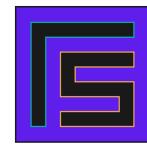


PROYECTO INTERMODULAR



MOVE
ON

MIGUEL HERNÁNDEZ MONLLOR
MIGUEL ÁNGEL GRIMAL



Índice

1. Introducción y nuestra ruta.....	4
1.1 Herramientas de la Aplicación.....	4
1.2 Inicio del Desarrollo y Prototipado.....	5
1.3 Desarrollo del Frontend.....	6
1.4 Desarrollo del Backend y Base de Datos.....	7
1.5 Pruebas, Integración y Despliegue.....	7
2. Diseño.....	8
2.1 Tecnologías y herramientas.....	8
2.1.1 FrontEnd Biblioteca de JS React.....	8
2.1.2 BackEnd API Restful en Laravel.....	9
2.1.3 Herramientas de desarrollo.....	10
2.1.4 Bases de datos.....	11
2.1.5 Control de versiones.....	11
2.1.6 Contenedores.....	12
2.2 Arquitecturas escogidas.....	13
2.3 Diseño de las interfaces entre los elementos.....	14
2.4 Diseño de la interfaz de usuario.....	15
2.4.1 Tipografía.....	15
2.4.2 Colores.....	16
2.4.3 Iconografía.....	17
2.4.4 Mockups.....	18
2.4.4.1 Bajo nivel.....	18
2.4.4.2 Alto nivel.....	19
2.4.5 Versión funcional.....	20



2.4.5.1 Página de inicio (Modo oscuro).....	20
2.4.5.2 Página de inicio (Modo claro).....	21
2.4.5.3 Página de inicio en móvil (Modo claro y oscuro).....	22
2.4.6 Utilidades de Tailwind CSS.....	23
3. Bases de datos.....	25
3.1 Modelo entidad-relación.....	25
3.2 Modelo relacional.....	26
4. Demostraciones de la APP.....	29
4.1 Registro e inicio de sesión.....	29
4.2 Perfil del usuario.....	31
4.3 Perfil del administrador.....	32
4.4 Dietas personalizadas.....	33
4.5 Rutinas y ejercicios.....	36
4.6 Dejar de fumar.....	39
4.7 Calendario (Beta).....	40
4.8 Mapamundi de ubicaciones destacadas.....	41
4.9 Entrenador personal AI.....	42
4.10 Asistente virtual Ramón AI.....	44
Unos claros ejemplos:.....	45
4.11 Pruebas de funcionamiento con usuarios en Postman.....	46
5. Subida a AWS.....	48
6. Agentes de IA (N8N).....	52
7. Futuras mejoras y ampliaciones.....	56
8. Comentarios y conclusiones.....	57
8.1 Valoraciones personales y técnicas.....	58
9. Bibliografía y anexos.....	59



1. Introducción y nuestra ruta

Somos **Safe Future**, una startup creada por dos estudiantes del IES Poeta Paco Molla. Se nos encomendó desarrollar una aplicación web funcional sobre cualquier temática. Tras una lluvia de ideas en la que surgieron tres propuestas, decidimos reservar dos de ellas para proyectos futuros y centrarnos en la opción más prometedora: “**Move On**”.

“**Move On**” es una aplicación diseñada para ayudar a las personas a llevar una vida saludable y, en definitiva, a rehacer su vida. La idea surge al observar cómo algunas personas se centran en comportamientos que no contribuyen a su bienestar —como salir de fiesta en exceso o involucrarse en actividades autodestructivas—, mientras que, en la actualidad, la depresión se ha convertido en una epidemia silenciosa y costosa de tratar. Con esta aplicación, buscamos aportar un beneficio social que trascienda lo meramente económico.

Para lograrlo, definimos desde el inicio que la base de una vida plena y el desarrollo personal se sustentan en pilares como las relaciones sinceras, el autoconocimiento, la gratitud, el propósito y el equilibrio. Creemos firmemente que cultivar el amor propio, disfrutar del presente y aprender de cada experiencia fortalece la resiliencia y el bienestar.

1.1 Herramientas de la Aplicación

Para ofrecer a nuestros usuarios una experiencia única, “Move On” incluirá varias herramientas principales a destacar:

- **Creador de rutinas de ejercicio:** El deporte favorece la liberación de dopamina, lo cual es especialmente beneficioso para quienes llevan una vida sedentaria, como suele pasar con los programadores.



- **Creador de dietas:** Permitirá a los usuarios gestionar su alimentación y mantener un control efectivo sobre su peso.

En resumen, la aplicación integrará un creador de dietas, un creador de ejercicios y recursos para ayudar a los usuarios.

1.2 Inicio del Desarrollo y Prototipado

El primer paso fue gestionar diversos aspectos: diseñar, plasmar la idea en papel y convertirla en una web funcional. Con la orientación de nuestra profesora de Diseño Web, **Rosa Aravid**, aprendimos a crear plantillas y a establecer una guía de estilos, lo que se convirtió en una práctica fundamental. Aunque al inicio las ideas eran algo vagas, nuestro objetivo siempre fue claro: crear la mejor web de Desarrollo Personal.

La inspiración llegó cuando a **Miguel** se le ocurrió comenzar por el **header**, definiendo las pestañas esenciales de la web. Con esta estructura base, repartimos las tareas: uno se encargó del prototipado y el otro de detallar aspectos como fuentes y colores. Tras plasmar nuestras ideas en papel, desarrollamos el diseño de bajo nivel, creando versiones tanto para dispositivos móviles como para escritorio. Una vez definidos los colores, avanzamos hacia un diseño de alto nivel, que se asemejaría al CSS final.

Al finalizar, presentamos nuestro prototipo a Rosa, quien nos otorgó una calificación de 9,5 confirmándose que estábamos en el camino correcto.

1.3 Desarrollo del Frontend

Una vez completada la guía de estilos, nos centramos en aprender a programar y conectar el backend con el frontend. Tras un breve descanso navideño,



retomamos las clases y, teóricamente, contábamos con un mes para finalizar el proyecto. Sin embargo, surgieron algunos contratiempos cuando nuestro profesor de Entorno Cliente, **Juan Carlos**, nos asignó la compleja tarea de crear una web funcional de lista de la compra.

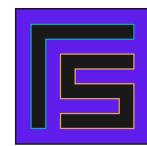
Inicialmente, **Miguel Ángel** pensó en reutilizar parte del código de una práctica anterior, pero pronto se nos indicó que no podíamos copiar ni utilizar su código, CSS o React. Fue entonces cuando decidimos adaptar el CSS basándonos en la guía de estilos que habíamos elaborado. Personalmente, me encargué de desarrollar el header —aun cuando el primer CSS no era perfecto, lo esencial era lograr un aspecto atractivo— y diseñé la página de inicio utilizando una plantilla de CSS para formularios de acceso que me resultó muy bonita.

Implementé la funcionalidad con **React** y, para realizar pruebas con la base de datos, utilizamos **Supabase**, herramienta que ya conocíamos por las clases de Juan Carlos y que facilitó el manejo de la parte de BBDD. La integración completa de la base de datos se realizará posteriormente con el profesor de Desarrollo Servidor, **Miguel Ángel Tomás Amat**. Una vez finalizado el diseño, creamos páginas adicionales y configuramos las rutas (routes) con un diseño coherente y atractivo.

1.4 Desarrollo del Backend y Base de Datos

Al finalizar el segundo trimestre y concluir las prácticas restantes, comenzamos a desarrollar el backend (API) y la base de datos del proyecto. Inicialmente, elaboramos un complejo modelo entidad-relación que, para la versión beta, tuvimos que simplificar, ya que aún nos faltaba conectar el backend con el frontend y desplegar la aplicación.

Realizamos pruebas de conectividad mediante SQL en Supabase y, una vez comprobado el funcionamiento, iniciamos el desarrollo de nuestra API utilizando



Laravel (framework PHP) y **MySQL** para la base de datos. Dividimos las tareas: uno se encargó de crear los controllers y definir las rutas, mientras el otro desarrolló los modelos, migraciones y demás elementos necesarios.

1.5 Pruebas, Integración y Despliegue

Después de realizar diversas pruebas y confirmar que todo funcionaba correctamente, nos percatamos de que la presentación de la beta se acercaba (el lunes siguiente). Por ello, era fundamental asegurar el correcto funcionamiento del inicio de sesión. Dado que **Miguel Ángel** debía viajar, se procedió a modificar el contexto y las funcionalidades para que, en lugar de conectarse con Supabase, se integrará la API de Laravel. A pesar de algunos errores iniciales, se lograron solucionar, incluso implementando una función de login en el backend para gestionar contraseñas encriptadas.

Durante el fin de semana, Miguel trasladó gran parte del proyecto a **TailWind**, corrigiendo problemas del CSS original y añadiendo mejoras como ventanas flotantes y configuraciones personalizadas, para mantener siempre un código limpio (clean code).

Finalmente, tras lograr que todo funcionara en **localhost**, quedó pendiente el despliegue de la aplicación. Esta tarea, que nos asignó nuestro tutor **David** (profesor de despliegue), consiste en hacer accesible la web mediante una URL pública y establecer un entorno de desarrollo adecuado. Debido a la sobrecarga de trabajo, dejamos esta parte para el final y, al principio desplegamos la aplicación en **Vercel**. Pero tras un arduo trabajo de investigación conseguimos desplegar la aplicación en **AWS**.



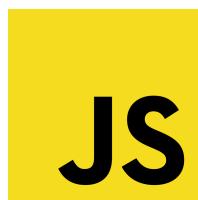
2. Diseño

2.1 Tecnologías y herramientas

2.1.1 FrontEnd

Biblioteca de JS React

Para nuestro frontend hemos empleado la biblioteca de JS React, que es el framework que más hemos utilizado durante el curso, lo que nos ha permitido familiarizarnos bastante con él. Nos gusta por su flexibilidad y su enfoque en utilizar componentes reutilizables Además, su integración con otras herramientas y librerías nos ha facilitado la creación de una interfaz dinámica y moderna.





2.1.2 BackEnd

API Restful en Laravel

Para nuestra API hemos empleado Laravel, el primer framework de PHP que aprendimos para gestionar el CRUD y que nos ha gustado bastante. Nos atrae por su robustez, sintaxis intuitiva y la facilidad que ofrece para implementar el patrón Modelo-Vista-Controlador (más tarde explicamos el porqué), facilitando así el desarrollo y mantenimiento de aplicaciones web seguras y de alto rendimiento.





2.1.3 Herramientas de desarrollo

Para el desarrollo del proyecto, hemos utilizado dos entornos de desarrollo integrados (IDE) diferentes: **Microsoft Visual Studio Code (VSCode)** y **PHPStorm** (IntelliJ en su defecto), cada uno asignado a una parte distinta del stack de la aplicación.



Visual Studio Code



JetBrains PhpStorm

VSCode ha sido nuestra elección para trabajar con **React** en el frontend, principalmente por nuestra familiaridad previa con esta herramienta. Este editor nos proporciona un entorno muy ágil gracias a la amplia variedad de extensiones disponibles, como aquellas que nos facilitan la creación de componentes React o la escritura de clases de Tailwind CSS de forma más rápida y visual. Además, con las terminales integradas hemos podido ejecutar comandos directamente desde el editor, lo que ha mejorado notablemente nuestro flujo de trabajo y nos ha hecho más productivos en tareas como el levantamiento del servidor de desarrollo, el uso de **npm** o el despliegue con Vite.

Por otro lado, **PHPStorm** ha sido clave para el desarrollo del **backend en Laravel**. Una de sus ventajas más destacables ha sido su **sistema de autocompletado inteligente**, que nos ha ayudado a escribir código más limpio y con menos errores, especialmente en estructuras complejas como controladores, rutas o migraciones. Otra funcionalidad crucial ha sido la integración nativa con **Docker**, que nos ha permitido ver y gestionar directamente los contenedores del proyecto (como los de Laravel Sail o MySQL) desde el propio IDE, así como explorar fácilmente nuestra base de datos sin necesidad de recurrir a herramientas externas como DBeaver o phpMyAdmin.

En resumen, el uso de dos editores distintos ha sido una decisión consciente que nos ha permitido **diferenciar claramente las responsabilidades de cada entorno y centrarnos en los aspectos específicos de frontend y backend**.



2.1.4 Bases de datos



Para la gestión de los datos de nuestra aplicación hemos optado por utilizar **MySQL** como **sistema gestor de bases de datos relacional**.

La principal razón de esta elección es que ya estábamos familiarizados con MySQL desde cursos anteriores, con el que hemos hecho la **creación de las tablas, relaciones y consultas**. Además, durante este curso hemos utilizado en la asignatura de DWS este mismo gestor de BD, por lo que se ha convertido en nuestra opción más cómoda para desarrollar este proyecto.

Además, para construir la base de datos de forma estructurada y mantenible, **hemos creado todas las tablas** a través del sistema de **migraciones de Laravel**, basándonos directamente en nuestro esquema entidad-relación previamente diseñado.

2.1.5 Control de versiones

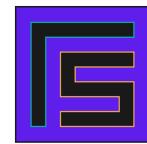


Durante el desarrollo del proyecto hemos utilizado **Git** como sistema de control de versiones y **GitHub** como plataforma para alojar nuestro repositorio de código.

Con ellos hemos podido ir guardando el progreso del proyecto conforme íbamos haciendo funcionalidades, y en caso de que tocáremos algo del proyecto y dejará de funcionar, podíamos **volver a una versión anterior** para que todo volviera a funcionar como antes.

Hemos utilizado principalmente la rama de **develop** para integrar los avances que íbamos haciendo en otras ramas, y una vez listo el proyecto ya lo hemos subido todo a **master** como versión “final” y estable.

En resumen, el control de versiones ha sido una parte fundamental del proyecto, tanto para organizar el trabajo como para mantener un registro claro y seguro del desarrollo.



2.1.6 Contenedores

Para el entorno de desarrollo hemos utilizado Docker ya que con él podemos levantar fácilmente todos los servicios necesarios sin tener que configurarlos manualmente en el sistema de cualquier dispositivo.

Hemos trabajado principalmente con varios contenedores, en concreto con el de **Laravel Sail**, que es un entorno de desarrollo ligero para Laravel. Con él hemos podido utilizar estos dos contenedores esenciales:

Servidor de desarrollo (Laravel)

Este contenedor ejecuta nuestra API en un entorno PHP preparado con todas las dependencias necesarias.

Base de datos MySQL

Este contenedor se conecta con Laravel y es el gestor de base de datos relacional que contiene todas las tablas y relaciones del proyecto.

La conclusión es que gracias al uso de contenedores hemos podido adaptar nuestro proyecto en nuestros ordenadores portátiles también siendo esto un gran punto a favor para poder trabajar en cualquier sitio evitando problemas de configuración.





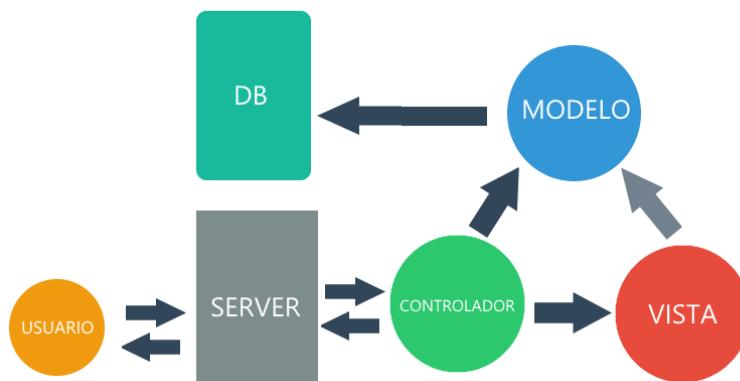
2.2 Arquitecturas escogidas

Para el desarrollo de nuestra aplicación hemos optado por una arquitectura basada en el patrón **Modelo-Vista-Controlador (MVC)**, que es el modelo que hemos aprendido y trabajado durante el curso, especialmente con Laravel y porque **se adapta perfectamente a las necesidades de nuestro proyecto**.

Esta arquitectura nos ha permitido **separar claramente la lógica del negocio (controladores y modelos), la gestión de datos (base de datos y migraciones) y la presentación**.

Como hemos comentado hemos separado completamente el frontend y el backend. El **frontend** está **desarrollado con React**, donde realizamos llamadas con los endpoints definidos en el **backend de Laravel**, que actúa como una API RESTful. Con esto conseguimos tener flexibilidad y escalabilidad si en un futuro quisiéramos continuar o seguir desarrollando la app.

También hemos hecho uso de una **API externa pública** para la importación de los ejercicios que complementan las rutinas en la aplicación. Esta integración se hace desde el backend y podemos crear una tabla de ejercicios muy extensa fácilmente.





2.3 Diseño de las interfaces entre los elementos.

Esta es la comunicación entre las distintas partes de nuestra arquitectura:

Frontend ↔ Backend

La parte del cliente, desarrollada con React, se comunica con el backend de Laravel a través de **llamadas HTTP**. Estas llamadas se realizan **usando fetch con await**, y la **URL base de la API** está definida en un archivo de entorno (**.env**), lo que nos permite modificarla fácilmente según el entorno (desarrollo y producción).

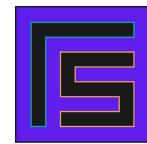
Toda **la información** se intercambia **en formato JSON**, tanto en las peticiones (body con application/json) como en las respuestas que envía el backend.

Backend ↔ Base de datos

Cuando una petición llega al backend (por ejemplo, para crear un nuevo usuario), Laravel primero utiliza una clase de validación “FormRequest” que comprueba que los datos enviados cumplen ciertos requisitos (por ejemplo, que la contraseña tenga la longitud mínima). **Si los datos no son válidos, se devuelve un error en formato JSON**. **Si son correctos**, se procesan (por ejemplo, se encripta la contraseña) y **se pasan a un modelo Eloquent**, que representa la tabla correspondiente en la base de datos.

Gracias a Eloquent, se pueden usar métodos como `create()` para insertar un nuevo registro **sin necesidad de escribir consultas SQL manuales**. Al obtener información con un GET, el controlador puede llamar a métodos como `Usuario::find($id)` y Eloquent se encarga de generar la consulta SQL necesaria (SELECT).

Laravel luego **envía** estos **datos al frontend en formato JSON**. Este flujo permite mantener el código organizado, seguro y limpio, con una clara separación entre lógica de negocio y acceso a datos.



Backend ↔ Servicios externos

Además de la base de datos interna, el backend se comunica también con una **API externa** pública: ExerciseDB. Esta API nos permite obtener datos sobre ejercicios deportivos. La integración **se realiza mediante llamadas HTTP con un token de autenticación incluido en los headers**, lo que nos permite acceder de forma segura a los recursos que ofrece la plataforma externa y complementar la información disponible en nuestra propia base de datos.

2.4 Diseño de la interfaz de usuario

2.4.1 Tipografía

Vamos a usar una tipografía moderna, limpia, y geométrica, conocida como Roboto, unos ejemplos en mayúsculas y minúsculas del tipo de la fuente:

Los Misterios Fascinantes de los Gatos Domésticos

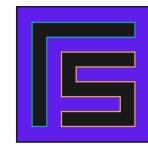
Los gatos han sido compañeros en la vida humana desde tiempos inmemoriales, pero su naturaleza independiente y misteriosa sigue intrigando a muchas personas. Estos animales, conocidos por su elegancia y agilidad, son criaturas llenas de curiosidades.

DESDE SU CAPACIDAD PARA MOVERSE SIGILOSAMENTE HASTA SUS INCREÍBLES HABILIDADES PARA SALTAR, LOS GATOS CONTINÚAN SIENDO ADORADOS TANTO POR SU BELLEZA COMO POR SU COMPORTAMIENTO INTRIGANTE. ADEMÁS, A PESAR DE SU FAMA DE SER DISTANTES, MUCHOS GATOS PUEDEN SER INCREÍBLEMENTE AFECTUOSOS Y LEALES.

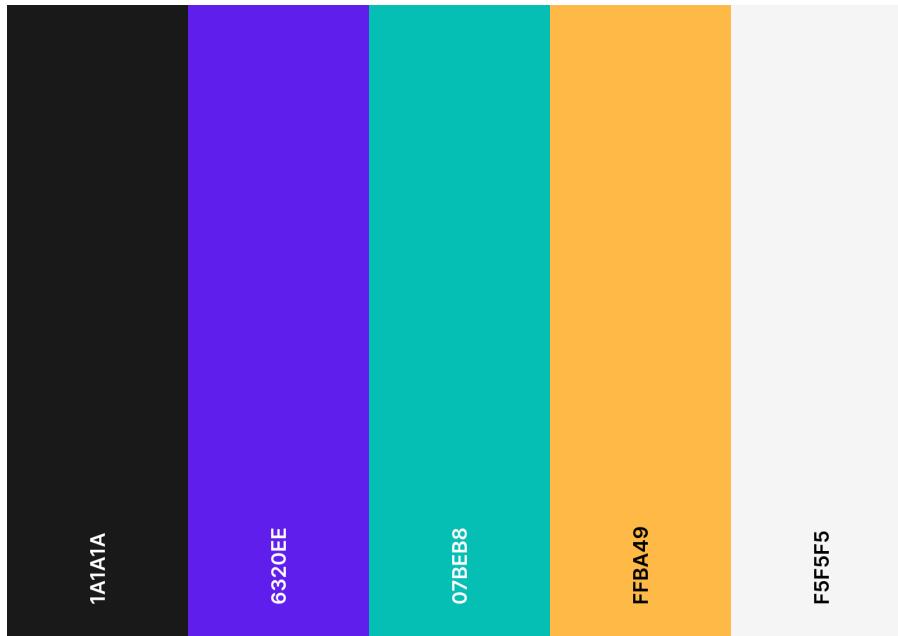
En cuanto los tamaños, usaremos para el título principal **h1** (misma fuente), un tamaño de **2.5 em** y la fuente estará en negrita.

Para los títulos **h2**, seguiremos respetando la fuente de Roboto, tienen un tamaño de **2 em** y la fuente estará también en negrita.

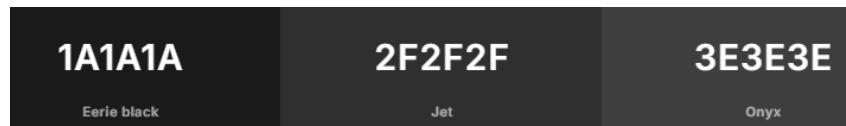
Además hemos aumentado el tamaño de la fuente en secciones para resaltar su contenido, o bien subrayando dicho texto.



2.4.2 Colores



Para la **cabezera y el footer**, utilizamos el color **negro** principalmente. Si el tema es oscuro hemos puesto diferentes tonalidades de negro/grises para diferenciar bien estas secciones frente al fondo, aquí el código de estos colores:



El **color blanco humo** prácticamente se utilizará para el **color del texto**, y en el tema claro posiblemente sea el color principal del fondo de la página dando lugar a que la letra pase a ser negra. Al igual que en con el color negro hemos incluido otras tonalidades claras, aquí el código de colores:



El **color púrpura** profundo se utiliza para **destacar elementos** en los que vayas pasando por encima, y también para destacar en que sitio estas en el menú o **para resaltar** algunas secciones o contenedores como color de fondo.



El color amarillo dorado y azul turquesa son para los detalles como lo pueden ser los estilos de las listas, títulos, iconos y algunas palabras para llamar la atención.

2.4.3 Iconografía

En nuestra aplicación vamos a utilizar iconos para **mejorar la usabilidad y la identificación visual** de cada sección, así representamos de manera intuitiva cada funcionalidad dentro de la aplicación. Los iconos han sido implementados utilizando Font Awesome.

Iconos en el Navbar

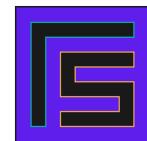
- Tema →
- Perfil del usuario →

Iconos en el contenido de la App

- Rutinas y entrenamiento →
- Alojamiento y vuelos →
- Dietas y alimentación →
- Trabajos disponibles →
- Calendario de tareas →
- Otros más →

Iconos en el Footer

- Redes sociales →



2.4.4 Mockups

2.4.4.1 Bajo nivel

Versión escritorio:

The first wireframe (left) shows the main landing page with a header, a 'CUESTIONARIO INICIAL' form, and three data visualizations: a map with a highlighted area, a scatter plot, and a bar chart.

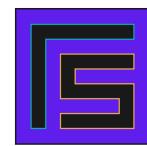
The second wireframe (right) shows the forum page with sections for 'General' and 'Comunidad', each containing posts and user interactions. A 'Footer' section is at the bottom.

Versión móvil:

The first mobile wireframe shows the main screen with a large empty box, a title, and a text block.

The second mobile wireframe shows a screen for creating routines with various category icons.

The third mobile wireframe shows a contact form with fields for name, email, and message, along with a reCAPTCHA and a 'ENVIAR MENSAJE' button.



2.4.4.2 Alto nivel

Versión escritorio:

The screenshot shows a desktop browser window with the following layout:

- Header:** Includes a yellow hourglass icon, a purple navigation bar with links to "Inicio", "¿Qué Somos?", "Servicios", "Planea tu rutina", "Foro", and "Contáctanos", and a search bar.
- Main Content Area:** A dark purple section containing:
 - Título principal** (Main Title)
 - Subheading**
 - Body text: "Body text for your whole article or post. We'll put in some lorem ipsum to show how a filled-out page might look."
 - A yellow button labeled "Cuestionario Inicial" (Initial Survey).
 - A large yellow circle with a play button icon.
- Footer:** Contains sections for "Contacto" (Contact), "Información" (Information), and "Redes sociales" (Social Media). It also includes a map showing the location of "I.E.S. Poeta Paco Mollà".
- Page Bottom:** A copyright notice: "Todos los derechos reservados. Copyright © 2024 - MoveOn".

Versión móvil:

The screenshot shows two mobile phone screens displaying the Safe Future website:

- Left Screen:** Shows the main landing page with a large yellow play button icon, the title "Título principal", a short text snippet, and a "Cuestionario Inicial" button.
- Right Screen:** Shows a navigation menu with "Inicio" selected, followed by links to "¿Qué Somos?", "Servicios", "Planea tu rutina", "Foro", and "Contacto". The "Contacto" screen is expanded, showing "Preguntas frecuentes", "Dirección" (Address), "Teléfono: 697 123 456", and email "(Petrer) - 03610 moveon@vida.com".



2.4.5 Versión funcional

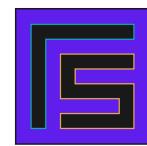
2.4.5.1 Página de inicio (Modo oscuro)



En esta versión funcional podemos ver que nos hemos adaptado casi al 100% del mockup de alto nivel, hemos incluido un borde para resaltar el video y el mapa.

También hemos modificado lo del cuestionario con un botón que lleva a los usuarios a una página donde puede descubrir las funcionalidades que puede hacer en la página web, para posteriormente redirigir al usuario para que se registre.

Hemos incluido el **botón para cambiar tambien el tema de modo oscuro a modo claro**, en el siguiente punto mostramos el cambio.



2.4.5.2 Página de inicio (Modo claro)

[Inicio](#) [¿Qué Somos?](#) [Servicios](#) [Plantea tu rutina](#) [Contacto](#)

Cambio de Vida

Introducción

Safe Future es una innovadora plataforma diseñada para brindar soluciones prácticas a personas que desean transformar sus vidas, ya sea encontrando empleo en toda Europa, trasladándose a nuevas ubicaciones o adoptando hábitos saludables. Con un enfoque centrado en la reintegración social, recompensamos a los usuarios por sus logros a través de un sistema motivador de puntos. Además, proporcionamos recursos y apoyo para facilitar el cambio hacia un futuro más prometedor. Aquí puedes ver un video que explica nuestra misión y cómo funcionamos.



[Descubre más](#)

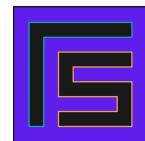
Contacto	Información	Redes sociales
Contacto	Acerca de nosotros	
WhatsApp	Preguntas frecuentes	
Teléfono: 697 123 456 (Petren) 03610 moveon@gmail.com	Política de Privacidad	
	Política de cookies	
	Sitios recomendados	



© 2025 - Creado por Miguel Ángel Grimal López y Miguel Hernández Monllor

Con el tema claro activado, se puede observar cómo todos los elementos visuales se han adaptado correctamente al nuevo esquema de colores. Los textos, botones e iconos mantienen su legibilidad y contraste, garantizando una buena experiencia de usuario también en ambientes más iluminados.

21 de 60



2.4.5.3 Página de inicio en móvil (Modo claro y oscuro)

The image shows two side-by-side screenshots of a mobile application's home screen. Both screens feature a black header bar at the top with three icons: a square with a white 'X' (Close), a stylized letter 'S' (Logo), and a user profile icon. Below the header is a vertical menu with five items: 'Inicio', '¿Qué Somos?', 'Servicios', 'Plantea tu rutina', and 'Contacto'. A 'Cambiar tema a claro' button with a sun icon is located at the bottom of the menu. The background of the left screen is dark, while the right screen has a light gray background. Both screens display a large image at the bottom showing four people (two men and two women) holding food items like burgers and salads, with a yellow arrow pointing from the text above to the image. Below the image, the text 'Cambio de Vida' and 'Introducción' is displayed in yellow, followed by a paragraph of text in white.

Cambio de Vida

Introducción

Safe Future es una innovadora plataforma diseñada para brindar soluciones prácticas a personas que desean transformar sus vidas, ya sea encontrando empleo en toda Europa, trasladándose a nuevas ubicaciones o adoptando hábitos saludables. Con un

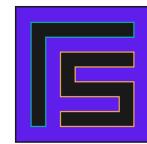
Cambio de Vida

Introducción

Safe Future es una innovadora plataforma diseñada para brindar soluciones prácticas a personas que desean transformar sus vidas, ya sea encontrando empleo en toda Europa, trasladándose a nuevas

En dispositivos móviles se ha diseñado un **menú desplegable** (hamburguesa), donde siguen estando los botones para acceder a los distintos sitios de la aplicación y el botón para cambiar de tema.

También hemos distribuido verticalmente lo que no cabía bien en horizontal para mejorar la visibilidad de las secciones.



2.4.6 Utilidades de Tailwind CSS

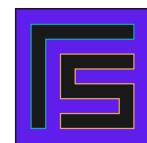
En este punto hay que destacar que gracias al uso de **Tailwind CSS** hemos podido construir la interfaz de nuestra aplicación completamente personalizada. Una de las ventajas más destacables ha sido la posibilidad de **gestionar tanto el modo claro como el modo oscuro** con gran facilidad, activándolo mediante la clase dark en el HTML.

Ejemplo en el que hemos utilizado estas propiedades:

```
<header className="min-h-24 w-full flex  
justify-around items-center border-b-2  
■ bg-white □ border-black □ dark:bg-black  
■ dark:border-white ">
```

También hemos definido **nuestros propios colores** en la configuración (**tailwind.config.ts**) adaptandolos a los de nuestra guía de estilos, para poder aplicarlos en la parte visual de nuestra aplicación, con tonos **como el púrpura, dorado y turquesa**, además de las variantes de blanco y negro que hemos comentado anteriormente para mantener el contraste y la elegancia del diseño:

```
colors: {  
  gold: '#FFBA49',  
  purple: '#6320EE',  
  purpleOp: '#6520ee70',  
  turq: '#07BEB8',  
  white: '#F5F5F5',  
  whiteOp: '#F5F5F570',  
  white2: '#DDDDDD',  
  white3: '#CECECE',  
  black: '#101010',  
  black1: '#1A1A1A',  
  black2: '#2F2F2F',  
  black3: '#3E3E3E'  
},
```



Otra característica fundamental han sido los **breakpoints personalizados** para poder adaptar partes de la aplicación en dispositivos móviles y garantizar una **experiencia responsive adaptada** a cualquier pantalla. Con '**hsm**' hemos definido que sea para móviles (max 640px) y '**sm**' para versión escritorio (min 640px).

```
screens: [
  'hsm': { 'max': '640px' },
  'sm': { 'min': '640px' },
],
```

Además, la **sistemática de espaciados** (**padding, margin**) basada en valores numéricos nos ha ayudado a mantener una **consistencia visual** en todos los componentes, facilitando el mantenimiento y haciendo que todo el diseño se vea uniforme.

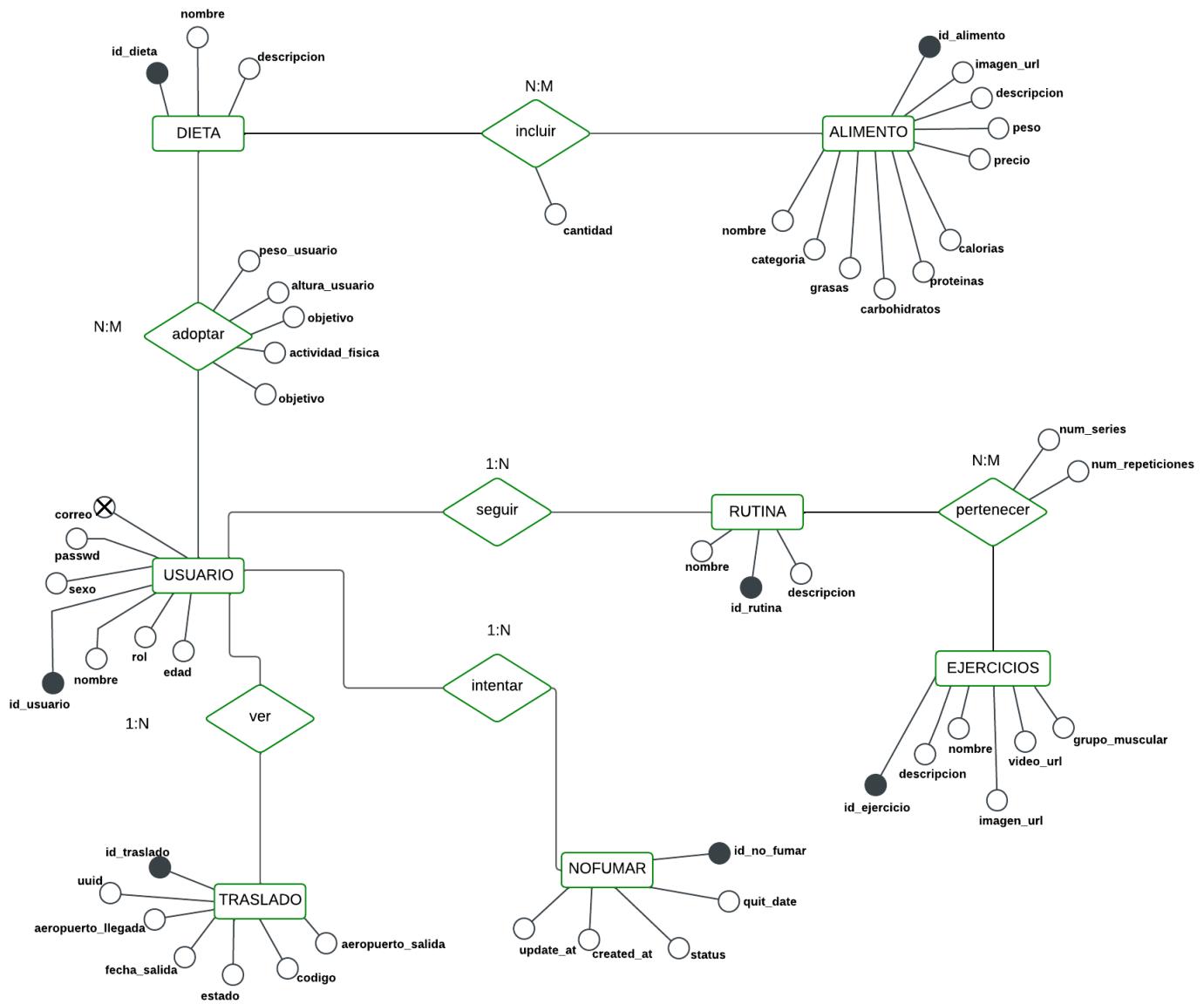
También hemos extendido algunas utilidades como escalado (scale: 1.02) y animaciones personalizadas (fadeInUp), que ha aportado **transiciones suaves y modernas** en algunas partes de la aplicación.

```
keyframes: {
  fadeInUp: {
    '0%': { opacity: '0', transform: 'translateY(1.5rem)' },
    '100%': { opacity: '1', transform: 'translateY(0)' },
  },
},
animation: {
  fadeInUp: 'fadeInUp 0.6s ease-out forwards',
},
```



3. Bases de datos

3.1 Modelo entidad-relación





3.2 Modelo relacional

TABLA	ATRIBUTOS	DEFINICIÓN	DESCRIPCIÓN
USUARIOS	id_usuario	uuid	PK
	email	string (100)	VNN
	rol	enum ['Usuario', 'Moderador', 'Administrador']	UK
	edad	integer	VNN
	sexo	enum ['hombre', 'mujer']	VNN
	contraseña	string (100)	VNN
DIETAS	id_dieta	uuid	PK
	nombre	string (100)	VNN
	descripción	string (100)	VNN
ALIMENTOS	id_alimento	id (autonumérico)	PK
	imagen_url	string (255)	-
	nombre	string (100)	VNN
	descripción	string (255)	-
	peso	decimal (5,2)	-
	precio	decimal (5,2)	-
	calorías	decimal (5,2)	-
	proteínas	decimal (5,2)	-
	carbohidratos	decimal (5,2)	-
	grasas	decimal (5,2)	-
	categoria	enum ['Proteínas', 'Carbohidratos', 'Grasas', 'Vitaminas']	VNN

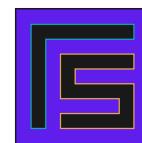


TABLA	ATRIBUTOS	DEFINICIÓN	DESCRIPCIÓN
RUTINAS	id_rutina	id (autonumérico)	PK
	nombre	varchar (100)	VNN
	descripción	varchar (255)	-
	id_usuario	uuid	FK - VNN
EJERCICIOS	id_ejercicio	id (autonumérico)	PK
	nombre	varchar (100)	VNN
	descripción	varchar (255)	-
	imagen_url	varchar (255)	-
	video_url	varchar (255)	-
	grupo_muscular	varchar (100)	-
TRASLADOS	id_traslado	id (autonumérico)	PK
	uuid	varchar (36)	VNN
	aeropuerto_salida	varchar (3)	VNN
	aeropuerto_llegada	varchar (3)	VNN
	fecha_salida	varchar (100)	VNN
	estado	varchar (100)	VNN
	codigo	varchar (100)	VNN
	id_usuario	uuid	FK -VNN
USUARIOS_DIETAS	id_usuario_dietas	id (autonumérico)	PK
	id_usuario	id	FK - VNN
	id_dieta	id	FK - VNN
	peso_usuario	decimal (5,2)	VNN
	altura_usuario	decimal (5,2)	VNN
	actividad_fisica	enum ['Sedentario', 'Ligero', 'Moderado', 'Activo',	VNN



TABLA	ATRIBUTOS	DEFINICIÓN	DESCRIPCIÓN
		'Muy-activo']	
	objetivo	enum ['Perder', 'Mantener', 'Ganar']	VNN
ALIMENTOS_DIETA	id_alimento	id	PK - FK - VNN
	id_dieta	uuid	PK - FK - VNN
	cantidad	float	VNN
RUTINA_EJERCICIOS	id_rutina_ejercicio	id (autonumérico)	PK
	id_rutina	id	FK - VNN
	id_ejercicio	id	FK - VNN
NO_FUMAR	id	id (autonumérico)	PK
	id_usuario	uuid	FK - VNN
	quit_date	dateTime	VNN





4. Demostraciones de la APP

4.1 Registro e inicio de sesión

Para esta funcionalidad tenemos en cuenta como se introducen los datos, en los campos hemos definido validaciones tanto en el frontend como en el backend para que coincidan a la hora de guardar los datos. Hemos verificado estos casos:

Para el registro

- Que el nombre tenga mínimo 3 caracteres.
- Que el email tenga un formato válido.
- Comprobación de que el email exista.
- Que la contraseña tenga mínimo 8 caracteres y tenga una letra mayúscula.
- Que la edad esté entre 18 y 64 años.
- Que se seleccione un sexo válido en el select.

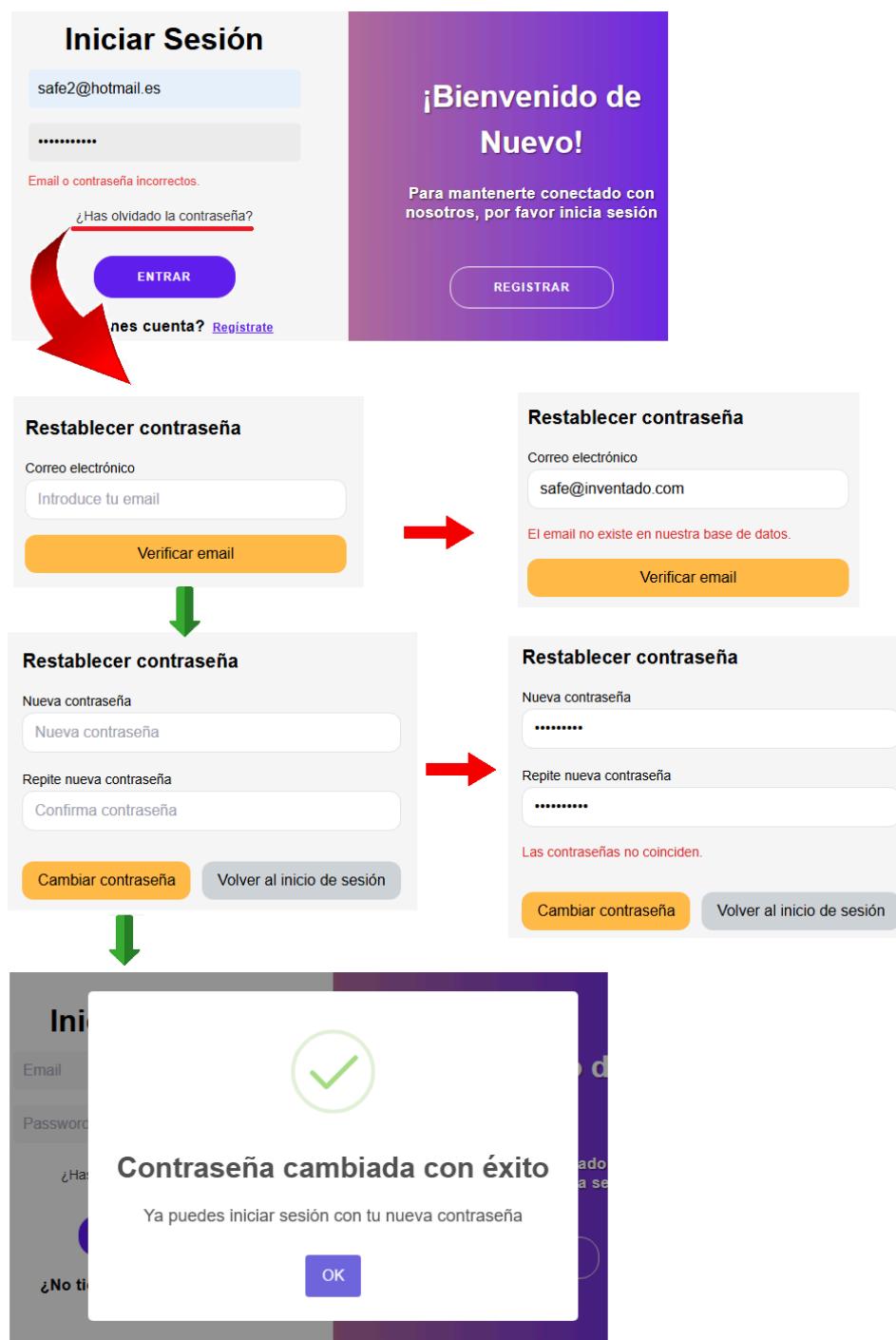
Si el registro sale exitoso, nos daremos cuenta porque veremos una confirmación en la esquina superior derecha





Para el inicio de sesión

- Que el email tenga un formato válido
- Que la contraseña sea válida y tenga 8 caracteres.
- Si alguno de los datos, o ninguno de ellos es válido decimos email o contraseña incorrectos sin especificar que está mal para no dar pistas a nadie.
- Posibilidad de recuperar la contraseña escribiendo el email.





4.2 Perfil del usuario

En este apartado el usuario puede modificar sus datos personales, incluyendo tambien el cambio de contraseña e incluso la eliminación de su propia cuenta. En las siguientes capturas de pantalla se muestra como cambiamos el nombre personal mediante el método **PATCH** para no tener que pasar todos los datos del usuario en la petición.

Detalles de la cuenta

Cerrar sesión

Nombre personal	
Nombre	Prueba usuario

Detalles de la cuenta

El nombre debe contener más de 3 caracteres.

Nombre personal	
Nombre	<input type="text" value="Nuevo Nombre"/>
Guardar Cancelar	



Actualizado

El campo fue actualizado con éxito

Nombre personal	
Nombre	Nuevo Nombre



4.3 Perfil del administrador

El administrador tiene una sección que otros usuarios no tienen, y es la posibilidad de administrar los usuarios, pudiéndoles asignar un rol distinto en la aplicación o eliminandolos del sistema.

El administrador no aparecerá en la lista de usuarios, pero si podrá ver a los demás administradores en la lista.

Lista de usuarios registrados	
Admin Principal	Administrador
Ramón	Usuario
El Dandy de Barcelona	Usuario
Aitana	Usuario
Asunción	Usuario

Gestión del usuario:

Nombre: Aitana
Correo: aitana@gmail.com
Rol: Usuario

Seleccionar cambio de rol

Cambiar Rol

Eliminar Usuario





4.4 Dietas personalizadas

En esta funcionalidad el usuario podrá crear tantas **dietas personalizadas** como quiera, en la que les preguntaremos a la hora de crearla los datos de su altura, estatura, su actividad física y su objetivo y dependiendo de estos datos **le recomendaremos unos macros** para que pueda cumplir en su día y así lograr su objetivo.

Si el usuario no añade sus datos o la información correcta, se le informará con un mensaje en dicho paso.

The flowchart illustrates the process of creating a new diet. It begins with the 'Tus Dietas' screen, which shows a message: 'Aún no tienes ninguna dieta, crea una dieta ahora'. A green arrow points from this screen to the 'Crear una nueva dieta' screen. This second screen has fields for 'Nombre de la dieta' (set to 'Dieta Calórica') and 'Descripción de la dieta' (set to 'Muchos carbohidratos'). It includes a 'Crear dieta' button and a 'Volver a tus dietas' link. A blue arrow points from the 'Crear dieta' button to a red error box at the bottom right, which contains validation messages: 'El campo nombre no tiene valores', 'El nombre debe contener al menos 3 caracteres', 'El campo descripción no tiene valores', and 'La descripción debe tener mínimo 10 caracteres'.

The flowchart shows the steps to set diet goals. It consists of three screens: 'Primer paso', 'Segundo paso', and 'Último paso'. The 'Primer paso' screen has four circular options labeled A, B, C, and D. The 'Segundo paso' screen also has four circular options labeled A, B, C, and D. The 'Último paso' screen shows a dropdown menu with 'Perder peso' selected. It includes a 'Anterior' button and a 'Terminar' button. A red arrow points from the 'Último paso' screen back to the 'Primer paso' screen.



El usuario podrá una vez creada la dieta podrá elegir los alimentos que mas le gusten, pudiendolos filtrar por categoria, e incluso por sus carbohidratos, grasas, calorías y proteínas para poder ajustarlos a los macros que le hemos recomendado en la tabla.

The diagram illustrates the user flow through the Safe Future app's meal selection and tracking features:

- Dieta para perder peso**: The user has created a diet named "Perder peso" with the description "Ligera para el dia a dia".
- Filtros para encontrar tus alimentos**: The user is using filters to search for foods based on various nutritional parameters like Carbohydrates, Proteins, Calories, and Fats.
- Selección de alimentos**: The user is selecting specific foods for their diet:
 - Pechuga de pollo: 1.5 € / 0.2 kg, Calorías: 110, Hidratos: 0, Grasas: 1, Proteinas: 23. Action: Seleccionar.
 - Huevo: 0.25 € / 0.05 kg, Calorías: 75, Hidratos: 0.5, Grasas: 5, Proteinas: 6. Action: Seleccionar.
 - Tofu: 1.2 € / 0.15 kg, Calorías: 130, Hidratos: 2, Grasas: 7, Proteinas: 12. Action: Seleccionar.
 - Lentejas cocidas: 0.5 € / 0.2 kg, Calorías: 116, Hidratos: 20, Grasas: 0.4, Proteinas: 9. Action: Seleccionar.
 - Salmón: 2 € / 0.15 kg, Calorías: 280, Hidratos: 0, Grasas: 18, Proteinas: 22. Action: Seleccionar.A green arrow points to the "Seleccionar" button for the Salmon entry.
- Macros Acumulados**: The user is viewing the accumulated macros for their selected items:

Objetivos de tu dieta		Macros actuales
Proteinas	150 g	23 g •
Carbohidratos	0 g	0 g •
Grasas	9.62 g	1.00 g •
Calorías	346 kcal	110 kcal •
- Alimentos seleccionados**: A summary table of the selected items:

Alimento	Cantidad	Acciones
Pechuga de pollo	1	• (green), ○ (yellow), □ (red)

At the bottom of the interface are three buttons: "Volver a tus dietas" (Return to your diets), "Guardar cambios" (Save changes), and "Ver los detalles de la dieta" (View diet details).



Una vez seleccionados los alimentos podrá verlos en los detalles de su dieta para ver más fácilmente los alimentos que puede consumir al día y desde los detalles si quiere puede volver para ver o crear otras dietas y editar la misma.

Alimentos que puedes consumir en el dia

[Editar dieta](#) [Volver a tus dietas](#)

Pechuga de pollo Cantidad: 1 Descripción: Fuente de proteína magra	Precio aproximado: 1.5€ Peso por unidad: 0.2 Kg
Yogur natural entero Cantidad: 1 Descripción: Lácteo con grasas beneficiosas	Precio aproximado: 0.7€ Peso por unidad: 0.15 Kg
Nueces Cantidad: 1 Descripción: Ricas en omega-3 vegetales	Precio aproximado: 0.7€ Peso por unidad: 0.05 Kg
Lechuga romana Cantidad: 1 Descripción: Rica en folato y fibra	Precio aproximado: 0.2€ Peso por unidad: 0.15 Kg

En caso de que el usuario no haya seleccionado nada y quiera ver los que puede consumir en el dia, le diremos que no tiene nada de momento.

Alimentos que puedes consumir en el dia

[Editar dieta](#) [Volver a tus dietas](#)

No has añadido ningún alimento a tu dieta



4.5 Rutinas y ejercicios

Para los ejercicios se ha creado una especie de **mini-app** en la que se el usuario puede elegir sus propios ejercicios favoritos y ponerse las series y repeticiones que considere conveniente.

Una vez seleccionados todos los ejercicios que le gusten, el usuario creará la rutina de ejercicios que tendrá disponible en la sección de “Mis entrenamientos”.

Aquí cuando el usuario quiere crear un entrenamiento tiene a su disposición la lista completa de **los ejercicios** y conforme se van seleccionando **se van incluyendo en la lista de la derecha**.

The screenshot shows a mobile application interface for creating a workout routine. On the left, there's a sidebar with options: 'Crear Entrenamiento' (Create Training), 'Mis Ejercicios' (My Exercises) which is selected and highlighted in blue, 'Mis Entrenamientos' (My Trainings), 'Plantillas' (Templates), and 'Entrenador Personal "beta"' (Personal Trainer "beta"). The main area has fields for 'Nombre de la Rutina:' (Routine Name) set to 'Para abdominales' and 'Descripción:' (Description) also set to 'Para abdominales'. Below this, a section titled 'Elige Ejercicios:' (Choose Exercises:) shows a search bar 'Busca un ejercicio...' (Search for an exercise...) and a list of exercises with icons: 'cable reverse crunch', 'cable standing crunch (with rope attachment)', 'band alternating biceps curl', 'band alternating v-up', 'band assisted pull-up', and 'band assisted wheel rollout'. To the right, a detailed view of the 'cable reverse crunch' exercise is shown in a purple box. It includes the exercise name, group ('Nombre: cable reverse crunch Grupo Muscular: abs'), an illustration of a person performing the exercise, and buttons for 'Series' (3) and 'Repeticiones' (10). A green arrow points from the list on the left towards this detailed view.



Una vez añadidos los ejercicios, el usuario puede ver toda la lista con sus rutinas de ejercicios personalizados.

En cada una de las rutinas puede:

- Ver todos los ejercicios detallados con sus series y repeticiones.
- Personalizar la propia rutina pudiendo modificar el nombre y la descripción.
- Personalizar los propios ejercicios añadidos anteriormente y añadir nuevos a esta rutina en el buscador, donde están toda la lista de ejercicios.

The image displays two screenshots of the Safe Future mobile application interface. The top screenshot shows the 'Entrenamientos' (Workouts) screen. On the left is a sidebar with navigation options: 'Crear Entrenamiento', 'Mis Ejercicios', 'Mis Entrenamientos' (which is highlighted in purple), 'Plantillas', and 'Entrenador Personal "beta"'. The main area is titled 'Entrenamientos' and contains a search bar labeled 'Filtrar por nombre' and a '+' button. Below the search bar is a list of routines. The first routine in the list is 'Para abdominales' with the description 'Para abdominales'. To the right of this routine are three icons: a trash can, a pencil, and a right-pointing arrow. The second routine listed is 'Para fuerza' with the description 'Brazitos fuertes'. To its right are similar icons. A large blue arrow points from the bottom-left towards this list. The bottom screenshot shows the 'Editar Rutina' (Edit Routine) screen. It has a similar sidebar on the left. The main area is titled 'Editar Rutina' and shows a form with 'Nombre:' set to 'Para abdominales' and 'Descripción:' set to 'Para abdominales'. Below this is a search bar with the placeholder 'Buscar Ejercicios:' and a text input field containing 'Busca un ejercicio...'. A large blue arrow points from the top-right towards this screen. To the right of the edit screen is a third screenshot titled 'Para abdominales' which lists two exercises: '45° side bend' (Series: 3, Repeticiones: 10) and '3/4 sit-up' (Series: 3, Repeticiones: 12). Each exercise card includes a small image of a person performing the movement and the text 'EXERCISES'.



Luego en la sección de “Plantillas” estarán **rutinas que hayan podido crear otros usuarios** y lo mejor de todo es que **puedes guardarte estas rutinas** de otras personas en las tuyas propias para que no las pierdas de vista.

The screenshot shows the Safe Future app's interface. On the left is a sidebar with options: Crear Entrenamiento, Mis Ejercicios, Mis Entrenamientos, **Plantillas** (which is selected and highlighted in purple), and Entrenador Personal "beta". The main area is titled "Plantillas entrenamientos" and lists three routines:

- Nombre: Para abdominales
Descripción: Para abdominales
- Nombre: Para fuerza
Descripción: Brazitos fuertes
- Nombre: Conviértete en HULK
Descripción: Esta rutina te pondrá a 100

A blue curved arrow points from the bottom right of the sidebar towards the "Plantillas" section. Below this, a modal window has a large question mark icon and the text: "¿Aregar rutina a tus entrenamientos?". It also contains the message: "Se copiará la rutina 'Conviértete en HULK' a tus rutinas." and the question "¿Continuar?". At the bottom are two buttons: "Sí, copiar" (in purple) and "Cancelar". A green curved arrow points from the bottom left of the sidebar towards the "Mis Entrenamientos" section. This section shows the copied routines integrated into the user's own training plan:

- Mis Entrenamientos
- Plantillas (highlighted in purple)
- Entrenador Personal "beta"

The copied routines are listed here:

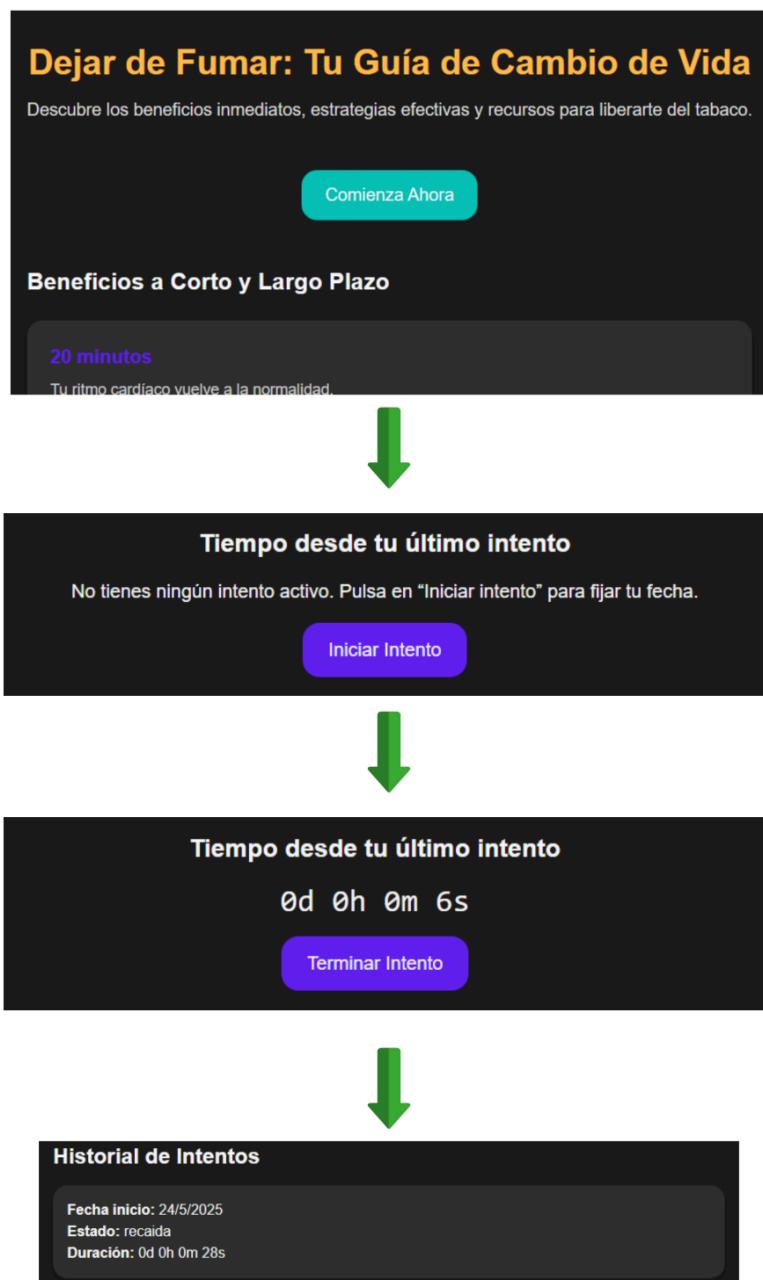
- Nombre: Para abdominales
Descripción: Para abdominales
- Nombre: Para fuerza
Descripción: Brazitos fuertes
- Nombre: Conviértete en HULK
Descripción: Esta rutina te pondrá a 100



4.6 Dejar de fumar

Esta funcionalidad se ha centrado en dar información sobre lo malo que es fumar y como coger buenos hábitos y para ello proponemos de nuestra parte **un contador que se iniciará y llevará el conteo del tiempo desde que se inicio**, así, cuando decidan dejarse de fumar, de esta manera se motivarán viendo como van pasando los días.

Si el usuario decide **terminar el intento** se quedará registrado su intento para ver las veces que ha recaído y que se de cuenta de las veces que lo ha intentado.





4.7 Calendario (Beta)

La funcionalidad de este calendario por el momento es que el usuario pueda **guardar sus eventos el día que seleccione** mediante una ventana modal que le permite establecer el nombre de dicho evento, reunión, objetivo, lo que sea.

Si dicho evento se desea eliminar por cualquier motivo o porque ya ha ocurrido, basta con hacer clic encima del evento y darle a “**Eliminar**”.

Aquí tenemos el calendario de eventos:

May 2025						
Anterior		Mes actual	Siguiente			
Mon	Tue	Wed	Thu	Fri	Sat	Sun
28		29	30	01	02	03
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	01

Si por ejemplo se desea añadir un evento el día 26 de mayo se le da clic encima y si lo queremos eliminar, se le da clic en el evento generado:

The diagram illustrates the interaction with the calendar. A blue arrow points from the 'Crear nuevo evento' (Create new event) button on the left to a modal window on the right. Inside the modal, another blue arrow points from the event title 'Entregar primera versión del proy...' back to the same button, forming a loop. Below the modal, a circular icon with an 'i' inside represents information. At the bottom, another blue arrow points from the 'Evento: Entregar primera versión del proyecto' (Event: Deliver first version of the project) text back to the 'Crear nuevo evento' button, further emphasizing the creation process.

Crear nuevo evento

¿Cómo quieres llamar a este evento?

Entregar primera versión del proyecto

Guardar Cancelar

26

Entregar primera versión del proy...

i

Evento: Entregar primera versión del proyecto

Inicio: Mon May 26 2025 00:00:00 GMT+0200 (hora de verano de Europa central)

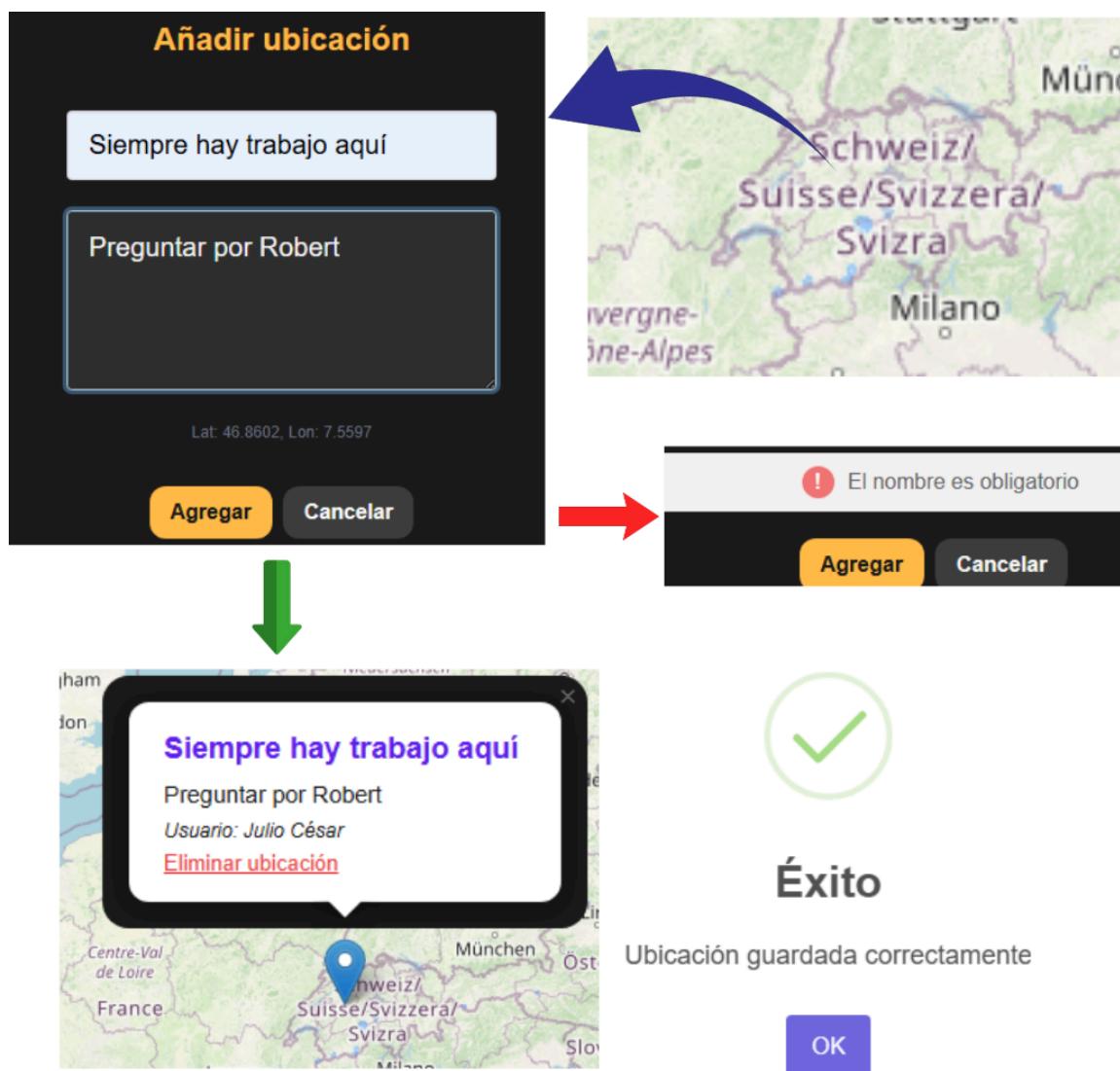
Eliminar Cancelar



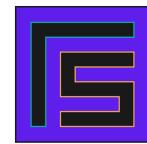
4.8 Mapamundi de ubicaciones destacadas

En esta funcionalidad el usuario va a poder ver las ubicaciones destacadas o de lugares interesantes donde otros usuarios han publicado información en un lugar del mapa relevante. El usuario actual también puede participar en poner en algún lugar del mapa algún dato importante para ayudar a los demás.

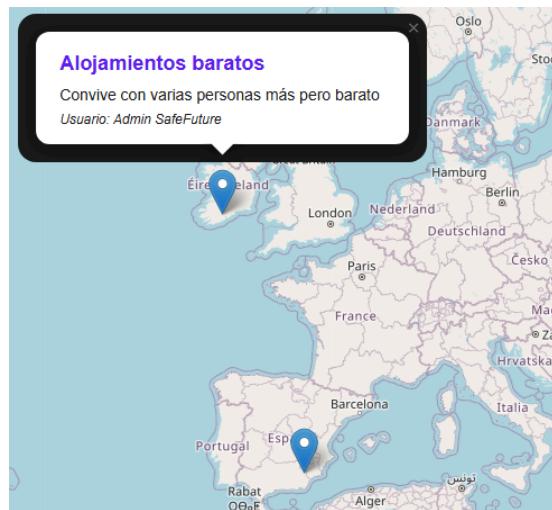
Vamos hacer la prueba de **añadir una ubicación** nosotros en Suiza:



En el caso de no introducir un nombre se le avisa al usuario, y una vez creado, el usuario puede eliminar la ubicación cuando quiera.



Un ejemplo de **otro usuario que ha publicado algo** en Irlanda:

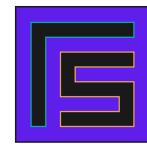


4.9 Entrenador personal AI

En este **chat** se le va a poder pedir al **entrenador personal cualquier rutina de ejercicios que quieras hacer** por si no tienes mucha idea, en este ejemplo le hemos pedido que nos haga una rutina de cuerpo completo y tras unos segundos el entrenador personal cogerá el concepto de tu rutina y se pondrá manos a la obra.

The interface shows a sidebar with navigation options: 'Crear Entrenamiento', 'Mis Ejercicios', 'Mis Entrenamientos', 'Plantillas', and 'Entrenador Personal "beta"'. The main area has a dark background with white text for messages. The conversation goes as follows:

- User message: 'Hola, soy tu asistente virtual para ayudarte a crear tus rutinas personalizadas.'
- AI response: 'Quiero una rutina completa para entrenar el cuerpo entero'
- AI message: 'Rutina creada correctamente. ¡Éxito en tu entrenamiento!'
- User message: '¡Genial! Me alegra que estés contento con la rutina. Si tienes cualquier duda o necesitas hacer algún ajuste en los ejercicios, aquí estaré para ayudarte. ¿Hay algo más en lo que te pueda asistir?'
- AI response: 'Perfecto ¡Gracias!'



Además el **entrenador** esta creado para **ayudar solamente con el tema de rutinas y ejercicios**, si te pones a hablarle de otros temas, enseguida te lo comunicará.

The screenshot shows a dark-themed chat application. On the left, there's a sidebar with navigation options: 'Crear Entrenamiento', 'Mis Ejercicios', 'Mis Entrenamientos' (which is highlighted in purple), 'Plantillas', and 'Entrenador Personal "beta"' (which is also highlighted in purple). The main area is a conversation between the user and the AI. The user's messages are in blue boxes, and the AI's responses are in white boxes with a black border. The AI's first message is: 'Hola, soy tu asistente virtual para ayudarte a crear tus rutinas personalizadas.' The user replies with 'hola'. The AI's second message is: '¡Hola! ¿Cómo te va? Cuéntame, ¿en qué puedo ayudarte hoy? ¿Buscas consejos sobre ejercicios, nutrición o alguna pregunta de entrenamiento? Estoy aquí para ayudarte a alcanzar tus metas.' The user replies with 'me puedes dar informacion sobre emigrar a otro pais'. The AI's third message is: 'Esa consulta está fuera de mi área de especialización. Me enfoco en entrenamiento y ejercicios. ¿Quieres ayuda con alguna duda específica de deporte o rutina?' At the bottom, there's a text input field with 'Escribe tu mensaje...' and a blue 'Enviar' button.

Una vez el **entrenador personal** lo tenga todo listo nos habrá añadido a nuestros **entrenamientos la rutina que le hemos especificado**:

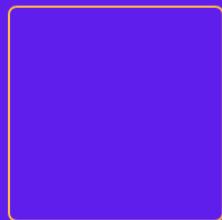
The screenshot shows the 'Entrenamientos' (Workouts) section of the application. The sidebar on the left has the same navigation options as the previous screenshot, with 'Mis Entrenamientos' highlighted in purple. The main area displays a list of routines. One routine is shown in detail: 'Nombre: Rutina Completa (IA)' and 'Descripción: Rutina completa para entrenar el cuerpo entero'. To the right of this detail box are three icons: a trash can, a pencil, and a right-pointing arrow. The overall interface is dark-themed.

Si vemos los **detalles de la rutina**, podemos ver los **ejercicios que ha creado el entrenador personal** en base a lo que le hemos pedido (Las imágenes funcionan con un token de la API y en esta captura no estaba actualizado).



Rutina Completa (IA)

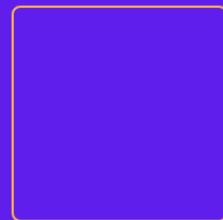
Nombre: wide grip rear pull-up
Grupo Muscular:



Series: 3

Repeticiones: 12

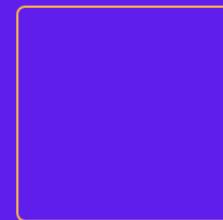
Nombre: suspended push-up
Grupo Muscular:



Series: 3

Repeticiones: 15

Nombre: barbell full zercher squat
Grupo Muscular:



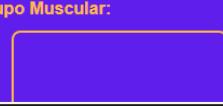
Series: 4

Repeticiones: 10

Nombre: cable one arm lateral raise
Grupo Muscular:



Nombre: dumbbell lying alternate extension
Grupo Muscular:



Nombre: smith deadlift
Grupo Muscular:

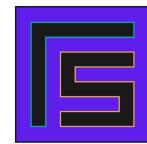


Volver

4.10 Asistente virtual Ramón AI

Este asistente virtual estará a disposición de todos los usuarios que se registren en nuestra aplicación web, y se trata de nada mas ni nada menos que Ramón AI, un chatbot a la API de n8n que esta dispuesto a realizar cualquier cosa (dentro de los límites) que le pidamos.





Unos claros ejemplos:

- Aquí pidiéndole información para emigrar a otros países.

Hola, soy Ramon tu asistente personal, ahora mismo solo te puedo ayudar con información sobre emigrar a otro país.

Quiero información sobre lugares que no sean peligrosos

He encontrado información sobre lugares que son considerados seguros y con baja peligrosidad:

1. Noruega: Es un país muy seguro con índices de criminalidad bajos y una sociedad pacífica. Además, ofrece una alta calidad de vida, buen sistema de salud y educación, y un entorno ideal para proyectos de turismo, naturaleza, y desarrollo humano. Es un destino excelente para quienes buscan seguridad y bienestar.

2. Estados Unidos: Aunque en general tiene una variabilidad en la seguridad dependiendo de la zona, muchas ciudades medianas y zonas suburbanas son seguras. El país cuenta con un sistema de seguridad profesional aunque los niveles de criminalidad pueden ser...

Escribe tu mensaje... Enviar

- Y aquí pidiéndole hasta una receta para hacer croquetas caseras.

Hola, soy Ramón tu asistente personal, ¿En qué puedo ayudarte?

Quiero hacer croquetas caseras

¡Qué deliciosa elección! Para ayudarte mejor, ¿podrías decirme si quieres una receta clásica de croquetas o tienes algún ingrediente especial en mente?

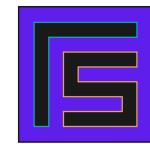
Croquetas caseras estaría bien

¡Perfecto! Aquí tienes una receta básica y deliciosa de croquetas caseras. La podrás adaptar según tus preferencias.

Ingredientes:

- 200 g de jamón serrano, pollo, o el ingrediente que prefieras (opcional)
- 50 g de mantequilla
- 3 cucharadas de harina

Escribe tu mensaje... Enviar



4.11 Pruebas de funcionamiento con usuarios en Postman

POST | Crear un usuario

The screenshot shows a POST request to `http://localhost:8089/api/usuarios`. The Body tab is selected, containing the following JSON payload:

```
1 {  
2   "nombre": "Juan Pérez",  
3   "correo": "juanperez@example.com",  
4   "contrasena": "JP123456",  
5   "edad": 30,  
6   "sexo": "hombre",  
7   "rol": "Usuario"  
8 }  
9
```

The response status is 201 Created, with a duration of 13.70 s and a size of 413 B. The response body is:

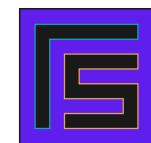
```
{ } JSON ▾ ▷ Preview ⚡ Visualize ▾  
1 {  
2   "message": "Usuario creado",  
3   "usuario": {  
4     "id_usuario": 61,  
5     "nombre": "Juan Pérez",  
6     "correo": "juanperez@example.com",  
7     "contrasena": "$2y$12$fEH/lZHv5j1qQ0ZdI4QbUeBomohVheRa/hQW77VJ.Jaf2283/w8D2",  
8     "edad": 30,  
9     "sexo": "hombre",  
10    "rol": "Usuario"  
11  },  
12  "status": 200  
13 }
```

GET | Obtener un usuario

The screenshot shows a GET request to `http://localhost:8089/api/usuarios/61`. The Body tab is selected, containing the following JSON response:

```
1 {  
2   "usuario": {  
3     "id_usuario": 61,  
4     "nombre": "Juan Pérez",  
5     "correo": "juanperez@example.com",  
6     "contrasena": "$2y$12$fEH/lZHv5j1qQ0ZdI4QbUeBomohVheRa/hQW77VJ.Jaf2283/w8D2",  
7     "edad": 30,  
8     "sexo": "hombre",  
9     "rol": "Usuario"  
10   },  
11   "status": 200  
12 }
```

The response status is 200 OK, with a duration of 9.06 s and a size of 443 B.



PUT | Editar un usuario

PUT http://localhost:8089/api/usuarios/61

Params Authorization Headers (8) Body Scripts Settings

Body raw binary GraphQL JSON

```
1 {  
2   "nombre": "Juan Pérez",  
3   "correo": "juanperez@example.com",  
4   "contrasena": "JP123456",  
5   "edad": 60,  
6   "sexo": "hombre",  
7   "rol": "Usuario"  
8 }  
o
```

Body Cookies Headers (7) Test Results 200 OK • 7.70 s • 4

{ } JSON ▾ Preview Visualize

```
1 {  
2   "message": "Usuario actualizado",  
3 }
```

PATCH | Editar un campo del usuario

PATCH https://sfmoveon.es/api/usuarios/905ad258-91a8-46aa-8cc4-a3068f8fb450

Params Authorization Headers (9) Body Scripts Settings

Body raw binary GraphQL JSON

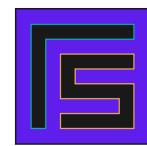
```
1 {  
2   "rol": "Administrador"  
3 }  
4  
5 // "rol": "Usuario" - "rol": "Moderador"
```

Body Cookies Headers (8) Test Results 200 OK

{ } JSON ▾ Preview Visualize

```
1 {  
2   "message": "Usuario actualizado parcialmente",  
3   "usuario": {  
4     "id_usuario": "905ad258-91a8-46aa-8cc4-a3068f8fb450",  
5     "nombre": "DIeta MALA",  
6     "correo": "hachexd@gmail.com",  
7     "edad": 32,  
8     "sexo": "hombre",  
9     "rol": "Administrador"  
10   },  
11   "status": 200  
12 }
```

✓ 200 OK
Request successful. The server responded as required



DELETE | Eliminar un usuario

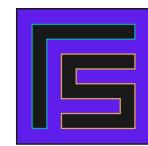
```
DELETE http://localhost:8089/api/usuarios/61

Params Authorization Headers (6) Body Scripts Settings
Body Cookies Headers (7) Test Results 200 OK
[] JSON ▾ Preview Visualize ▾
1 {
2   "message": "Usuario eliminado",
3   "status": 200
4 }
```

5. Subida a AWS

Como bien sabemos tenemos que subir nuestra aplicación a AWS para poder desplegarla, y que sea visible en la web para que todo el mundo pueda acceder a ella.

Para ello hemos creado una instancia de EC2 en AWS, con un servidor ubuntu.



Luego en la página de configuración añadimos los protocolos SSH, HTTP y HTTPS.

Tipo	Protocolo	Rango de puertos
SSH	TCP	[22]
HTTPS	TCP	[443]
HTTP	TCP	[80]

Añadir regla

Luego antes de lanzar la instancia tenemos que generar el par de claves para poder comunicarnos con el servidor desde nuestros ordenadores, procedemos a definir el par de claves.

Crear un nuevo par de claves

Nombre del par de claves

laravel8

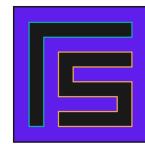
Descargar par de claves

Una vez hecho esto ya podremos darle al botón de lanzar instancia y la tendremos en ejecución.

Instancias (1/1)		Información	Última actualización	Conectar	Estado de la instancia	Acciones	Lanzar instancias
<input type="text"/>	<input type="text"/>	Buscar Instancia por atributo o etiqueta (case-sensitive)	Hace less than a minute	(C)	Todos los ...	< 1 >	ξ
<input checked="" type="checkbox"/> Name	<input type="checkbox"/> ID de la instancia	<input type="checkbox"/> Estado de la i...	<input type="checkbox"/> Tipo de inst...	<input type="checkbox"/> Comprobación de	<input type="checkbox"/> Estado de la al:	<input type="checkbox"/> Zona de dispon...	

API SF i-0dcda8b134856c852 En ejecución t3.micro Inicializando Ver alarmas eu-north-1a

Una vez hecho podemos conectarnos al servidor siguiendo las instrucciones.



AWS Buscar [Alt+S] Europa (Estocolmo) mangel

EC2 > Instancias > i-0dcda8b134856c852 > Conectarse a la instancia

Conectarse a la instancia [Información](#)

Conéctese a la instancia i-0dcda8b134856c852 (API SF) mediante cualquiera de estas opciones

Conexión de la instancia EC2 | Administrador de sesiones | **Cliente SSH** | Consola de serie de EC2

ID de la instancia i-0dcda8b134856c852 (API SF)

1. Abra un cliente SSH.
2. Localice el archivo de clave privada. La clave utilizada para lanzar esta instancia es Conexion.pem
3. Ejecute este comando, si es necesario, para garantizar que la clave no se pueda ver públicamente.
 chmod 400 "Conexion.pem"
4. Conéctese a la instancia mediante su DNS público:
 ec2-16-16-207-185.eu-north-1.compute.amazonaws.com

Ejemplo:
 ssh -i "Conexion.pem" ubuntu@ec2-16-16-207-185.eu-north-1.compute.amazonaws.com

Nota: En la mayoría de los casos, el nombre de usuario adivinado es correcto. Sin embargo, lea las instrucciones de uso de la AMI para comprobar si el propietario de la AMI ha cambiado el nombre de usuario predeterminado de la AMI.

[Cancelar](#)

Para conectarnos al servidor de AWS lo que hacemos es utilizar SSH proporcionandole la key (del par de claves) para poder tener el acceso.

```
PS C:\Users\hache> ssh -i "laravel8.pem" ubuntu@ec2-13-36-168-253.eu-west-3.compute.amazonaws.com
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Mon May 26 18:04:36 UTC 2025

 System load:  0.03           Processes:          122
 Usage of /:   9.3% of 28.02GB  Users logged in:    0
 Memory usage: 35%            IPv4 address for enX0: 172.31.6.9
 Swap usage:   0%

Expanded Security Maintenance for Applications is enabled.

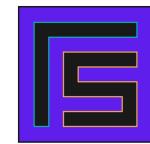
0 updates can be applied immediately.

Last login: Sat May 24 20:08:38 2025 from 185.95.59.132
ubuntu@ip-172-31-6-9:~$
```

Desde aquí hemos ya hemos instalado Apache (Con las configuraciones aprendidas durante los cursos), MySql y establecido el entorno de PHP con laravel.

Y para acabar de desplegarlo todo **hemos adquirido un dominio .es** para ahora en el certificado que vamos a instalar poder **desplegarlo en HTTPS** y así evitar problemas con los cors de laravel.

Instalamos con **sudo apt install certbot python3-certbot-apache -y** y luego ejecutamos certbot especificando el dominio de nuestra página **sfmoveon.es**



```
ubuntu@ip-172-31-6-9:~$ sudo certbot --apache -d sfmoveon.es
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for sfmoveon.es

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/sfmoveon.es/fullchain.pem
Key is saved at:      /etc/letsencrypt/live/sfmoveon.es/privkey.pem
This certificate expires on 2025-08-14.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for sfmoveon.es to /etc/apache2/sites-available/000-default-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://sfmoveon.es

-----
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                   https://eff.org/donate-le
-----
```

```
ubuntu@ip-172-31-6-9:~$
```

Ya teniendo esto instalado solamente nos hizo falta copiar **nuestro React**, primero lo comprimimos todo **haciendo build**.

```
>> npm run build
> moveon@0.0.0 build
> vite build

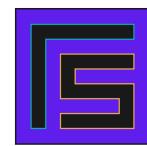
vite v6.3.4 building for production...
✓ 2332 modules transformed.
```

Y ya después mediante un **scp copiar todo el build al servidor en la carpeta public de laravel**.

```
PS C:\Users\hache> scp -i ./laravel8.pem -r C:\Users\hache\OneDrive\Desktop\2DAW\ProyectoIntermodular\FrontEnd\MoveOn\dist/* ubuntu@ec2-13-36-168-253.eu-west-3.compute.amazonaws.com:/home/ubuntu/laravel/public/
borrar-D00ke8yP.png                                         100% 4153    130.8KB/s  00:00
calendario-BoFE_yn.png                                     100% 1942KB   5.3MB/s  00:00
Card1-C1DxbvRp.webp                                      100%   21KB 725.2KB/s  00:00
Card2-CPwaW8K3.webp                                      100%   69KB   1.2MB/s  00:00
dietas-alimentos-BmTqRoag.png                            100% 2178KB 15.0MB/s  00:00
ejercicios-rutinas-D-0ThL7W.png                           100% 1592KB 16.5MB/s  00:00
food-CoWMEIIf.png                                         100%   10KB 355.0KB/s  00:00
index-BmnSFxUA.css                                       100% 105KB   1.7MB/s  00:00
index-Ne7FnbkT.js                                         100% 871KB 13.1MB/s  00:00
logo-D6mskC-1.svg                                         100%   364    12.7KB/s  00:00
no-fumar-D8LtvrCx.png                                    100% 2087KB 22.4MB/s  00:00
ver-CL2BatyA.png                                         100% 6088   220.2KB/s  00:00
index.html                                               100%   551    19.2KB/s  00:00
move-on.mp4                                              0%     0     0.0KB/s  --:-- ETAmove-o
n.mp4                                                    100% 2762KB 24.1MB/s  00:00
vite.svg                                                 100% 1497    52.2KB/s  00:00
```

Tras eso y unas configuraciones que hicimos en el **.htaccess en la carpeta public de laravel**, conseguimos hacer que las rutas de la aplicación de React funcionasen, para que no interfiriera con las de la API.

```
ubuntu@ip-172-31-6-9:~/laravel/public$ ls -a
.  .. .htaccess assets favicon.ico index.html index.php
```



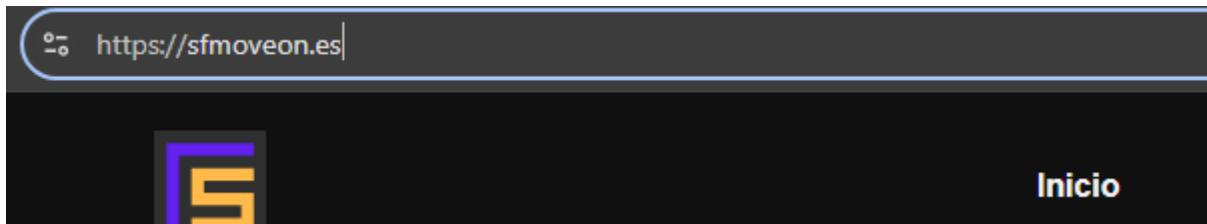
```
GNU nano 7.2 .htaccess *
<IfModule mod_rewrite.c>
  Options -MultiViews -Indexes
  RewriteEngine On

  # Si la petición es a /api, la maneja Laravel (index.php):
  RewriteCond %{REQUEST_URI} ^/api
  RewriteRule ^ index.php [L,QSA]

  # Si el fichero o carpeta existe, servirlo directamente:
  RewriteCond %{REQUEST_FILENAME} -f [OR]
  RewriteCond %{REQUEST_FILENAME} -d
  RewriteRule ^ - [L]

  # Y Para cualquier otra ruta (SPA), devolver el index.html de React:
  RewriteRule ^ index.html [L]

</IfModule>
```



6. Agentes de IA (N8N)

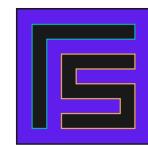
Aquí os voy a explicar cual es el **cerebro de la inteligencia artificial**, al principio como desarrolladores estamos acostumbrados a pensar que para trabajar con la IA es solo conectarse a una api de open ia y ya. Pero aquí os voy a demostrar que es más que eso.

Primer paso ya que se me dificulta por tema de tiempo usar **N8N** una plataforma de low code. Esto es muy bien ya que te ahorra horas al programar pero a la hora de la verdad es mejor usar una biblioteca de open AI o cualquiera otro para poder ejecutarlo, **n8n funciona** pero si no quieres pagar no es muy buena opción, **es buena para crear un diagrama de flujo de tu código** y luego es simplemente pasarlo a código.

Pero bueno como lo he hecho así, os voy a enseñar como crear un agente desde 0 sin conocimientos.

En primer lugar tienes que crearte una cuenta de n8n y una vez creada tienes dos opciones desplegarlo en local (pruebas sin pagar) o en la nube, en este caso **lo voy a hacer en local**.

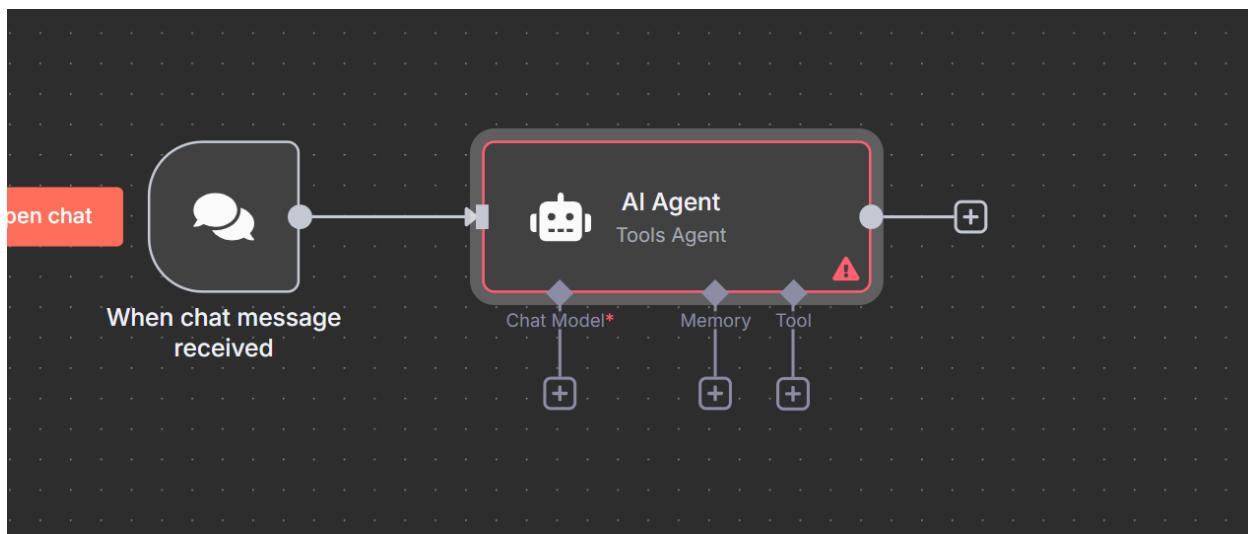
Para desplegarlo en local es fácil ya que al ser desarrolladores tenemos [node.js](#) instalado y simplemente en tu equipo tendrás que hacer `npm install n8n`, luego solo **ejecutaras en la terminal el comando “n8n”** y le das a la tecla enter y se te **abrirá el menu**, una vez hay le das a crear flujo.



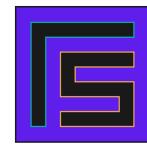
Se nos abrirá la siguiente pantalla:

The screenshot shows the n8n workflow editor interface. The main area is labeled 'My workflow'. On the left, there's a sidebar with 'Templates', 'Variables', and 'Help'. The central workspace has a large '+ Add first step...' button. On the right, a panel titled 'What triggers this workflow?' lists several options: 'Trigger manually', 'On app event', 'On a schedule', 'On webhook call', 'On form submission', 'When Executed by Another Workflow', 'On chat message', and 'Other ways...'. Each option has a brief description and a small icon.

Una vez aquí como ejecutor de código que será lo que ejecute el flujo pondremos “**chat message**”, esto es como un componente que te da un chat ya hecho, seguidamente le daremos a más y elegiremos “**IA Agent**”.



Aquí tenemos 3 parámetros que ponerle, **uno es el cerebro en sí** que será el modelo que lo conectaremos en mi caso open IA, pero hay que recalcar que hay muchos con propuestas muy interesantes y más baratas.

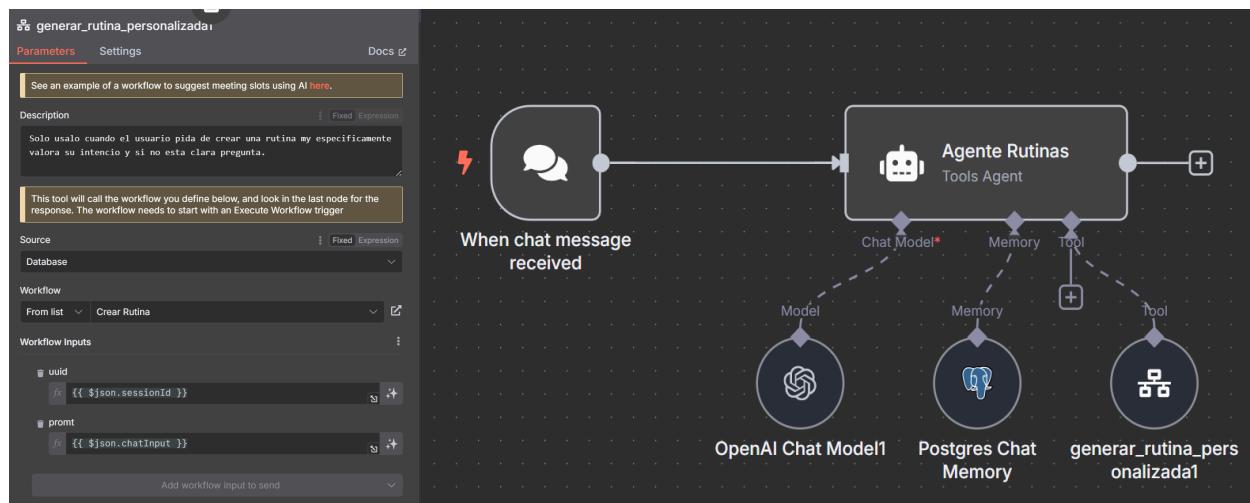


Luego, tenemos la memoria que es donde le pondremos que se guarde el chat en la memoria del agente, como nosotros hemos usado una llamada a la API de n8n necesitamos guardarlo en una base de datos, el problema de n8n es que solo deja Postgres, para ello yo **lo he conectado a un Postgres de supabase**, ya que las conversaciones en nuestra pagina solo se guarda la que está hablando, ya que si pusiera el uuid del usuario al no reiniciar nunca la memoria los agentes fallan.

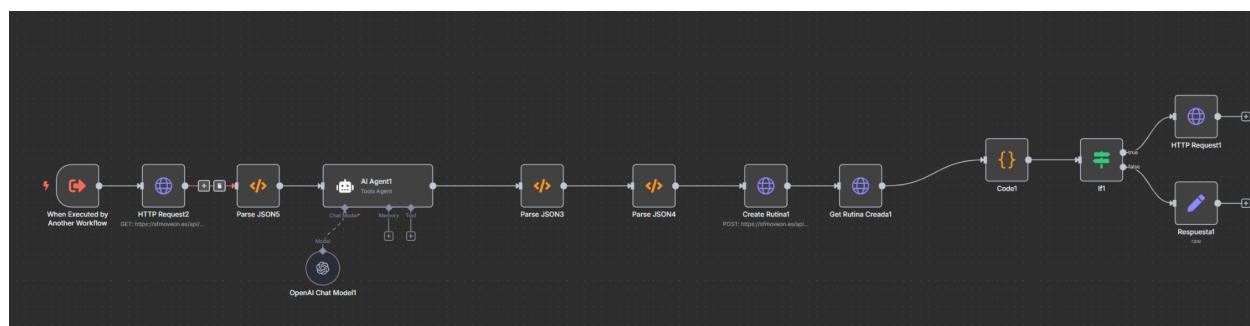
Por último las herramientas que es donde entra el juego lo que el agente puede hacer, aquí voy a explicar solo dos. Uno es un función de programación para que la ia elija cuándo hacerlo con sus parámetros y el otro es embedding.

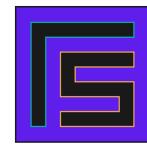
Funcion

Aquí le he puesto otro flujo para que cuando vea que el usuario quiera crear una rutina la use y tenga estos parámetros.



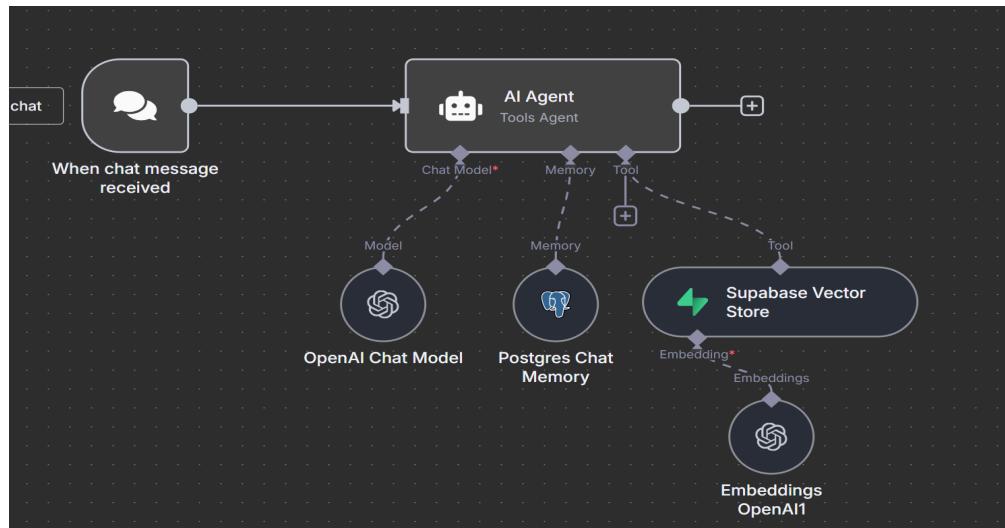
La función a la que **se conecta es a un sub agente no razonador** que recibe todos los ejercicios de mi base de datos y elige de ahí con que hacer la rutina.



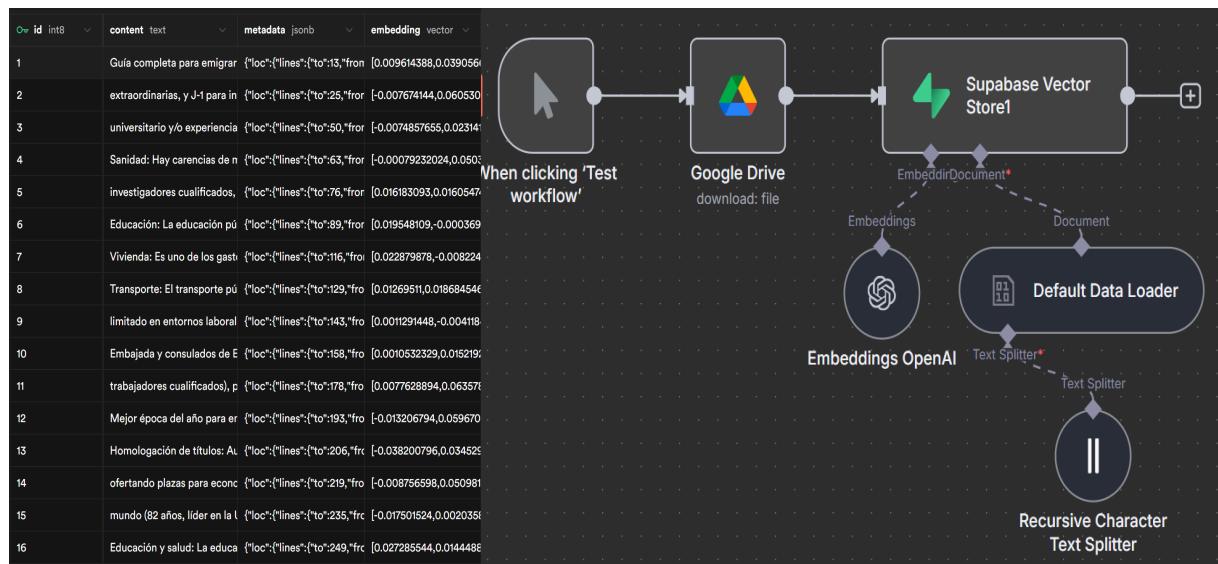


Embedding

Ese es el cerebro de Ramón IA que es simplemente dar vida a tus documentos, pdfs o cualquier tipo de documento que contenga texto.



Lo que hice fue vectorizar con la siguiente herramienta un pdf de 20 páginas que hice con un Deep Research de 20 minutos que me puedo toda la información actualizada de información de emigrar a países.



En conclusión, veo esto que es una rama de programación con mucho futuro y más complicado de lo que parece ya que los agentes son muy tontos para entender el contexto de lo que le estas pidiendo, tienen muchos fallos pero se puede decir que es algo normal ya que es una tecnología que tiene menos de 2 años en el mercado.



7. Futuras mejoras y ampliaciones

A lo largo del desarrollo de esta primera versión de la aplicación hemos pensado e identificado varias **mejoras** que nos gustaría implementar en **futuras versiones**, tanto para mejorar la seguridad como para ampliar las funcionalidades y la experiencia del usuario. Estas son algunas de las más destacadas:

- **Verificación adicional en el registro:** Añadir un campo de “confirmar contraseña” al momento de registrarse, para reducir errores y que el usuario sepa bien la contraseña que ha elegido.
- **Gestión segura del cambio de contraseña:** Incluir una verificación para modificar la contraseña desde el perfil del usuario, que primero se deba introducir la contraseña actual. Así añadimos una capa de seguridad ante accesos no autorizados.
- **Recuperación de contraseña más robusta:** Implementar un sistema de recuperación de contraseña mediante **preguntas de seguridad** o **verificación por correo electrónico**, con el envío de un enlace seguro para restablecer la contraseña.
- **Mejoras generales en la seguridad:** Revisar e implementar medidas de seguridad adicionales a nivel de validación, autenticación y gestión de sesiones, garantizando la protección de los datos del usuario mediante tokens por ejemplo.
- **Sistema de notificaciones:** Evaluar la posibilidad de enviar **notificaciones internas** o **por correo electrónico**, por ejemplo, para recordar al usuario que realice su rutina, su dieta o incluso un evento en el calendario.
- **Integración con más APIs externas:** Nos gustaría añadir más fuentes de datos, por ejemplo, para **ampliar la base de alimentos disponibles en las dietas** o incorporar rutinas personalizadas desde servicios especializados.
- **Foro comunitario:** Una idea que ya contemplábamos desde el inicio es la creación de un **foro dentro de la aplicación**, donde los usuarios puedan compartir experiencias, dudas, avances y motivarse mutuamente. Esto ayudaría a crear una comunidad activa en torno al objetivo de mejora personal.



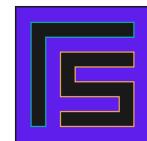
- **Roles avanzados y permisos:** Actualmente, ya contamos con un sistema básico de roles que distingue entre usuarios normales, moderadores y administradores. Sin embargo, en futuras versiones queremos ampliar y refinar estos permisos para que **cada rol tenga acceso solo a las funcionalidades que le correspondan** según su nivel de responsabilidad.

8. Comentarios y conclusiones

Hemos optado por un **diseño con colores oscuros** para nuestro sitio web, con el fin de transmitir una imagen **anónima** y evitar la fatiga visual. No obstante, hemos incorporado un **modo claro** también para los usuarios que se sientan más cómodos con colores más iluminados.

En esta fase inicial, hemos demostrado el funcionamiento básico de la aplicación, abarcando aspectos como:

- **Navegación intuitiva e iconografía personalizada**
- Un diseño coherente del **footer** y el **header**.
- Hemos implementado correctamente el sistema de **autenticación de usuarios** y la conexión con la **base de datos**.
- La posibilidad de **modificar cada uno de los datos del usuario**.
- Un **creador de rutinas y ejercicios**.
- Un **creador de dietas personalizadas**.
- Un **contador para ver el recuento de días, horas y minutos** que lleva una persona sin fumar.
- Un **mini recopilatorio** de información **sobre los próximos vuelos** en países como suiza, irlanda, países bajos... etc.
- Un **mapamundi** donde los usuarios señalan **ubicaciones destacadas** con una ligera descripción.



Además, durante el desarrollo hemos hecho incorporación de **agentes de inteligencia artificial**. Hemos integrado **n8n** con nuestra **API de ejercicios y rutinas**, de forma que un agente puede **generar rutinas personalizadas en función de una descripción que le proporciona el usuario**.

Por ejemplo, el usuario puede escribir algo como "*Quiero una rutina para perder peso sin usar pesas*", y n8n interpreta esta descripción, consulta los ejercicios disponibles a través de nuestra API (con datos provenientes de la API externa de ExerciseDB) y **devuelve una rutina adaptada** a esas necesidades.

Por último, queríamos implementar un **foro** para nuestros usuarios, pero durante el desarrollo de nuestra primera versión decidimos finalmente no incluirla.

8.1 Valoraciones personales y técnicas

Desde un punto de vista **personal**, este proyecto nos ha servido para reforzar y practicar muchos de los conocimientos que hemos adquirido a lo largo del ciclo.

Esto nos ha ayudado a **aplicar los conceptos aprendidos** en clase de forma práctica, a enfrentarnos a problemas reales y a buscar soluciones eficientes. Uno de los aspectos que más hemos disfrutado ha sido el diseño de la interfaz y la construcción de una experiencia visual clara y coherente ya que **estábamos utilizando CSS** en ficheros aparte, y **empezamos a migrarlo todo a Tailwind**. A la vez, esto ha sido muy motivador ver cómo todo el sistema iba funcionaba en conjunto, desde la base de datos hasta las respuestas que llegaban al cliente.

También hemos aprendido a **organizarnos mejor como equipo**, a distribuir tareas según nuestras fortalezas y a mantener una comunicación constante para resolver obstáculos, a pesar de tener a veces nuestros problemillas.

Desde el punto de vista **técnico**, nos sentimos orgullosos de haber trabajado con las tecnologías de **Laravel, React, Tailwind CSS y Docker**, ya que nos han permitido crear una aplicación moderna, funcional y fácil de mantener. Hemos aprendido a configurar correctamente los entornos de desarrollo (al principio no paraban de surgir problemas), a trabajar con migraciones y bases de datos relacionales, y a mantener el **control de versiones con Git**.

Uno de los retos más importantes que hemos superado fue **adaptarnos a trabajar con Laravel y su sistema de API REST**, ya que inicialmente utilizábamos **Supabase como backend**. Aunque Supabase nos sirvió para las primeras pruebas y para



comprender cómo funcionan las APIs, decidimos dar el salto a Laravel para tener mayor control, robustez y seguridad en el desarrollo del backend. Esto implicó **aprender a estructurar rutas, controladores y validaciones personalizadas** desde cero, lo cual ha sido un gran aprendizaje.

La conclusión es que ha sido una experiencia muy completa y a la vez exigente, tanto a nivel técnico como personal, y creemos que este proyecto refleja todo lo que hemos aprendido, así como nuestra capacidad para adaptarnos, trabajar en equipo y seguir creciendo como desarrolladores.

9. Bibliografía y anexos

ELABORACIÓN DE LA WEB:

Ventanas flotantes: <https://sweetalert2.github.io/>

Para navegar haciendo scroll: <https://www.npmjs.com/package/react-scroll>

Iconos de la app:

<https://fontawesome.com/>

<https://heroicons.com/>

Cookie Consent: <https://www.npmjs.com/package/react-cookie-consent>

Generación de imágenes con IA: <https://chatgpt.com/>

Video animado con IA: <https://app.klingai.com/>

Calendario: <https://www.npmjs.com/package/react-big-calendar>

Mapamundi: <https://leafletjs.com/>

Mapa de google: <https://www.google.es/maps?hl=es>

Búsqueda de vuelos: <https://www.google.com/>

Búsqueda de imágenes: <https://www.freepik.es/>

Agentes de IA: <https://n8n.io/>



DOCUMENTACIÓN:

Tailwind CSS Docs: <https://tailwindcss.com/docs>

Manual de PHP: <https://www.php.net/manual/es/>

W3Schools: <https://www.w3schools.com/>

DISEÑO / ESQUEMAS / MOCKUPS:

Paleta de colores: <https://coolors.co/>

Planificación de pantallas: <https://lucidspark.com/es>

Planificación Entidad Relación: <https://www.lucidchart.com/>

Mockup bajo nivel: <https://balsamiq.com/>

Mockup alto nivel: <https://www.figma.com/es-es/>

Diapositivas y edición de imágenes: <https://www.canva.com/>

SERVIDORES

Almacenamiento de imágenes: <https://console.cloudinary.com/>

Servidor en la web: <https://supabase.com/>

Despliegue de la APP:

<https://aws.amazon.com/es/>

<https://vercel.com/>

INSPIRACIONES EN OTRAS PÁGINAS

<https://www.blizzard.com/es-es/>

<https://www.flashscore.es/>