

PROYECTO INTEGRADO

OnlyFood



ONLY FOOD

Desarrollo de Aplicaciones Web.

Autor: Javier Galván Poley.

Tutor: Juan Sevillano.

Centro: I.E.S Francisco Romero Vargas, Jerez de la Frontera.

Curso: 2022/2023.

Tabla de contenido

1. Introducción	2
2. Objetivos	3
3. Tecnologías escogidas y justificación	4
3.1. Tecnologías escogidas y justificación	4
3.2. Motor de bases de datos	6
3.3. Frameworks seleccionados	6
4. Diseño de la aplicación	7
4.1. Casos de uso	7
4.2. Modelo de dominio	8
5. Arquitectura de la aplicación	9
5.1. Estructura del proyecto	9
5.2. Recursos externos	12
6. Manual de despliegue y usuario	13
6.1. Requisitos hardware y software aplicables.	13
6.2. Instrucciones de despliegue	14
6.3. Manual de Usuario	14
6.4. Instrucciones para despliegue	24
7. Gestión del Proyecto.	25
7.1. Problemas encontrados.	25
7.2. Modificaciones sobre el proyecto planteado inicialmente.	26
7.3. Posibles mejoras al proyecto.	26
8. Conclusión.	27
9. Bibliografía.	28

1. Introducción

Este proyecto consiste en el desarrollo de una red social de comida, la cual permite crear publicaciones en las que se pueden observar la foto del plato elegido, el título del mismo, los ingredientes utilizados y ubicación del sitio donde se encuentra.

La idea es que los usuarios puedan publicar los lugares en los que han comido y recetas para recrear.

La aplicación se desarrolla en web, no requiere ningún tipo de instalación de software, sólo requiere conexión a internet, ya que tiene un diseño adaptativo para que funcione tanto en ordenador como en móvil.

2. Objetivos

La razón por la que personalmente he decidido desarrollar esta red social es por la cantidad de veces que me han surgido dudas a la hora de decidir qué comer o a dónde ir.

La idea principal del desarrollo de esta red social es que cada usuario pueda añadir a su lista de Post aquellas recetas o sitios que le gustan, para, posteriormente, poder replicar esas mismas recetas en casa o visitar esos lugares.

3. Tecnologías escogidas y justificación

La aplicación dará servicio a distintos tipos de usuarios:

- **Anónimo:** Podrán acceder a la página principal de la aplicación dentro de la cuál podrán ver publicaciones, pero no tendrán acceso a ninguna otra funcionalidad. Si intentan realizar cualquier otra acción, se les requerirá que se registren o inicien sesión si ya cuentan con un usuario.
- **Usuario:** Podrán acceder mediante una autenticación a sus datos de carácter personal y podrán acceder a todas las funcionalidades que tiene la red social.

3.1. Tecnologías escogidas y justificación.

La aplicación se desarrolla en Next.js , usando React, typescript y con la base de datos en MongoDB, con la cuál se interactúa a través de prisma. A toda la web se le ha aplicado el diseño de estilos a través de TailwindCss, además, para facilitar y potenciar el desarrollo de esta aplicación, se han utilizado diversas librerías, tanto de react como de Next.js, entre las que se encuentran:

@next-auth/prisma-adapter@1.0.5

└─ @prisma/client@4.15.0

└─ @types/bcrypt@5.0.0

└─ @types/lodash@4.14.191

└─ @types/node@18.14.2

└─ @types/react-dom@18.0.11

└─ @types/react@18.0.28

└─ autoprefixer@10.4.13

└─ axios@1.4.0

└─ bcrypt@5.1.0

└─ cloudinary-react@1.8.1

└─ cloudinary@1.37.1

└─ date-fns@2.30.0

|— eslint-config-next@13.2.1
|— eslint@8.35.0
|— lodash@4.17.21
|— next-auth@4.22.1
|— next-cloudinary@4.12.0
|— next@13.2.4
|— postcss@8.4.21
|— prisma-client@0.0.0
|— prisma@4.15.0
|— react-dom@18.2.0
|— react-dropzone@14.2.3
|— react-hot-toast@2.4.0
|— react-icons@4.8.0
|— react-spinners@0.13.8
|— react-toastify@9.1.1
|— react@18.2.0
|— swr@2.1.5
|— tailwind-scrollbar-hide@1.1.7
|— tailwindcss@3.2.7
|— toast@0.5.4
|— typescript@4.9.5
└— zustand@4.3.8

3.2. Motor de bases de datos.

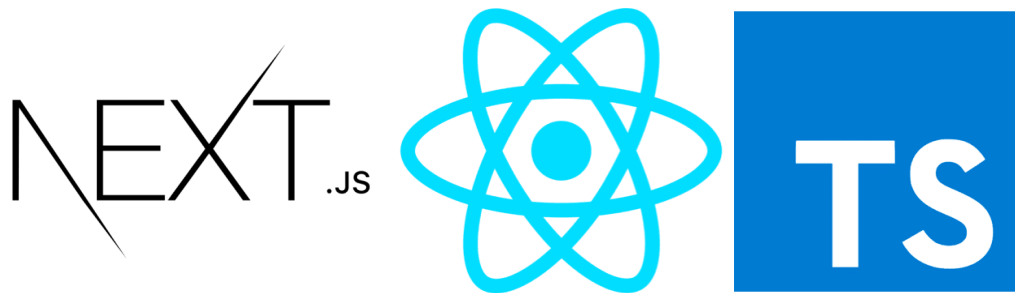
El motor de bases de datos usado es MongoDB Atlas. Las razones de su elección son las siguientes:

- Es un motor gratuito, por lo que no tiene coste.
- Presenta una gestión simple y sencilla desde el navegador.
- Es un motor muy extendido en la comunidad de desarrolladores, por lo que conseguir ayuda es muy sencillo.
- Mantenimiento sencillo y fácil para cualquier tipo de usuario.
- Es escalable, lo cual nos da una ventaja con vistas al futuro.

3.3. Frameworks seleccionados.

El framework usado es Nextjs. Las razones de su elección son las siguientes:

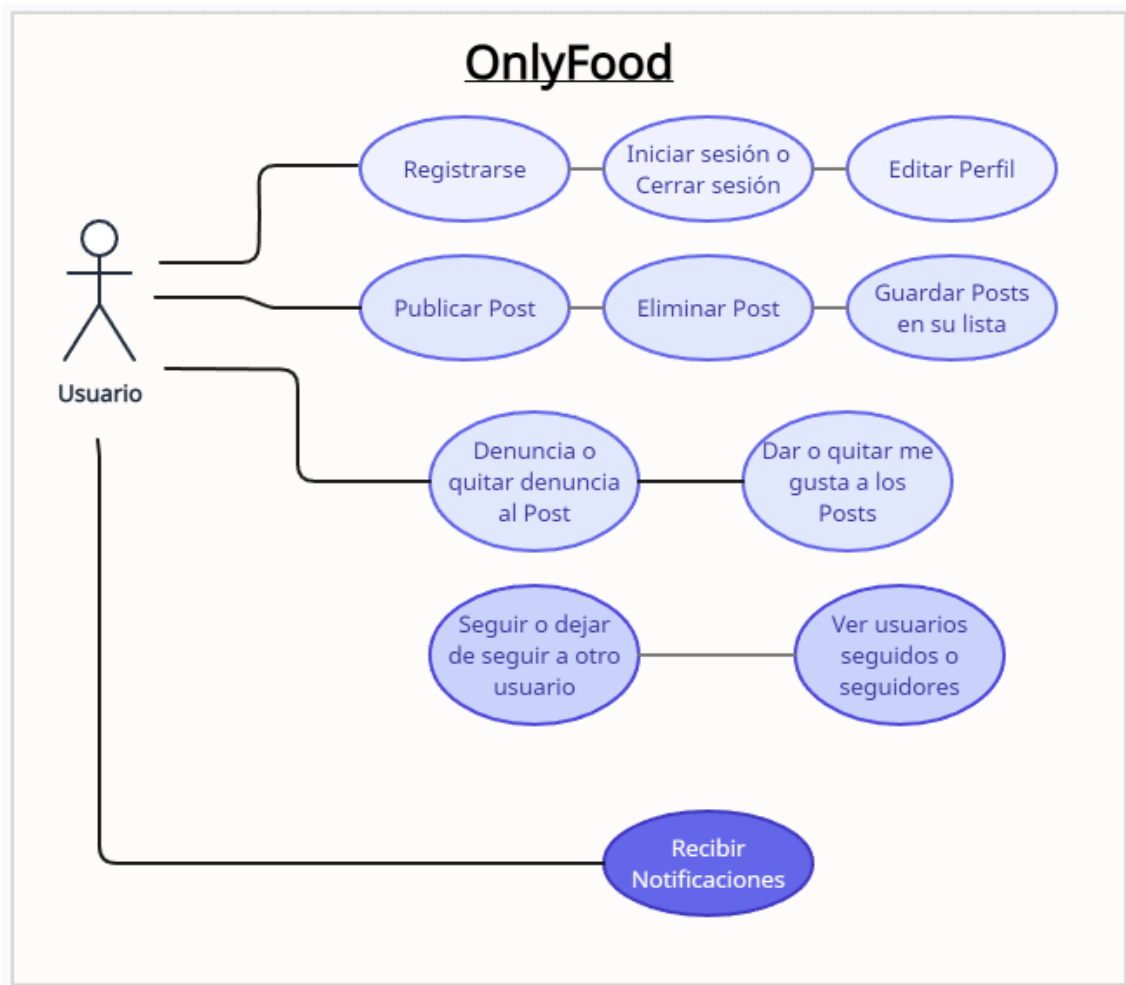
- Es un framework gratuito.
- Cuenta con una gran cantidad de librerías para poder funcionar con el lenguaje react, lo que facilita y simplifica muchas funcionalidades, además, aporta gran facilidad para su despliegue.
- Uso de React y TypeScript.



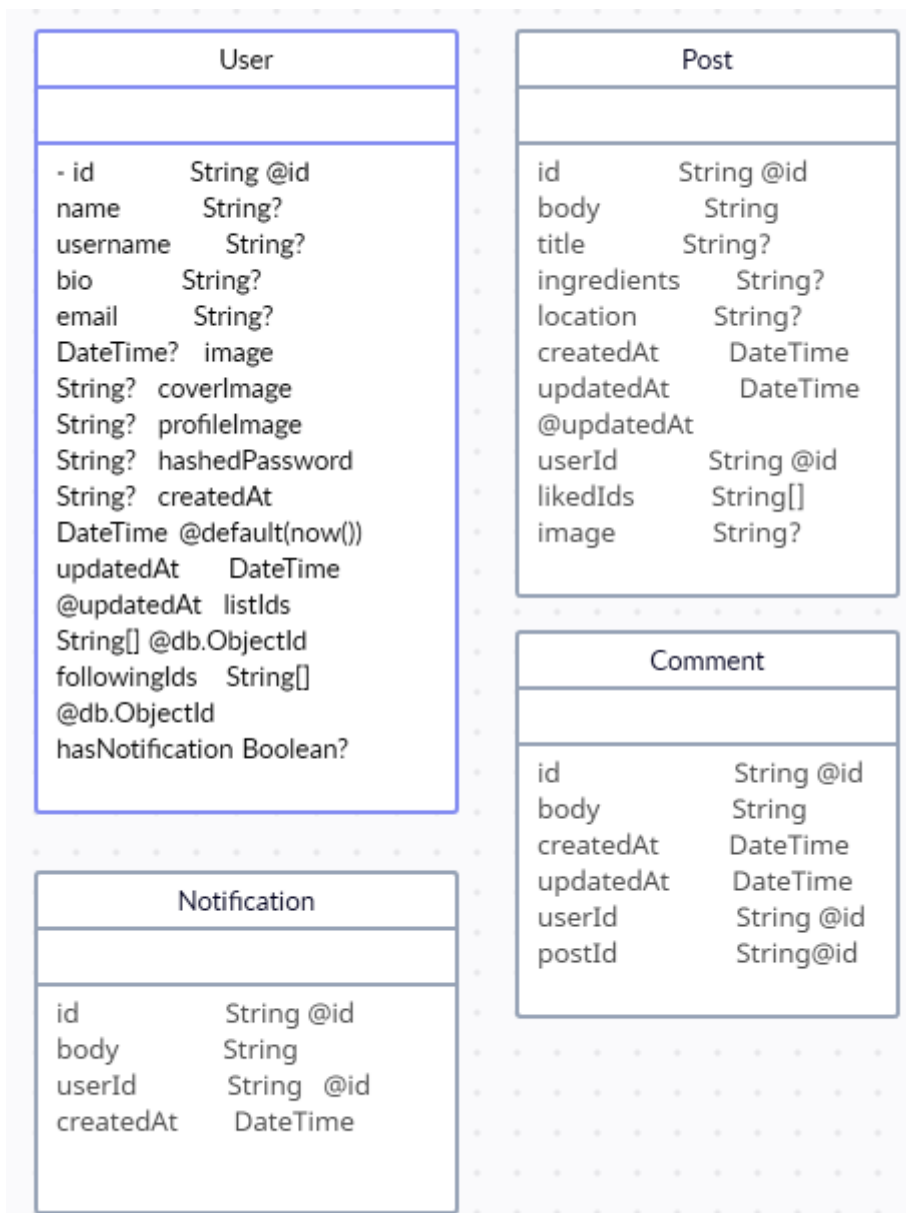
4. Diseño de la aplicación

4.1. Casos de uso.

Funcionalidades del usuario:



4.2. Modelo de dominio.

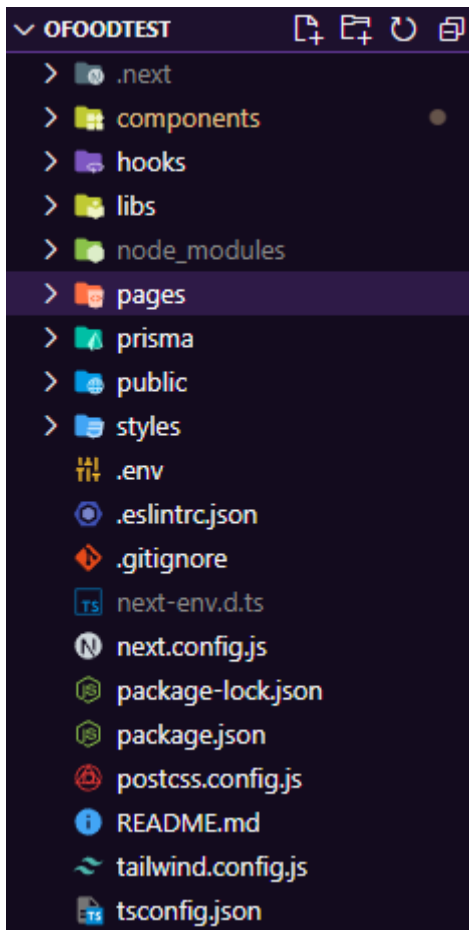


Representación del modelo de la base de Datos. En este caso el modelo de datos utilizado corresponde a uno no relacional, por lo tanto el esquema no muestra la relación entre las distintas tablas.

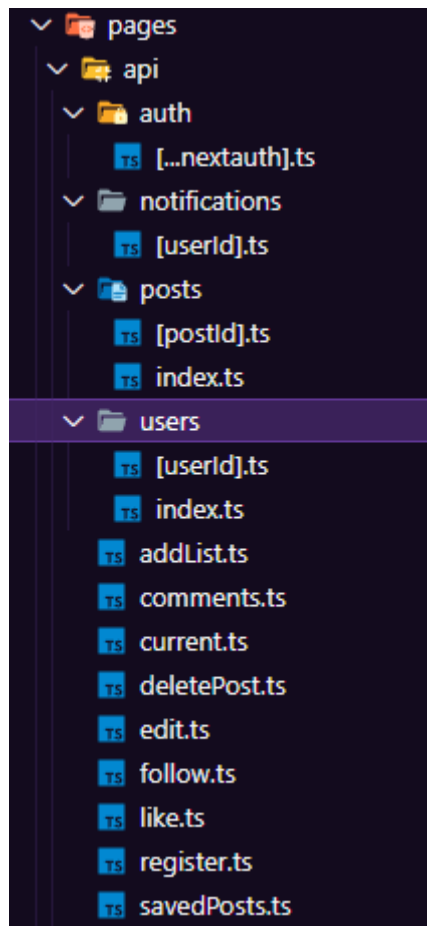
5. Arquitectura de la aplicación

5.1. Estructura del proyecto.

La estructura de la aplicación es la típica de un proyecto de NextJs con React. Las principales carpetas son:

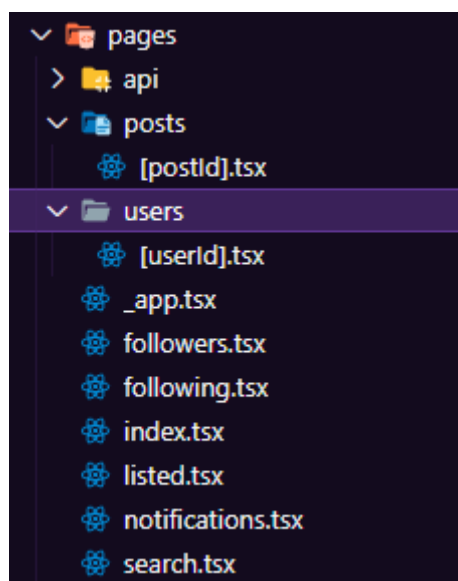


- **Components:** Hace referencia al sitio donde se almacenan todos los componentes que van a ser usados en la web con react.
- **Hooks:** En esta carpeta se encuentran todos los hooks para poder manejar los estados de los componentes y comunicar con las distintas apis.
- **Pages:** Dentro de esta se encuentra la carpeta de apis, donde se realizan todas las peticiones a la base de datos, las cuales son llamadas a través de los hooks.



Estas apis están organizadas como generales y tienen algunas subcarpetas en aquellas cuya llamada depende de si necesita algún dato extra para realizar la petición.

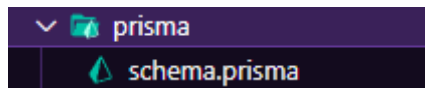
- **pages:**



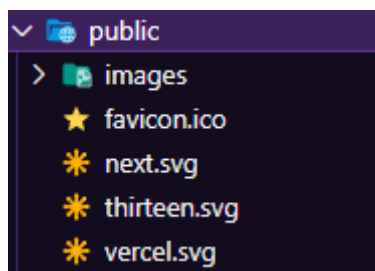
Posteriormente, dentro de pages, se encuentran las propias páginas de la web, compuestas de subcarpetas para usuarios y posts y el resto de páginas empleadas.

- **Prisma:**

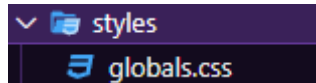
Carpeta que contiene el esquema de la base de datos en prisma, que a su vez va a comunicar con la base de datos de MongoDB.



- **Public:**



- **Styles:**



Hace referencia al sitio donde se cuenta con la configuración básica de estilos para el correcto funcionamiento de TailwindCss.

- **.env:**

```
DATABASE_URL="mongodb+srv://jgpdev8:siCK5V6bHYjAoLFn@clusterofood.i2gqkfy.mongodb.net/test"
NEXTAUTH_JWT_SECRET="NEXT_AUTH_JWT_SECRET"
NEXTAUTH_SECRET="NEXT_AUTH_SECRET"
NEXT_PUBLIC_CLOUDINARY_CLOUD_NAME="dv95degpi" You, 6 hours ago • added changes
```

Contiene el archivo .env donde se configura la base de datos y la api a Cloudinary, que es la nube utilizada para la subida de imágenes.

- **Resto de carpetas:**

Cabe mencionar también la existencia de otras carpetas que se crean al iniciar nuestro proyecto con NextJs, las cuáles almacenarán la configuración de NextJs del proyecto y la configuración de todos los paquetes o librerías que se han ido instalando durante el desarrollo del mismo.

- **.gitignore:**

Presenta una gran importancia su configuración, donde se especifican qué archivos no se subirán a github. Normalmente estos archivos se corresponden con los de configuración y paquetes, que son los que cada persona inicializa e instala personalmente antes de inicializar la aplicación, como se explicará próximamente en las instrucciones de instalación.

5.2. Recursos externos.

En esta aplicación se han usado diversos recursos externos, es decir, librerías y códigos no escritos por el desarrollador de esta aplicación. Los recursos externos usados son:

@next-auth/prisma-adapter@1.0.5

└─ @prisma/client@4.15.0

└─ @types/bcrypt@5.0.0

└─ @types/lodash@4.14.191

└─ @types/node@18.14.2

└─ @types/react-dom@18.0.11

└─ @types/react@18.0.28

└─ autoprefixer@10.4.13

└─ axios@1.4.0

└─ bcrypt@5.1.0

└─ cloudinary-react@1.8.1

└─ cloudinary@1.37.1

└─ date-fns@2.30.0

└─ eslint-config-next@13.2.1

└─ eslint@8.35.0

└─ lodash@4.17.21

└─ next-auth@4.22.1

└─ next-cloudinary@4.12.0

└─ next@13.2.4
└─ postcss@8.4.21
└─ prisma-client@0.0.0
└─ prisma@4.15.0
└─ react-dom@18.2.0
└─ react-dropzone@14.2.3
└─ react-hot-toast@2.4.0
└─ react-icons@4.8.0
└─ react-spinners@0.13.8
└─ react-toastify@9.1.1
└─ react@18.2.0
└─ swr@2.1.5
└─ tailwind-scrollbar-hide@1.1.7
└─ tailwindcss@3.2.7
└─ toast@0.5.4
└─ typescript@4.9.5
└─ zustand@4.3.8

6. Manual de despliegue y usuario

6.1. Requisitos hardware y software aplicables.

- Navegadores soportados en móviles:

	Chrom e	Firefox	Safari
Android	X	X	-
iOS	X	X	X

- o Navegadores soportados en Pc:

	Chrom e	Firefox	Opera	Safari
Mac	X	X	X	X
Windows	X	X	X	-
Linux	X	X	X	-

6.2. Instrucciones de Instalación.

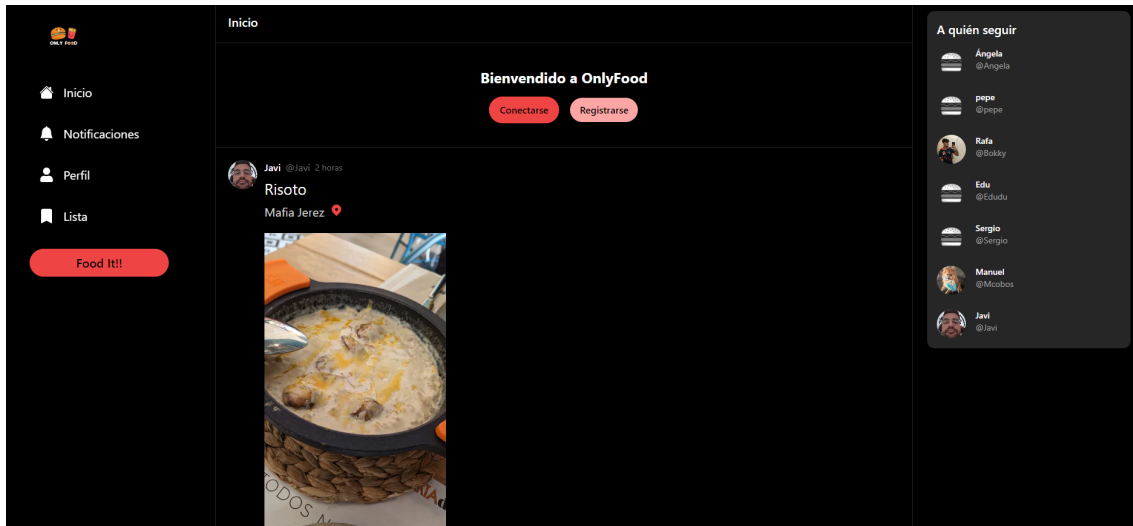
Para instalar la aplicación de manera local bastará con copiar el repositorio de forma local, y ejecutar el comando `npm i` para instalar todas las dependencias necesarias antes de poder ejecutar el proyecto, además, habrá que configurar el archivo `.env`, para asociarlo a una base de datos de MongoDB, y ejecutar el los comandos `npx prisma create`, `npx prisma generate` y `npx prisma db push`, para crear el esquema en la base de datos una vez enlazada.

6.3. Manual de Usuario.

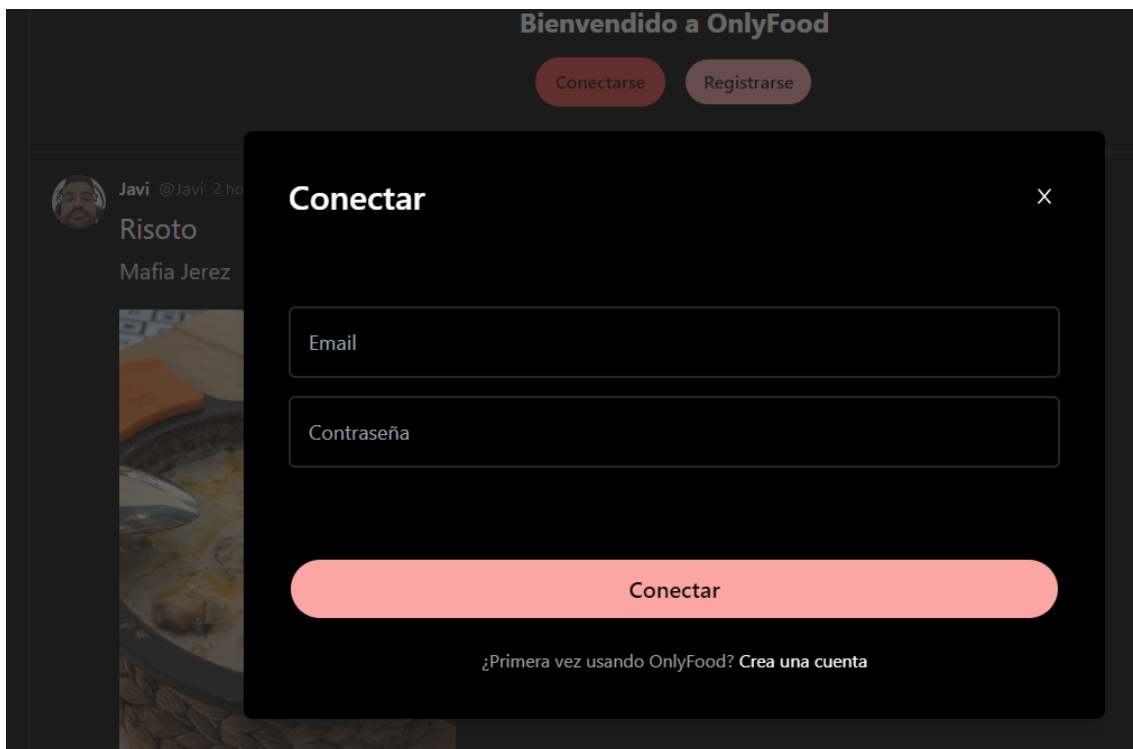
Instrucciones para el uso de la aplicación:

Aún su intuitividad, se va a comentar paso a paso cómo manejarse a través de la aplicación y todas las funcionalidades que presenta.

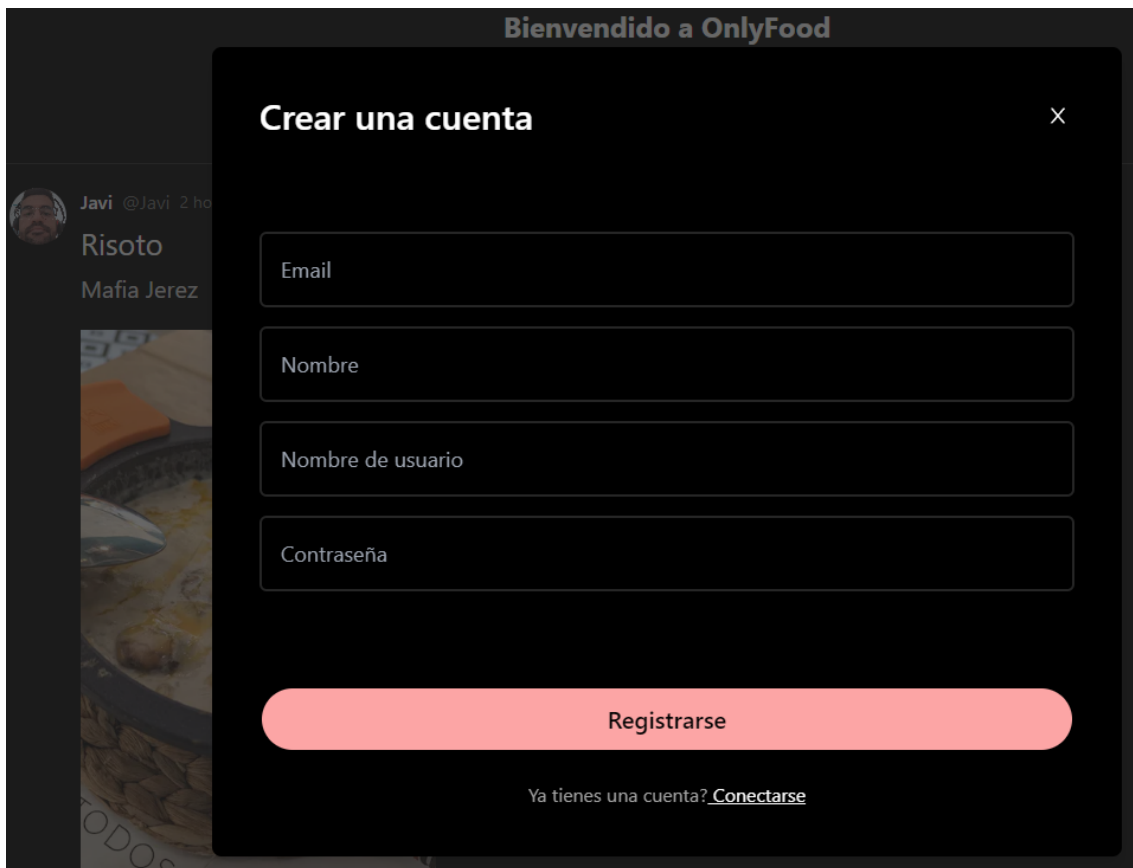
Al acceder a la web nos encontraremos con esta pantalla:



Para comenzar, en esta pantalla se puede observar inicialmente el logo de la aplicación, a continuación, una barra lateral de navegación con la cual se podrá interactuar para desplazarse por las distintas funcionalidades de la página. También se encuentra otra barra lateral en la parte derecha que muestra un listado de usuarios a los que seguir. Además, se encuentra un apartado central dónde se visualizan los Posts, estos, podrán ser visualizados junto a los perfiles de los usuarios, pero en el momento que se intente realizar cualquier acción que requiera de un usuario, aparecerá este modal por pantalla:

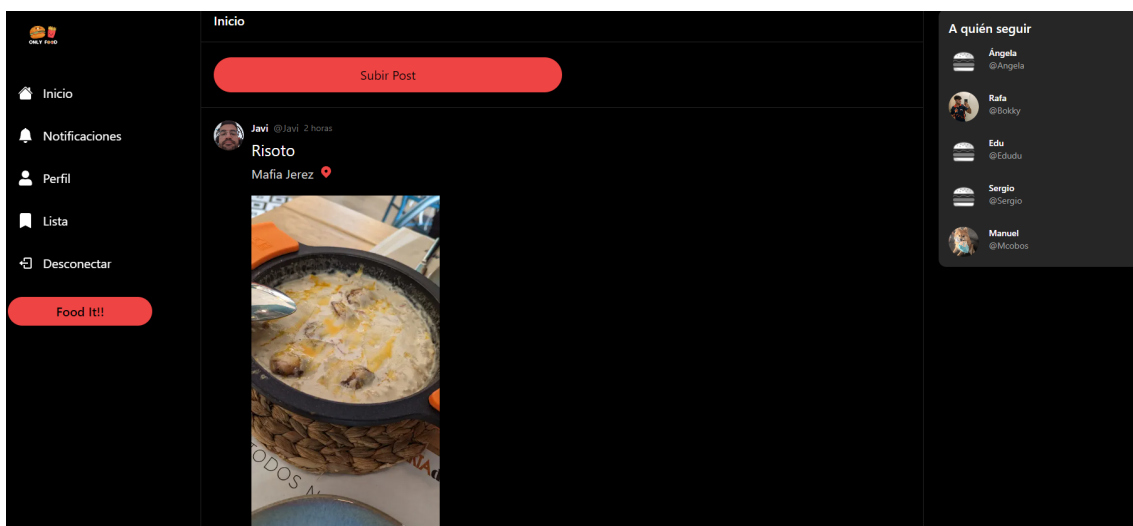


En la parte inferior de este modal se encuentra la opción de cambiar entre el apartado de conectar y de registrarse:

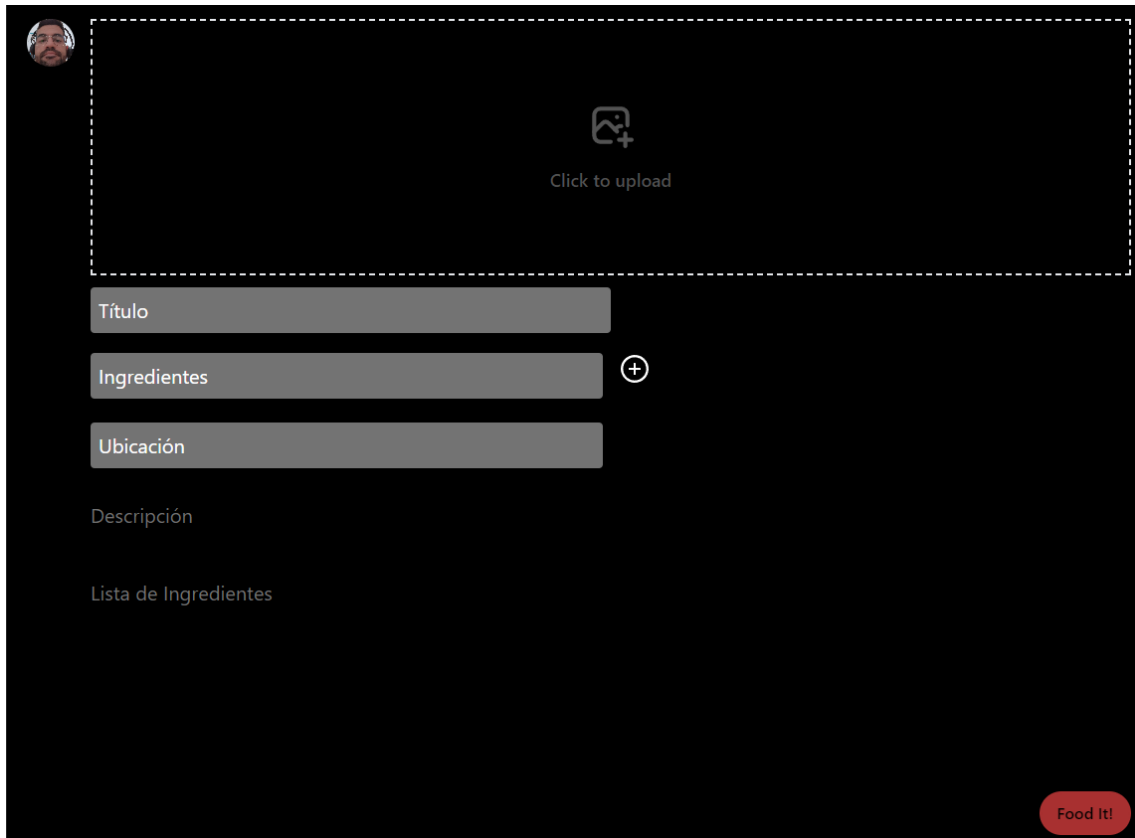


Inicialmente el usuario deberá registrarse en la red social para poder acceder a ella.

Lo primero que se observa tras acceder es la página principal con todos los Posts:



Se puede observar una pantalla similar a la de antes de acceder con un usuario, sin embargo, al acceder a la misma, se tendrá un total acceso de las funcionalidades. Lo primero que se diferencia es el botón para subir un Post, en el cual, si se pulsa, se mostrará el siguiente formulario para crear publicaciones:



The screenshot shows a user profile picture in the top left corner. Below it is a large dashed rectangular area for uploading a photo, containing a camera icon and the text "Click to upload". Underneath the photo area are three input fields: "Título", "Ingredientes" (with a plus icon to its right), and "Ubicación". Below these fields is a "Descripción" label followed by a large text area. At the bottom left, there is a "Lista de Ingredientes" label. In the bottom right corner, there is a red button labeled "Food It!".

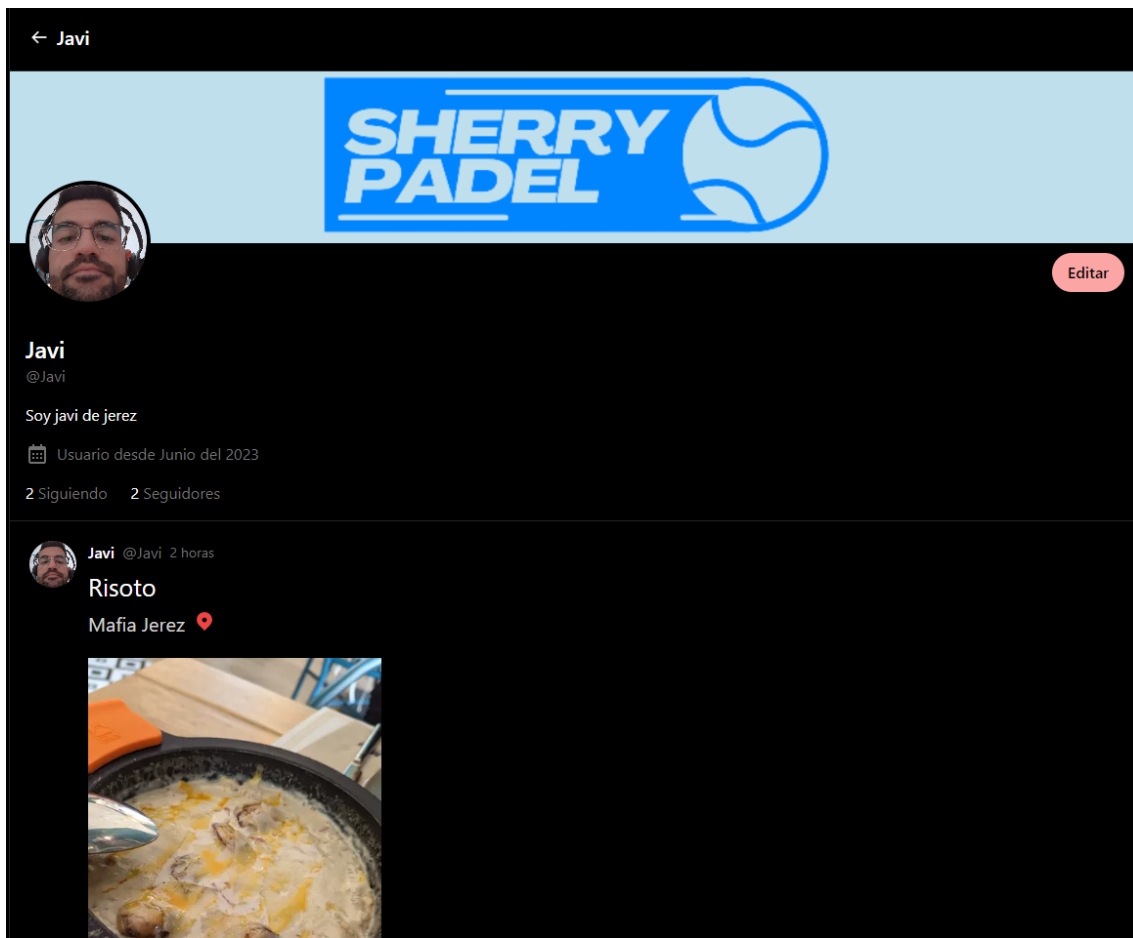
Se observan una serie de apartados con información a proporcionar, como ya se ha mencionado anteriormente, los cuales son:

- Foto
- Título
- Ingredientes
- Ubicación
- Descripción

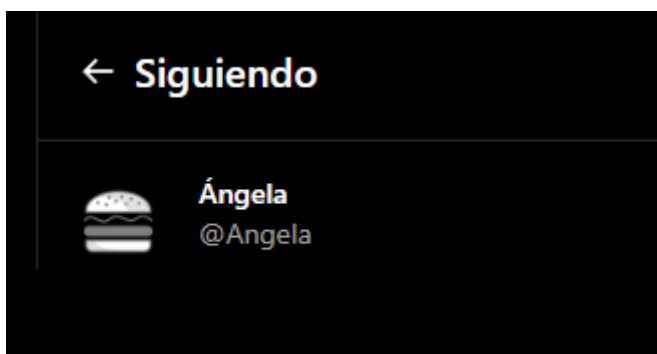
Una vez creado el primer Post, se van a enumerar todas las funcionalidades permitidas en la web:

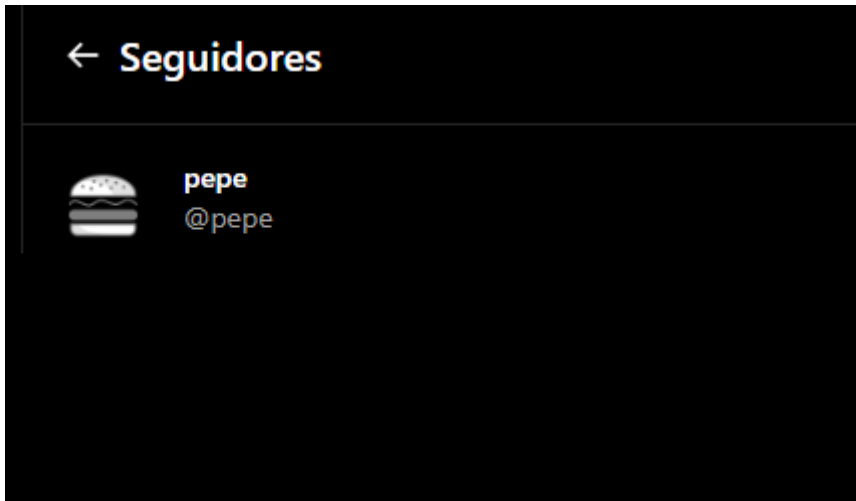
Editar el perfil:

- Primero se deberá acceder al apartado del perfil desde el botón de la barra de navegación lateral:

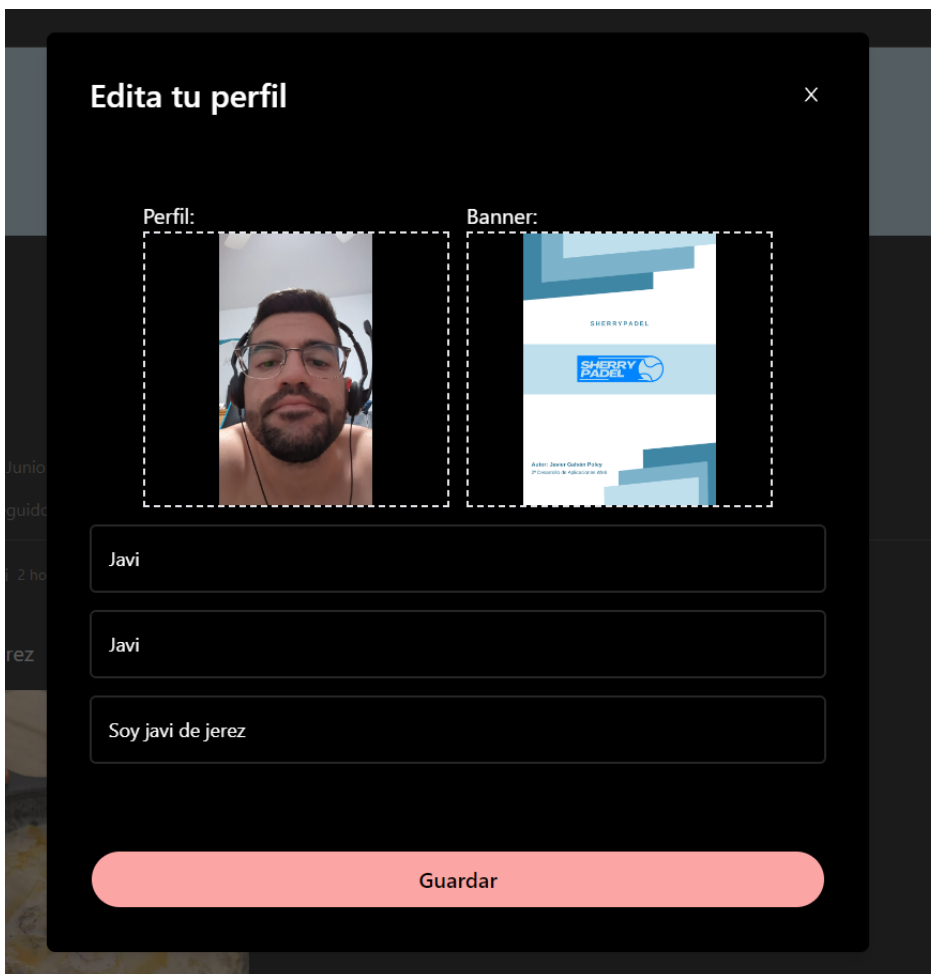


Una vez dentro del perfil, se observará el nombre, nombre de usuario, Bio, fecha de creación del usuario, cantidad de usuarios seguidos y seguidores, donde se podrán abrir aparte para ver información de los mismos.:





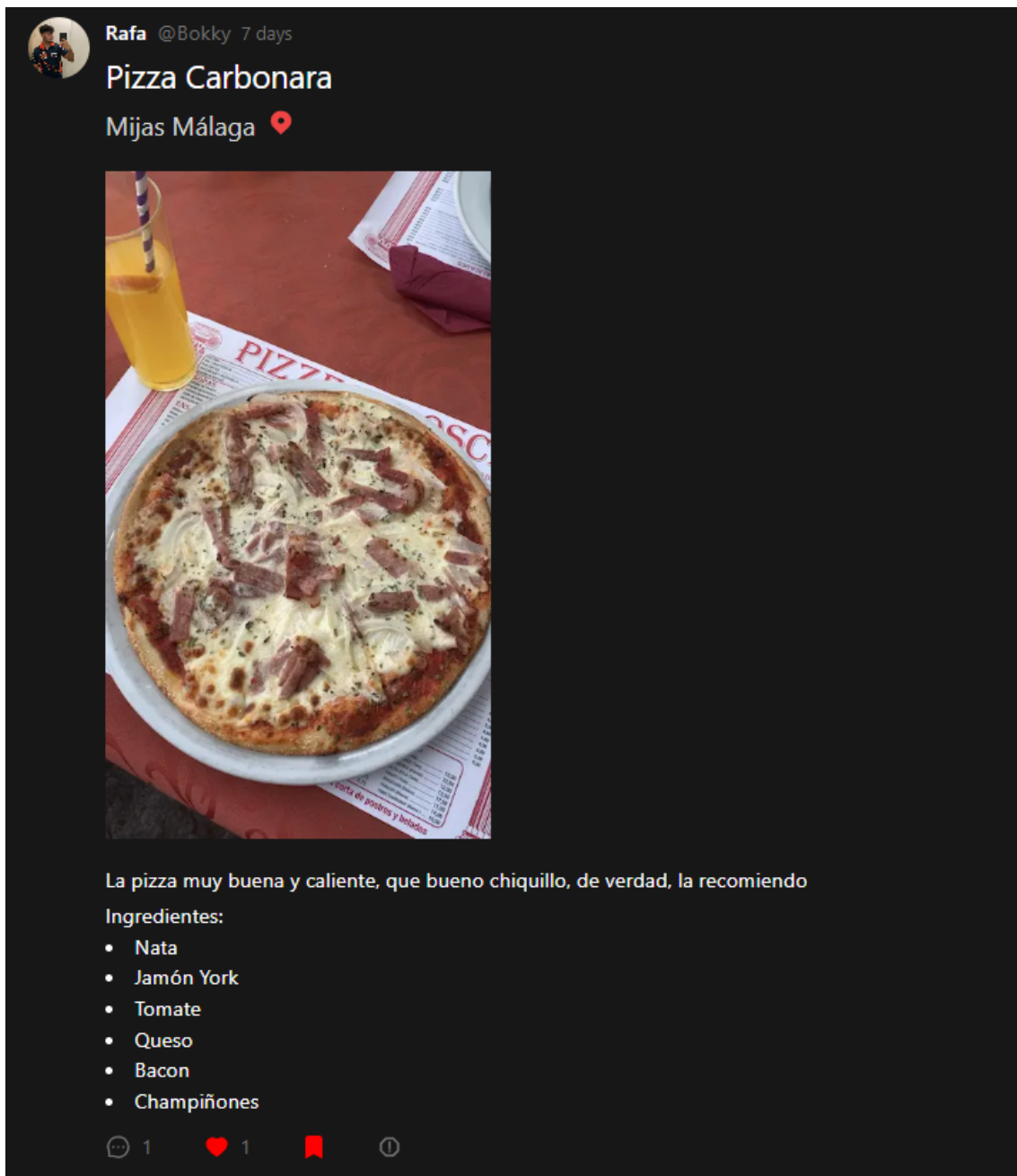
Además, arriba a la derecha se puede visualizar el botón de editar, a través del cuál se puede modificar la foto de perfil, la foto del Banner, el nombre y la Bio:



El Botón de editar mencionado anteriormente pasará de Editar a Seguir o dejar de seguir, dependiendo de si se sigue o no a ese usuario. Si se visita el perfil de otra persona:



Al volver al inicio y observar un Post, o publicación, se podrán hacer una serie de interacciones con el mismo:



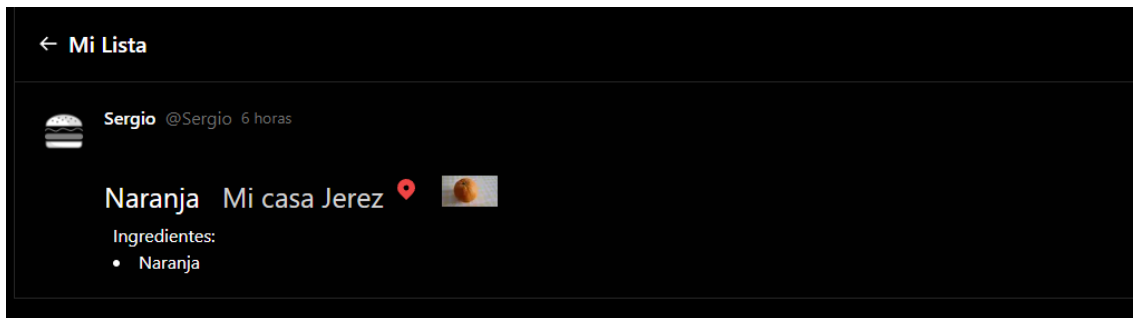
Se encuentra un primer botón, el cuál llevará a la publicación como tal y permitirá comentar en esta, además, se mostrará el número de comentarios que tiene la misma.

El segundo botón consiste en un corazón que permite dar o quitar el “me gusta” a la publicación, dependiendo de si ya se le había dado me gusta con anterioridad o no.

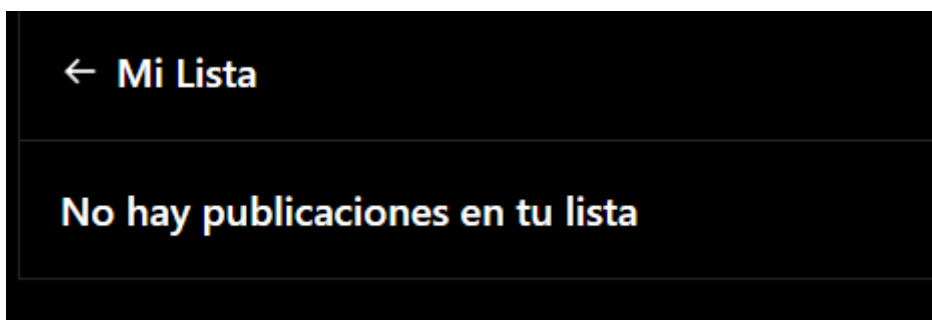
Un tercer botón en forma de pestaña que permite añadir la publicación a una lista personal, o eliminarla de forma similar que el botón de me gusta.

Finalmente, se encuentra un botón que permite denunciar o quitar la denuncia a un Post, actualmente está configurado para que a las tres denuncias el Post se elimine.

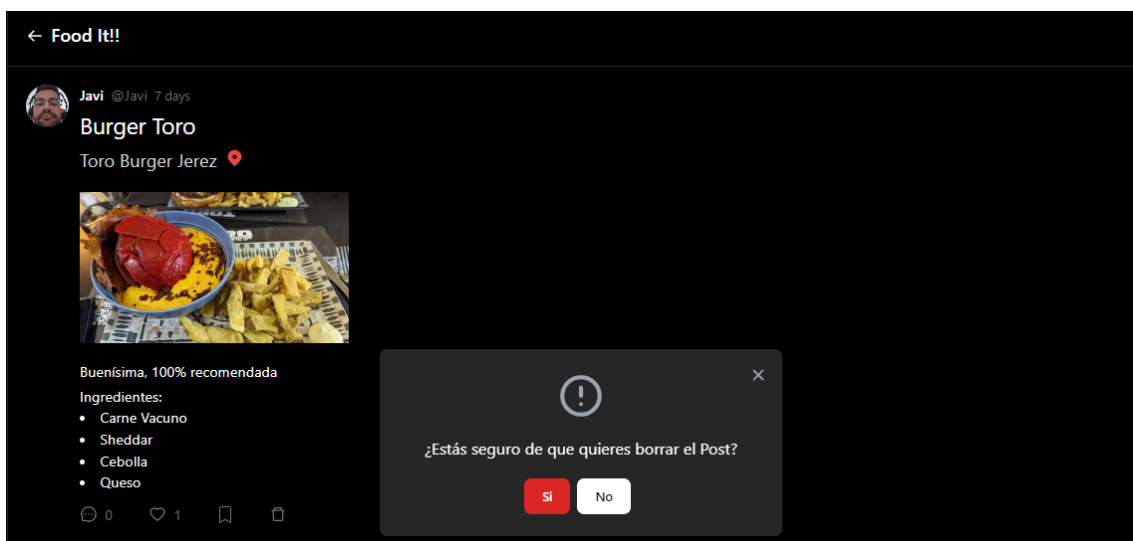
Desde la barra de navegación se cuenta con un botón para acceder a “mi lista” en la cuál se observan un resumen de las publicaciones que se tienen en la lista, que a su vez, es un enlace para acceder a la publicación para interactuar con ella.



En caso de no tener ninguna publicación en la lista, se verá lo siguiente:



Por último, se encuentra el botón de eliminar publicación, que es una papelera y sólo aparecerá en aquellos Post que son propios del usuario.

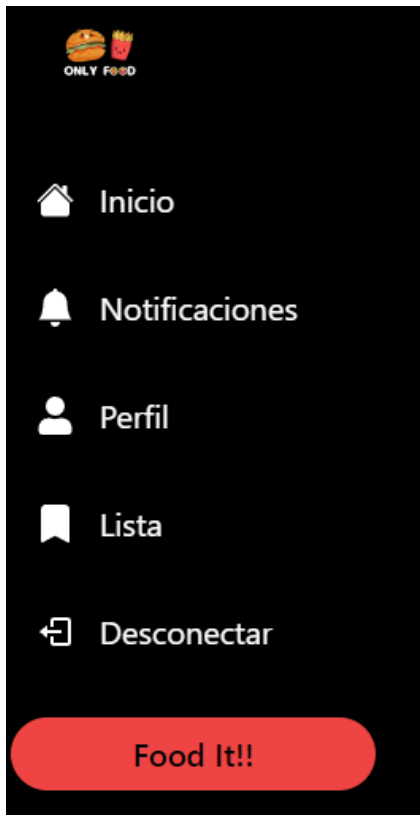


Al pulsar en él, se mostrará una ventana emergente que realizará una pregunta de confirmación al eliminar el post, y en caso de pulsar en el sí, este quedará eliminado permanentemente.

Volviendo a la barra de navegación, se observan otros dos últimos botones; el de notificaciones, que llevará a la página dónde se ven todas las interacciones con nuestro usuario, ya sean me gusta a las publicaciones o personas que nos han seguido:

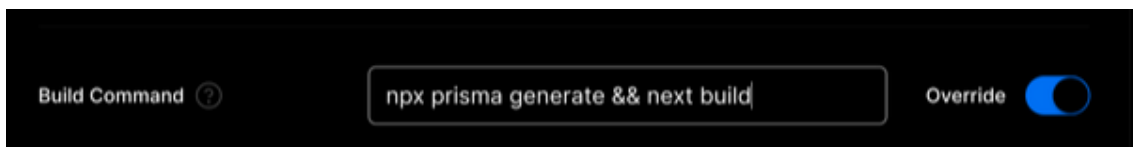


Por último, se encuentra el botón de FoodIt, el cual manda de nuevo a la sección principal, para subir un Post, invitando, de esta forma, a los usuarios a publicar lo máximo posible:



6.4. Instrucciones para despliegue.

El despliegue de la aplicación en este caso se ha realizado de forma gratuita a través de Vercel, y es muy simple, sólo se necesita contar con el repositorio de github con el proyecto, registrarse en Vercel y vincular la cuenta. Posteriormente, vincular ese repositorio de github a vercel, y al realizar el despliegue se tendrá que aportar la configuración con la que contamos en nuestro archivo .env, además de aportar un script por el funcionamiento de prisma, que es el siguiente:



7. Gestión del Proyecto.

1. Diseño de la página de inicio, con su barra de navegación
2. Diseño del modal para inicio de sesión y registro de usuario.
3. Creación de base de datos y conexión con la misma.
4. Creación de API para creación de usuarios y encriptación de contraseña.
5. Creación de API para inicio de sesión y `CurrentUser` para contar con usuario actual.
6. Prueba del modal de registro e inicio de sesión y correcto funcionamiento.
7. Diseño del Perfil del usuario y desarrollo del modal para editar el mismo.
8. Implementación en la API para modificar el usuario.
9. Diseño del Post.
10. Diseño de formulario para crear un Post.
11. Creación de la API para crear el Post.
12. Prueba de creación de Post y comprobación de funcionamiento correcto.
13. Implementación de botón editar o seguir dependiendo del usuario.
14. Creación de API para seguir a otros usuarios.
15. Prueba de funcionamiento seguir y dejar de seguir otros usuarios.
16. Creación de la API para comentar los Post.
17. Creación de la API para dar y quitar me gusta a los Post.
18. Prueba de funcionamiento de ambas funcionalidades.
19. Implementación de las API para creación de notificaciones.
20. Implementación de subida de Fotos para los Post, que hasta el momento funcionaban sin imagen.
21. Implementación de la lista de ingredientes.
22. Desarrollo del botón para eliminar Posts.
23. Desarrollo de lista de seguidores y siguiendo.
24. Mostrar y ocultar formulario de creación de posts a través de botón
25. Mejora de las notificaciones, mostrando username de la persona.
26. Implementación de botón para denunciar el Post, y eliminación del Post según el número de denuncias.
27. Mejoras de diseño finales y ajustes para el diseño adaptativo a móvil.

7.1. Problemas encontrados.

- Problema al subir imágenes, que en un principio se subían a través de transformación a base64, pero esto provocaba que imágenes con un cierto peso no pudieran ser subidas, por lo que se acaba implementando nube de fotos, en este caso Cloudinary.
- Problema para refrescar lista de los Posts o cambiar el estado de los botones, resuelto a través de algunos video tutoriales y la ayuda de Chat GPT.
- Problema con el diseño adaptativo para móvil que acaba provocando un rediseño y la implementación de la barra lateral desplegable.
- Problema con el formato de fechas para España, la librería date-fns con la fecha local española daba fallos que interrumpen el funcionamiento de la web, solución mediante modificación manual del texto.
- Problema con el despliegue, ya que al funcionar con prisma hay que generar la base de datos junto antes de desplegar, investigación para dar lugar al uso del script en vercel antes del despliegue.

7.2. Modificaciones sobre el proyecto planteado inicialmente.

Respecto al anteproyecto, el resultado final ha sido bastante similar a lo que se plantea en un principio, se ha conseguido el objetivo inicial del desarrollo del proyecto, aunque las tecnologías utilizadas no han sido exactamente las mismas que fueron listadas al inicio.

Se ha utilizado TypeScript en lugar de JavaScript. Como framework se ha utilizado NextJs en lugar de Vite.

Se ha implementado prisma para interactuar con la base de datos que sí ha sido en MongoDB.

7.3. Posibles mejoras al proyecto.

Qué nos gustaría añadir al proyecto para mejorarlo o ampliarlo, pero que no venía indicado en el anteproyecto. Son ideas de mejora para implementar en un futuro.

- Implementación de un chat en vivo para que los usuarios puedan interactuar entre sí.
- Opción de iniciar sesión a través de la cuenta de google.
- Búsqueda de publicaciones por ubicación o nombre.
- Edición de publicaciones ya creadas.
- Descarga de lista de Ingredientes para compra en PDF.
- Desarrollo de aplicaciones para móvil en React Native.
- Ver personas que han dado me gusta a una publicación.
- Ajustes de privacidad para los perfiles, para que puedan visitarse o no dependiendo de si es público o privado.
- Mejoras visuales para móvil o desarrollo de aplicación en versión móvil a través de React Native.

8. Conclusión.

Bajo mi opinión personal, considero que el desarrollo del proyecto ha sido bastante positivo, ya que se ha cumplido el esquema inicial que se planteaba en el anteproyecto. Además, se han utilizado lenguajes no utilizados previamente como son TypeScript con React, prisma para interactuar con la base de datos y el uso de diversas librerías para simplificar ciertas tareas, al igual que el continuo uso de git para el desarrollo del mismo, dando lugar a un amplio aprendizaje y una mejora a la hora de afrontar problemas y buscar soluciones para los continuos fallos que iban surgiendo durante el desarrollo, dejando aún así muchas mejoras que implementar para poder seguir aprendiendo y mejorando.

9. Bibliografía.

- Chat GPT.
- <https://www.prisma.io/docs/guides/other/troubleshooting-orm/help-articles/vercel-caching-issue>
- <https://react-icons.github.io/react-icons/>
- <https://nextjs.org/docs>
- <https://react-typescript-cheatsheet.netlify.app/docs/basic/setup/>
- <https://handsonreact.com/docs/labs/react-tutorial-typescript>
- https://docs.google.com/document/d/1kdIVlt5uPAGmsRjsrQFaPdVyg9R_vR4Z/edit#
- <https://www.youtube.com/watch?v=15VKbky2gB4>