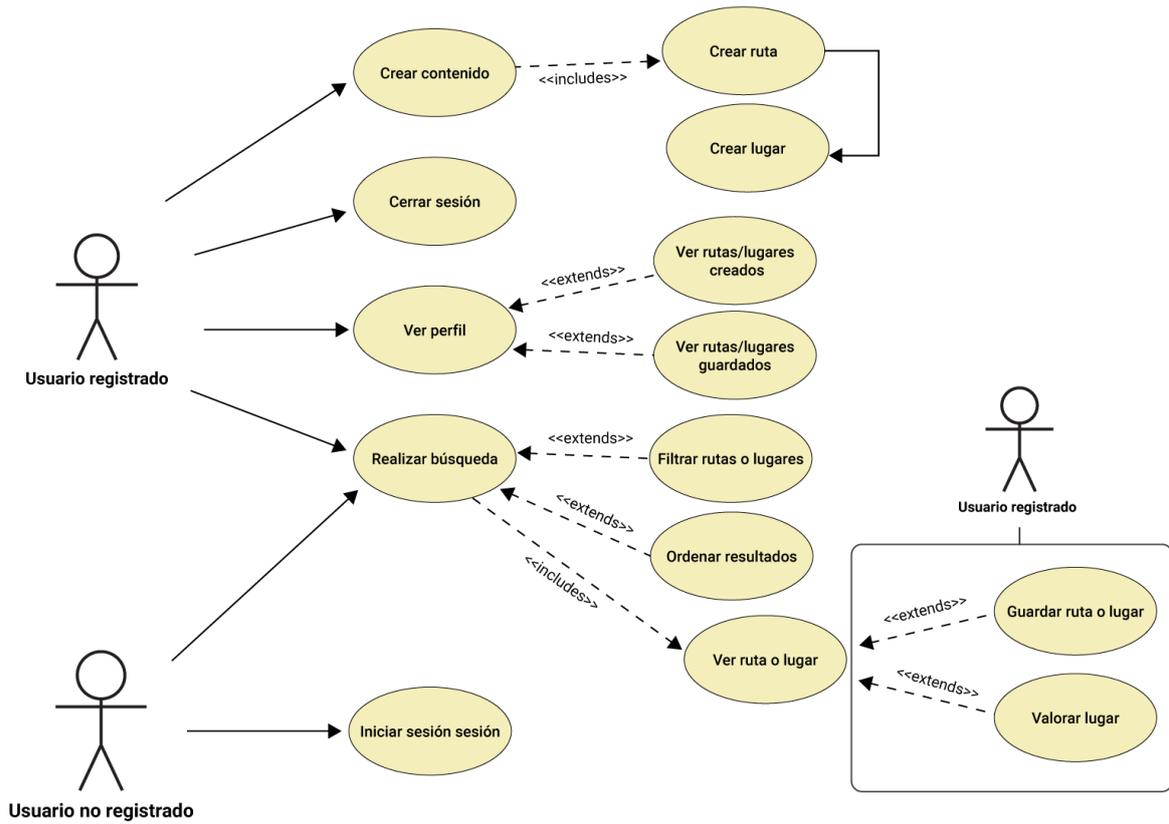


Ilustración 3. Casos de uso

Detalles

CU1	Iniciar sesión
Tipo	Primario
Actor	Usuario sin identificar
Descripción	Permite al usuario identificarse y pasar a ser un usuario identificado, ampliando así las funciones a las que tiene acceso.
Casos relacionados	
Precondición	Debe usarlo el usuario sin identificar.
postcondición	El usuario pasa a ser identificado y se habilitan más funciones.

CU2	Cerrar sesión
Tipo	Primario
Actor	Usuario identificado
Descripción	Permite al usuario pasar a ser usuario no identificado y por tanto pierde funciones en la aplicación.
Casos relacionados	
Precondición	Debe usarlo el usuario identificado.
postcondición	El usuario deja de estar identificado y pierde funciones.

CU3	Realizar búsqueda
Tipo	Primario
Actor	Usuario identificado, Usuario no identificado
Descripción	El usuario utiliza filtros para buscar información.
Casos relacionados	Ver ruta o lugar
Precondición	Ninguna
postcondición	Una vez hecha la búsqueda el usuario puede consultar los resultados y pasar a inspeccionar cada uno.

CU4	Ver ruta o lugar
Tipo	Secundario
Actor	Usuario identificado, Usuario sin identificar
Descripción	El usuario consulta los datos de una ruta o lugar. En el caso de la ruta, podrá ver los lugares individuales relacionados con la misma.
Casos relacionados	Realizar búsqueda, (Guardar, Valorar)
Precondición	Se debe haber mostrado un listado desde donde seleccionar un lugar o ruta.
postcondición	En caso de ser un usuario identificado se pueden realizar las acciones de Guardar o Valorar.

CU5	Crear contenido
Tipo	Primario
Actor	Usuario identificado
Descripción	Permite al usuario dar de alta nuevos elementos en el sistema, que estarán relacionados con su perfil.
Casos relacionados	Crear ruta, crear lugar
Precondición	Debe usarlo el usuario identificado
postcondición	El usuario selecciona el tipo de contenido a crear (ruta o lugar)

CU6	Crear ruta
Tipo	Secundario
Actor	Usuario identificado
Descripción	Permite al usuario dar de alta una ruta, que es una colección de lugares. Se habilita la edición de una nueva ruta.
Casos relacionados	Crear contenido, Crear lugar
Precondición	Debe usarlo el usuario identificado y haber pasado por la selección de tipo de contenido en el CU5.
postcondición	El usuario puede añadir lugares

CU7	Crear lugar
Tipo	Secundario
Actor	Usuario identificado
Descripción	Permite al usuario dar de alta un nuevo lugar, que puede estar relacionado con una ruta o ser independiente. Se habilita la edición de un nuevo lugar.
Casos relacionados	Crear contenido, Crear ruta
Precondición	Debe usarlo el usuario identificado y haber pasado por la selección de tipo de contenido en el CU5.
postcondición	

CU8	Ver perfil
Tipo	Secundario
Actor	Usuario identificado
Descripción	Consulta de información básica del usuario
Casos relacionados	Ver rutas/lugares creados, Ver rutas/lugares guardados
Precondición	Debe usarlo el usuario identificado.
postcondición	El usuario puede modificar el contenido relacionado con su perfil (eliminar elementos creados, dejar de guardar elementos o modificar su información de perfil)

2.2 Diagrama de flujo

Se han diseñado tres diagramas de flujo, en base a las actividades principales interacciones del usuario con la aplicación.

D1. Crear contenido

Representa la creación de contenido, que son los lugares y las rutas de usuario. Este flujo se relaciona con las pantallas de Edición de lugar y Edición de ruta.

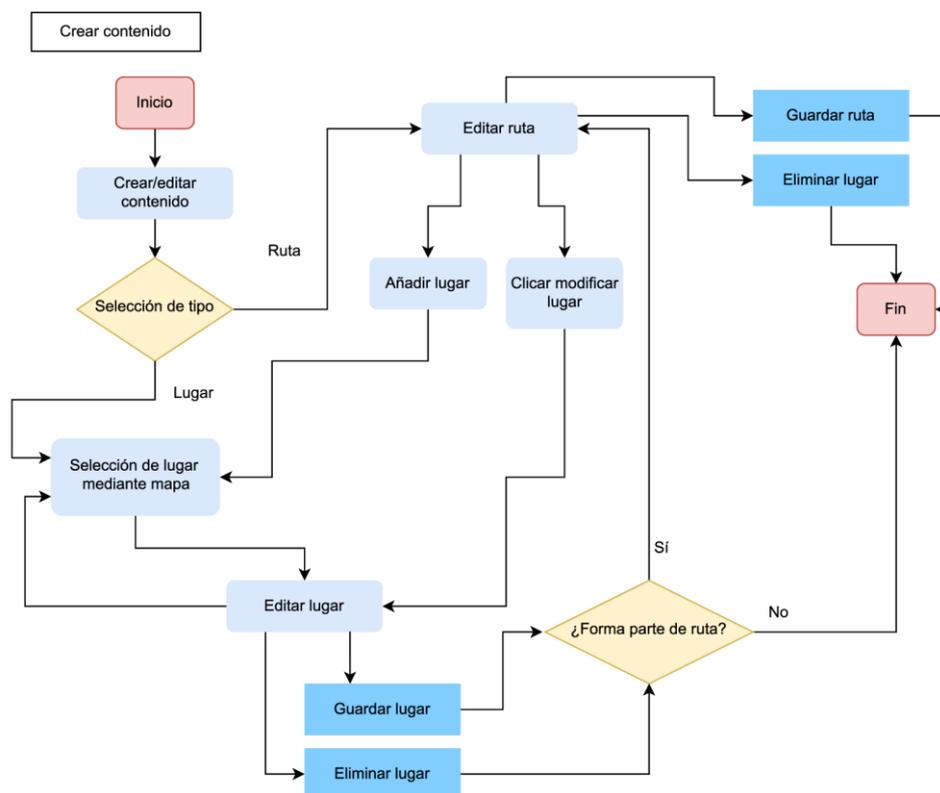


Ilustración 4. Diagrama de flujo – Crear contenido

D2. Realizar búsqueda

Representa la búsqueda de contenido, el usuario puede utilizar unos filtros y ordenar los resultados. Una vez mostrados, puede acceder a ellos. Este flujo se relaciona con las pantallas de Búsqueda, Visor de ruta y Visor de lugar.

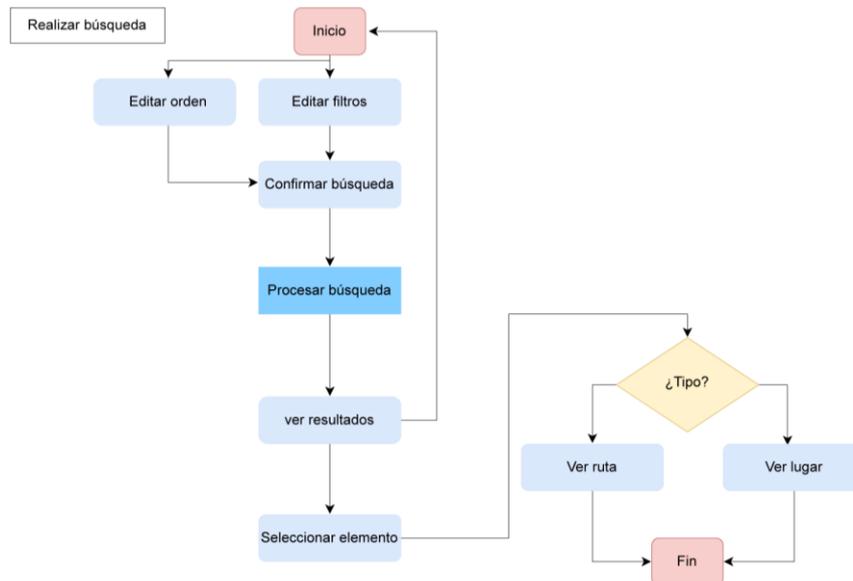


Ilustración 5. Diagrama de flujo – Realizar búsqueda

D3. Ver datos de elemento

Representa las acciones de las que dispone un usuario en un Visor de contenido, ya sea de lugar o de ruta.

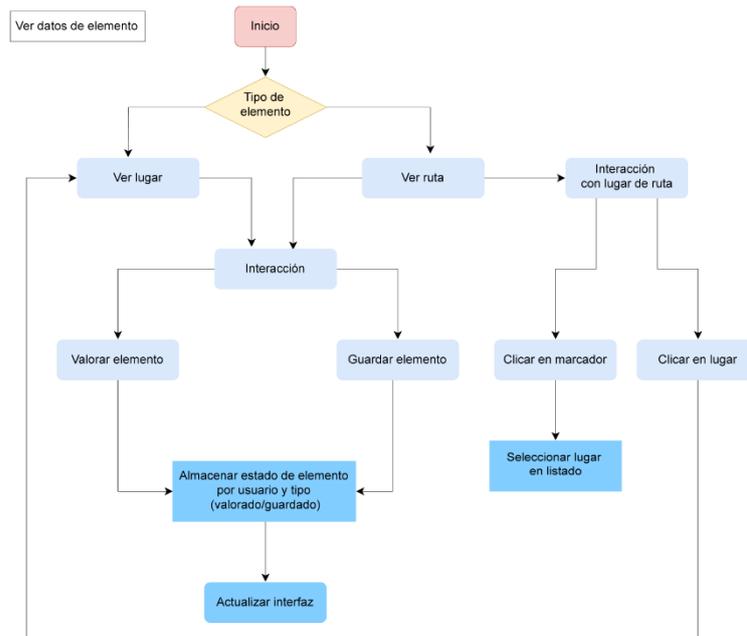


Ilustración 6. Diagrama de flujo – Ver datos de elemento

2.3 Diseño de base de datos

Para el desarrollo del backend se ha optado por utilizar Supabase. Se utiliza esta opción ya que optimiza el tiempo de desarrollo al facilitar un sistema de autenticación de usuarios y una api mediante la cual acceder y modificar los datos necesarios.

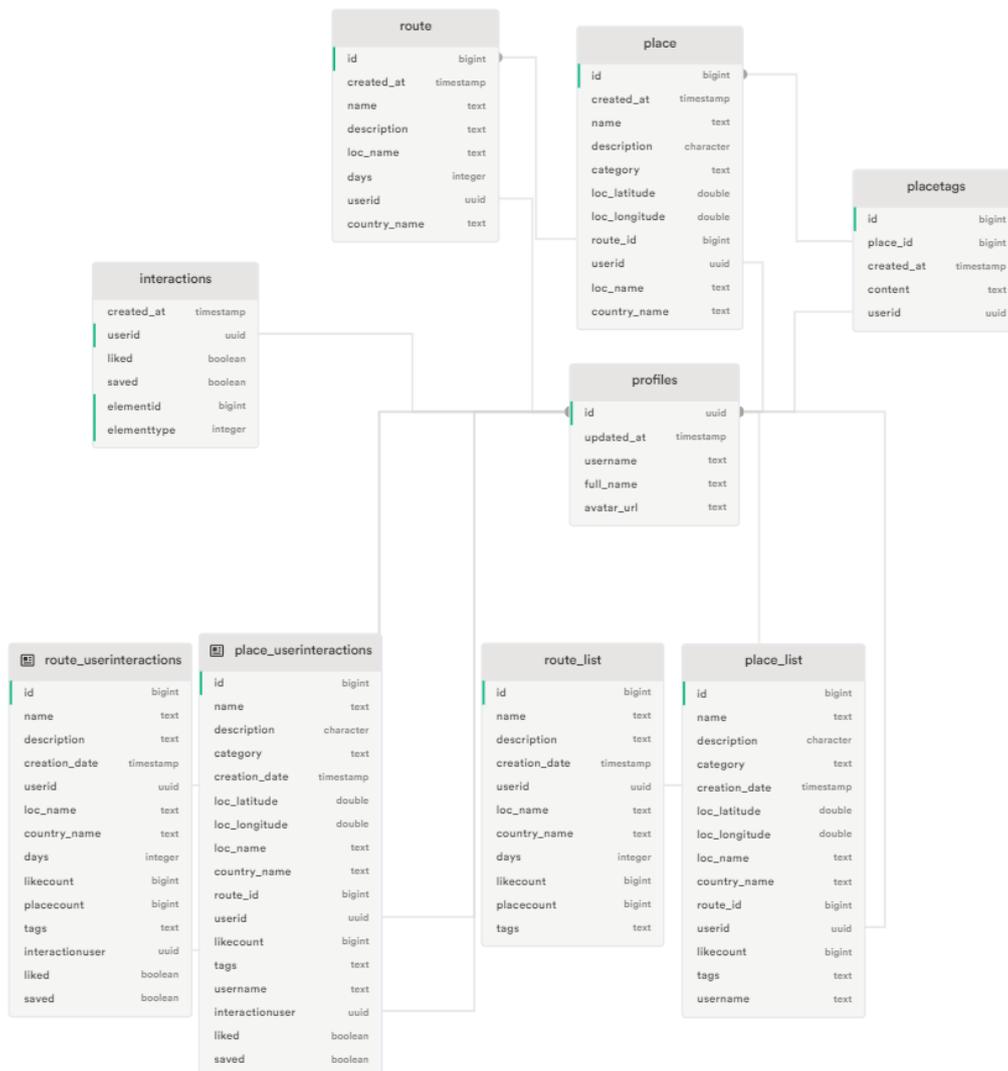


Ilustración 7. Estructura de base de datos

Se ha obtenido el esquema inicial de la base de datos mediante Supabase Schema Visualizer. En el vemos las tablas creadas: place, route, placetags, profiles e interactions, y las vistas place_list, route_list, place_userinteractions y route_userinteractions, que contiene los datos necesarios para la visualización de datos de un lugar o ruta en la aplicación así como las interacciones del usuario con cada elemento.

La tabla 'interactions' almacena la relación entre un usuario y un lugar o ruta, donde userid identifica el usuario, elementid el identificador el elemento con el que interactúa y elementtype determina el tipo de elemento (ruta o lugar)

La tabla **profiles** está directamente relacionada mediante el id, con la tabla auth.users, que es la que ofrece supabase para manejar los usuarios de la aplicación. Mediante profiles, configuramos los datos adicionales que necesitemos de un usuario.

Las imágenes se gestionan mediante los Buckets de Supabase, que permiten organizar los archivos por carpetas.

En el Bucket images se almacenan las imágenes de lugares, que siguen la siguiente estructura: places/[id de usuario]/[id de lugar]/[nombre de archivo]

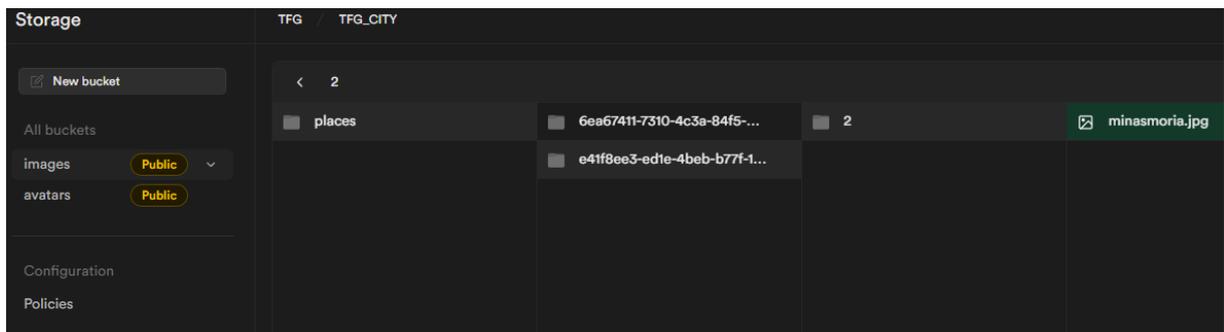


Ilustración 8. Estructura Bucket de lugares

El Bucket 'avatars' almacena las imágenes de perfil de usuario, según la estructura: [id de usuario]/[nombre de archivo].

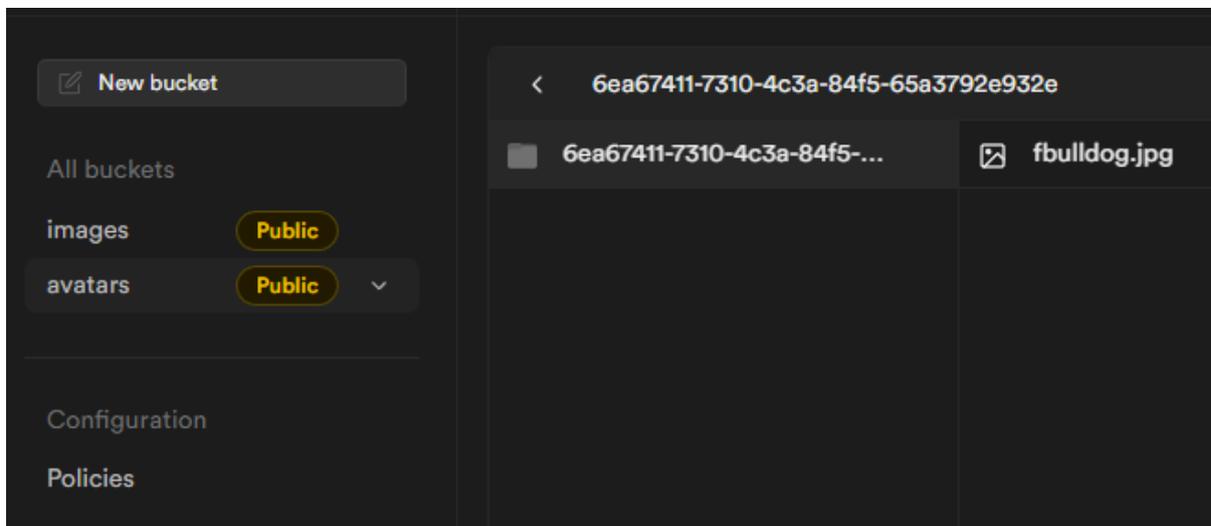


Ilustración 9. Estructura Bucket de avatares

Vistas

Se han realizado una serie de vistas en la base de datos con fin de facilitar la consulta de datos y el filtrado.

La vista **place_list** permite obtener todos los datos necesarios para mostrar un lugar en pantalla, además de los campos necesarios para realizar los filtrados y ordenes pertinentes. La vista **route_list** permite cumplir la misma función, pero con los datos de rutas.

```
CREATE VIEW place_list
AS
SELECT p.id AS id
,p.name AS name
,p.description AS description
,p.category AS category
,p.created_at AS creation_date
,p.loc_latitude AS loc_latitude
,p.loc_longitude AS loc_longitude
,p.loc_name AS loc_name
,p.country_name AS country_name
,coalesce(p.route_id, 0) AS route_id
,p.userid AS userid
,(
SELECT count(*)
FROM interactions i
WHERE i."elementid" = p.id
AND i."elementtype" = 1
AND liked = true
) AS likecount
,(
SELECT ';' || string_agg(content, ';') || ';'
FROM placetags t
WHERE t.place_id = p.id
) AS tags
,(
SELECT prof.username
FROM profiles prof
WHERE prof.id = p.userid limit 1
) AS username
FROM place p
```

Ilustración 10. Vista SQL de lugares

```
create VIEW route_list
AS
SELECT r.id AS id
,r.name AS name
,r.description AS description
,r.created_at AS creation_date
,r.userid AS userid
,r.loc_name AS loc_name
,r.country_name AS country_name
,r.days AS days
,(
SELECT count(*)
FROM interactions i
WHERE i.elementid = r.id
AND i.elementtype = 2
) AS likecount
,(
SELECT count(*)
FROM place p2
WHERE p2.route_id = r.id
) AS placecount
,(
SELECT ';' || COALESCE(string_agg(content, ';'), '') || ';'
FROM placetags t
INNER JOIN place p ON t.place_id = p.id
WHERE p.route_id = r.id
) AS tags,
(
SELECT ';' || COALESCE(string_agg(category, ';'), '') || ';'
FROM place p
WHERE p.route_id = r.id
) AS categories
FROM route r
```

Ilustración 11. Vista SQL de rutas

Finalmente, las vistas **place_userinteractions** y **route_userinteractions**, muestran todos los datos de la vista de lugar o ruta además de las interacciones de un usuario con el mismo, es decir, si lo ha marcado con 'Guardado' o 'Me gusta'.

Estas vistas se emplean en para cargar los elementos guardados de un usuario en la pantalla Profile.

```
create view
  public.place_userinteractions as
select
  p.id,
  p.name,
  p.description,
  p.category,
  p.creation_date,
  p.loc_latitude,
  p.loc_longitude,
  p.loc_name,
  p.country_name,
  p.route_id,
  p.userid,
  p.likecount,
  p.tags,
  p.username,
  i.userid as interactionuser,
  i.liked,
  i.saved
from
  place_list p
join interactions i on i.elementid = p.id
and i.elementtype = 1;
```

Ilustración 10. Vista SQL de interacciones con lugar

```
create view
  public.route_userinteractions as
select
  r.id
  ,r.name
  ,r.description
  ,r.creation_date
  ,r.userid
  ,r.loc_name
  ,r.country_name
  ,r.days
  ,r.likecount
  ,r.placecount
  ,r.tags
  ,i.userid as interactionuser,
  i.liked,
  i.saved
from
  route_list r
join interactions i on i.elementid = r.id
and i.elementtype = 2;
```

Ilustración 13. Vista SQL de interacciones con ruta

Políticas de base de datos

Se han configurado en Supabase, las políticas Row Level Security de forma que se restrinjan las acciones dependiendo del usuario que accede a ellas, de forma que los usuarios autenticados puedan modificar sus registros, y los no autenticados sólo puedan consultar.

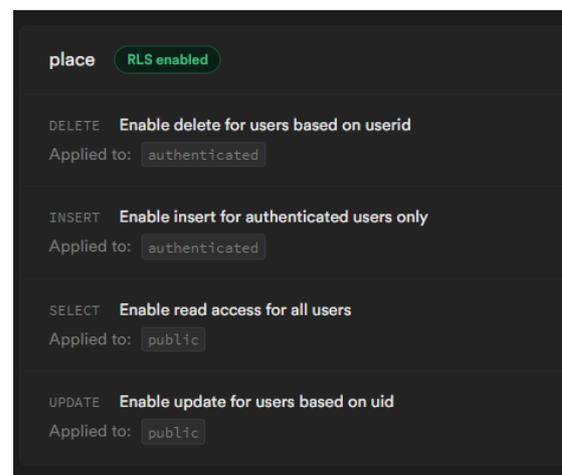


Ilustración 11. Políticas RLS en Supabase

2.4 Prototipo

Se ha empleado Adobe Xd para el diseño de los prototipos de cada pantalla.

Búsqueda

Es la pantalla inicial de la aplicación, desde ella se pueden utilizar filtros para consultar los lugares o rutas almacenados en la base de datos así como ordenar los resultados.

Permite mostrar u ocultar los filtros mediante un botón y muestra los resultados separados según su tipo (ruta o lugar).

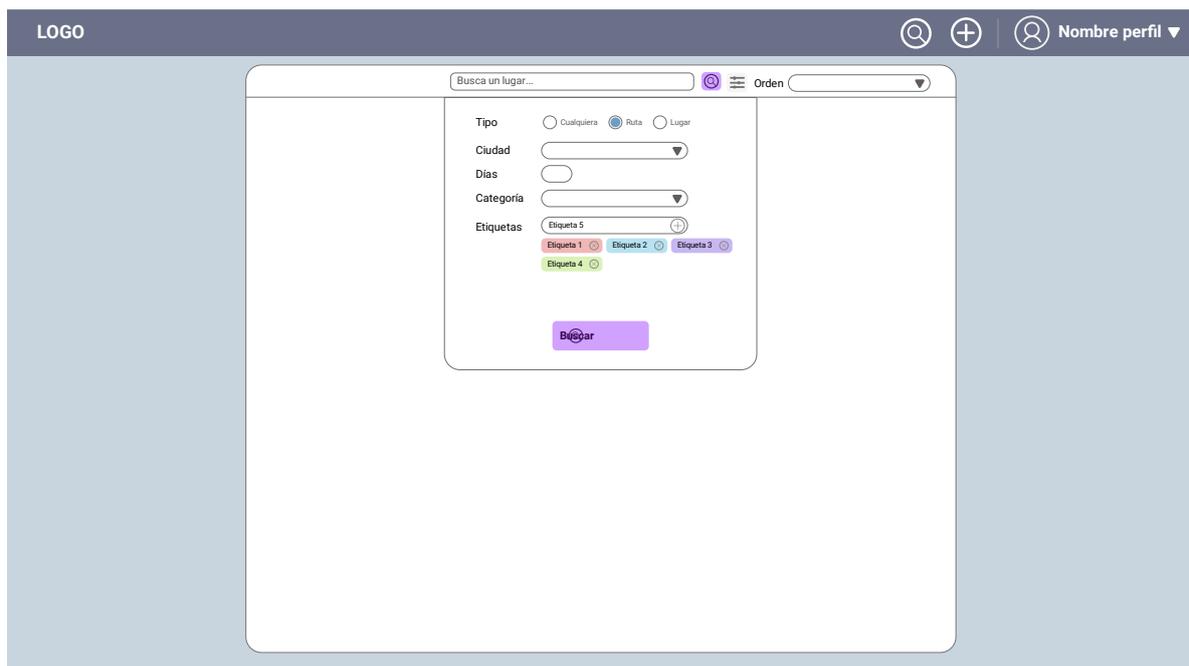


Ilustración 12. Pantalla de búsqueda. Visualización de filtros

Interacciones de usuario:

- Modificar filtros de búsqueda
- Ver ruta o lugar de listado

Con respecto al diseño final, no ha variado en gran medida. Principalmente el control de Orden, la disposición del panel de filtros plegable y la eliminación del botón de búsqueda desde el panel de filtros por redundancia.

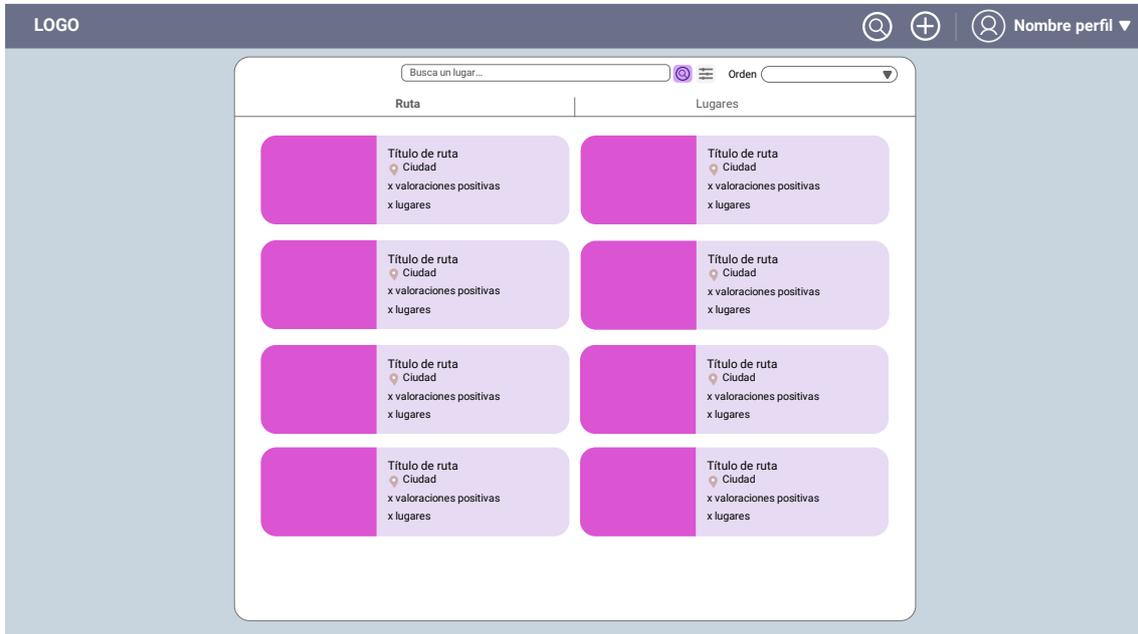


Ilustración 13. Pantalla de búsqueda. Resultados de búsqueda de rutas.

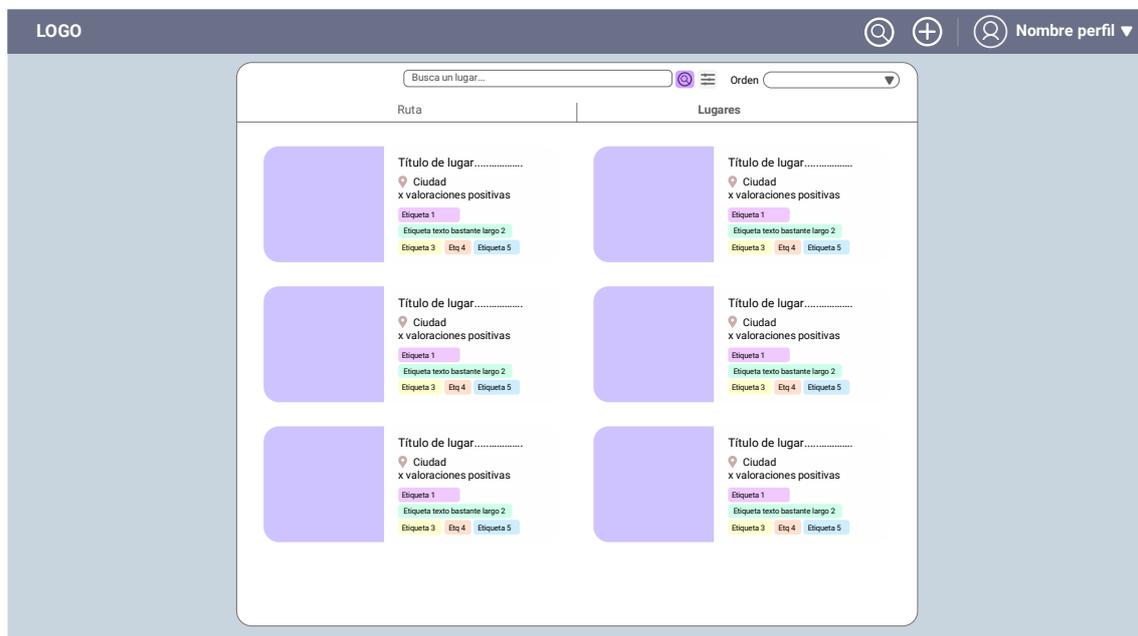


Ilustración 14. Pantalla de búsqueda. Resultados de búsqueda de lugares.

Visualizador de lugar

Esta pantalla permite visualizar los datos de un Lugar, contiene controles que aparecen de forma condicional, el botón de Editar si el lugar pertenece al usuario que lo visualiza y el botón de Ver ruta si el lugar pertenece a una ruta.

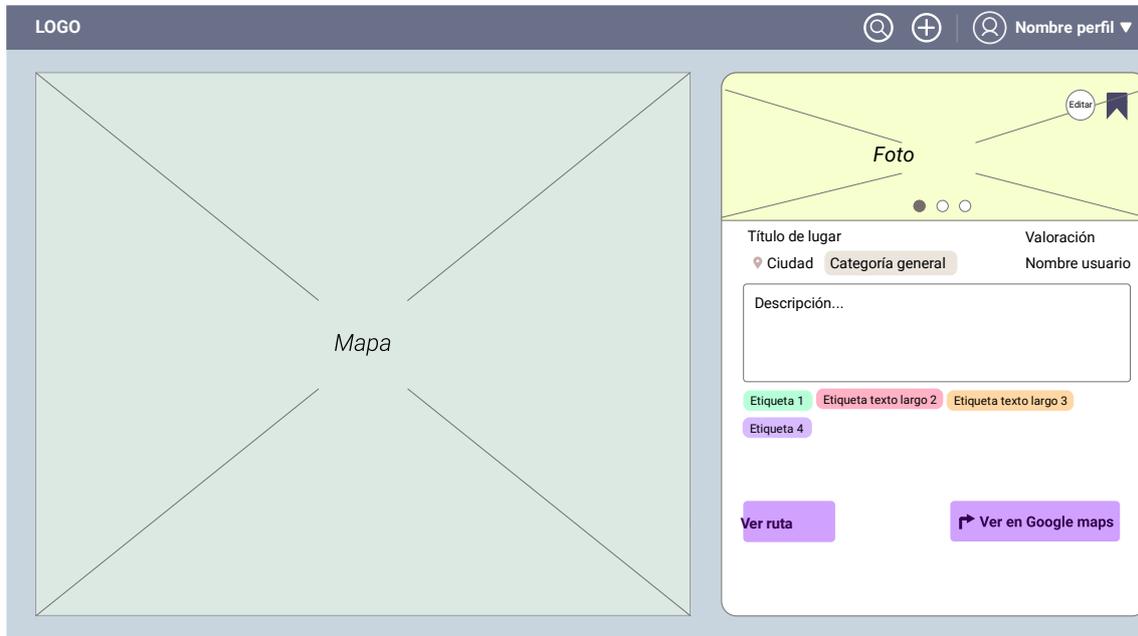


Ilustración 15. Visualizador de lugar

Con respecto al diseño final, ha cambiado la posición del nombre de usuario propietario y el posicionamiento del contenedor para tener más consistencias con respecto al contenedor del visualizador de ruta.

Visualizador de ruta

Permite visualizar los datos de una ruta y su listado de lugares relacionado. Desde el listado de lugares podemos visualizar el lugar en el mapa o acceder al detalle del lugar.

Interacciones de usuario:

- Valorar lugar
- Guardar lugar
- Ver mapa
- Direccionar a google maps en nueva pestaña
- Volver a ruta (en caso de pertenecer a una)

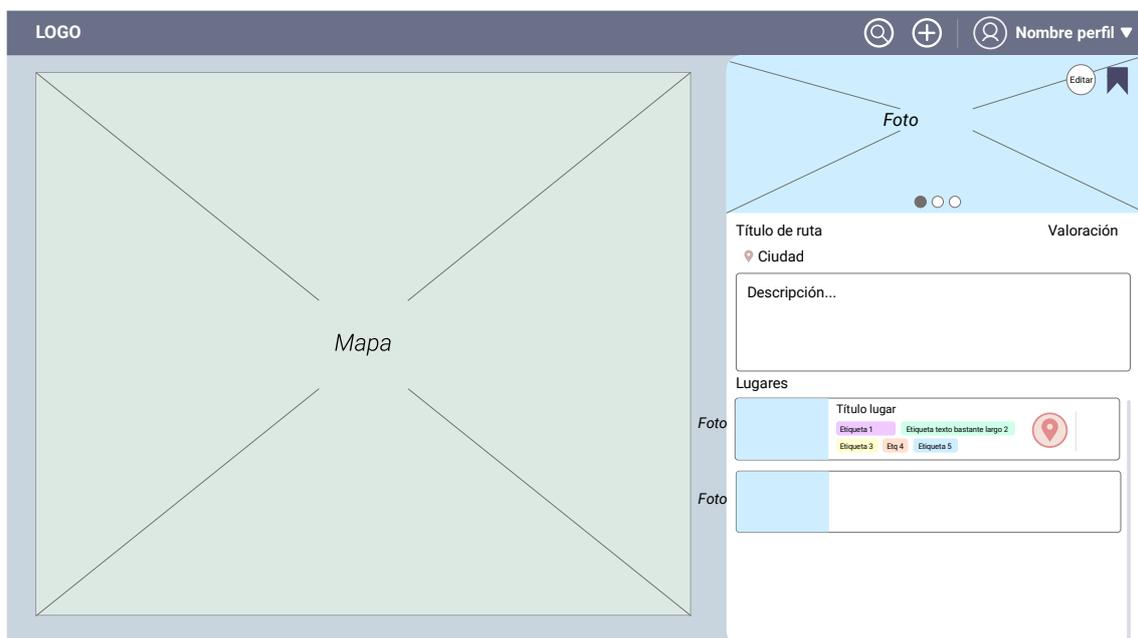


Ilustración 16. Visualizador de ruta

Perfil de usuario

Desde el perfil de usuario se pueden consultar las interacciones del usuario con la aplicación mediante dos listados: Mis sitios, que muestra los lugares o rutas que hemos subido al sistema, y Sitios guardados, que muestra los que hemos guardado de otros usuarios.

Interacciones de usuario:

- Ver rutas/lugares creados y guardados.
- Modificar rutas/lugares creados.

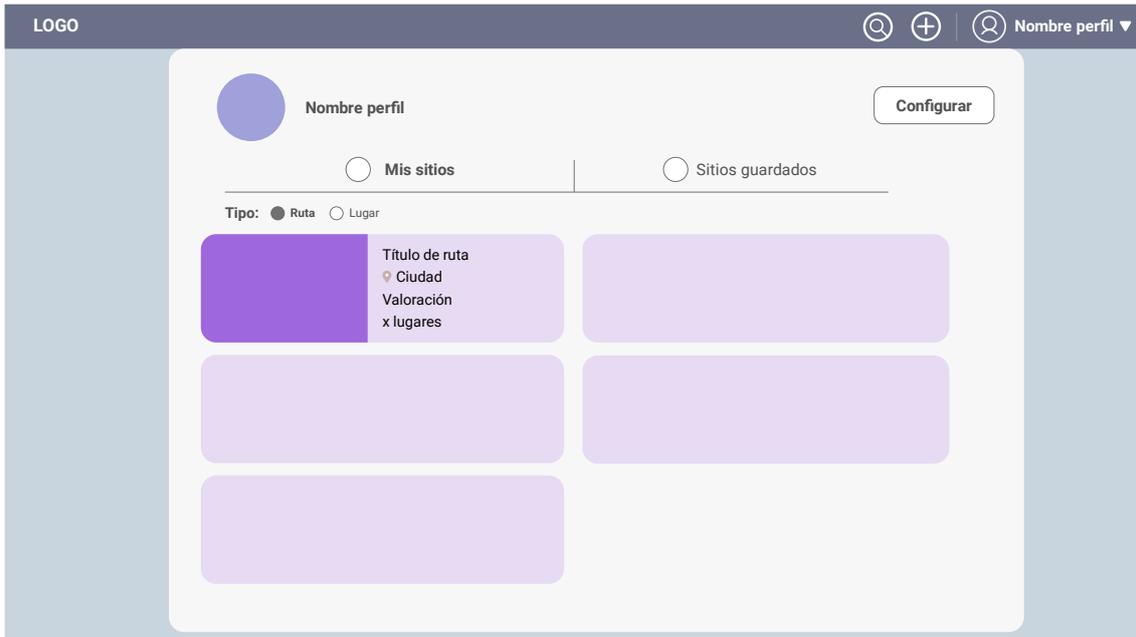


Ilustración 17. Perfil de usuario (Listado de rutas)

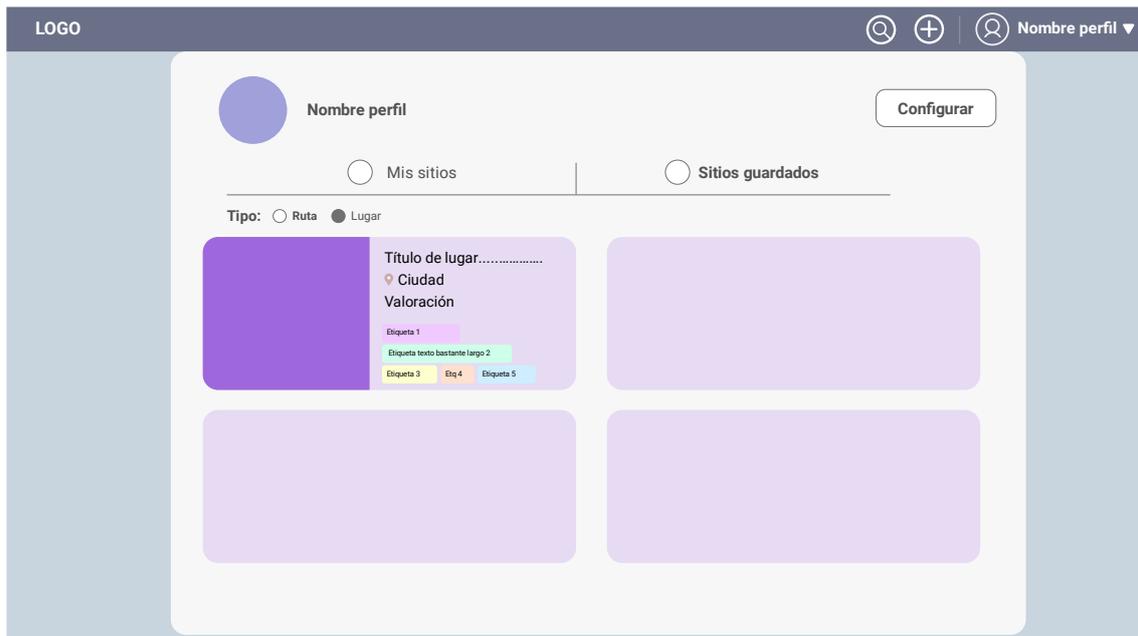


Ilustración 18. Perfil de usuario (Listado de lugares)

En el diseño final se ha añadido el visualizador de perfil para mostrar también el correo electrónico y un botón para cerrar sesión, que en principio se había pensado para ser una opción de menú del botón de la barra de navegación. Además se ha incluido una pestaña más de sitios valorados, ya que no implicaba un desarrollo extenso y se detectó la necesidad en la fase de pruebas.

Edición de ruta

Permite crear o modificar una ruta. Es similar al visualizador de ruta, pero contiene controles específicos para la edición de valores.

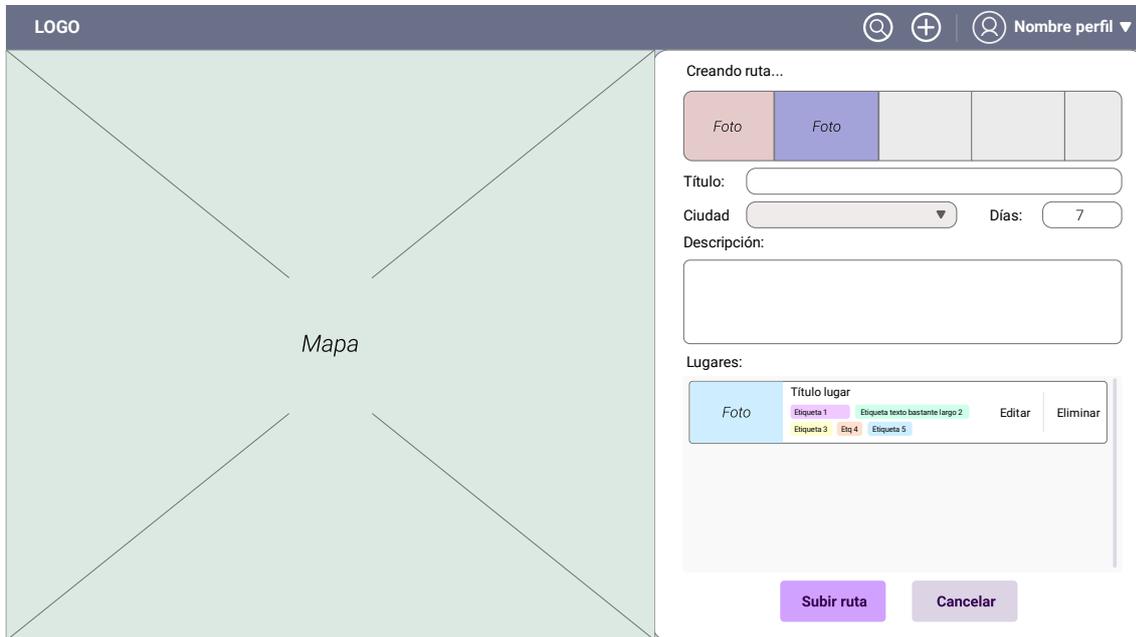


Ilustración 19. Edición de ruta

Interacciones de usuario:

- Modificar valores
- Añadir nuevo lugar de listado
- Eliminar lugar de listado
- Editar lugar de listado

Edición de lugar

Permite crear o modificar un lugar. No contiene un visualizador de mapa ya que se selecciona mediante un paso previo.

Interacciones de usuario:

- Modificar valores

LOGO

Creando lugar...

Foto Foto

Título:

Ciudad Categoría

Descripción:

Etiqueta: +

Etiqueta 1 Etiqueta 2 Etiqueta 3

Etiqueta 4 Etiqueta 5

Subir lugar Cancelar

Ilustración 20. Edición de lugar

Barra de navegación

La barra de navegación se compone de los botones de búsqueda, creación de contenido y perfil de usuario.

En caso de que el usuario no tenga sesión iniciada, tiene acceso a búsqueda y a iniciar sesión.



Ilustración 21. Barra de navegación sin sesión iniciada

4. Resultados

3.1 Implementación del proyecto

Estructura

El proyecto se ha construido a partir de la estructura de NextJS. Por tanto, en la carpeta **src**, encontramos el núcleo de la aplicación.

La carpeta **components** aloja los componentes reutilizables empleados en la aplicación.

La carpeta **db** contiene las funciones necesarias para realizar la gestión de la base de datos.

La carpeta **images** almacena los iconos y gráficos empleados en la aplicación.

La carpeta **maps** Contiene los componentes y funciones relacionadas con la api de Google Maps y Google Places.

La carpeta **options** contiene constantes de la aplicación, como las categorías de lugar disponibles o las opciones de orden de resultados de búsqueda.

La carpeta **pages** contiene todas las páginas de la aplicación.

La carpeta **redux** gestiona el estado de la aplicación mediante la librería Redux.

La carpeta **styles** contiene las hojas de estilo, tanto el fichero de estilos global, como los ficheros de estilo modular, ya sea para componentes o para páginas.

Finalmente, la carpeta **node_modules** almacena los paquetes y módulos externos empleados en el proyecto, y en el fichero. **env.production** se almacenan las Claves de API de Supabase y Google Maps Platform como variables de entorno.

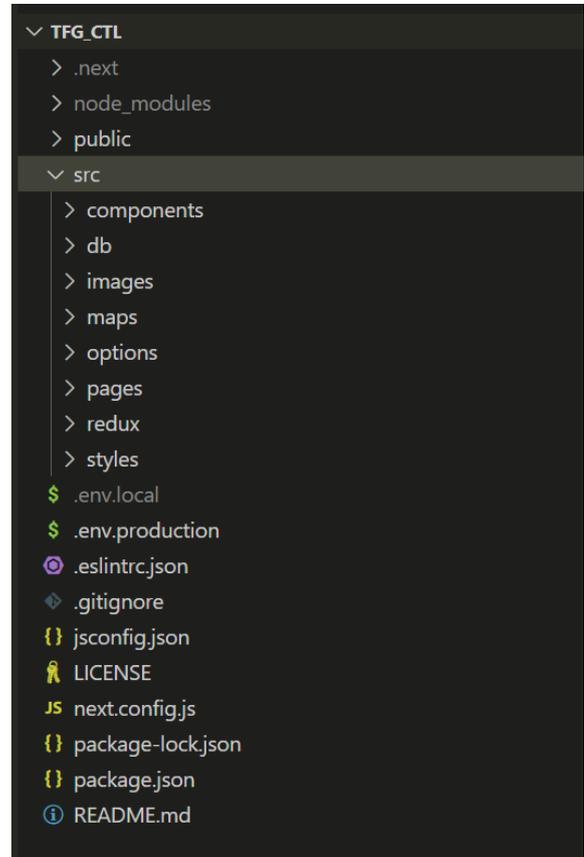


Ilustración 22. Estructura de proyecto

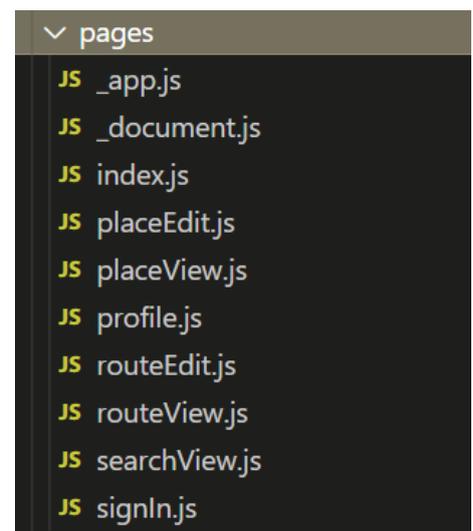


Ilustración 23. Páginas del sitio web

Componente MyApp

El componente MyApp define la estructura común de toda la aplicación. Se emplea NextUIProvider hacer uso de los componentes de la librería de NextUI, Provider permite utilizar la librería Redux, SessionContextProvider permite aplicar la lógica de la autenticación de Supabase, MesaggeWrapper es el componente que permite mostrar avisos a través de cualquier página de la aplicación y por último, la estructura base de todas las páginas, un contenedor, la barra de navegación y el contenido de la página.

```
const MyApp = ({ Component, pageProps }) => {
  const [supabase] = useState(() => createBrowserSupabaseClient());

  const { isBrowser } = useSSR()

  return (
    isBrowser && (
      <NextUIProvider>
        <Provider store={reduxStore}>
          <SessionContextProvider supabaseClient={supabase} initialSession={pageProps.initialSession}>
            <MessageWrapper>
              <div className='mainContainer'>
                <NavBar />
                <Component {...pageProps} />
              </div>
            </MessageWrapper>
          </SessionContextProvider>
        </Provider>
      </NextUIProvider>
    )
  )
}
```

Ilustración 24. Componente MyApp

Mapa

El fichero MapsApi contiene el componente de mapa y las funciones necesarias para visualizar los datos que controla la aplicación.

Mediante la librería @react-google-maps/api se tiene acceso al componente principal de mapa.

Las funciones del mapa son las siguientes:

- Centrar el mapa según los marcadores de lugares disponibles
- Creación de marcadores
- Consulta de nombre de lugar al añadir marcador mediante la librería react-google-places-autocomplete
- Búsqueda de ciudad y centrado en mapa de esta
- Selección de marcador

Utiliza el componente `MarkerPopup` para mostrar una ventana flotante que permita añadir un nuevo lugar al mapa.

La API se carga mediante el script correspondiente en el fichero `Document`, que define la estructura html de las páginas.

```
return (  
  <Html lang="es">  
    <Head >  
      <Script type="text/javascript" src={googleMapsApiSrc} strategy="beforeInteractive" onLoad={onLoadMapsApi} />  
    </Head>  
    <body>  
      <Main />  
      <NextScript />  
    </body>  
  </Html>  
)
```

Ilustración 25. Componente Document

El atributo `src` hace referencia a la constante de la fuente de la API con referencia a las claves como variables de entorno.

```
export const googleMapsApiSrc = `https://maps.googleapis.com/maps/api/js?key=${process.env.NEXT_PUBLIC_GOOGLE_MAPS_API_KEY}&libraries=places`
```

Ilustración 26. Ruta de API de Google Maps

Por otra parte, el mapa utiliza un ID de estilo de mapa, mediante el cual se obtiene el diseño de mapa previamente configurado desde Google Maps Cloud Platform con fin de presentar el control con unos colores acordes al sitio web.

En cuanto a la gestión de la creación de marcadores, en caso de clicar sobre un lugar específico, se realiza una llamada a `PlacesService` a partir del `PlaceId`, obteniendo así la información geográfica (ciudad, país y coordenadas) y el nombre de lugar con fin de mostrarlo en la ventana flotante de creación de lugar, sobre el nuevo marcador. En caso contrario, se emplea la función `geocodeByLatLng` de la librería `react-google-places-autocomplete` para obtener los datos de geolocalización.

Mensajes de aplicación

Se ha creado un mecanismo de notificaciones de usuario, a través de un componente gráfico, un componente funcional y un estado de `Redux` para manejar el momento en el que se deben mostrar mensajes.

InfoMessage contiene el componente visual que representa la notificación, a través del `c` componente `Snackbar` de la librería `@material-ui`, que se muestra sobrepuesto a todos los elementos de cada página a través de sus estilos `css`.

```

/* Gestiona las notificaciones de la aplicación */
export default function MessageWrapper({ props, children }) {
  const messageDispatch = useDispatch();
  const messageListener = useSelector(state => state.showMessage); // Escucha los cambios en los mensajes de la aplicación
  const [textMessage, setTextMessage] = useState("") // Almacena el mensaje a mostrar
  const [messageOpen, setMessageOpen] = useState(false); // Indica si se muestra mensaje en pantalla
  useEffect(() => {
    if (messageListener && messageListener.message.length > 0) {
      setMessageOpen(true) //Activa la ventana
      setTextMessage(messageListener.message)
    }
  }, [messageListener])

  function onMessageClose(e) {
    setMessageOpen(false) // Cierra ventana
    messageDispatch(setMessage("")) // Elimina mensaje pendiente de mostrar
    setTextMessage("") //Elimina mensaje en pantalla
  }
  return <>{children} <InfoMessage messageOpen={messageOpen} setMessageOpen={setMessageOpen} message={textMessage} onMessageClose={onMessageClose}/> </>
}

```

Ilustración 27. Componente MessageWrapper

El componente **MessageWrapper** engloba todos los componentes de la aplicación y está a la escucha de los cambios en el estado de Redux que contiene el mensaje a mostrar.

De esta forma, se consigue gestionar notificaciones en todas las páginas de la aplicación web.

3.2 Componentes

Durante el desarrollo de las páginas web, se han creado componentes que permitan modular, estructurar y reutilizar el código de forma más eficiente.

El componente **Profile** contiene los datos de usuario registrado, su información básica y la carga de sus interacciones (elementos guardados y creados).

El componente **Navbar** contiene la barra de navegación, que es común a toda la aplicación. Utiliza el componente **NavProfile**, que cambia dependiendo del estado de inicio de sesión; en caso de tener la sesión iniciada, se muestra la imagen y nombre de usuario.

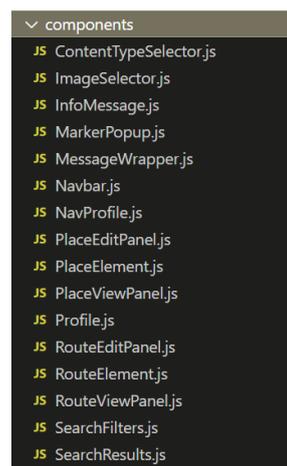


Ilustración 28. Listado de componentes

PlaceElement y **RouteElement** representan los elementos que se muestran en listados, mientras que **PlaceViewPanel** y **RouteViewPanel** son los paneles que muestran toda la información de un lugar y ruta así como o las acciones que se pueden realizar sobre ellos (añadir a me gusta, guardar, editar).

PlaceEditPanel y **RouteEditPanel**, muestran los datos de lugar y ruta de forma editable y contienen la funcionalidad de actualización de datos en supabase.

Para la pantalla de búsqueda, **searchView**, se emplean los componentes **SearchFilters** y **SearchResults**. El primero contiene los filtros de búsqueda y gestiona su lógica, mientras que el segundo presenta los listados de lugares y/o rutas. La propia página **searchView** es la que se ocupa de utilizar los filtros

empleados en SearchFilters y obtener los datos de Supabase para presentarlos en SearchResults.

ImageSelector es el control que permite editar las imágenes asociadas a un lugar, a través del botón de edición a la izquierda se añaden las imágenes seleccionadas al listado, además de permitir eliminarlas de forma individual.

ContentTypeSelector contiene un control cambiante que permite la selección de tipo de contenido (ruta y lugar). Mediante props se puede indicar si se muestra como control de tipo check o de tipo Radio, ya que se emplea en filtros y en el perfil de usuario con una función distinta.

3.3 Base de datos

El fichero **dbmanager** crea el cliente de supabase a partir de la URL y clave requeridas.

El fichero **dbContent** gestiona las consultas a base de datos, rutas, lugares, interacciones e imágenes.

El fichero **dbUpdate** gestiona las actualizaciones de datos, modificaciones y eliminaciones.

Datos secundarios

En la carpeta options se almacena el fichero **categories**, con la información de las categorías de lugar disponibles, con su valor y descripción visible, mientras que **orderOptions** almacena las posibilidades de orden de resultados de búsqueda.

La carpeta **redux** contiene ficheros para el manejo de varios estados de la aplicación a través de la librería del mismo nombre. En los estados encontramos: **messageReducer**, que almacena el mensaje a mostrar al usuario y **placeReducer** junto a **routeReducer** almacenan un objeto lugar o ruta que permita utilizarse cuando se esté en edición y sea necesario conservar esa información entre páginas sin la necesidad de la inserción y posterior consulta en base de datos.

Librerías de componentes utilizadas

Se han empleado los siguientes componentes en la aplicación

De **NextUI**: Checkbox, Radio, Textarea, Input, Form, Avatar, Tab, Tabs, Pagination y Modal.

De **React Bootstrap**: CloseButton, Nav, Navbar, NavDropdown, Container, Typography, Card y Carousel.

De **Material UI**: Snackbar y Button.

De **Next**: Image.

De **react-google-maps/api**: GoogleMap, InfoWindowF y MarkerF.

De **supabase/auth-ui-react**: Auth yThemeSupa.

Componentes **independientes**: GooglePlacesAutocomplete (react-google-places-autocomplete) y TagsInput (react-tag-input-component).

3.4 Estilos

En cuanto al estilo css, se han importado estilos por defecto de **bootstrap** que permiten aplicar una apariencia básica.

De forma general, se ha optado por un estilado modular, de forma que cada componente y página utilicen un fichero css que emplea clases exclusivas del componente sobre el que se aplica.

Finalmente, en el fichero globals.css se ha aplicado estilo a elementos que estén presentes en la mayor parte de las páginas del sitio web.

3.5 Logo

A pesar de que en la planificación no se preveía el diseño de un logo, en las etapas finales se decidió crear un logotipo que otorgara personalidad a la página web, así como una selección de colores con tonalidades azules.



Ilustración 29. Logotipo City Tour



Ilustración 31. Favicon de sitio

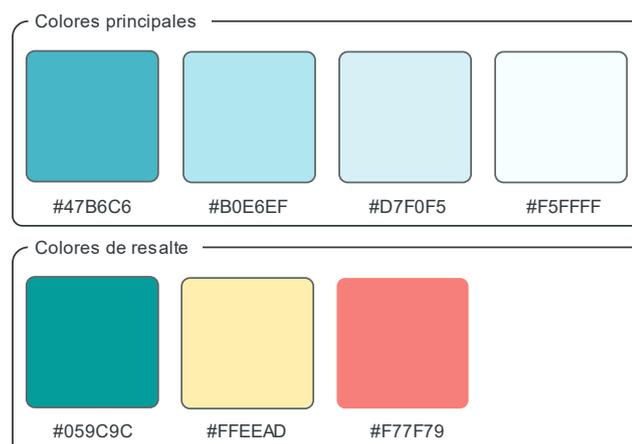


Ilustración 30. Colores de marca

3.6 Funcionamiento

Barra de navegación

La barra de navegación superior está compuesta del logo de la aplicación a la izquierda, que permite acceder a la página principal de búsqueda.

A la derecha, se muestran los botones de búsqueda, creación y perfil de usuario.

El botón de búsqueda lleva a la página de búsqueda, el botón de creación permite elegir si dar de alta un lugar o una ruta y el botón de perfil cambia dependiendo del estado de *login* del usuario.

Si el usuario no tiene sesión iniciada, aparece una etiqueta de Iniciar sesión que permite acceder a la página de inicio de sesión o registro. Si el usuario tiene la sesión iniciada, verá su avatar y nombre de usuario, a través de este enlace podrá acceder a su perfil.



Ilustración 32. Barra de navegación (Sesión iniciada)

Cuando se reduce la anchura de pantalla, la barra de navegación modifica su apariencia; desaparecen los botones de la izquierda para mostrar el botón de tipo 'Hamburguesa'. Al clicar en ese botón se despliegan las opciones en vertical.



Ilustración 33. Barra de navegación (Móvil)

En caso de que no haya usuario con sesión iniciada, el menú se modifica para no mostrar el botón de creación de contenido y para mostrar la opción de inicio de sesión.



Ilustración 34. Barra de navegación (Sin sesión iniciada)

Búsqueda

En la pantalla de búsqueda se ve en la parte superior una fila de opciones de búsqueda básica.

A la izquierda, la caja de texto permite realizar una búsqueda por texto, buscando entre los títulos, descripciones o ciudad de los lugares o rutas.

A la derecha, el control de orden permite modificar el orden de disposición de los resultados de búsqueda.

El botón central despliega un panel debajo de filtros adicionales: En él, podemos filtrar por tipo de contenido (rutas o lugares), ciudad (a través de un control que realiza sugerencias de ciudad según el texto escrito), categoría y etiquetas (pudiendo indicar varias a la vez, permitiendo mostrar resultados que contengan al menos una de las etiquetas filtradas).



The image shows a search filter panel with the following elements:

- Search bar: "Introduce tu destino..." with a magnifying glass icon.
- Order control: "Orden" with a list icon and "Valoración" with a downward arrow icon.
- Content type: "Tipo de contenido" with radio buttons for "Rutas" and "Lugares".
- City: "Ciudad" with a dropdown menu showing "Seleccione una ciudad...".
- Days: "Días" with a range selector.
- Category: "Categoría" with a dropdown menu showing "Cualquiera*".
- Tags: "Etiquetas" with a text input field showing "Nueva etiqueta...".

Ilustración 34. Sección de filtros de búsqueda

A medida que se modifican filtros en este panel, podemos ver reflejado la cantidad de filtros empleados en el botón correspondiente mediante un círculo naranja y un número.

A continuación, el botón de lupa ejecuta la búsqueda y despliega los resultados encontrados utilizando los filtros actuales, mientras que al clicar en los botones de paginación se visualizan los resultados de la última búsqueda, independientemente de que se hayan modificado los filtros después de realizar la búsqueda.

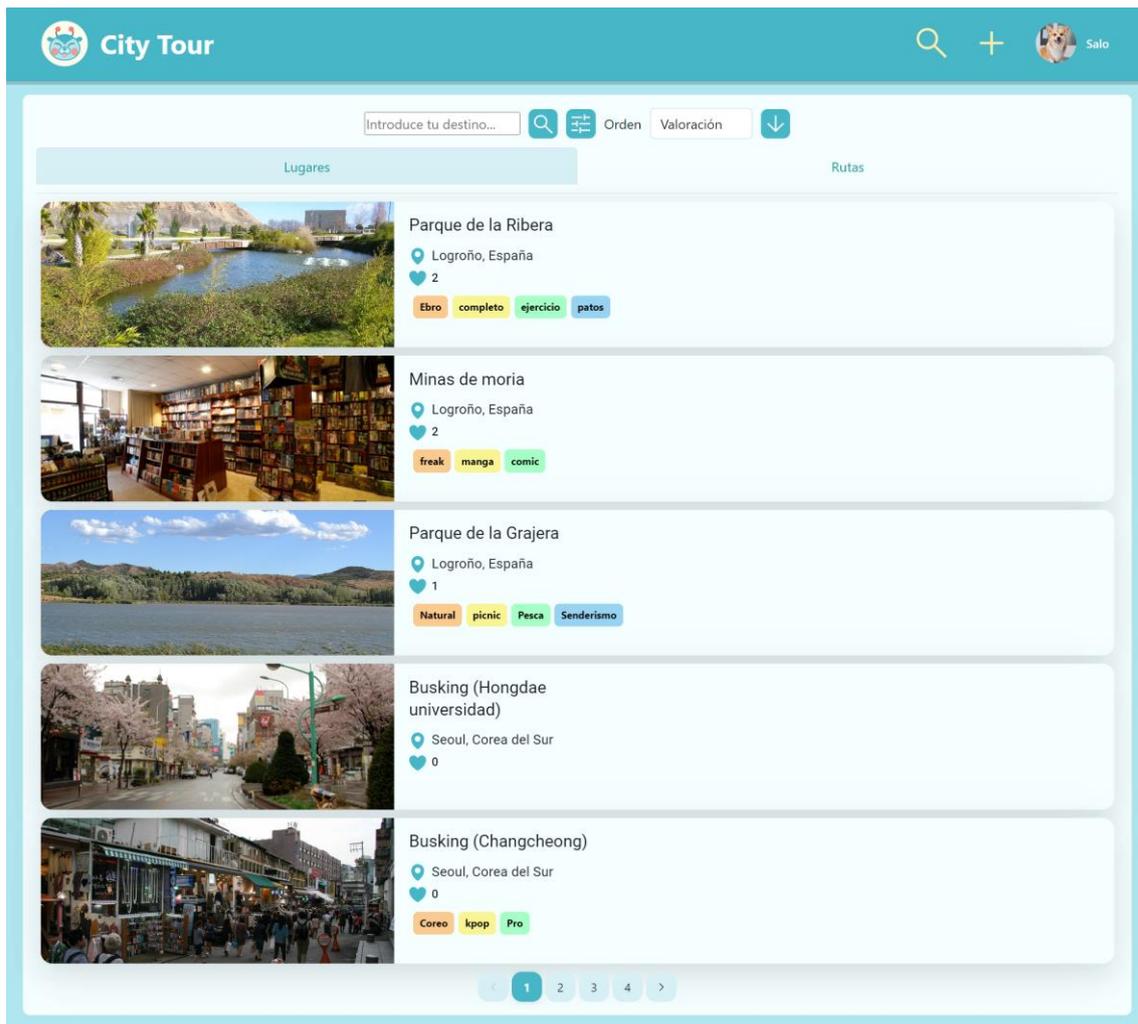


Ilustración 35. Pantalla de búsqueda

Finalmente, en la zona central se muestran los resultados. Esta área contiene un control formado por dos pestañas, donde una está dedicada a los resultados de lugares, y la otra, a rutas. En caso de que sólo haya un tipo de contenido disponible, se muestra sólo la pestaña correspondiente.

Los elementos de lugares están formados por una imagen, su título, ciudad, número de interacciones 'Me gusta' y listado de etiquetas.

Los elementos de ruta están formados por una imagen de su primer lugar, su título, ciudad, número de interacciones de 'Me gusta' y número de lugares. Al clicar en un elemento, se accede al visor de lugar o ruta correspondiente.

Visor de lugar

La pantalla que visualiza un lugar está formada por un mapa interactivo con un marcador indicando la posición del sitio.

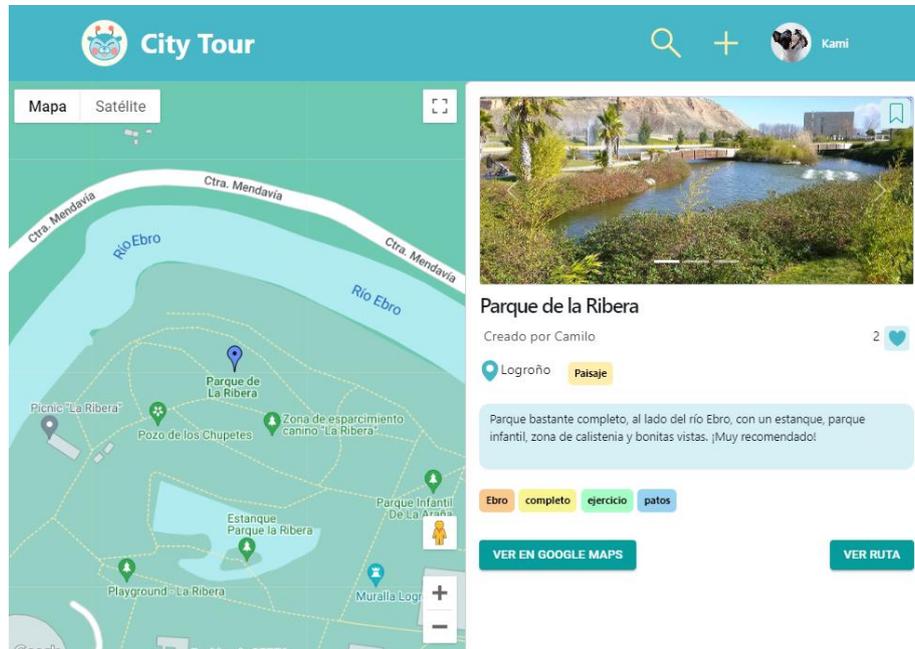


Ilustración 35. Pantalla de visor de lugar

A la derecha, el panel lateral muestra todos los datos del lugar: Un control de tipo Carrusel, el título, el nombre de usuario de creación, la ciudad, categoría principal, número de interacciones, descripción, etiquetas y un botón 'Ver en Google Maps' que permite acceder a Google Maps con una ruta calculada desde la ubicación actual del usuario al punto geográfico del lugar.



Al clicar en la imagen podremos ampliar la imagen actual del carrusel mediante una ventana emergente que emplea el componente Modal de NextUI y el mismo control de carrusel de imágenes.

Ilustración 36. Ventana modal de imagen de lugar

Finalmente, dependiendo del usuario se muestran unas u otras opciones:

- Usuario propietario del lugar, puede acceder al botón de 'Edición' y al de 'Me gusta'
- Usuarios registrados que no son propietarios, tienen acceso al botón de 'Edición' que guarda el lugar.
- Usuarios que no tiene sesión iniciada, no tiene acceso a 'Guardar', 'Editar' ni 'Me gusta'.

Visor de ruta

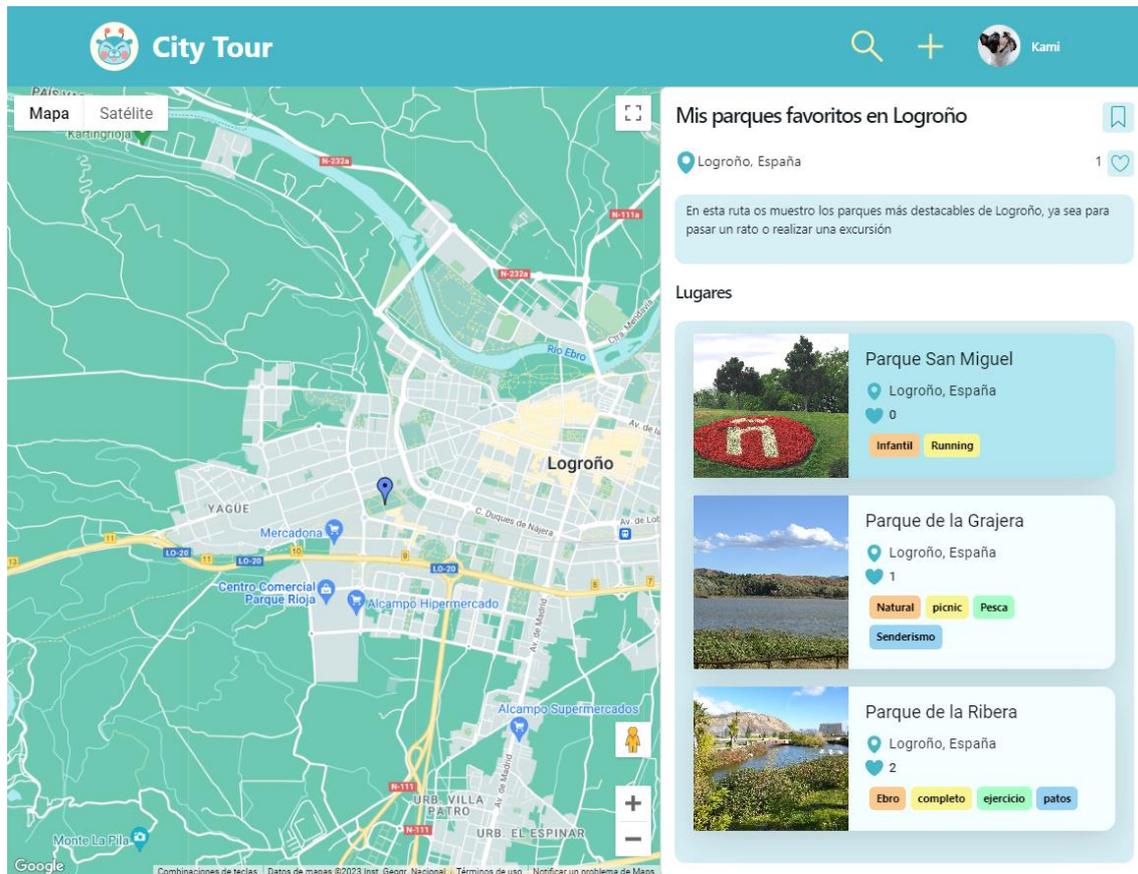


Ilustración 37. Pantalla de visor de ruta

La pantalla que visualiza una ruta, como la de lugar, está compuesta del mapa interactivo y un panel lateral con su información.

El panel lateral muestra, el título, ciudad, número de interacciones 'Me gusta', descripción y el listado de lugares.

De igual forma, los botones de 'Edición', 'Me gusta' y 'Guardar' cambian su visibilidad según el tipo de usuario.

El mapa interactivo muestra los marcadores correspondientes a los lugares que forman la ruta, centrando el mapa de forma que se visualicen todos los marcadores a la vez. Al clicar en cada marcador, se selecciona en el listado de lugares el elemento correspondiente.



Para pantallas de anchura reducida o móvil, se adapta la disposición de los componentes, mostrando el mapa en la parte superior y el panel debajo.

Además, los lugares también se adaptan su composición para adaptarse a la visualización vertical.

Ilustración 38. Pantalla de visor de ruta (formato móvil)

Editor de lugar

Al acceder a la edición de lugar, si es un lugar nuevo, se muestra el mapa interactivo con controles de selección de ciudad y de activación de pin marcador lugar.

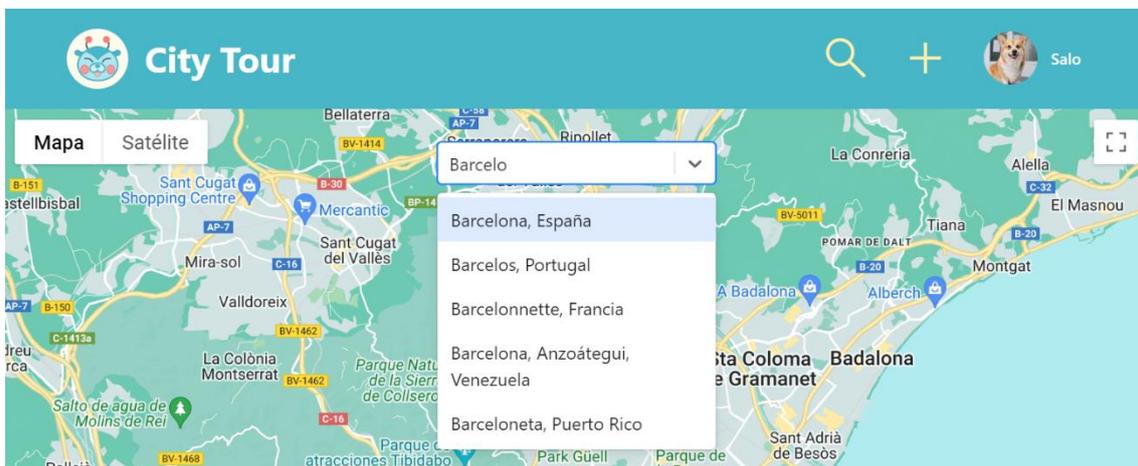
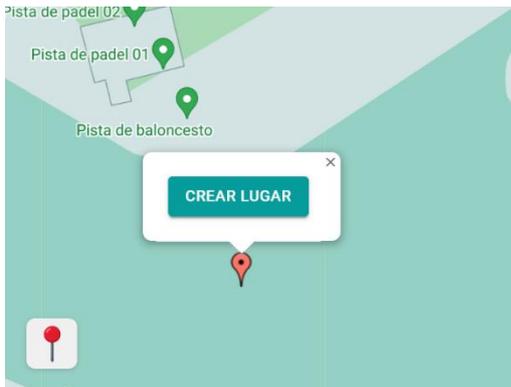


Ilustración 39. Edición de lugar. Selección de ciudad

La selección de lugar permite centrar el mapa rápidamente en la ciudad seleccionada, mientras que el pin modifica el cursor para establecer la posición del lugar a crear.

Al colocar el pin, aparece una ventana flotante sobre el mismo, que muestra el nombre del lugar (si lo tuviera) a través de una consulta a Google Places, y un botón que permite confirmar la asignación de esa posición al lugar a crear.

Mientras la ventana flotante esté activa, se puede cancelar la asignación de ese punto, clicando en la 'x' que la ventana o añadiendo un pin en una nueva posición.



Una vez se confirme la posición mediante el botón de 'Crear lugar', se accede a la edición de sus datos. Todo este proceso se realiza desde la misma página, renderizando el panel de datos en lugar del mapa, si la posición ya ha sido asignada.

Ilustración 40. Edición de lugar. Selección de coordenadas

En el panel podemos establecer los datos del lugar, donde el control que tiene algo más complejidad es el de selección de imágenes, permitiendo elegir hasta cinco imágenes almacenadas en el equipo, así como descartar alguna de las ya seleccionadas.

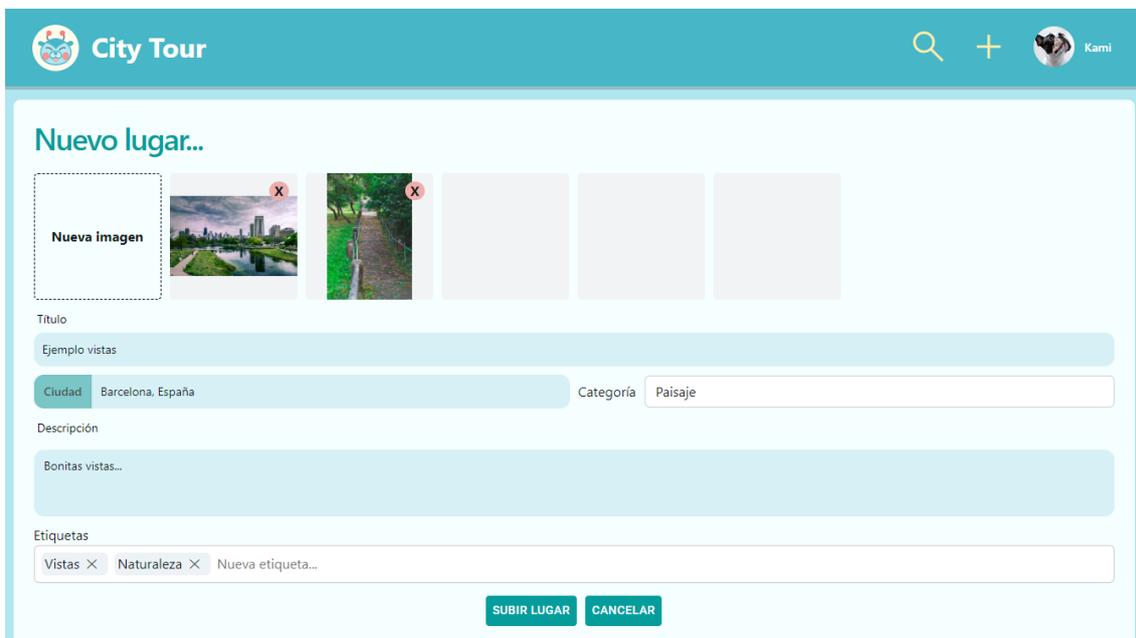


Ilustración 41. Edición de lugar. Introducción de datos

En la parte inferior, el botón cancelar anularía el proceso de creación y redireccionaría a la página principal. El botón de subir lugar ejecuta un proceso de validación de los datos introducidos, y una vez validado, se inserta en base de datos, lanzando un mensaje de confirmación al usuario si se ha insertado correctamente.

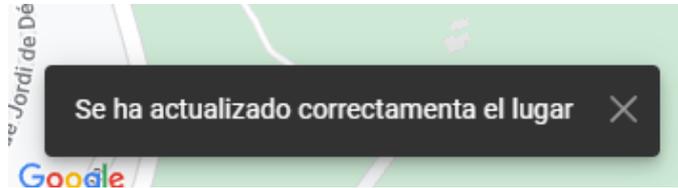


Ilustración 42. Edición de lugar. Mensaje de actualización con éxito

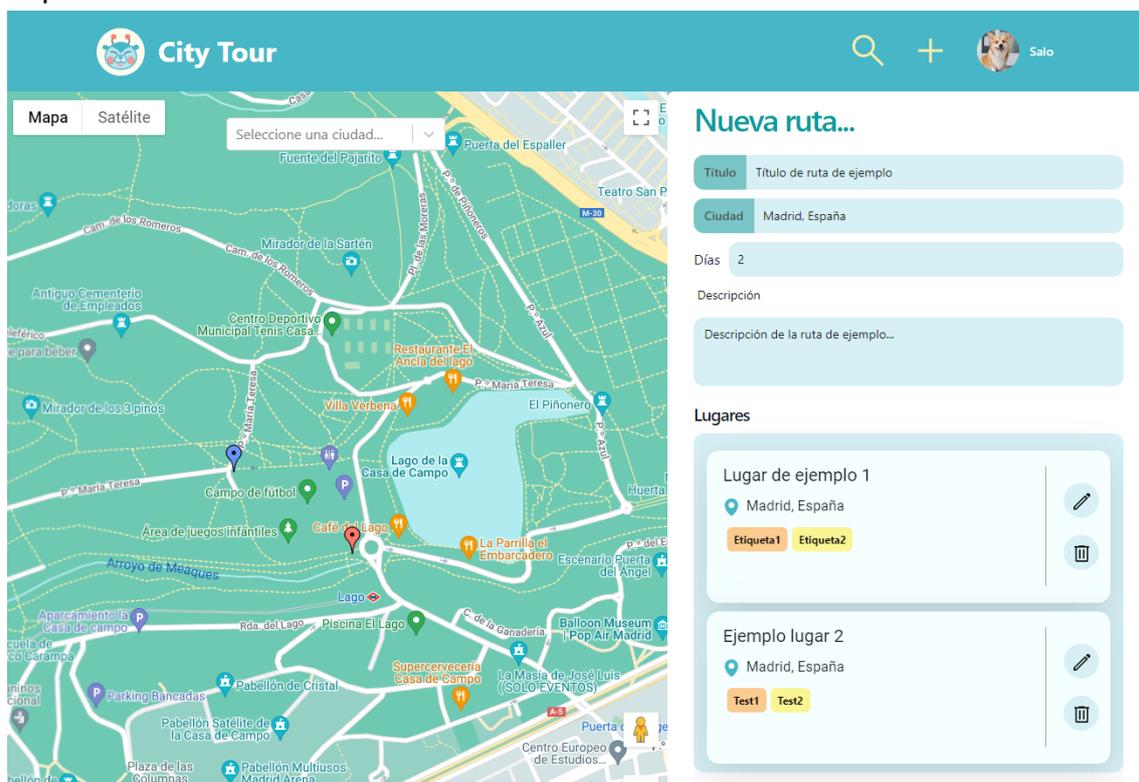
Las validaciones realizadas son las siguientes:

- Se debe especificar al menos una imagen
- Se debe especificar un título
- Se debe especificar una categoría

Esta pantalla altera su funcionamiento dependiendo de si el lugar es nuevo o se está modificando uno existente, ya que en el primer caso se debe realizar una inserción y en el segundo una modificación del registro en base a su id.

Editor de ruta

La pantalla de edición de ruta se compone del mapa interactivo a la izquierda y el panel de edición de datos a la derecha.



Mediante el panel lateral se editan los datos de la ruta y se consultan los lugares que la componen en ese momento.

A través del mapa se pueden añadir pines de la misma forma que el editor de lugar, una vez se confirma un lugar se accede a la edición de lugar. Desde ella podremos confirmar la creación del lugar mediante 'Subir lugar' o cancelar.

En esta última opción se realiza una confirmación de la cancelación mediante una ventana modal (componente NextUI), ya que es conveniente proteger esta acción frente a una posible cancelación por error, dada la capacidad de una ruta de contener gran cantidad de datos a través de los lugares que la componen.

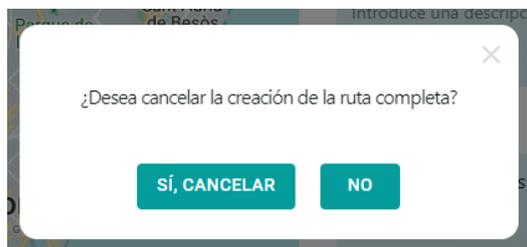


Ilustración 44. Ventana de confirmación de cancelación de ruta

Es importante tener en cuenta que el botón 'Subir lugar' en este caso no inserta el lugar en base de datos, si no que se almacena como estado de la aplicación con Redux. Esta acción redirigirá a la edición de ruta (que también se ha almacenado como estado) y se verá reflejado el lugar añadido en la lista de lugares del panel lateral.

En el listado de lugares, los elementos cuentan con unas opciones diferentes a los elementos de lugar que se muestran en los listados de búsqueda. En este caso, no se muestra su imagen y permiten acceder a la edición del lugar o a la eliminación de este del listado de lugares de la ruta.

Al clicar en 'Subir ruta' se validan los datos y se insertan en base de datos tanto la ruta, como todos los lugares que la componen.

Las validaciones a realizar con las siguientes:

- Se debe especificar un título
- Se debe especificar una ciudad
- La ruta debe contener al menos un lugar.

Al igual que en la edición de lugar, se altera el funcionamiento dependiendo de si la ruta es nueva o ya existente. Además, las modificaciones o eliminaciones en lugares se realizan al instante en caso de ser una ruta existente, a diferencia de las rutas nuevas, que insertan todos sus datos y lugares en bloque al guardar la misma.

Durante la fase de desarrollo, se tomó la decisión de no implementar la funcionalidad de eliminación de lugares o rutas creadas por el usuario, ya que, pese a considerarse un requisito principal, no aportaba, a nivel personal, novedades en la capa de programación. Por ello, se priorizó la optimización de otras áreas que requerían correcciones u optimizaciones.

Perfil

La pantalla de Perfil muestra la información relacionada con el usuario, su nombre de usuario, correo electrónico y los elementos con los que ha interactuado.

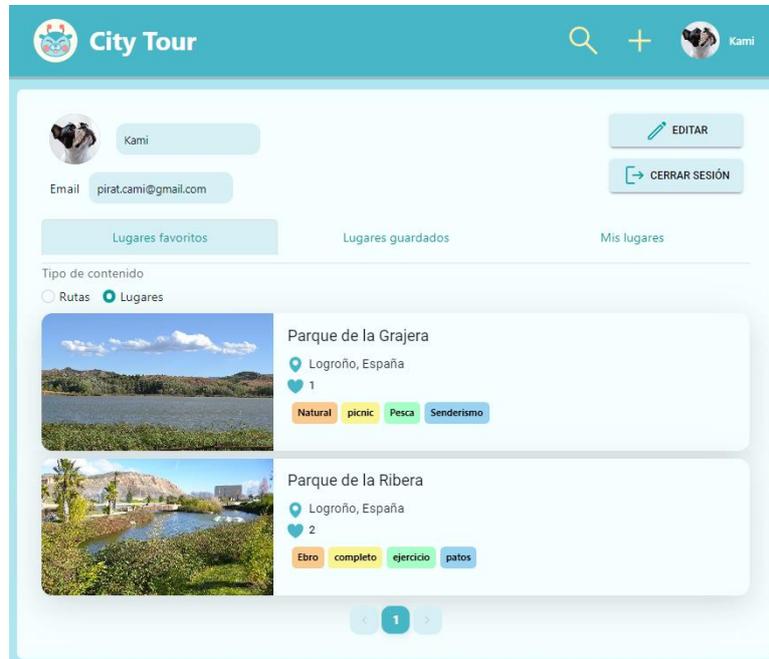


Ilustración 45. Pantalla de Perfil de usuario

En la pestaña 'Lugares favoritos' se listan los elementos marcados con 'me gusta' por el usuario, en 'Lugares guardados' se ven los elementos marcados con el marcapáginas y en la pestaña 'Mis lugares' se muestran los lugares o rutas creadas por el usuario.

Los listados de contenido se cargan al cambiar de pestaña o modificar el control de tipo de contenido y cuentan con un control de paginación que recuerda la página que se estaba consultando a medida que nos movemos a través de las pestañas y filtros de tipo de contenido. Esto se realiza mediante la gestión del estado de un objeto que almacena los datos de página actual y total de páginas por tipo de pestaña y por tipo de contenido.



Ilustración 46. Perfil de usuario, edición activada

Mediante el botón 'Editar' podemos modificar el nombre de usuario y la imagen de avatar. Mientras está en modo edición, aparece otro botón a la derecha del de guardar, para anular los cambios realizados o confirmarlos mediante el botón correspondiente. Finalmente, con 'Cerrar sesión', se cierra la sesión del usuario actual.

En caso de no tener cuenta creada o sesión iniciada, se puede acceder mediante el botón de la barra de navegación de 'Iniciar sesión'. Se trata de un componente facilitado por Supabase que permite realizar el registro, inicio de sesión o inicio de sesión mediante correo en caso de olvidar la contraseña.



Ilustración 47. Inicio de sesión

Este apartado no se ha explorado a fondo, ya que el sistema permite darse de alta e iniciar sesión con contraseña o confirmación en email, pero tras registrarse se ha de asignar nombre de usuario e imagen de forma manual desde el perfil, y, por otra parte, no se da la posibilidad de modificar la contraseña o correo electrónico.

Con respecto al prototipo realizado y a las funciones planteadas en la planificación, se ha incorporado el listado de 'lugares favoritos' ya que su funcionamiento es similar al de 'lugares guardados'. Por otra parte, el botón de deshacer edición se incorporó durante la fase de pruebas, el cual no se preveía desarrollar la edición del perfil, sin embargo, la planificación temporal permitió agregar funcionalidad básica en esta área.

3.7 Acceso a sitio web

El sitio web es accesible desde el siguiente enlace, a través de Vercel:

<https://tfqct13.vercel.app/>

5. Conclusiones y trabajos futuros

A través de este proyecto se ha logrado presentar un prototipo con las funciones básicas presentadas en el planteamiento inicial. Además, se ha logrado el objetivo personal de desarrollar un proyecto web mediante NextJS y React a la vez que se asimilaba el aprendizaje de este nuevo entorno.

Con respecto a la planificación del trabajo, se han realizado estimaciones de tiempo ajustadas que me han permitido lograr los aspectos esenciales pero no expandir las funciones u optimizar el funcionamiento. Por otra parte, las fases finales de testing, correcciones y optimizaciones han posibilitado manejar las contingencias que se preveían, como la curva de aprendizaje del nuevo entorno, la gestión de errores o la limitación de tiempo de desarrollo.

En esta línea, el límite temporal del proyecto no ha permitido expandir, optimizar y explorar al completo todas las posibilidades de React. Como consecuencia, el entregable presenta funcionalidades con alcance limitado, especialmente en el área de registro y perfil, o en la carencia de características sociales (comentarios, consulta de perfiles...).

Durante la etapa de implementación se comprendieron mejor las funciones requeridas y, por tanto, se realizaron modificaciones de algunas prioridades. De esta forma, se dio prioridad a la gestión del login de usuario, al menos en sus operaciones básicas, y al diseño responsive. Además, se dedicó un tiempo adicional a un apartado no planeado al inicio del proyecto, el diseño de identidad visual.

Mejoras futuras

En cuanto a posibles optimizaciones para el sitio web se encuentran, en primer lugar, la implementación de las funciones complementarias proyectadas en la planificación y no realizadas en este prototipo debido al límite de tiempo de desarrollo, así como numerosas validaciones de datos que han quedado fuera del alcance del proyecto.

En segundo lugar, la ejecución de una fase de testing con usuarios que permita obtener un *feedback* para realizar modificaciones sobre el funcionamiento actual, detectar mejoras, errores y nuevas funciones no planteadas.

En tercer lugar, optimizaciones en dispositivo móvil, tales como mejoras en la creación de marcadores en mapa mediante arrastre, la actualización de la navegación entre páginas, ya que actualmente se pierden los estados de la aplicación al retroceder, especialmente en la pantalla de búsqueda con el estado de los filtros o la personalización de la apariencia (modo claro y oscuro).

En cuarto lugar, una adaptación de los componentes y las páginas para garantizar la accesibilidad web, especialmente con los lectores de pantalla.

Por último, un replanteamiento del área de registro y login de usuarios, ya que al haberse implementado mediante las APIs de Supabase se pierde el control de los eventos y otros aspectos, como la personalización de los emails de confirmación de registro o el restablecimiento de contraseña.

6. Glosario

Front-end: Diseño de interfaz de usuario y desarrollo de la interacción del usuario con la información

Back-end: Desarrollo de la lógica de la aplicación que engloba la gestión de la base de datos y el procesamiento detrás de la aplicación.

Bucket: Sistema de almacenamiento de imágenes propio de Supabase.

API: Interfaz de programación de aplicaciones que ofrece datos o funciones específicas.

DOM: Interfaz de programación que permite crear, modificar o eliminar elementos de un documento Web.

HTML: Lenguaje de marcas que define la estructura de las páginas web.

CSS: Lenguaje que describe la presentación de un documento html.

Github: Plataforma empleado para almacenar código fuente de un proyecto y realizar un seguimiento de los cambios realizados por uno o más desarrolladores.

NextJS: Framework de React que permite desarrollar una aplicación web, ofreciendo una estructura, funciones y optimizaciones.

React: Biblioteca de código abierto basada en javascript que permite construir interfaces y aplicaciones web de forma rápida y eficiente.

Componente: Fragmentos de código reutilizable e independiente que cumple con una función determinada.

Componente funcional: Es una función que recibe unos parámetros (props) y devuelve un nodo React. Se puede emplear para definir la estructura y funcionamiento de un componente individual o una página compuesta de varios componentes.

Hook: Funciones de React que permiten tener estados en los componentes funcionales.

Framework: Marco de trabajo empleado en programación para el desarrollo de software.

Vista: Tabla virtual cuyo contenido procede de una consulta de base de datos.

PostgreSQL: Sistema de gestión de base de datos de código abierto y orientado a objetos.

Supabase: Plataforma orientada al desarrollo de aplicaciones web y móviles que ofrece las funciones de base de datos, login de usuarios o almacenamiento de documentos.

Redux: Librería que permite almacenar estados de una aplicación web que pueden ser reutilizados desde cualquier componente de la aplicación.

Javascript: Lenguaje de programación interpretado, orientado a objeto y débilmente tipado. Se utiliza principalmente para dotar de interactividad a las páginas web.

Landing page: Página web diseñada específicamente para atraer la atención de usuarios potenciales o para dirigirlos a realizar alguna acción en concreto.

Diseño responsive: Técnica de diseño web cuyo objetivo es crear sitios web que se adapten y respondan automáticamente a diferentes dispositivos y tamaños de pantalla.

7. Bibliografía

- 1- Los españoles viajarán igual o más que antes de la pandemia de la COVID-19. Carmen Porras. 18 de agosto de 2021. [Fecha de consulta: 6 de marzo de 2023]. Disponible en: https://www.hosteltur.com/146075_los-espanoles-viajaran-igual-o-mas-que-antes-de-la-pandemia-de-la-covid-19.html
- 2- POSTPANDEMIA Y TURISMO SLOW: ANÁLISIS Y ESTUDIO EMPÍRICO. Nuria Portela Gabian. 2020. [Fecha de consulta: 6 de marzo de 2023]. Disponible en: https://ruc.udc.es/dspace/bitstream/handle/2183/29222/PortelaGabian_Nuria_TFM_2021.pdf
- 3- Post-COVID-19 Tourists' Preferences, Attitudes and Travel Expectations: A Study in Guayaquil, Ecuador. Miguel Orden-Mejía. 15 de abril de 2022. [Fecha de consulta: 6 de marzo de 2023]. Disponible en: <https://www.mdpi.com/1660-4601/19/8/4822>
- 4- Tourists' Travel-Related Information Search Channels. Sunny Sun. 20 de diciembre de 2018. [Fecha de consulta: 6 de marzo de 2023]. Disponible en: <https://www.tandfonline.com/doi/full/10.1080/15256480.2020.1727809?needAccess=true>
- 5- Channels for Searching Hotel and Travel Information. Sunny Sun. 5 de enero de 2017. . [Fecha de consulta: 6 de marzo de 2023]. Disponible en: https://link.springer.com/chapter/10.1007/978-3-319-51168-9_32
- 6- Conoce el impacto al medio ambiente del uso de internet. Diario ecología. [Fecha de consulta: 10 de marzo de 2023]. Disponible en: <https://finanzasparamortales.es/conoce-el-impacto-al-medio-ambiente-del-uso-de-internet/>
- 7- Turismo cultural y planificación del viaje: Un estudio de caso. Marrero Rodríguez, J. Rosa; Abdul-Jalbar, Beatriz. Tourism & Management Studies, 2012. [Fecha de consulta: 11/03/2023]. Disponible en: <https://www.redalyc.org/pdf/3887/388743870005.pdf>
- 8- Tourists' Travel-Related Information Search Channels. Sunny Sun. 30 de diciembre de 2018. [Fecha de consulta: 11/03/2023]. Disponible en: <https://www.tandfonline.com/doi/full/10.1080/15256480.2020.1727809?needAccess=true>
- 9- Página web de Minube. Disponible en: <https://www.minube.com/>
- 10- Aplicación AroundMe. Disponible en: <https://apps.apple.com/es/app/aroundme/id290051590>
- 11- Documentación de Supabase. Disponible en: <https://supabase.com/docs/reference/javascript/installing>
- 12- Distribución porcentual de los viajes realizados por los hombres y las mujeres residentes en España en 2021, por edad. Statista. Consultado el: 05/03/2023. Disponible en: <https://es.statista.com/estadisticas/702206/viajes-de-los-hombres-y-mujeres-residentes-en-espana-distribuidos-por-edad/?locale=es>

- 13- Número de viajes por ocio, recreo y vacaciones de turistas residentes en España entre 2016 y 2021. Statista. Consultado el: 05/03/2023. Disponible en: <https://es.statista.com/estadisticas/515135/viajes-de-ocio-recreo-y-vacaciones-de-residentes-en-espana/>
- 14- Why travel prolongs happiness: Longitudinal analysis using a latent growth model. Jangwook Kwon, Hoon Lee. Febrero de 2020. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0261517719301347>
- 15- Documentación API Supabase. Disponible en: <https://supabase.com/docs/reference/javascript/installing>
- 16- Documentación ReactJs. Disponible en: <https://legacy.reactjs.org/>
- 17- Documentación Google Maps Platform. Disponible en: <https://developers.google.com/maps/documentation?hl=es-419>
- 18- Documentación API Redux. Disponible en: <https://redux.js.org/api/api-reference>
- 19- Documentación componentes React Bootstrap. Disponible en: <https://react-bootstrap.github.io/components/alerts>
- 20- Documentación componentes NextUI. Disponible en: <https://nextui.org/docs/guide/getting-started>
- 21- Documentación componentes Material UI. Disponible en: <https://mui.com/material-ui/getting-started/overview/>
- 22- Documentación control Google Places Autocomplete. Disponible en: <https://www.npmjs.com/package/react-google-places-autocomplete>
- 23- Documentación Componente GoogleMap. Disponible en: <https://react-google-maps-api-docs.netlify.app/#googlemap>
- 24- Documentación Componente react-tag-input. Disponible en: <https://www.npmjs.com/package/react-tag-input-component>

8. Créditos

- Imagen Grajera 1. Realizada por Edu1975. Disponible en: https://commons.wikimedia.org/wiki/File:Parque_de_la_Grajera_Logro%C3%B1o_-_panoramio.jpg
- Imagen Grajera 2. Realizada por Elsiloesar. Disponible en: https://es.wikipedia.org/wiki/Archivo:Logro%C3%B1o_Parque_de_la_Grajera.jpg
- Imagen Parque San Miguel. Realizada por LBM1948. Disponible en: https://commons.wikimedia.org/wiki/File:Logro%C3%B1o,_%C3%91_1.jpg
- Imágenes parque de la Ribera 1. Realizadas por zarateman. Disponibles en: https://commons.wikimedia.org/wiki/Category:Parque_de_la_Ribera_Logro%C3%B1o#/media/File:Logro%C3%B1o_-_Parque_de_la_Ribera_25.jpg, https://commons.wikimedia.org/wiki/Category:Parque_de_la_Ribera_Logro%C3%B1o#/media/File:Logro%C3%B1o_-_Parque_de_la_Ribera_15.jpg, https://commons.wikimedia.org/wiki/Category:Parque_de_la_Ribera_Logro%C3%B1o#/media/File:Logro%C3%B1o_-_Parque_de_la_Ribera_21.jpg
- Imagen de Tienda Minas de Moria. Realizada por Juan Kim en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipN_BYdOm7HjjLzjQfkq6LI73bDHTn1izOqu0B1g=s680-w680-h510
- Imagen de calle Leganitos. Realizada por Luis García. Disponible en: [https://commons.wikimedia.org/wiki/Category:Calle_de_Leganitos,_Madrid#/media/File:Calle_de_Leganitos_\(Madrid\)_02.jpg](https://commons.wikimedia.org/wiki/Category:Calle_de_Leganitos,_Madrid#/media/File:Calle_de_Leganitos_(Madrid)_02.jpg)
- Imagen de Tienda SugoiStation 1. Realizada por Family Apple en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipMiAT_Uy17i68F0rnGIHtesM_qsM8RyLlqCSG_Q=s680-w680-h510
- Imagen de Tienda SugoiStation 2. Realizada por Kansuke Yamamoto en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipMXiS_0pNUdMGIP_ZXs7IPRZCghWvxb1_IIEk-J=s680-w680-h510
- Imagen de Tienda SugoiStation 3. Realizada por Merime M en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipOBdK3pgHaKR2H-0AP0dPboID2P_W9ASCzE06jY=s680-w680-h510
- Imagen de Generación X (Conde de Romanones, Madrid) 1. Realizada por Generación X en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipNyogIsvt38xcvKQIV_z5HVXQsSmi9Jck_ZNIw=s680-w680-h510
- Imagen de Generación X (Conde de Romanones, Madrid) 2. Realizada por Ruben Pandiella en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipOvYpbkY6jOT9HE-5Zx_kL05dkyA7EVzFlwrBwZ=s680-w680-h510
- Imágenes de tienda Zacatrus (Fernández de los Ríos, Madrid). Realizadas por Vicente Corralero en Google. Disponibles en: <https://lh3.googleusercontent.com/p/AF1QipPMIYNk4e2DHqDSE2qzEaUfiySTxi--WCUCmfO=s680-w680-h510>, <https://lh3.googleusercontent.com/p/AF1QipOQI8ueZbVOM-V7WHhudyflwhLqXaVMLd3Gqn3u=s680-w680-h510>

- Imagen de Restaurante Gourmet Experience (Gran vía, Madrid). Realizada por Alesion Guiggi en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipP6AQfwXzgYRW_tbWFLtj4kEcZidJxxQGzxrZi=s680-w680-h510
- Imagen de Restaurante Gourmet Experience (Gran vía, Madrid). Realizada por Gourmet Experience Gran Vía en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipPlmVpY0pVpBvYDb_0wKx7s2dxKCzACkSX66lk=s680-w680-h510
- Imagen de Restaurante Gourmet Experience (Gran vía, Madrid). Realizada por Trachemysta en Google. Disponible en: https://lh3.googleusercontent.com/p/AF1QipOEuyJyd2iRdolnel4kgdF1YNHSdoJQUI_nWfoV=s680-w680-h510
- Imagen vistas parque 1 (Utilizada en la demo de creación de lugar). Realizada por Nancy Bourque en pexels. Disponible en: <https://www.pexels.com/es-es/foto/rio-cerca-de-los-edificios-de-la-ciudad-bajo-el-cielo-nublado-1209978/>
- Imagen vistas parque 2 (Utilizada en la demo de creación de lugar). Realizada por Markus Spiske. Disponible en: <https://www.pexels.com/es-es/foto/naturaleza-bosque-arboles-hierba-192320/>
- Imagen Busking Hongdae (cantante). Realizada por 달보드레 (Dalbodre). Disponible en: https://commons.wikimedia.org/wiki/File:160909_%ED%99%8D%EB%8C%80%EB%B2%84%EC%8A%A4%ED%82%B9_%EC%98%A4%EB%B8%8C%EB%A1%9C%EC%A0%9D%ED%8A%B8_%EC%98%A4%ED%83%9C%EC%84%9D_5.jpg
- Imagen Busking Hongdae (grupo). Realizada por Seoul Arts-Occupational Training Collage. Disponible en: [https://commons.wikimedia.org/wiki/File:%EC%8B%A4%EC%9A%A9%EC%9D%8C%EC%95%85%EA%B3%BC_%EC%8B%A0%EC%B4%8C_%EB%B2%84%EC%8A%A4%ED%82%B9_%EA%B3%B5%EC%97%B0_%ED%98%84%EC%9E%A5!_\(3\).png](https://commons.wikimedia.org/wiki/File:%EC%8B%A4%EC%9A%A9%EC%9D%8C%EC%95%85%EA%B3%BC_%EC%8B%A0%EC%B4%8C_%EB%B2%84%EC%8A%A4%ED%82%B9_%EA%B3%B5%EC%97%B0_%ED%98%84%EC%9E%A5!_(3).png)
- Imagen Busking Hongdae (cantantes). Realizada por Taehyung Jin. Disponible en: [https://commons.wikimedia.org/wiki/File:Busking_\(121004051\).jpeg](https://commons.wikimedia.org/wiki/File:Busking_(121004051).jpeg)
- Imagen Busking Changcheong. Realizada por Estelle Pizer en Flickr. Disponible en: <https://www.flickr.com/photos/estellephotos/5742667222>
- Imagen Comida callejera Seúl 1. Realizada por LinasD. Disponible en: https://commons.wikimedia.org/wiki/File:Seoul_street_food_2011_march_02.jpg
- Imagen Comida callejera Seúl 2. Realizada por tragrpx. Disponible en: [https://commons.wikimedia.org/wiki/File:Tornado_gamja_\(Seoul_street_food\)_tornado_potato.jpg](https://commons.wikimedia.org/wiki/File:Tornado_gamja_(Seoul_street_food)_tornado_potato.jpg)
- Imagen Comida callejera Seúl 3. Realizada por Piotrus. Disponible en: https://commons.wikimedia.org/wiki/File:Ansan_fall_2014_11.JPG
- Foto de avatar Bulldog Francés. Realizada por Martin Dufosset en Pexels. Disponible en: <https://www.pexels.com/es-es/foto/bulldog-de-pelo-corto-blanco-y-negro-3930940/>
- Foto de Avatar Corgi. Realizada por Freepik. Disponible en: https://www.freepik.es/foto-gratis/perro-esperando-oficina-veterinario_20823326.htm
- Icono vectorial Corazón. Realizado por juici__fish en freepik. Disponible en:

- Icono de marcapáginas. Realizado por Nigar Novruzova en vecteezy. Disponible en: <https://www.vecteezy.com/vector-art/21916475-bookmark-social-media-icon-isolated-on-background>
- Icono de localización. Realizado por juicy_fish en freepik. Disponible en: https://www.freepik.es/vector-gratis/vector-plano-basico-corazon_41285191.htm