

Red social culinaria orientada a los procesadores de alimentos

Memoria de Proyecto Final de Máster

Máster Universitario en Desarrollo de Sitios y Aplicaciones Web

Área de Informática, Multimedia y Telecomunicación

Autor: Aníbal Santos Gómez

Consultor: Miguel Calvo Matalobos

Profesor: César Pablo Córcoles Briongos

Junio 2022

Créditos/Copyright

Licencias recomendadas



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)

Abstract

Este proyecto busca crear una comunidad de usuarios que tengan en común el gusto por la cocina y el uso de procesadores de alimentos para elaborar recetas. Se pretende recoger las recetas que vayan creando los usuarios según el robot utilizado, categorizar dichas recetas y ofrecer la posibilidad de crear un listado de ingredientes con el objetivo de tener una lista de la compra exacta a las recetas que se vayan a preparar. Se podrá dar feedback a las aportaciones y se incentivará el uso de la aplicación mediante diversas estrategias como la creación de secciones o una pequeña página de producto con un blog. Esta pequeña comunidad conseguirá concentrar a personas que utilizan diariamente los procesadores de alimentos. Para su desarrollo utilizaremos un stack MERN¹ con un diseño basado en la técnica Mobile First y servicios escalables.

Palabras clave: cocina, recetas, alimentos, procesadores de alimentos, robots de cocina, red social, aplicación web.

¹ Mongoddb, Express.js, React y Node.js

Abstract (english version)

This project seeks to build a community of users who have in common a love for cooking and using food processors to prepare recipes. It is intended to store the recipes created by users according to the processor used, categorize these recipes and offer the possibility of adding a list of ingredients in order to have an exact shopping list for the recipes to be prepared.

It will be possible to give feedback to the contributions and the use of the application will be encouraged through various strategies such as the creation of sections or a small product page with a blog. This small community will bring together people who use food processors on a daily basis. For its development we will use a MERN² stack with a design based on the Mobile First technique and scalable services.

Keywords: cooking, recipes, food, food processors, food robots, social network, web application.

² Mongoddb, Express.js, React and Node.js

Agradecimientos

Gracias a mis **padres (Isabel y Silvio)** que me han apoyado siempre en mi camino y en este apasionante cambio de planes. A **Marta**, que soporta y admira día tras día mi pasión por seguir creciendo. A **Miguel, Paula, Alejandra, Vera y Carlos**, por el apoyo durante estos años de transformación constante. A **Sergio, Manuel y a David**, porque juntos nos empujamos.

Gracias a **Luis** y a **Javi** por sus horas de clases.

Gracias también a todo el equipo de **Iberdrola Comercial** (muy especialmente a **Laura, Nuria, Sacris, Sesi, Miguel, Victor y Edu**) la oportunidad que me dieron y con la que empecé a volar, estaré eternamente agradecido por ello. Y por supuesto no pueden faltar en estas líneas mis agradecimientos a la **comunidad Open Source**.

Y sobre todo a mi fiel compañero, **Congo**.

“Hay que creer en la posibilidad de la felicidad para ser feliz.” (Lev Tolstoi, Guerra y Paz, 1867).

Índice

1. Introducción	9
2. Descripción	10
3. Objetivos	13
3.1 Principales	13
3.2 Secundarios	13
4. Marco teórico/Escenario	15
4.1. Los medios de comunicación	15
4.2. El marketing social	16
4.3. Las redes sociales	16
5. Contenidos	19
6. Metodología	23
7. Planificación	26
7.1. Epics	26
7.2. Sprints y tareas	26
7.3. Diagrama de Gantt	28
8. Proceso de trabajo	29
8.1 MVP	29
8.2. Sprints	29
9. Arquitectura de la aplicación	31
9.1. Patrones de diseño	31
9.2. Tecnologías	34
9.3 Arquitectura de la base de datos	36
10. Plataforma de desarrollo	37
10.1 Entorno de desarrollo	37
10.2 Hardware	38
11. Estudio de viabilidad	39
12. Diseño y usabilidad	42
12.1 Conceptualización	42
12.2 Diseño de la experiencia	45
12.2.1 Requisitos	45
12.2.2 Sitemap y user flows	47
12.3 Diseño de la interacción	53
12.3.1 Páginas de error	54
12.3.2 Inicio de sesión / registro	55
12.3.3 Página de inicio	56
12.3.4 Trending / Robousers	57
12.3.1 Categorías	58

12.3.1 Recetas	59
12.3.1 Receta	60
12.3.1 Perfil	61
12.3.1 Crear / Editar / Eliminar receta	62
12.3.1 Ajustes / Editar perfil	63
12.3.1 Colecciones	64
12.3.1 Mensajes	67
12.3.1 Favoritos	68
13. Implementación	69
13.1 Desarrollo	69
13.2 Seguridad	69
13.3 Testing	69
13.4 Versionado	69
13.5 Despliegues	69
14. Instalación y uso	70
14.1 Requisitos e instalación	70
14.2 Configuración de entornos	70
14.3 Herramientas recomendadas	70
15. Proyección a futuro	71
15.1 Roadmap	71
15.2 Contribuciones	71
16. Conclusiones	73
Anexo 1. Entregables del proyecto	74
Anexo 2. Código fuente (extractos)	75
Anexo 3. Librerías	76
Anexo 4. Capturas de pantalla	78
Anexo 5. Bibliografía	79
Anexo 6. Vita	80

Figuras y tablas

Índice de figuras

Figura 1: Diagrama de Gantt - Planificación	29
Figura 2: Clean Architecture	33
Figura 3: Principio de Inversión de Dependencia	34
Figura 4: Arquitectura y herramientas	35
Figura 5: Vercel	35
Figura 6: Github	35
Figura 7: Nextjs	36
Figura 8: React	36
Figura 9: Atlas MongoDB	36
Figura 10: Arquitectura de la base de datos	37
Figura 11: Las actividades en la cocina del futuro	41
Figura 12: Cambios tecnológicos en las cocinas.	41
Figura 13: Sitemap	48
Figura 14: Flujo de interacción sin autenticación	49
Figura 15: Flujo de interacción con autenticación	50
Figura 16: Flujo de creación de una receta	51
Figura 17: Flujos de edición y eliminación de una receta	52
Figura 18: Flujos de añadir, editar y eliminar una colección	53
Figura 19: Flujos de interacción social sobre una receta	54
Figura 20: Páginas error.	55
Figura 21: Inicio de sesión / registro.	56
Figura 22: Página de inicio.	57
Figura 23: Trending / Robousers.	58
Figura 24: Categorías.	59
Figura 25: Recetas.	60
Figura 26: Receta.	61
Figura 27: Perfil.	62
Figura 28: Crear receta.	63
Figura 29: Eliminar receta.	63
Figura 29: Ajustes / Editar perfil.	64
Figura 30: Colecciones.	65
Figura 31: Colecciones / Ingredientes.	66
Figura 32: Colecciones / Recetas.	67
Figura 33: Mensajes.	68
Figura 34: Favoritos.	69

Índice de tablas

Tabla 1: Resumen de entregas del Trabajo de Fin de Máster	27
Tabla 2: Sprints, tareas y planificación del Trabajo de fin de Máster	28
Tabla 3: User persona - Ana Sánchez	44
Tabla 4: User persona - Carlos Hernández	45
Tabla 5: Requisitos Funcionales	47
Tabla 6: Requisitos no funcionales	47

1. Introducción

En pleno 2022, nos encontramos con ante un escenario digital donde abunda la creación de contenido. Podemos ver como plataformas como Youtube, Twitch, Instagram o Linkedin, están llenas de usuarios que generan grandes cantidades de contenido. Más de 500 horas de video son subidas a YouTube cada minuto (Kinsta.com), 500 millones de personas usan Instagram diariamente (Oberlo.es).

No cabe duda de que el contenido es el gran protagonista en las plataformas y de que están abandonando los canales de comunicación tradicionales como la radio o la televisión; en virtud de las plataformas, que de alguna manera generan un contenido que atrae a otros usuarios.

Por otro lado, estamos asistiendo a un proceso de domotización de nuestros hogares, donde se están integrando sistemas de automatización de tareas y nuevos medios de comunicación y servicios. El objetivo de este proceso, no es otro que, mejorar nuestra calidad de vida, el ahorro de energía y el tiempo.

La cocina está sufriendo también una revolución, los procesadores de alimentos, comúnmente llamados, robots de cocina empezaron a ocupar un lugar en los hogares y con los años han ido tomando protagonismo no sólo a nivel doméstico sino también profesional.

Este tipo de herramientas han sido susceptibles a los cambios y las mejoras tecnológicas. Nos ahorran tiempo y concentran en una herramienta funcionalidades que se encuentran repartidas en multitud de utensilios culinarios.

Con esta pequeña situación, podemos situarnos en el contexto de que existen infinidad de personas generando contenido, en campos como el culinario. Y el principal problema que suelen tener los robots de cocina, es la comunidad que tienen detrás. Normalmente esta comunidad de usuarios se centra en el modelo de robot que tienen y en las recetas disponibles que les proporciona la marca o distribuidor, mediante su plataforma de contenido, que normalmente es privativa.

Este proyecto tratará de abrir un camino a que los usuarios de internet puedan crear libremente recetas de cocina adaptadas para sus procesadores, y tengan un sitio de referencia donde encontrar nuevas creaciones compartidas por otros usuarios. Gracias a esta comunidad, podremos beneficiarnos de recetas relativas al modelo de robot que utilicemos sin pasar largas horas buscando en internet.

Se pretende generar una comunidad viva, participativa, y además una API de datos que puede ser explotada posteriormente con otro tipo de negocios.

2. Descripción

Con este Trabajo Final de Máster se plantea la creación de una aplicación web que responda, **en primer lugar** a la problemática que se comentaba en el punto anterior; crear una comunidad de usuarios bajo una red social, que permita a cada uno de ellos crear recetas de cocina adaptadas a su robot y compartirla con el resto de miembros, agregar recetas a su colección, que les parezcan atractivas y generar una lista de la compra con la totalidad de ingredientes ya calculada.

En segundo lugar un sistema de creación de producto digital ágil, que nos permita desarrollar un ecosistema satélite a nuestro producto principal, mediante páginas de promoción, blog con contenido, y otra serie de servicios.

Esta aplicación será pública y podrá acceder al contenido sin registro previo. El usuario podrá revisar las recetas, sus detalles, elaboraciones, ingredientes, compartir las en otras redes sociales o por correo electrónico.

Los usuarios registrados, además podrán crear, editar y eliminar sus propias recetas, añadir a colecciones aquellas recetas que quieran cocinar o a favoritos aquellas que más les hayan gustado.

Podrán consultar la lista total de ingredientes de las recetas en colecciones para obtener una lista de la compra de todo aquello que desean preparar.

También podrán generar interacciones de algún tipo con otros usuarios, mediante el feedback directo a cada receta, comentando, siguiendo a otras personas o escribiendo un mensaje privado a estas.

De forma general podemos listar las secciones y funcionalidades que tendrá la aplicación:

- Página de inicio: donde encontraremos un buscador y diversas secciones destacadas, como secciones categorizadas, sección por robot, últimas publicaciones y una sección de los usuarios más activos.
- Página de recetas: donde encontraremos un listado de todas las recetas ordenadas por la fecha de creación más reciente.
- Página de categoría: tendremos un listado de recetas, donde veremos las últimas recetas publicadas de esta categoría, y las cinco más cocinadas.

- Página de detalle de receta: aquí podremos consultar la información de la receta, ingredientes, tiempo, dificultad, fotografía, comentarios, interacciones, el usuario que la ha creado.

Sólo mostraremos a los usuarios registrados los pasos de creación de la receta así como las acciones de interacción sobre la receta (comentar, agregar a favoritos o guardar en colecciones).

- Página de perfil de usuario: aquí podremos ver las recetas que ha publicado un usuario, seguirle/dejar de seguirle, enviarle un mensaje.
- Página de creación de receta: donde el usuario registrado podrá llenar un formulario con toda la información de la receta.
- Página de edición de receta: donde el usuario registrado podrá editar o eliminar la receta que ha creado.
- Página de cuenta de usuario: para usuarios registrados, donde tendremos una vista, con las opciones disponibles:
 - Perfil: para editar la información del usuario.
 - Recetas favoritas: las recetas favoritas que vaya añadiendo el usuario.
 - Colecciones: recetas organizadas por colecciones, con la opción de exportar todos los ingredientes a una lista de la compra.

Para el desarrollo del proyecto se ha elegido un stack MERN, basado en Mongodbd, Express.js React y Node.js, se integrará bajo el framework Next.js, basado en React en la capa Frontend y en la capa Backend en Express.js, para la capa de datos, se trabajará con Mongodbd y Mongoose como ORM.

Además para asegurarnos de poder realizar un desarrollo de producto digital completo, realizaremos una librería de componentes genéricos aislados con React y Storybook. El conjunto total de herramientas a utilizar, son las siguientes:

- **Nextjs.**
- **React.**
- **TypeScript.**
- **Tailwindcss.**
- **Axios.**
- **Jest.**
- **Testing library.**
- **Redux Toolkit.**
- **Storybook.**
- **Node.js.**
- **Express.js.**
- **Mongodb.**
- **Mongoose.**
- **AWS S3 Bucket.**
- **Vercel.**
- **Github.**
- **Atlas Mongodb.**

3. Objetivos

El presente trabajo tiene como objetivo principal, el desarrollo de una aplicación web, para su utilización como red social que permita a los usuarios la creación y gestión de recetas orientada a procesadores de alimentos. Crear una comunidad de usuarios y promover la publicación y consumo de un contenido accesible tanto por usuarios tanto registrados como no registrados.

Concretamente podemos realizar una división entre **objetivos principales**, aquellos que fundamentan la sostenibilidad del actual proyecto, y **secundarios**, aquellos que nos permiten la consecución de los primeros.

3.1 Principales

- Diseñar, desarrollar e implementar una aplicación web que nos permita construir una comunidad de usuarios y una API propia, a través de una red social
- Poner en práctica los conocimientos adquiridos a lo largo de la realización de cada una de las asignaturas del presente Máster.
- Crear un sistema de desarrollo eficaz para la elaboración de productos digitales.

3.2 Secundarios

- Determinar los requisitos para el desarrollo de la aplicación web, mediante un análisis integral del producto y las necesidades de los usuarios.
- Diseñar una aplicación responsive bajo el enfoque Mobile First, utilizando los requisitos obtenidos durante el análisis, aplicación de guías de diseño, usabilidad y experiencia.
- Desarrollo de un ecosistema de desarrollo integral orientado a la producción de bloques de código reutilizables que nos permitan ampliar el producto digital, mediante un marco de trabajo efectivo, basado en librerías propias reutilizables, que nos permita reducir el tiempo de producción.
- Elaborar un plan de calidad del software bajo el ecosistema de desarrollo integral, mediante la implementación de algún tipo de prueba, ya sea unitaria, integración o end-to-end³.

³ Metodología de pruebas de software que consiste en probar una aplicación desde el punto de vista del usuario final.

- Utilizar herramientas modernas de despliegue e integración continua, para centrarnos en el desarrollo del producto.
- Utilizar algún patrón de diseño de ingeniería del software, que nos permita minimizar los efectos secundarios en la lógica de negocio.

4. Marco teórico/Escenario

4.1. Los medios de comunicación

En los medios clásicos de comunicación, el receptor de esa comunicación es un receptor parcialmente pasivo. Llamamos **nuevos medios de comunicación** a aquéllos que se ponen al servicio del cliente y que permiten la interacción con el mismo, por medio de un canal que está a disposición tanto de la empresa como del cliente.

Understanding Media (McLuhan, 1964) sienta las bases para el estudio y la comprensión de los medios de comunicación de masas. McLuhan habla de que todos los medios de comunicación:

Son una prolongación humana, física o psíquica. Entendiendo medio en su aceptación más amplia, no sólo las creaciones tecnológicas sino las facultades humanas: la palabra hablada, la escritura, la mano, el puño, el pie y la piel. Y todo medio mecánico o tecnológico es, a su vez, extensión de las facultades humanas.

El medio es el que condiciona el mensaje. Se trata de poner énfasis en el efecto de la forma, más que en el efecto del contenido. Medio y mensaje funcionan simbióticamente puesto que uno suele contener una representación del otro. El contenido de la televisión es la electricidad, que contiene, a su vez, la imagen. El contenido del cine es la fotografía, que contiene la pintura, etc.⁴

Conocer las características del medio y la forma en que se transmite el mensaje, al mismo tiempo que las características físicas que potencia el ser humano y la implicación cognitiva en el uso de los nuevos medios de comunicación, se convertirá en el germen justificativo de este proyecto.

El uso del teléfono ha evolucionado hasta convertirlo en un objeto del que se depende constantemente: para comunicar que se está llegando, para descargarse sonidos personalizados, para narrar la vida en el momento presente.

El uso de Internet por las corporaciones ha sido espectacular y los motivos son muy diversos. Por ejemplo, muchas empresas creen que Internet es la respuesta para conquistar el mundo y llegar a segmentos de clientes que no eran posibles para una organización.

⁴ Sempere, P. *McLuhan en la era de Google: Memorias y profecías de la Aldea Global*. Madrid: Editorial Popular, 2007.

El conocimiento del ser humano en cuanto al uso de Internet y de las nuevas tecnologías se convirtió en el mantra de muchas de estas nuevas empresas nacidas en Internet, mientras que, para las empresas tradicionales, no se innovó internamente en cuanto a la comprensión del cliente.

4.2. El marketing social

En los últimos años, el marketing social se ha aplicado al ahorro de energía, a dejar de fumar, a una conducción de vehículos más segura. Muchas veces se trata de convencer a la gente de que adopte hábitos nuevos. El proceso de difusión queda definido por Rogers como:

Un proceso en el que una innovación se comunica a través de determinados canales de comunicación durante un determinado tiempo a los miembros de un sistema social.⁵

El proceso de decisión sobre una innovación es una actividad que representa una búsqueda y procesado de información, en la que un individuo se encuentra motivado para reducir su incertidumbre sobre las ventajas y desventajas de dicha innovación. Las innovaciones que proporcionan mayores ventajas, que son compatibles, que se pueden comprobar, observar y que son menos complejas se adoptarán más rápidamente que otras.

La comunicación es un proceso por el cual los participantes crean y comparten información entre ellos, para alcanzar una comprensión mutua.

4.3. Las redes sociales

En primer lugar deberíamos señalar que, la tecnología parece haber enraizado profundamente en nuestro sentido común y actualmente es fácil detectar su presencia implícita o explícita en la mayor parte de los discursos u opiniones de los medios de comunicación⁶.

Asumiendo este discurso y los puntos anteriores en nuestro marco teórico, respecto a los canales de comunicación y su respectiva evolución, no cabe duda, que podemos resaltar, como las herramientas tecnológicas han cambiado la forma en la que nos comunicamos y recibimos la información. Han entrado en esfera nuevos agentes y canales de comunicación, y la proliferación de los mismos ha generado una pluralidad a nivel exponencial tanto de emisores como receptores.

⁵ Rogers, E. M. *Diffusion of Innovations*. New York: Free Press, 2003.

⁶ Aibar, E. *La visión constructivista de la innovación tecnológica. Una introducción al modelo SCOT*. Barcelona: UOC, 2006.

Las redes sociales propician la interacción de miles de personas en tiempo real. Las redes sociales no son otra cosa que máquinas sociales diseñadas para fabricar situaciones, relaciones y conflictos. No hay duda de que suponen un nuevo agente comunicativo.

Definidos claramente estas bases y antecedentes, podemos determinar que la herramienta red social, sirve como canal de comunicación en una aplicación cuya misión es estar viva, crecer compartir conocimiento y tener una autonomía interna, controlada, pero menos privativa que las plataformas de empresas ligadas a los procesadores de alimentos.

Las plataformas donde proliferan amplias cantidades de recetas creadas por usuarios, son **Instagram** y **Youtube**, donde podemos ver influencers que se dedican a promocionar recetas de cocina con algún tipo de recompensa comercial. Fuera de estas encontramos algunos ejemplos más centrados en la cocina en sí, como serían:

- **Funcook**, <https://funcook.com/>, es una red social donde se pueden conseguir cientos de recetas que la gente comparte. Los usuarios explican paso a paso los ingredientes y utensilios que han utilizado para crearlas. Además, estas recetas están divididas en secciones según del tipo o categoría que sean: ensaladas, postres, pastas, carne etc.

Se pueden puntuar, valorar y comentar los platos de los demás. Permite ver las recetas mejor puntuadas y los usuarios mejor valorados, puedes seguirles, guardar sus mejores recetas y preguntarles dudas. También puedes elegir los ingredientes con los que quieras cocinar o no, para filtrar las recetas que más se adapten a tus gustos. Esta red social incluye una lista de la compra para que no olvides ningún ingrediente en tu próxima compra.

- **Cookpad**, <https://cookpad.com/es>, es una aplicación utilizada por personas de todo el mundo que comparten el gusto por cocinar. Existen unas 60,000 recetas de miles de chefs y cocineros aficionados.

Esta aplicación se desarrolló por primera vez en Japón y presentaba una variedad de recetas de Cookpad en japonés. Se extendió a Occidente a lo largo de los años, aunque tiene pocos seguidores en Estados Unidos. Está dirigida a todos los chefs caseros, sin importar en qué parte del mundo se encuentren. Podemos buscar recetas, publicarlas y conectarnos con otros usuarios.

Respecto a las comunidades que giran entorno a los procesadores de alimentos, están han ido generando en mucha medida por el tipo o marca, como es el claro ejemplo en:

- **Thermomix**, <https://www.recetario.es/>, es la comunidad que rodea al procesador Thermomix, donde encontramos recetas creadas por la marca y un foro que congrega a todos los usuarios que poseen un robot de cocina Thermomix.
- **Cecotec Mambo**, <https://cecotec.es/recetas-mambo/>, esta plataforma es creada por Cecotec, propietaria del procesador Mambo, aquí podemos ver las recetas que crea la propia marca, categorizadas y en cada detalle la preparación de la receta y comentarios que pueden dejar usuarios mediante el envío de un formulario sencillo sin registro previo.

5. Contenidos

A continuación se detallarán los contenidos de nuestra aplicación. En el enfoque del desarrollo, crearemos una librería de componentes propia, que utilizaremos a lo largo de toda la aplicación, donde tendremos un conjunto de elementos básicos para acelerar nuestro desarrollo, estos elementos serán comunes a todas las secciones, páginas y pantallas, y serán los siguientes:

- **Alerta:** muestra información contextual a sus usuarios. Puede utilizarse para proporcionar información a los usuarios, como mensajes de éxito o de error, pero también información destacada que complemente el flujo normal de los párrafos y los encabezados de una página.
- **Accordion:** es una colección de elementos de cabecera y cuerpo que se contraen verticalmente y que pueden utilizarse para mostrar y ocultar información.
- **Badge:** puede utilizarse para complementar otros elementos, como botones o elementos de texto, a modo de etiqueta, o para mostrar el recuento de un dato determinado.
- **Breadcrumb:** puede utilizarse para mostrar la ubicación actual de una página en una estructura jerárquica de páginas.
- **Buttons:** puede utilizarse para lanzar una acción pero también para enlazar con otras páginas.
- **Button group:** puede utilizarse para apilar varios botones y enlaces dentro de un mismo elemento.
- **Card:** se utilizará para mostrar entradas de datos e información a usuarios en múltiples formas y contextos, como por ejemplo, perfiles de usuario.
- **Carousel:** puede utilizarse para recorrer un conjunto de elementos utilizando opciones, controles e indicadores.
- **Dropdowns:** nos sirven para mostrar un menú al hacer clic en un elemento como un botón.
- **Forms:** los utilizaremos para recopilar información de los usuarios utilizando inputs, checkboxes, radiobuttons o text areas.
- **List Group:** puede utilizarse para mostrar una serie de elementos, botones o enlaces dentro de un único componente de tarjeta similar a una barra lateral.

- **Typography:** serán componentes reutilizables para categorizar nuestros tipos de texto, como encabezados o párrafos.
- **Modal:** diálogo interactivo del sitio web para mostrar notificaciones y recopilar información mediante elementos de formulario.
- **Tabs:** se puede utilizar como una jerarquía de navegación adicional que complementa la barra de navegación principal.
- **Navbar:** nos mostrará una barra común de navegación de la aplicación a las diferentes rutas o secciones.
- **Footer:** puede utilizarse como una forma de intentar convencer a los usuarios para que sigan navegando en la aplicación.
- **Sidebar:** elemento complementario a la barra de navegación que se muestra a la izquierda o a la derecha de la página utilizada para la navegación.
- **Pagination:** se puede utilizar para navegar a través de una serie de contenidos y conjuntos de datos para varias páginas.
- **Timeline:** nos ayudará a mostrar series de datos en orden cronológico para casos de uso como feeds de actividad.
- **Progress bar:** un indicador para mostrar la tasa de finalización de los conjuntos de datos.
- **Tables:** un conjunto de elementos formados por filas y columnas como celdas de tabla que pueden utilizarse para mostrar conjuntos de datos.
- **Spinner:** puede ser utilizado como un indicador de carga.
- **Toast:** se utilizará como componente para las notificaciones a los usuarios.
- **Tooltips:** mostrar contenido adicional al pasar el ratón por encima de un elemento.

Respecto a los contenidos principales, podemos citar las siguientes pantallas o rutas que compondrán las páginas:

- **Página de inicio:** será nuestra página inicial, donde encontraremos información promocionada de la aplicación. Secciones destacadas, que contendrán un listado de recetas con una imagen de las mismas y contadores de las interacciones sociales; un buscador principal donde podremos introducir una palabra para poder buscar una receta o un ingrediente y que nos dirija a una página de resultados. Listas de usuarios que participarán de forma activa, destacando a los usuarios que más publicaciones realizan. Listas de las últimas recetas publicadas. Y si el usuario está registrado y autenticado tendremos un listado de las últimas recetas que han subido las personas a las que sigue ordenadas por la fecha cronológica más reciente.
- **Página de recetas:** donde tendremos un listado de todas las recetas de la aplicación, ordenadas inicialmente por fecha más reciente de creación, y un buscador que nos permitirá buscar por receta.
- **Página de categoría:** donde tendremos un listado de recetas según la categoría seleccionada ordenadas por fecha de creación y las cinco más cocinadas.
- **Página de detalle de receta:** Aquí encontraremos la información relativa a la receta, ingredientes, dificultad, tiempo de preparación, raciones, pasos a seguir, imagen, autor de la receta con enlace a su perfil de usuario, acciones de interacción, como añadir a favoritos o guardar en una colección y bloque de comentarios. Las interacciones, los comentarios y los pasos para poder seguir la receta estarán disponibles únicamente si el usuario se encuentra autenticado. Si el usuario está autenticado y la receta ha sido publicada por él podrá editarla y se le redirigirá a una página de edición de la receta.
- **Página de creación de receta:** en esta página podrá crear una receta añadiendo todos los campos obligatorios, como la imagen, el nombre, el tiempo, los pasos, los ingredientes, el tipo de procesador de alimentos, la dificultad.
- **Página de edición de receta:** aquí el usuario registrado podrá editar cualquier de los campos creados en el registro. También podrá eliminar la publicación.
- **Página de perfil de usuario:** mostrará información pública del usuario, relativa a su perfil, como nombre de usuario, seguidores y número de personas seguidas. Así como el listado de las publicaciones que ha ido creando, con acceso al detalle de la misma, que redirigirá a la página de detalle de receta, anteriormente mencionada. Tendremos también la opción de seguir al usuario si

estamos autenticados.

- **Página de cuenta de usuario:** en este apartado tendremos varias secciones, para poder administrar la información del usuario registrado:
 - Información del perfil: que contendrá los datos personales editables, así como la posibilidad de poder eliminar su cuenta.
 - Colecciones de recetas: donde podrá crear una colección y añadir una receta. Podrá calcular el total de ingredientes de una colección.
 - Favoritos: recetas favoritas añadidas por el usuario.
 - Mensajes: donde encontraremos la lista de usuarios que nos han enviado un mensaje privado a nuestra cuenta.
- **Página de registro:** donde un usuario podrá registrarse completando un formulario que recogerá sus datos personales.
- **Página de inicio de sesión:** donde un usuario ya registrado podrá iniciar sesión en la aplicación y poder acceder a las opciones limitadas.

6. Metodología

Respecto a la forma de trabajar o metodología de trabajo, este proyecto de fin de Máster, se basará en la implementación de Agile o metodologías ágiles, que describiremos a continuación.

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.⁷

Ahora bien, las metodologías ágiles nos permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad en la respuesta para amoldar el proyecto a las circunstancias del entorno.

Las ventajas que nos proporciona este tipo de forma de trabajar, son:

- **Mejor calidad del producto:** se fomenta la integración, comprobación y mejora continua de las propiedades del producto.
- **Satisfacción del cliente:** mediante varias demostraciones y entregas, el cliente vive a tiempo real las mejoras introducidas en el proceso.
- **Motivación en el trabajo:** a través del trabajo autogestionado se facilita el desarrollo de la capacidad creativa y la innovación.
- **Trabajo colaborativo:** la división del trabajo, permite una mejor organización del mismo.
- **Utilización de métricas relevantes:** gracias a la división en pequeñas y fases podemos ser más conscientes de lo que está sucediendo.
- **Control y capacidad de predicción:** La oportunidad de revisar y adaptar el producto a lo largo del proceso nos permite ejercer un mayor control sobre el trabajo y esto permite mejorar la capacidad de predicción en tiempo y costes.
- **Reducción de costes:** se elimina prácticamente la posibilidad de fracaso absoluto en el proyecto, porque los errores se van identificando a lo largo del desarrollo.

⁷ Wikipedia, *Desarrollo ágil de software*, 2022

Existen diversas metodologías ágiles y en este proyecto utilizaremos dos para establecer el siguiente punto de esta memoria. Nos centraremos en:

- **Kanban:** es una estrategia que consiste en la elaboración de un cuadro o diagrama en el que se reflejan tres columnas de tareas; pendientes, en proceso o terminadas. En este cuadro estableceremos todas las tareas que integrarán las diversas fases del desarrollo. Con ello evitaremos la repetición de las tareas. Sus ventajas fundamentales son: planificación de las tareas, la mejora en el rendimiento del trabajo, métricas visuales y plazos de entrega continuos.
- **Scrum:** se basa en una estructura de desarrollo incremental o iterativo; cualquier ciclo de desarrollo del proyecto se desgrana en pequeños proyectos divididos en distintas etapas: análisis, desarrollo y testing. En la etapa de desarrollo encontramos las iteraciones o sprints, que son, entregas regulares y parciales del producto final. Scrum nos permite abordar proyectos complejos que exigen flexibilidad, es probable que podamos fallar en algún sprint, en este caso analizaremos el porqué y buscaremos otra solución flexible que nos permita adaptarnos a las circunstancias y no bloquear el flujo de trabajo.

En los dos próximos puntos (**planificación y proceso de trabajo**), daremos forma a los hitos y fases de desarrollo del proyecto, mediante el uso de estas dos técnicas.

Para ello catalogaremos las tareas en un **roadmap** u hoja de ruta, con plazos de entrega donde tendremos los siguientes procesos:

- **Epics:** son una forma de agrupar tareas y planificar a un nivel global, incluyendo las fases de análisis, desarrollo y pruebas.
- **Sprints:** son empujones con límite de tiempo para completar un conjunto de tareas, en las fases de desarrollo.
- **Tareas:** son las acciones que conforman los epics y los sprints

Las tareas, son los átomos que compondrán nuestros sprints en las fases de desarrollo y los epics a nivel más global, incluyendo otras fases que no conforman el desarrollo, como el análisis, las pruebas o la agrupación de bugs o errores o mantenimiento y/o soporte.

Tendrán como mínimo las siguientes propiedades:

- Título.
- Descripción.
- Fecha de creación.
- Tipo: Epic o tarea.
- Estado: Sin empezar, en progreso, completada, o bloqueada.
- Sprint: sprint al que pertenece.
- Prioridad: Alta, media, baja.
- Timeline: inicio y fin del periodo de entrega.

7. Planificación

La planificación del Trabajo Final de Máster, como comentábamos en el punto anterior está dividida en cuatro **epics**, cada **epic** corresponde a las diferentes PECs establecidas en el calendario académico del semestre, donde constan los **sprints** (si los hubiese) y el conjunto de tareas entregables y sus respectivas fechas.

7.1. Epics

Epic	Título	Timeline
PEC 1	Propuesta y definición formal del proyecto	16/02/2022 - 01/03/2022
PEC 2	Fundamentación y análisis del software	02/03/2022 - 30/03/2022
PEC 3	Desarrollo, POC y MVP	31/03/2022 - 08/05/2022
PEC 4	Entrega final del proyecto, documentación y publicación	09/05/2022 - 06/06/2022

Tabla 1: Resumen de entregas del Trabajo de Fin de Máster

7.2. Sprints y tareas

En la siguiente tabla se presentan los sprints de desarrollo, así como las tareas que no se engloban en dichos sprints, pero que sí forman parte de los **epics** anteriormente mencionados:

Epic	Título	Sprint	Timeline
PEC 1	Elección del tema	-	16/02/2022 - 17/02/2022
	Propuesta formal y elección de tecnologías	-	18/02/2022 - 18/02/2022
	Fundamentación teórica: introducción, descripción, objetivos, marco teórico	-	19/02/2022 - 23/02/2022
	Fundamentación teórica: contenidos, metodología, planificación, proceso de trabajo	-	24/02/2022 - 28/02/2022
PEC 2	Definición de la arquitectura y plataforma de desarrollo	-	02/03/2022 - 03/03/2022
	Diseño del modelo y entidades de datos	1	04/03/2022 - 05/03/2022
	Usabilidad: Conceptualización	2	06/03/2022 - 07/03/2022
	Usabilidad: diseño de la experiencia	3	08/03/2022 - 09/03/2022
	Usabilidad: diseño de la interacción	4	10/03/2022 - 12/03/2022
	Desarrollo: creación de repositorios de la aplicación y librería de componentes, primeros componentes base.	5	13/03/2022 - 14/03/2022
	Despliegues: Entorno de desarrollo e integración continua	5	13/03/2022 - 14/03/2022

	Librería: Desarrollo completo de librería de componentes	6	15/03/2022 - 23/03/2022
	Librería: Refactorización de la librería de componentes y revisión de cobertura de pruebas unitarias	6	24/03/2022 - 27/03/2022
	Documentación de la memoria del trabajo	-	28/03/2022 - 30/03/2022
PEC 3	Backend: Autenticación y registro de usuarios	7	31/03/2022 - 02/04/2022
	Backend: Creación de capa API Rest y servicios crud	8	03/04/2022 - 06/04/2022
	Backend: Revisión y primera refactorización	9	07/04/2022 - 09/04/2022
	Frontend: Estructuración inicial y primeras integraciones con backend	10	10/04/2022 - 13/04/2022
	Frontend: Estructuración de rutas vistas y guards	11	14/04/2022 - 15/04/2022
	Frontend: integración de funcionalidades básicas	12	16/04/2022 - 23/04/2022
	Frontend: refactorización y desarrollo completo de las funcionalidades	13	24/04/2022 - 30/04/2022
	Desarrollo: Revisión y refactorización de Frontend y Backend	14	01/05/2022 - 04/05/2022
	MVP: Despliegue y presentación	15	05/05/2022 - 08/05/2022
	Documentación de la memoria del trabajo	-	05/05/2022 - 08/05/2022
PEC 4	Deuda técnica y refactorización	16	09/05/2022 - 15/05/2022
	Revisión de cobertura y testing	16	16/05/2022 - 20/05/2022
	Documentación de la memoria del trabajo	-	21/05/2022 - 25/05/2022
	Presentación académica	-	26/05/2022 - 28/05/2022
	Presentación pública	-	29/05/2022 - 31/05/2022
	Video de presentación	-	01/06/2022 - 03/06/2022
	Autoinforme de evaluación	-	04/06/2022 - 06/06/2022

Tabla 2: Sprints, tareas y planificación del Trabajo de fin de Máster

7.3. Diagrama de Gantt

En el siguiente diagrama de Gantt se muestra la planificación mencionada en el punto anterior.

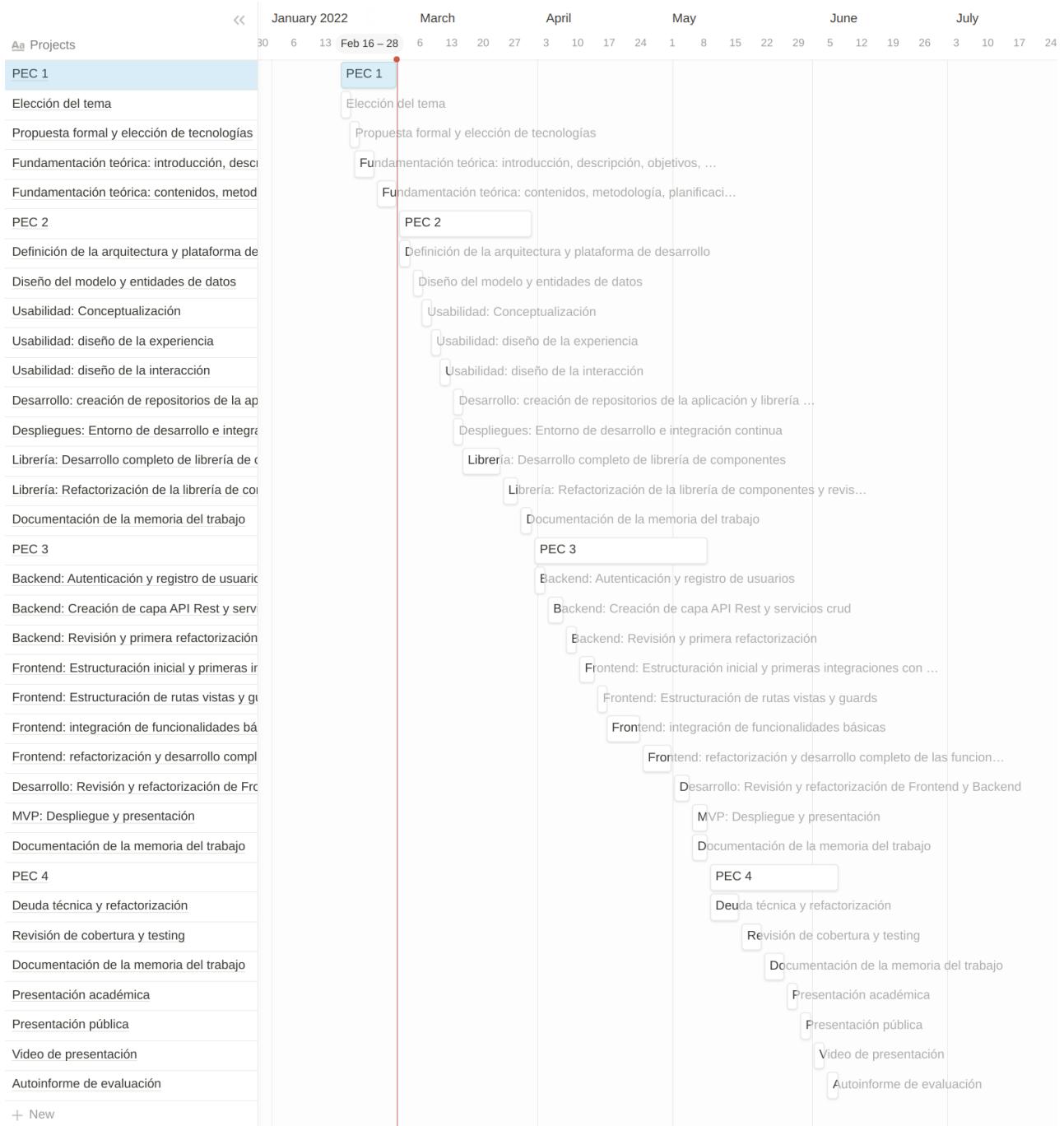


Figura 1: Diagrama de Gantt - Planificación

8. Proceso de trabajo

8.1 MVP⁸

El producto mínimo viable será una aplicación web que contendrá las siguientes funcionalidades:

- Autenticación de usuarios.
- Registro de usuarios.
- Edición y eliminación del usuario previamente registrado.
- Navegación entre las diferentes vistas.
- Creación, edición y eliminación de recetas de los usuarios.
- Búsqueda por nombre, ingrediente o categoría de la receta.
- Interacciones sociales “like” sobre la receta de un usuario.
- Comentarios sobre una receta.
- Seguir y dejar de seguir a un usuario.
- Crear colecciones de recetas.
- Calcular el total de ingredientes de las recetas añadidas previamente a una colección.
- Mensaje privado a un usuario.

8.2. Sprints

Las fases del proceso de desarrollo estarán contenidas por los dieciséis sprints, mencionados en la planificación:

1. Diseño del modelo y entidades de datos.
2. Conceptualización
3. Diseño de la experiencia
4. Diseño de la interacción
5. Creación de repositorios de la aplicación y librería de componentes, primeros componentes base.
Entorno de desarrollo e integración continua.
6. Desarrollo completo de librería de componentes. Refactorización de la librería de componentes y revisión de cobertura de pruebas unitarias.
7. Autenticación y registro de usuarios
8. Creación de capa API Rest y servicios crud
9. Revisión y primera refactorización del backend
10. Estructuración inicial del Frontend y primeras integraciones con backend

⁸Producto mínimo viable

11. Estructuración de rutas vistas y guards.
12. Integración de funcionalidades básicas del Frontend.
13. Refactorización y desarrollo completo de las funcionalidades del Frontend.
14. Revisión y refactorización de Frontend y Backend.
15. MVP: Despliegue y presentación.
16. Deuda técnica y refactorización. Revisión de cobertura y testing.

9. Arquitectura de la aplicación

En este capítulo justificamos la arquitectura de la aplicación atendiendo a dos puntos muy importantes, los **patrones de diseño** y las **tecnologías** que utilizaremos para llevar a cabo nuestro desarrollo.

Es fundamental sentar las bases de nuestro trabajo, bajo un patrón que nos permita organizar el trabajo, y con el que podamos medir el futuro del mismo. Esto nos ahorrará quebraderos de cabeza a la hora de plantear migraciones o cambios de requisitos imprevistos en la fase de análisis.

9.1. Patrones de diseño

En este punto trataremos de definir y resumir la adopción de un patrón de diseño para nuestra aplicación. Como ya se menciona en **Patterns.dev** podemos resumir qué es un patrón de diseño con el siguiente concepto explicado en dicho sitio:

Los patrones de diseño son una parte fundamental del desarrollo de software, ya que proporcionan soluciones típicas a problemas comúnmente recurrentes en el diseño de software. Más que proporcionar piezas de software específicas, los patrones de diseño son simplemente conceptos que pueden utilizarse para manejar temas recurrentes de forma optimizada.⁹

Si bien existen infinidad de patrones, casi como problemas a resolver, existe uno que nos permite encapsular la lógica de negocio, aislandola de dependencias externas. Estamos hablando del patrón **Clean Architecture**.

Realmente no es un patrón, sino más bien un conjunto de principios, como explica Jorge Sánchez Fernández en su artículo acerca de Clean Architecture:

[...] cuya finalidad principal es ocultar los detalles de implementación a la lógica de dominio de la aplicación. De esta manera mantenemos aislada la lógica, consiguiendo tener una lógica mucho más mantenible y escalable en el tiempo.¹⁰

⁹ A. Osmani. *Introduction*, Patterns.dev, 2022.

¹⁰ J.S. Fernández. *¿Por qué utilizo Clean Architecture?*, XurxoDev, 2016.

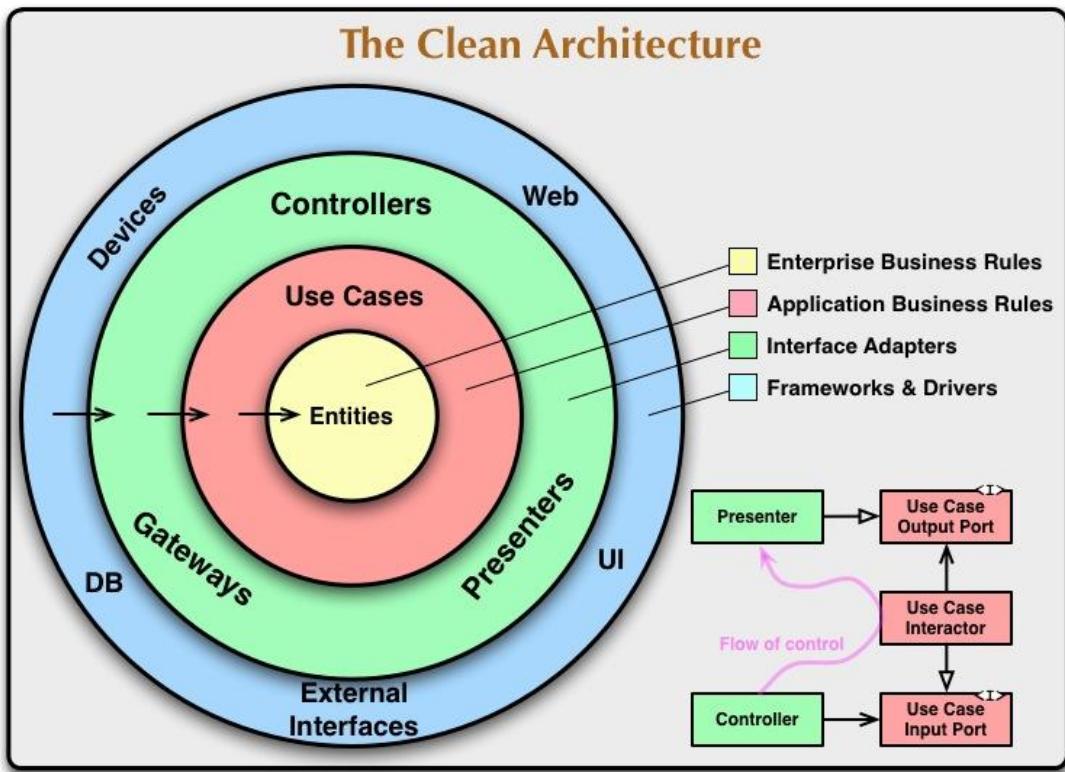


Figura 2: Clean Architecture

A modo de resumen podemos hablar de las siguientes capas en **Clean Architecture**¹¹:

- **Entidades**: donde se encuentra la lógica de negocio empresarial que existiría aunque no tengamos una aplicación para automatizar procesos. Por ejemplo, para realizar un pedido es necesario que deba existir un cliente con nombre, apellidos, NIF y una dirección.
- **Casos de uso**: representan la lógica de aplicación, que existe principalmente debido a la automatización de procesos mediante la aplicación y es inherente a cada aplicación. Por ejemplo para crear un pedido, el caso de uso entre la aplicación móvil y la API tiene sentido que sean diferentes.
- **Adaptadores**: se van a encargar de transformar la información como se entiende y es representada en los detalles de implementación o frameworks, drivers a como la entiende el dominio (entidades y casos de uso). Es habitual utilizar en este punto junto con clean architecture diferentes patrones de presentación como MVC, MVP, MVVM, donde los presenters, controllers, views, serían los

¹¹R. C. Martin. *Clean Architecture*, The Clean Code Blog, 2022.

adaptadores encargados de transformar la información de las vistas a información que necesitan los casos de uso.

- Frameworks y drivers: serían los detalles de implementación, todos aquellos frameworks, librerías que se suelen utilizar en una aplicación para mostrar o almacenar información (como por ejemplo React, Vue, etc...).

Los círculos de la *Figura[2]*, no pueden depender de los exteriores, pero los exteriores tienen que poder comunicarse entre ellos. Para ello utilizaremos el *Principio de Inversión de Dependencia (Figura[3])*.

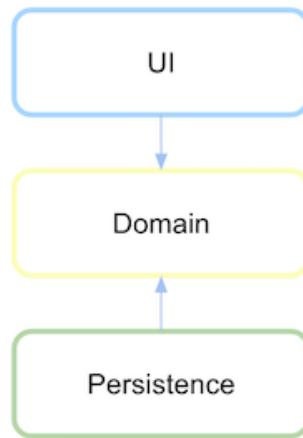


Figura 3: Principio de Inversión de Dependencia

Crearemos en la capa de casos de uso, las abstracciones que deben implementar los adaptadores. De esta forma los adaptadores dependen del dominio pero este no depende de ellos. Esto consigue que el dominio quede aislado y pueda ser sencillo reemplazar adaptadores. Y realmente podemos alejarnos de las librerías y frameworks como Angular, Vue, React, etc, que cambian con el tiempo.

9.2. Tecnologías

En este punto se definirá la arquitectura respecto a las tecnologías, donde se detalla la comunicación entre las mismas y el papel que tienen en los diferentes procesos.

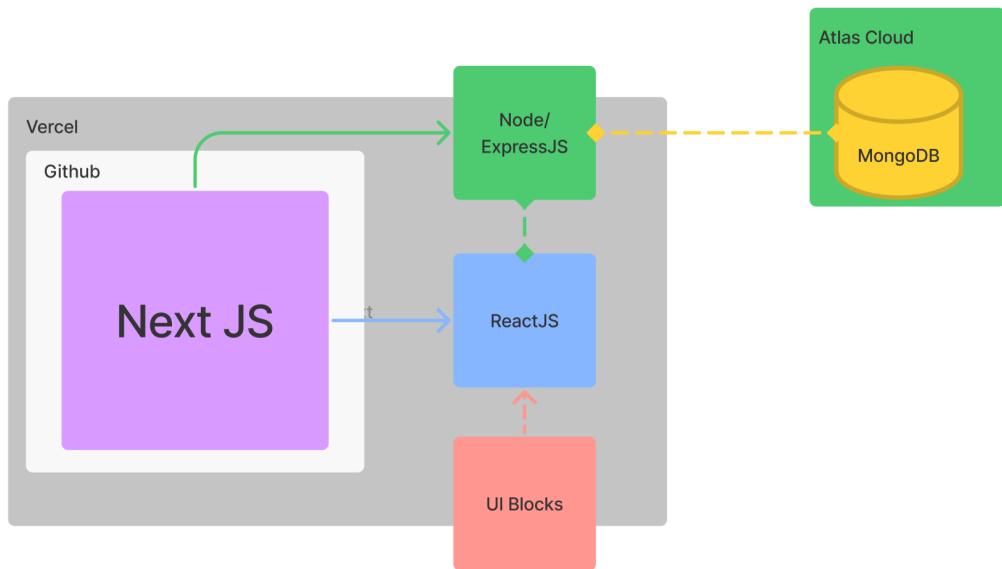


Figura 4: Arquitectura y herramientas

Para poder llevar a cabo el desarrollo completo de este proyecto, utilizaremos las herramientas mostradas en la *Figura[4]*:



Figura 5: Vercel

- **Vercel:** plataforma que nos permite desplegar aplicaciones orientadas a proyectos Frontend. Tenemos plena integración con Github y podemos configurar ramas para el despliegue automático tanto para producción como para versionado.



Figura 6: Github

- **Github:** plataforma donde albergamos los diferentes repositorios y que conectaremos con **Vercel**

para poder desplegar nuestro producto. Utilizamos Git y Gitflow como control de versionado del código.



Figura 7: Nextjs

- **Nextjs:** es el core de nuestra aplicación, Nextjs nos proporciona un ecosistema fullstack para el desarrollo frontend y backend. Por un lado tendremos las diferentes pantallas que se desarrollaran bajo React en la parte frontend; y por otro lado tendremos todo el sistema de servicios API Rest, que se desarrolla internamente bajo Expressjs en Node dentro del propio Nextjs. Esto nos permite ahorrar costes para el almacenamiento de nuestro proyecto y desarrollar ágilmente toda la lógica fullstack bajo esta herramienta, conocida ya como The Web SDK.

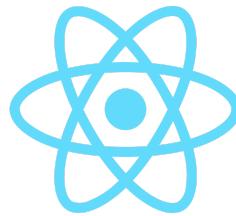


Figura 8: React

- **Librería de componentes:** desarrollaremos una librería aislada de componentes , en **React**, para nuestra aplicación, que publicaremos en **npm**, y consumiremos, desplegando para comprobar su desarrollo e interacciones en **Vercel**, mediante **Storybook**. Así tendremos el core de la aplicación por un lado, y los módulos reutilizables por otro.



Figura 9: Atlas MongoDB

- **Atlas MongoDB:** para la persistencia de los datos de nuestra aplicación utilizaremos MongoDB y para servir la base de datos utilizaremos Atlas Cloud como servicio, y así no tener servidores propios y perder tiempo en configuraciones innecesarias.

9.3 Arquitectura de la base de datos

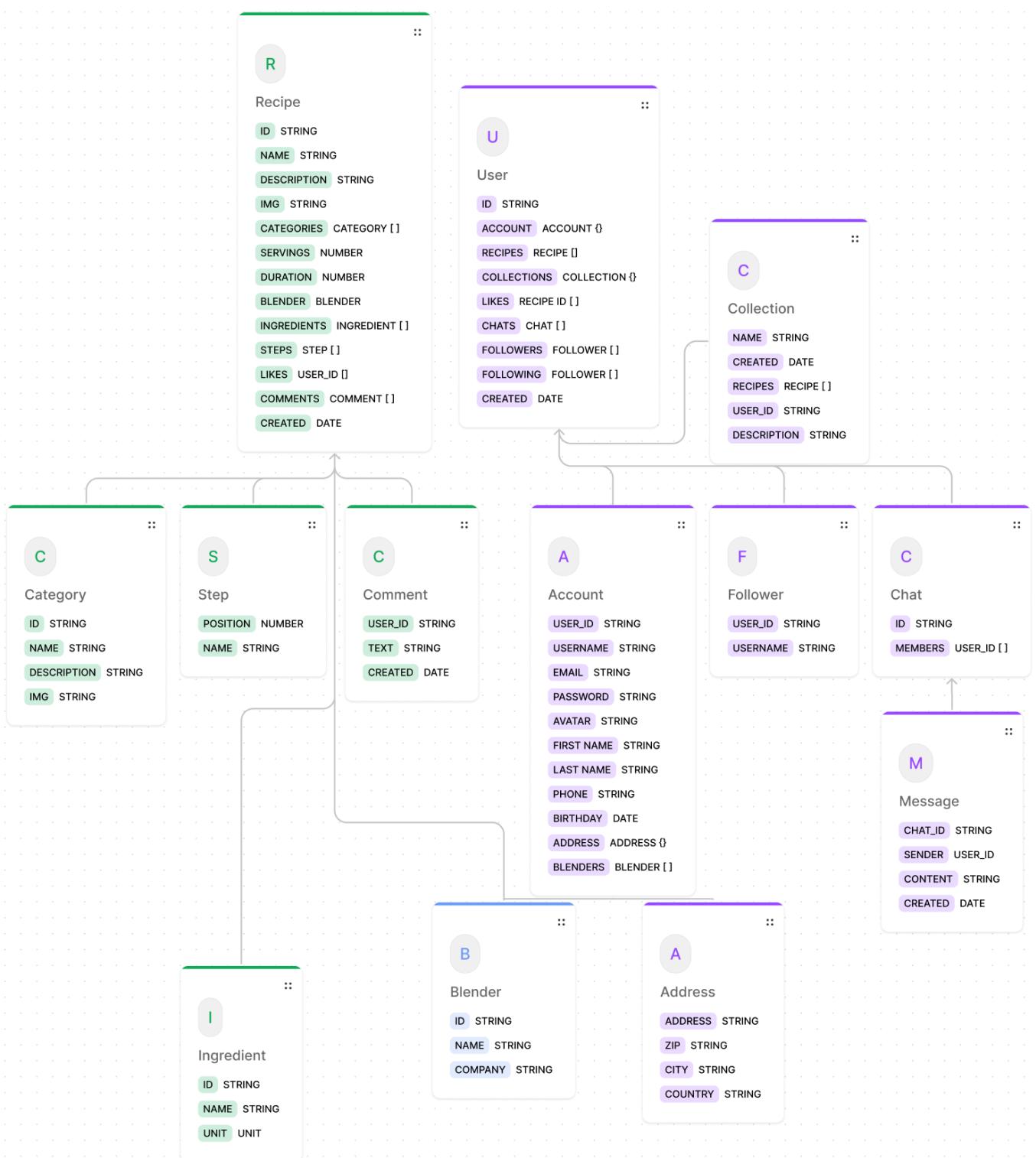


Figura 10: Arquitectura de la base de datos

10. Plataforma de desarrollo

10.1 Entorno de desarrollo

- **Ubuntu:** como sistema operativo utilizaremos Linux bajo la distro Ubuntu, que está basada en Debian. Es de las distros más utilizadas de Linux por su bajo nivel de complejidad a la hora de configuraciones y nos permitirá ahorrar recursos y tiempo.
- **VSCode:** editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, mediante la instalación de plugins, lo que lo convierte en casi un IDE. Es gratuito y de código abierto.
- **Git y Gitflow:** **Git** es un software de control de versiones de aplicaciones. Su propósito es llevar registro de los cambios en los diferentes archivos del proyecto, incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

Gitflow es un modelo alternativo de creación de ramas en Git en el que se utilizan ramas de función y varias ramas principales, así podremos ir realizando entregas mediante features a una rama de desarrollo y realizar las publicaciones a la rama principal por medio de release y organizar de forma eficiente los desarrollos.
- **npm:** ya que nuestro proyecto es full stack Javascript, utilizaremos **npm** como sistema de gestión de paquetes por defecto, para la instalación de todas las librerías que necesitaremos en nuestro proyecto.
- **Postman:** utilizaremos esta herramienta para hacer peticiones a APIs y generar colecciones de peticiones que nos permitan probarlas de una manera rápida y sencilla.
- **Robo 3T:** herramienta de administración de MongoDB multiplataforma, de código abierto y gratuita con un shell mongo integrado, soporte real de autocompletado y una gran cantidad de funciones que cualquiera puede usar para administrar las bases de datos de MongoDB.

10.2 Hardware

Para llevar a cabo el desarrollo de este proyecto se ha utilizado el siguiente hardware que conforma el equipo de desarrollo:

- Corsair iCUE 4000X RGB Cristal Templado USB 3.0 Blanca, como caja.
- MSI MAG Core Liquid 240R V2 Kit, como refrigeración principal del procesador.
- Nfortec Alcyon 512GB SSD M.2 NVMe, como disco duro principal.
- Gigabyte B660 GAMING X DDR4, como placa base.
- Corsair RMx White Series RM750x 750W 80 Plus Gold Full Modular, como fuente de alimentación.
- Corsair Vengeance RGB Pro DDR4 3200 PC4-25600 32GB 2x16GB CL16, como memoria RAM.
- Intel Core i5-12400 4.4 GHz, como procesador.
- LG 27UL550-W 27" LED IPS Ultra HD 4K FreeSync x2, como monitores.
- Keychron K6, como teclado.
- Logitech Pro Wireless Mouse, como ratón.

11. Estudio de viabilidad

Llegados a este punto realizaremos un análisis del mercado basado en el estudio publicado por Silestone Institute, Global Kitchen, *The home Kitchen in the globalization era*¹²

- **La función de la cocina en el hogar.**

Casi nueve de cada diez profesionales (87%) confirman la tendencia de que la cocina va a adquirir más importancia en el hogar, frente a un 9% que cree que la cocina conservará la misma importancia y un 3,5% que considera que otras estancias que otras estancias serán más importantes.

Esta opinión es más pronunciada en Brasil, Portugal y Reino Unido (por encima del 90%), mientras que Australia parece más reacia a reconocer este nuevo papel (74%).

Debido a la creciente relevancia de la cocina en el hogar, el 82,8% de los profesionales creen que el papel de esta estancia debería tener más importancia en la forma de planificar y construir los espacios colectivos de la vivienda.

- **¿Qué actividades tendrán lugar en las cocinas en el futuro?**

La cocina se consolidará en los próximos años como espacio de encuentro y ocio para los miembros de la familia. Además de la manipulación de alimentos y la cocción, la cocina será el centro de otras actividades que tradicionalmente se han llevado a cabo en otras zonas del hogar.

Actividades como, estar con nuestros amigos o familiares, comer, ver la televisión, navegar en internet o trabajar o estudiar, están empezando a formar parte de nuestros hábitos en esta estancia de nuestro hogar.

¹² G.Kitchen, *The home kitchen in the globalization era*, Silestone, 2019

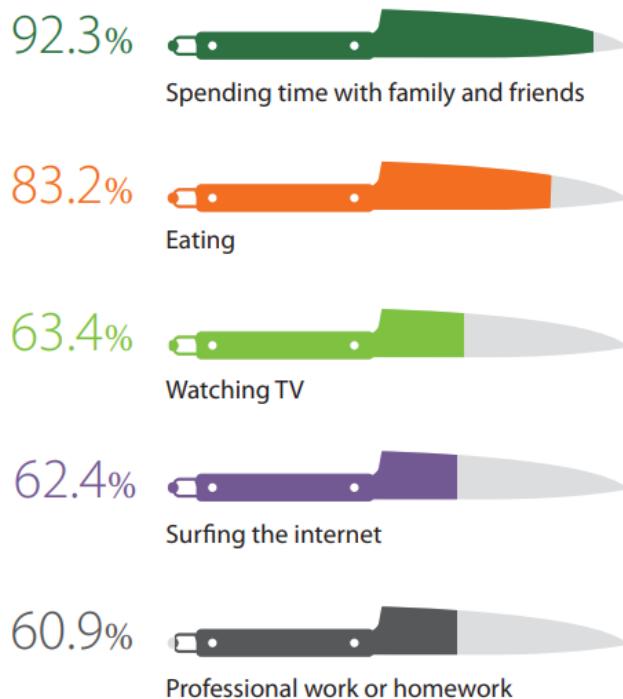


Figura 11: Las actividades en la cocina del futuro.

- **¿Cuáles son los principales cambios tecnológicos que afectarán a la cocina en el futuro?**

La conexión de la cocina a Internet y a los dispositivos y los electrodomésticos inteligentes destacan como las principales innovaciones tecnológicas a corto y medio plazo. Según los profesionales del sector, las encimeras del futuro deberían permitir a los usuarios cocinar directamente en la superficie, además de incorporar conectividad y actuar como un panel de control.

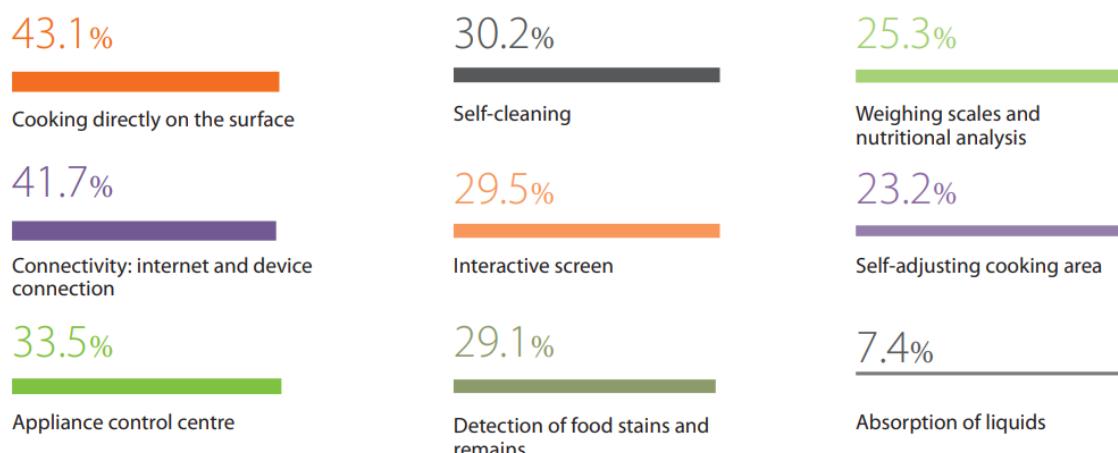


Figura 12: Cambios tecnológicos en las cocinas.

Bajo las premisas anteriormente expuestas, parece plausible, que la interconexión entre cocinas o usuarios que las manejan se digitalizará y que realizaremos en ellas otro tipo de actividades que además de incluir la cocina, nos permitirán socializar de alguna manera, tal es así que podremos seguramente, intercambiar opiniones con personas bien de forma remota (mediante alguna plataforma) o físicamente.

Como ya comentamos en el **Marco Teórico** del presente trabajo, existen diversas alternativas que si bien en su conjunto, cubren la necesidad de tener una comunidad orientada a los robots de cocina, no lo hacen de forma aislada.

Si bien **Cookpad** o **Funcook** nos permiten intercambiar recetas entre usuarios, utilizar estas plataformas orientadas a procesadores puede suponer una gran pérdida de tiempo para encontrar el contenido deseado. Esto es suplido por los foros y plataformas de las marcas de cada robot, pero obviamente no recogen las recetas de la competencia.

Podemos afirmar que el desarrollo de una red social de cocina, orientada a procesadores de alimentos es viable, centrándolo su foco objetivo en unir esa carencia en la información de las recetas de procesadores de alimentos y minimizar el gasto en tiempo para los usuarios que utilicen la plataforma.

12. Diseño y usabilidad

12.1 Conceptualización

Según el estudio Global Kitchen, la cocina se perfila como un espacio multifuncional e hiperconectado, en el que aparatos inteligentes no sólo facilitarán el cocinado sino también la organización y compra de alimentos en el hogar.

Si bien el cambio tecnológico de cocineros robots, hornos inteligentes, encimeras donde ver vídeos y proyectar recetas está aún por llegar. Lo que sí es ya es hoy una realidad es el uso de las redes sociales para la cocina.

Como sucede en muchos otros ámbitos de la vida cotidiana, las redes sociales se utilizan habitualmente para recabar información y compartir experiencias sobre nuevos platos e ingredientes, conocer nuevas recetas o introducir cambios en la dieta.

Las redes sociales no solo nos conectan con aquellos familiares y amigos que hoy, debido al confinamiento, los tenemos a la distancia, sino que también nos ofrecen soluciones y novedades para variar en comidas ricas, originales y dignas de cualquier chef.

En este contexto, tendremos por un lado usuarios habituados al uso de las redes sociales, consumidores y creadores de contenido, familiarizados con algún procesador de alimentos y otro tipo de dispositivos electrónicos y tecnológicos. Y por otro lado tendremos personas con aficiones o gustos orientados a la gastronomía, críticos de cocina o profesionales de la misma.

Para visualizar de una manera más clara las necesidades y expectativas del público objetivo, a continuación definiremos las siguientes **User Personas**:

	Bio
Ana Sánchez Edad: 28 años Ocupación: Youtuber Localidad: Barcelona Educación: Universidad	<p>Ana es una joven barcelonesa apasionada por la cocina, trabaja como influencer mediante la plataforma Youtube, a través de la cual distribuye contenido acerca de consejos para la cocina, promociona productos recomendados y ofrece a sus suscriptores listas de recetas de cocina.</p> <p>Está siempre conectada ya sea por medio del teléfono o el ordenador, y gracias al patrocinio de varias marcas recibe nuevas versiones de diferentes robots de cocina para hacer reviews de los mismos y compartir con sus suscriptores la experiencia de uso de estos.</p> <p>Le encanta categorizar todo su contenido para facilitar su posicionamiento y así garantizar a sus seguidores que encuentran aquello que buscan en su canal.</p>
Comportamiento	<ul style="list-style-type: none"> ● Utiliza las herramientas digitales de forma plena ● Es detallista y procura ofrecer una información real y sin sesgo en su contenido. ● Busca la sencillez en su forma de comunicar e informar ● Comparte mucho contenido visual ● Está actualizada con cada novedad tecnológica
Necesidades	<ul style="list-style-type: none"> ● Acceso intuitivo y moderno. ● Encontrar de forma sencilla lo que busca. ● Compartir el contenido que crea. ● Recibir buenas valoraciones de sus creaciones.
Pain points	<ul style="list-style-type: none"> ● Categorización ineficaz. ● Mapa web demasiado amplio. ● Aplicación web no adaptada a dispositivos móviles. ● Capacidad para compartir según tendencia de redes compleja.

Tabla 3: User persona - Ana Sánchez

 <p>Carlos Hernández</p> <p>Edad: 33 años</p> <p>Ocupación: Periodista</p> <p>Localidad: Vigo</p> <p>Educación: Universidad</p>	<p>Bio</p> <p>Carlos es un joven vigués, amante de la cocina, de los buenos productos gastronómicos y de salir a comer. Le encanta hacerle fotos a todo lo que come y cocina y sobre todo compartirlas con la gente. Es un “cocinillas”, siempre sorprende a sus amigos en las cenas con alguna receta nueva.</p> <p>Le encanta la tecnología, ha empezado a domotizar su casa recientemente y la cocina ha recibido un nuevo procesador de alimentos. Pasa muchas horas delante de las pantallas buscando ingredientes y recetas exóticas.</p> <p>Cocina de forma tradicional, pero gracias a tener un nuevo procesador de alimentos está mejorando su forma en la que cocina y sus conocidos no paran de repetirlo.</p> <p>Suele participar en foros, seguir a influencers del mundo de la cocina y a prestigiosos chefs</p>
	<ul style="list-style-type: none"> • Busca recetas en todo tipo de plataformas. • Atiende al detalle y sabe que es la clave del éxito. • Está siempre actualizado tecnológicamente. • Sus aficiones son el vehículo para socializar.
	<ul style="list-style-type: none"> • Acceso optimizado desde dispositivos móviles. • Encontrar la información de manera precisa. • Compartir el contenido que crea. • Buscar personas afines a sus gustos.
	<ul style="list-style-type: none"> • Demasiada información. • Mapa web demasiado amplio. • Interacciones complejas.

Tabla 4: User persona - Carlos Hernández

Las redes sociales, son nuestro principal punto de entrada. A través de ellas penetramos en nuestro público objetivo. Cómo hacerlo, a través de anuncios publicitarios, influencers a los que patrocinamos o cualquier

otra campaña en redes sociales que mueva a nuestro principal público objetivo. Como principal estrategia para el **User Engagement**:

- Cuidaremos la estética, que promoverá la atención focalizada y mantendrá una asociación de imagen de marca, en nuestro sitio y en cualquier otro medio publicitario. Tendremos en cuenta principios de simetría, equilibrio, prominencia y sencillez.
- Promoveremos la navegación para retener al usuario, creando una comunidad que permita a los usuarios construir, y recibir de otros usuarios aportaciones.
- En cuanto a la usabilidad, haremos un espacio simple, dinámico y sencillo donde los pain points de los usuarios como Ana y Carlos, se reduzcan a la mínima expresión. Evitando la sobreinformación, y un mapa web muy amplio.

12.2 Diseño de la experiencia

12.2.1 Requisitos

Categoría	Requisito Funcional
Registro	Registro de usuario mediante correo y contraseña.
Inicio de sesión	Autenticación de usuario mediante correo y contraseña.
Home	Visualización de página de inicio con recetas y categorías.
Home	Selección de contenido para cualquier tipo de usuario.
Home	Búsqueda de receta según palabra para cualquier usuario.
Home	Visualización listado de recetas de usuarios seguidos para usuarios autenticados.
Recetas	Visualización listado de recetas ordenadas por fecha de creación.
Recetas	Visualización listado de las recetas más cocinadas.
Categoría	Posibilidad de ver las recetas pertenecientes a una categoría por todos los usuarios
Detalle Receta	Visualización de información de la receta para todos los usuarios.
Detalle Receta	Visualización de pasos a seguir para usuarios autenticados.
Detalle Receta	Interacción social, like, comentario y guardar en colecciones para usuarios autenticados.
Editar Receta	Posibilidad de modificar una receta o eliminarla por un usuario autenticado al

	que pertenezca.
Añadir Receta	Creación de receta por un usuario autenticado mediante un formulario.
Perfil	Visualización del contenido creado por todos los usuarios.
Perfil	Acceso a edición de perfil por el propio usuario autenticado.
Perfil	Posibilidad de seguir o dejar de seguir o enviar mensaje privado por parte de otro usuario autenticado.
Cuenta / Editar perfil	Posibilidad de modificar los datos del perfil o su eliminación por un usuario autenticado
Cuenta / Colecciones	Visualización de colecciones de recetas agregadas por un usuario autenticado.
Cuenta / Colecciones	Visualización del total de ingredientes de una colección
Cuenta / Colecciones	Edición o eliminación de una colección de recetas
Cuenta / Favoritos	Visualización de recetas añadidas a favoritos
Cuenta / Favoritos	Eliminación de recetas de favoritos
Cuenta / Mensajes	Visualización de conversaciones con otros usuarios, por un usuario autenticado
Cuenta / Mensajes	Creación de mensaje en una conversación con otro usuario, por un usuario autenticado.
Cuenta / Mensajes	Eliminación de una conversación por parte de un usuario autenticado.

Tabla 5: Requisitos Funcionales

Categoría	Requisitos no funcionales
Frontend	El frontend debe estar desarrollado en React 17 bajo el Framework Nextjs 12.1
Frontend	Componentes UI en React desarrollados en una librería externa y visibles en Storybook
Frontend	La aplicación deberá ser responsive
Frontend	La aplicación deberá ser compatible con la mayoría de navegadores.
Backend	El backend debe de estar desarrollado con Express en Nextjs 12.1
Backend	Backend protegido con JWT
Base de datos	La base de datos de ser en MongoDB desplegada en Atlas Cloud
Servicios	Comunicaciones entre frontend y backend serán ser HTTPS
Servicios	Comunicaciones entre frontend y backend en formato JSON

Tabla 6: Requisitos no funcionales

12.2.2 Sitemap y user flows

En este subapartado se recoge el mapa de navegación de nuestra aplicación en un sentido amplio, es decir, para usuarios autenticados y también no autenticados, que son los que podrían acceder a todas las vistas disponibles.

Posteriormente detallaremos en los user flows los accesos a los procesos que tienen los usuarios autenticados y no autenticados, así como la creación, edición o eliminación de una receta, que son las funcionalidades básicas de la aplicación.

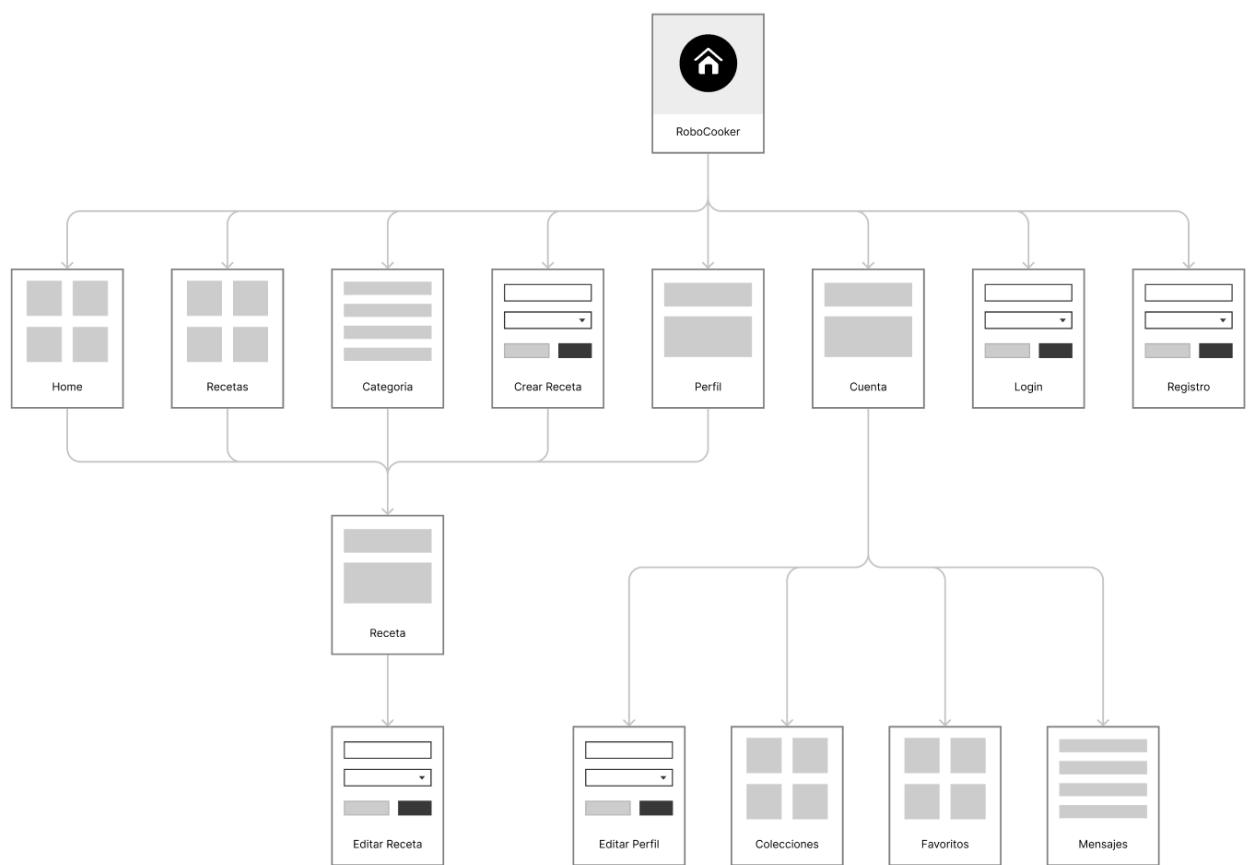


Figura 13: Sitemap

Como podemos ver accederemos a una vista Home y podremos navegar entre las diferentes vistas, Recetas, Categoría, Crear Receta o Perfil, ambas nos conducen a la vista en detalle de una receta seleccionada y su posible edición o eliminación en otra vista independiente.

En el apartado cuenta podremos editar el perfil del usuario, ver las colecciones que haya generado el usuario o los favoritos añadidos, así como los mensajes. Si el usuario no está autenticado podremos acceder a las vistas de Inicio de sesión o registro.

Un usuario no autenticado podrá acceder a la plataforma y ver las recetas, categorías y el perfil del usuario que ha creado la receta:

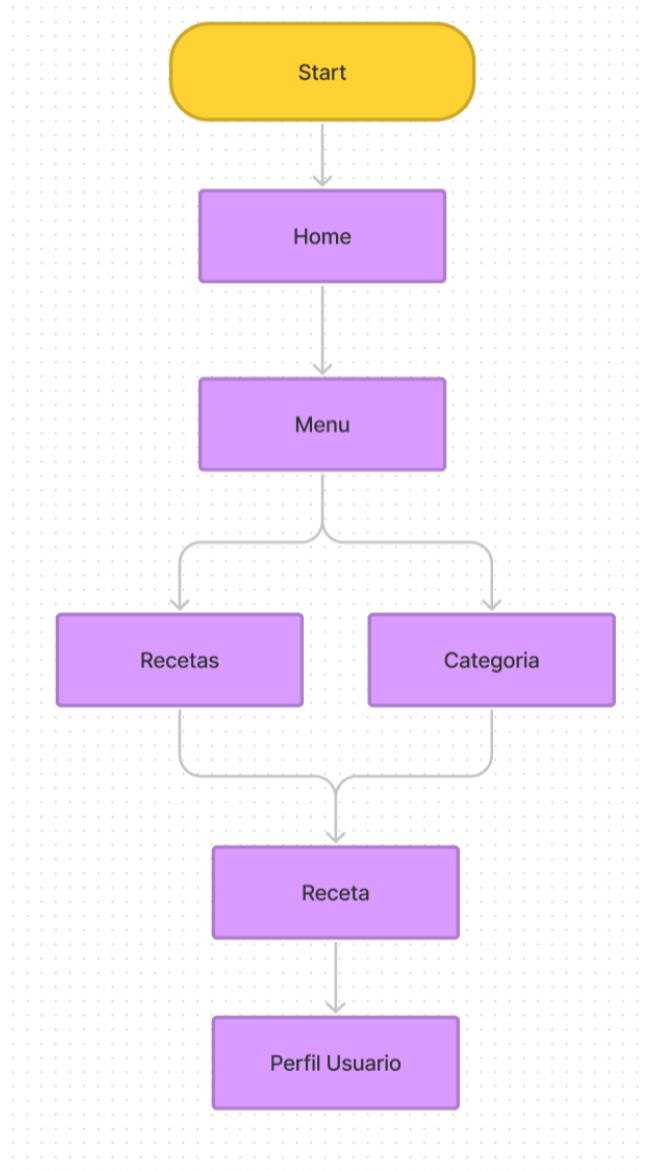


Figura 14: Flujo de interacción sin autenticación

En la siguiente figura podremos ver todas las acciones que puede realizar un usuario autenticado en la plataforma, incluyen las de la figura anterior, y además podrá salir de la sesión, acceder a su cuenta, colecciones, favoritos, mensajes, y editar su perfil. Podrá también comentar una receta, darle a like o unlike, y añadir a su colección la receta. Si dicha receta la ha creado él, podrá además editarla o eliminarla.

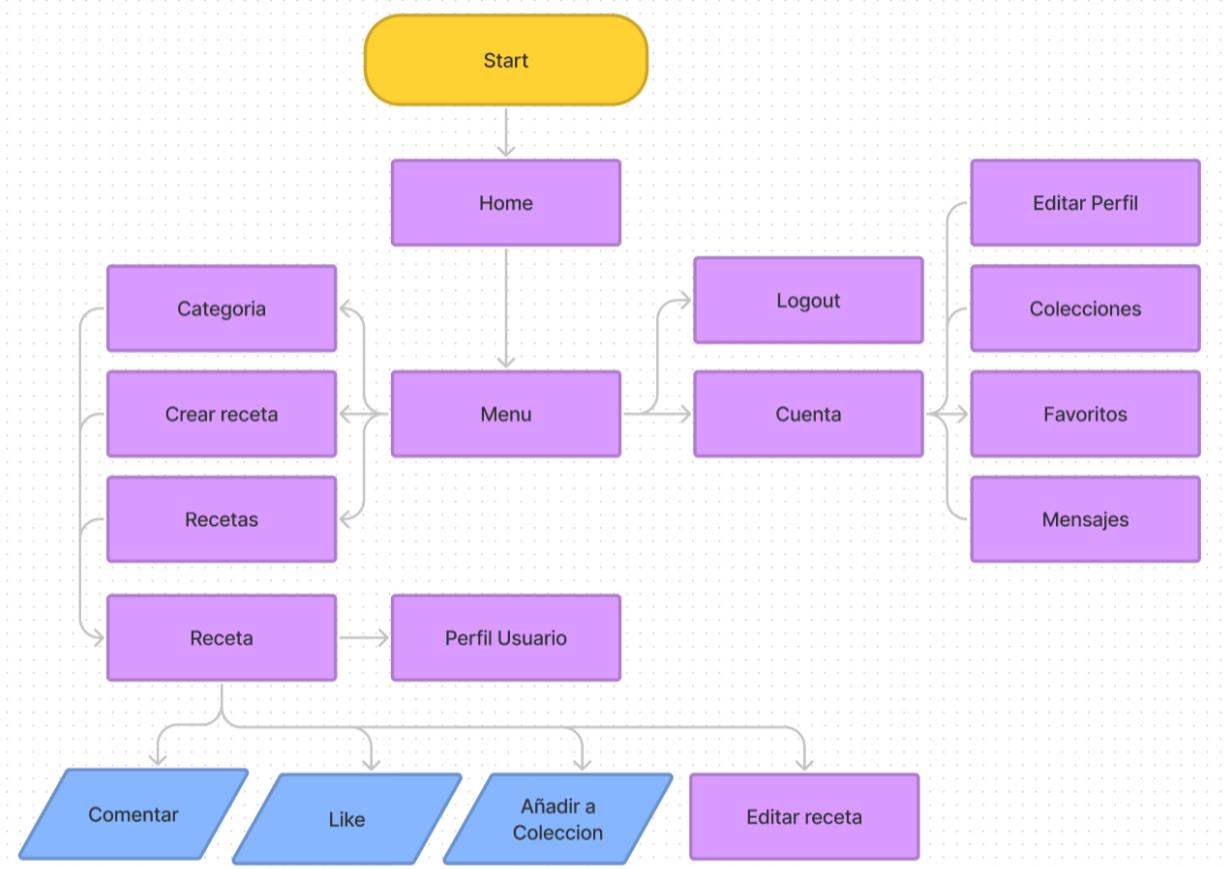


Figura 15: Flujo de interacción con autenticación

En el siguiente diagrama podemos ver como un usuario **creará una receta**, la aplicación comprobará si está logueado o no, y le permitirá realizar entonces la acción, para los siguientes casos, presentaremos al usuario como ya autenticado en la aplicación para simplificar un poco más el contexto.

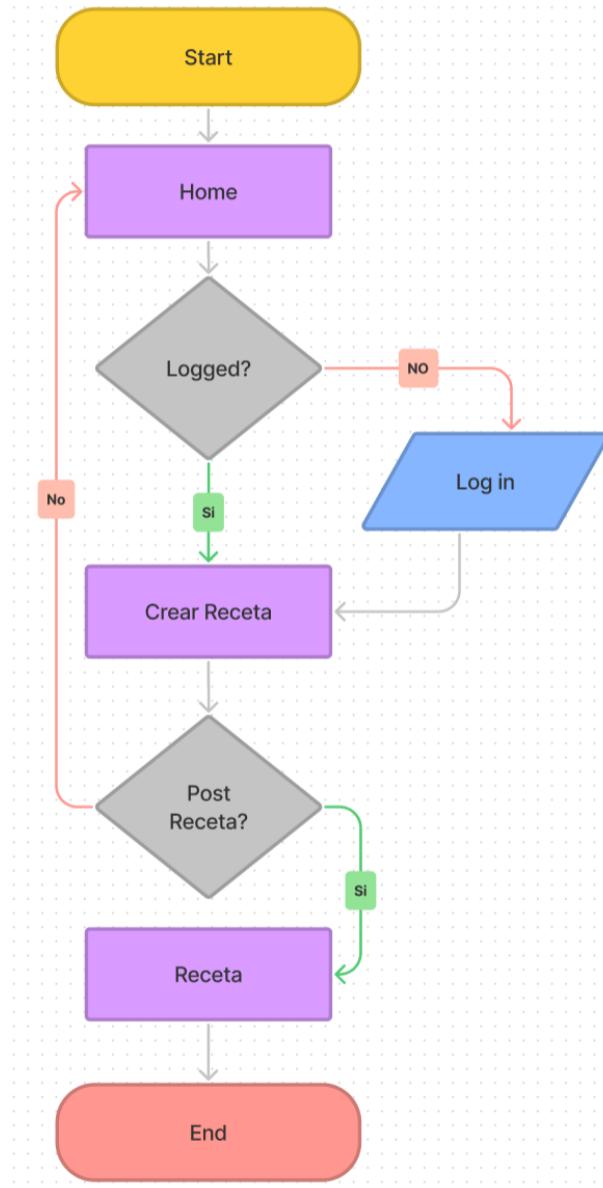


Figura 16: Flujo de creación de una receta

Un usuario autenticado podrá también **editar o eliminar una receta** desde la vista de Editar Receta. En el supuesto tambien de **añadir una colección, editarla o eliminarla** será muy similar y podrá hacerlo en la subvista Colecciones, como queda reflejado en las siguientes figuras:

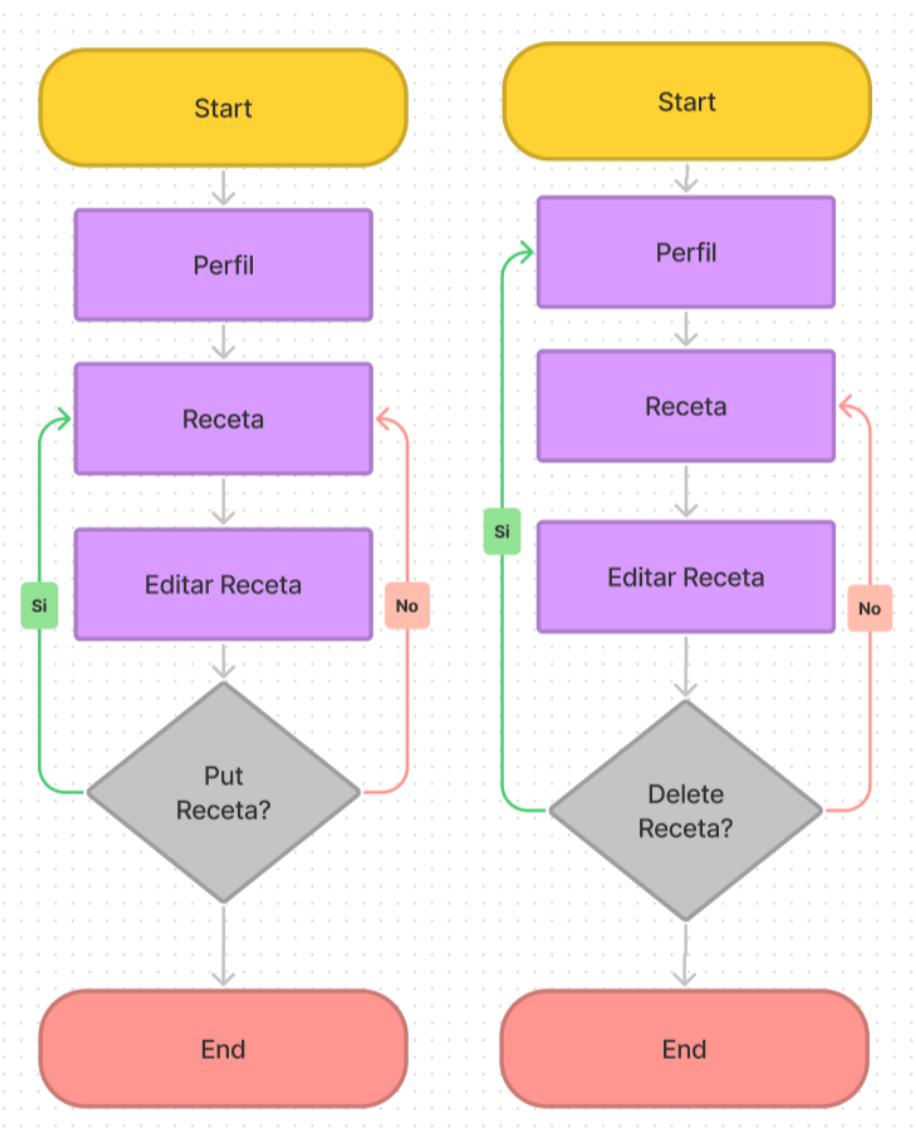


Figura 17: Flujos de edición y eliminación de una receta

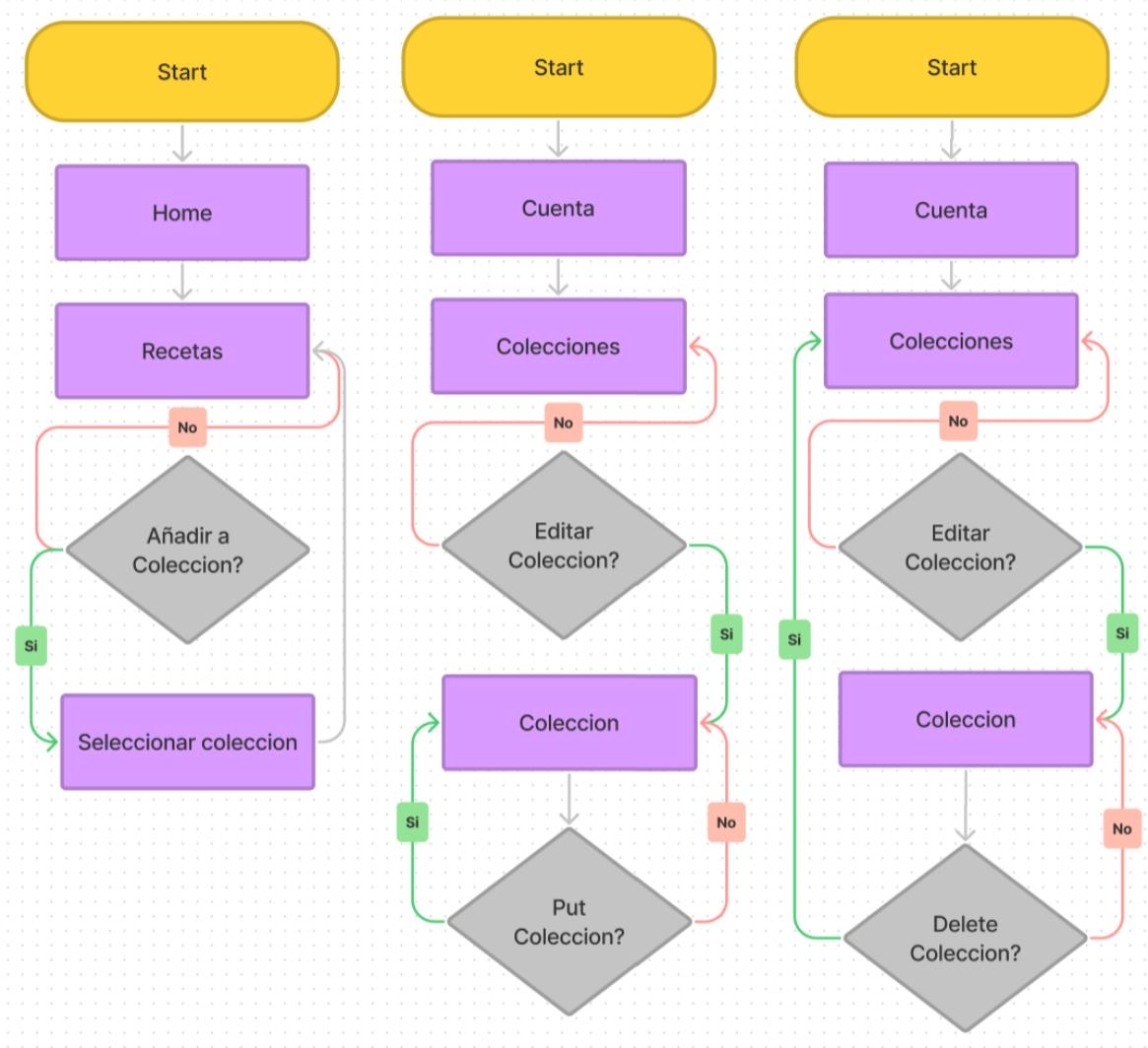


Figura 18: Flujos de añadir, editar y eliminar una colección

Por último, un usuario autenticado, podrá **comentar** una receta, en la vista de detalle de la receta. Podrá también darle a **like** o unlike a dicha receta, y finalmente podrá ver las recetas a las que le ha dado like en su Cuenta en el subapartado **Favoritos**, donde también podría eliminar esas recetas con like de la lista.

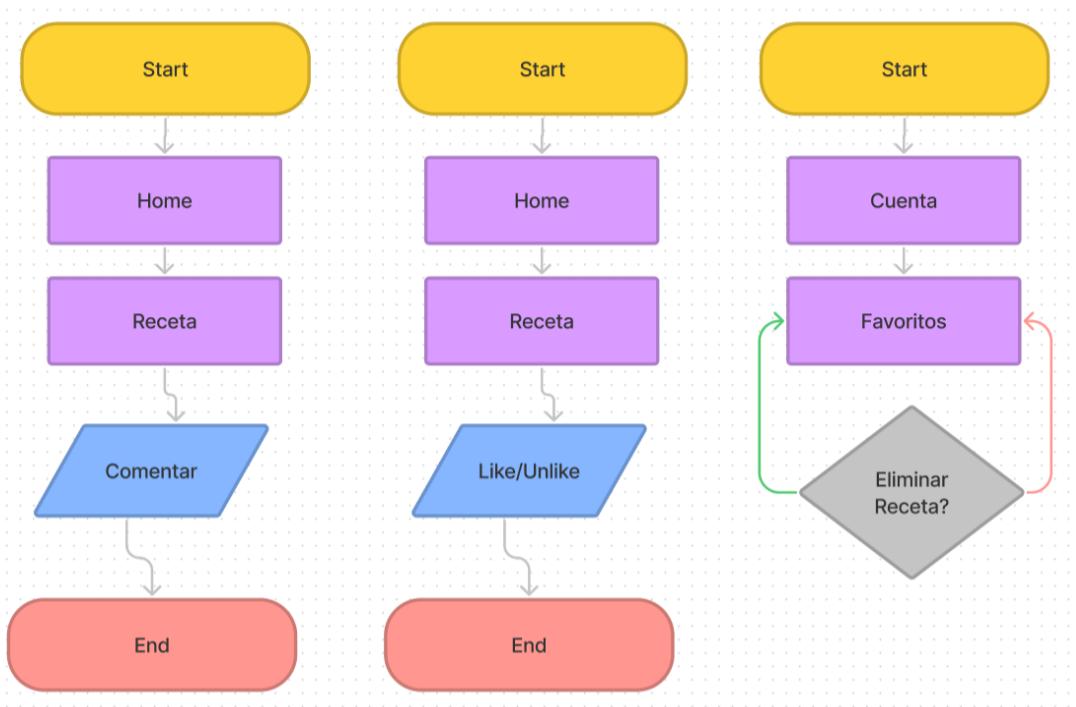


Figura 19: Flujos de interacción social sobre una receta

12.3 Diseño de la interacción

En esta fase procederemos a realizar el prototipado de la aplicación, que será una aproximación al diseño de las interfaces que tendrá el sistema.

Partiendo de los flujos y diagramas de interacciones planteados anteriormente, se ha procedido a diseñar la aplicación de manera que su utilización sea fácil y fluida para los potenciales usuarios y adaptando el contenido a dispositivos móviles y de escritorio.

A continuación dividiremos en subapartados cada una de las pantallas que formarán parte de la aplicación, junto con una breve descripción y explicación de las diferentes funcionalidades de cada una. Se incluyen tanto la vista de dispositivos móviles como desktop.

Podemos acceder a la totalidad de los diseños de las pantallas donde veremos realmente todo el flujo de un usuario a lo largo de la aplicación de forma muy detallada, en: [Figma](#).

12.3.1 Páginas de error

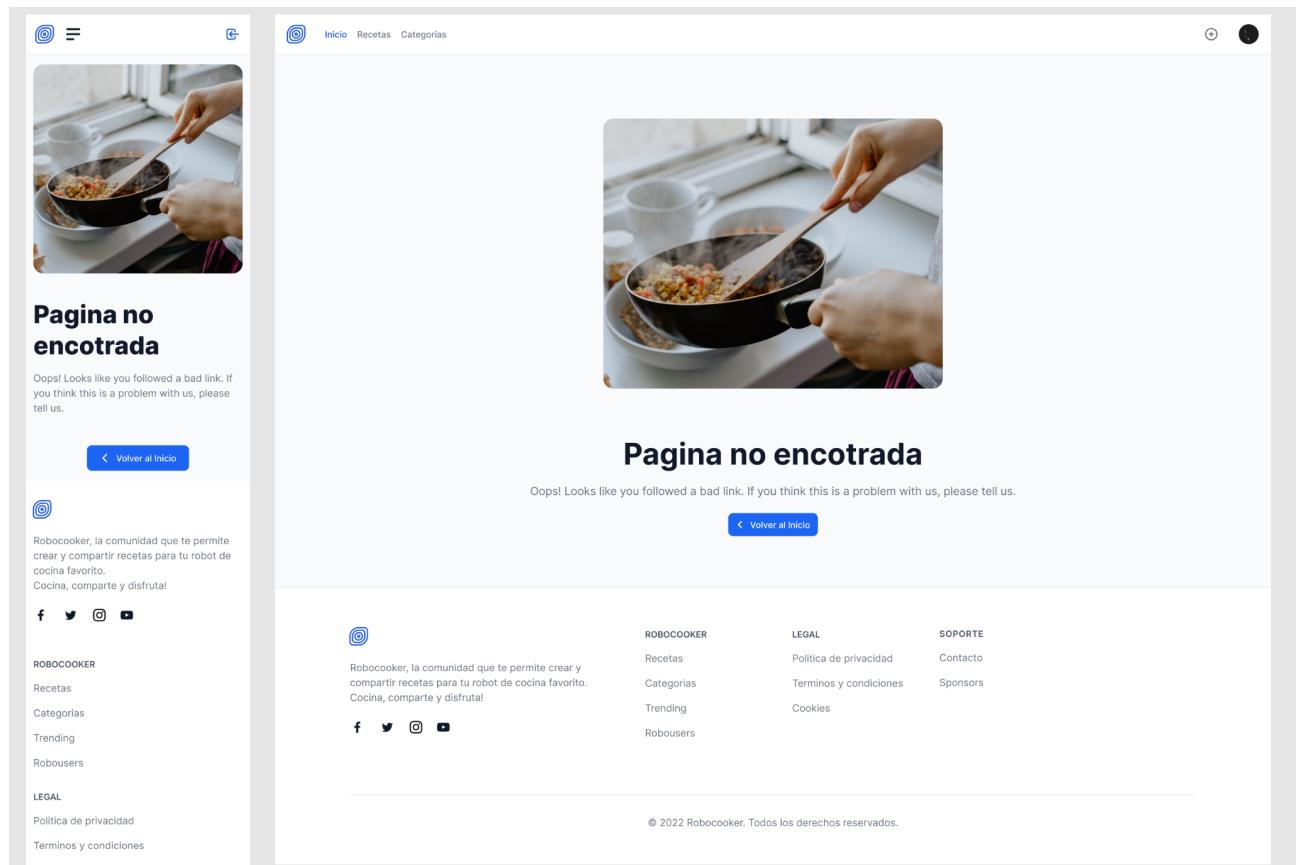


Figura 20: Páginas error. [Figma](#).

Son aquellas páginas a las cuales accede el usuario como consecuencia de introducir una ruta que no esté contemplada en el mapa de navegación o como cualquier otra respuesta de error del servidor.

12.3.2 Inicio de sesión / registro

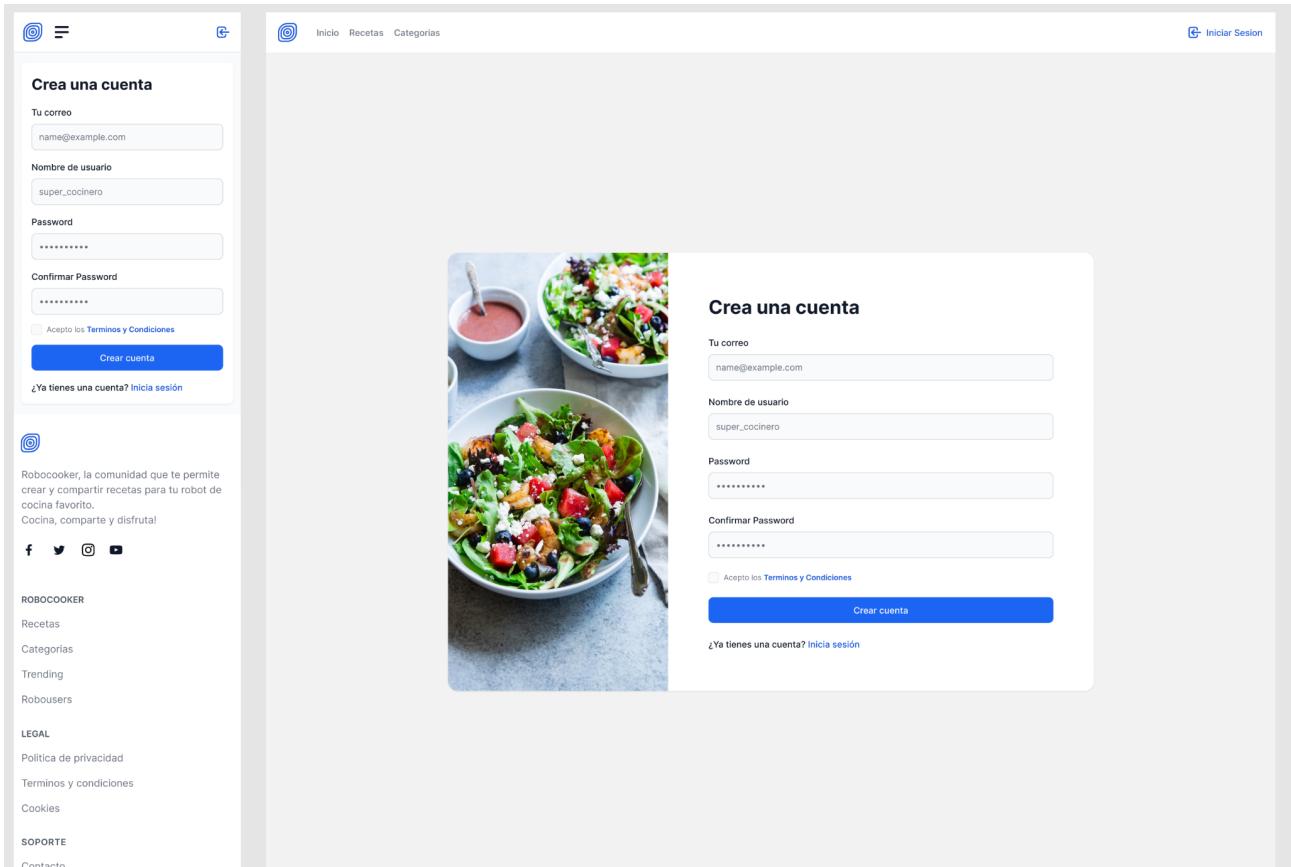


Figura 21: Inicio de sesión / registro. [Figma](#).

Tanto la página de inicio de sesión como de registro contendrá un formulario para el caso de uso correspondiente. Por simplicidad hemos incluido solo el diseño de registro de un usuario, accesible desde el enlace en la barra de navegación “Iniciar sesión”. Este enlace sólo será visible cuando el usuario no esté autenticado.

12.3.3 Página de inicio

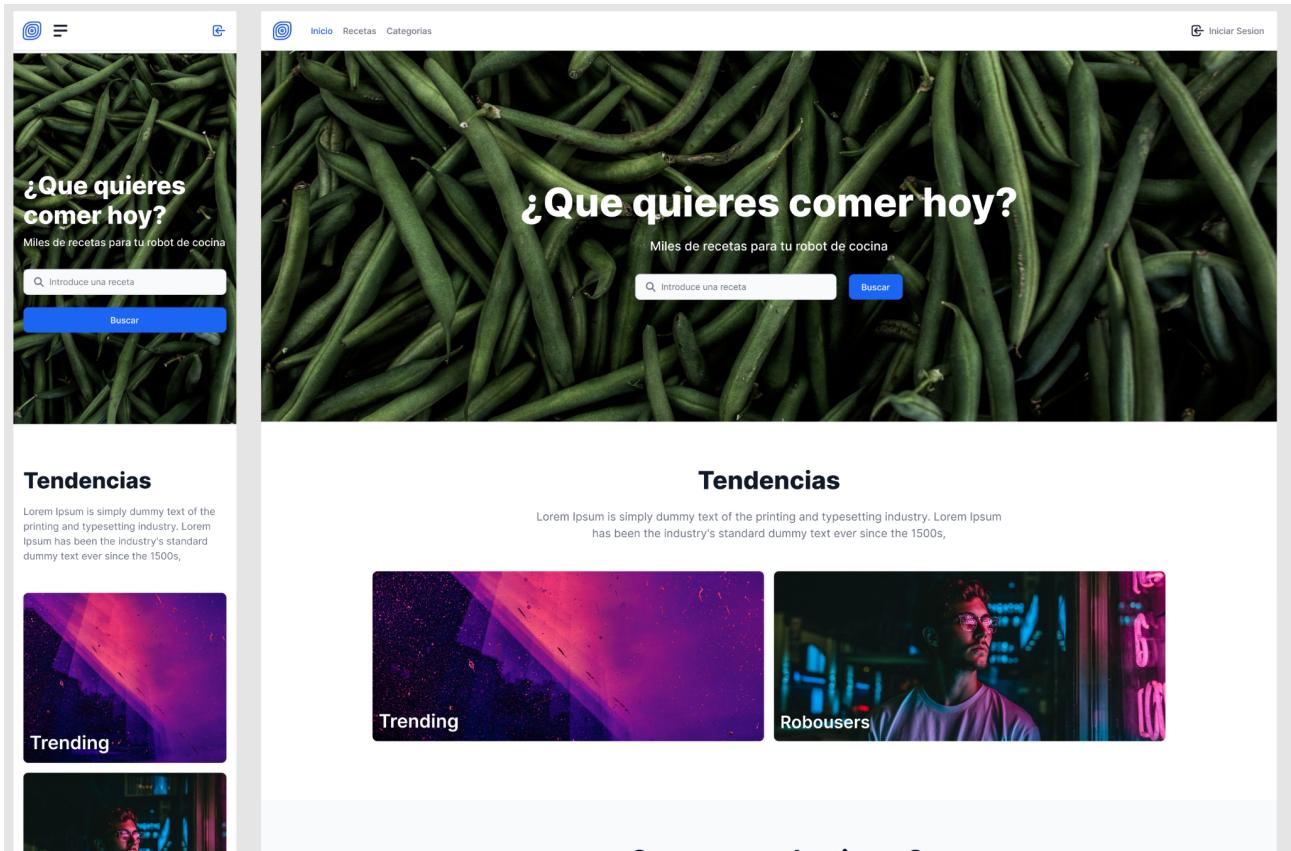


Figura 22: Página de inicio. [Figma](#).

En la página de inicio podremos ver en primer lugar un hero que contendrá un input de búsqueda de una receta que nos dirigirá a la página de recetas, donde podremos aplicar diferentes filtros. Seguidamente tendremos categorías según las tendencias y los usuarios que más colaboran en la plataforma, procesadores de alimentos, y las últimas recetas publicadas.

12.3.4 Trending / Robousers

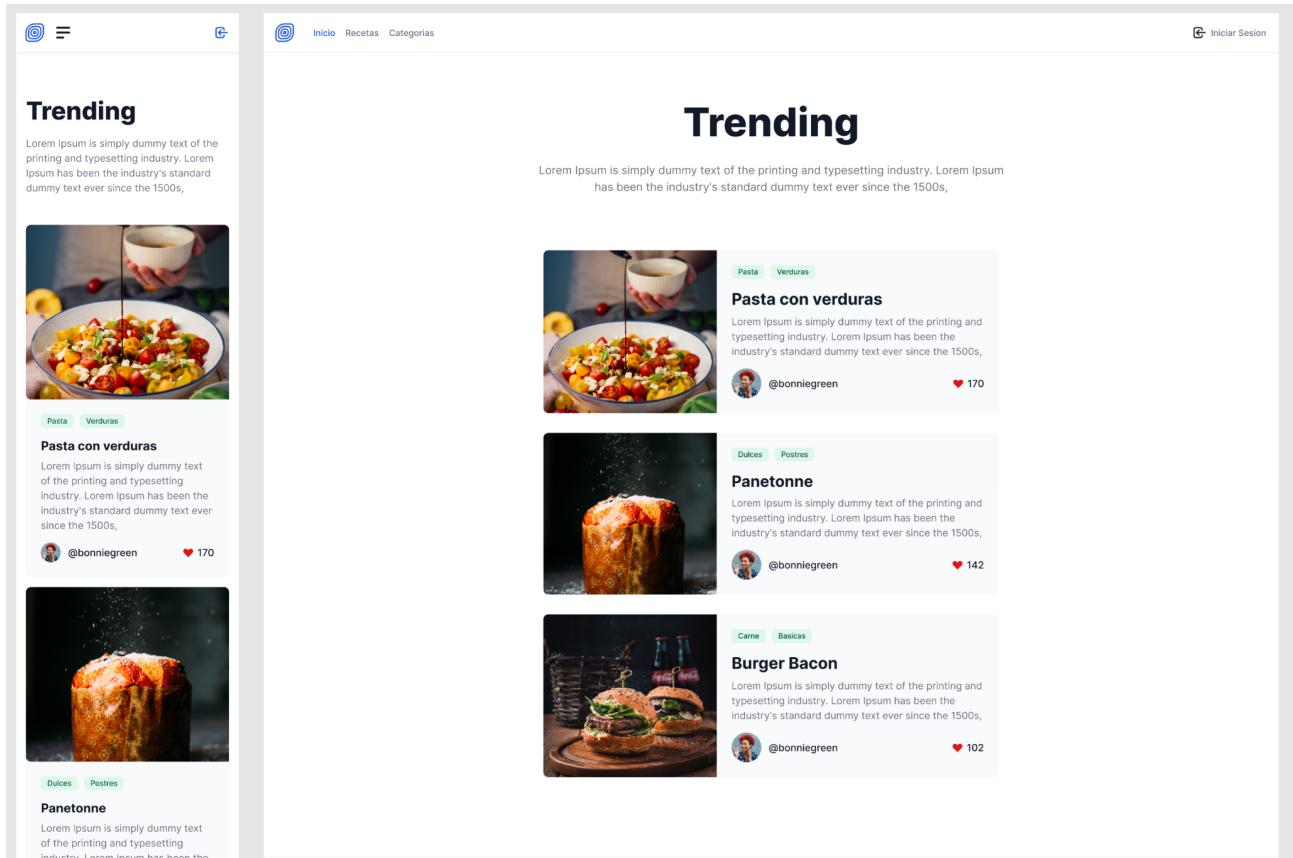


Figura 23: Trending / Robousers. [Figma](#).

En esta sección podremos encontrar las recetas que más “me gusta” han recibido a lo largo de un determinado periodo de tiempo. Tendremos un podio donde estarán las 3 más valoradas, para darles una mayor visibilidad y las siguientes en un grid a continuación. La sección de robousers será equivalente a las recetas trending pero con los usuarios que más aportan contenido.

12.3.1 Categorías

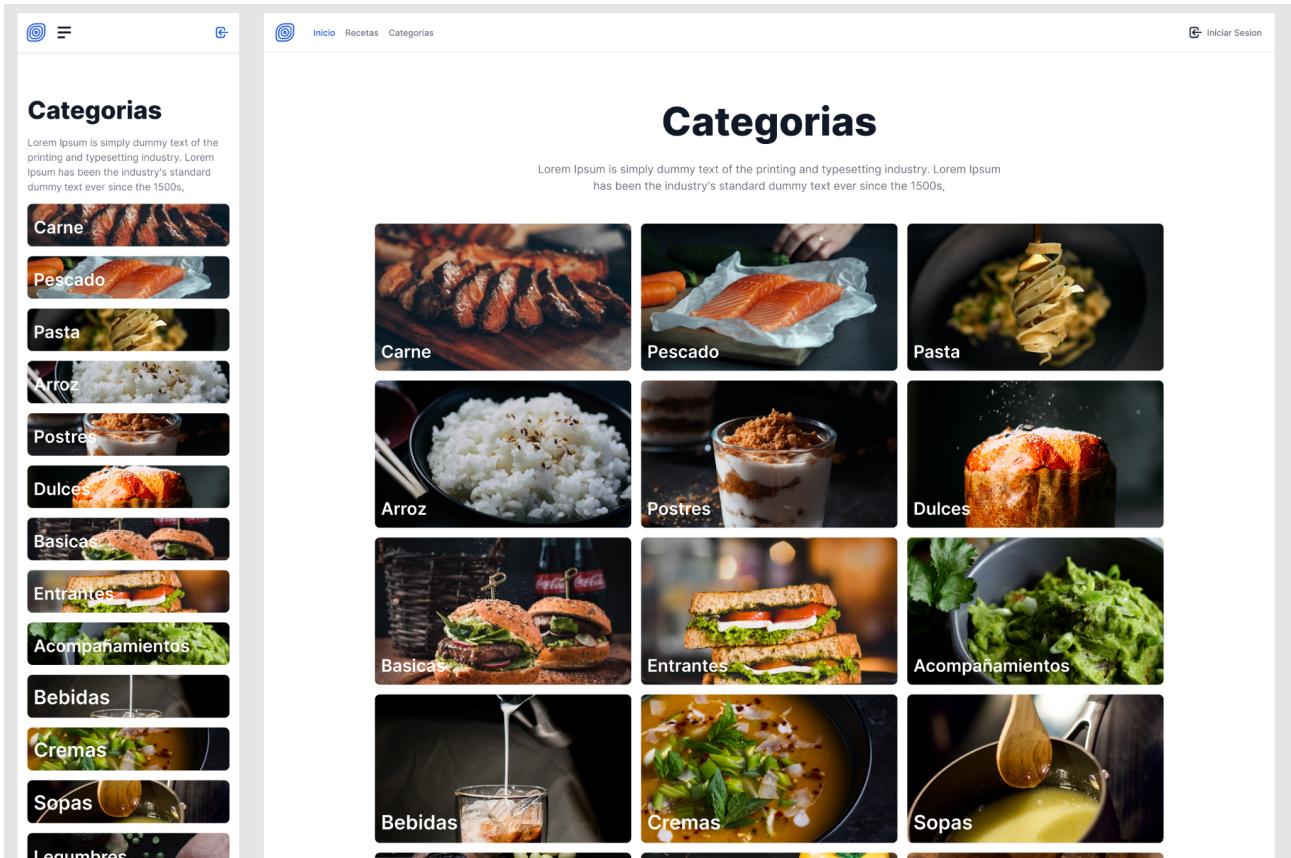


Figura 24: Categorías. [Figma](#).

En categorías podremos acceder a recetas según el tipo de alimento que queramos buscar, se redirigirá a la página de recetas con un filtro aplicado para obtener del servicio los resultados que coincidan en base de datos.

12.3.1 Recetas

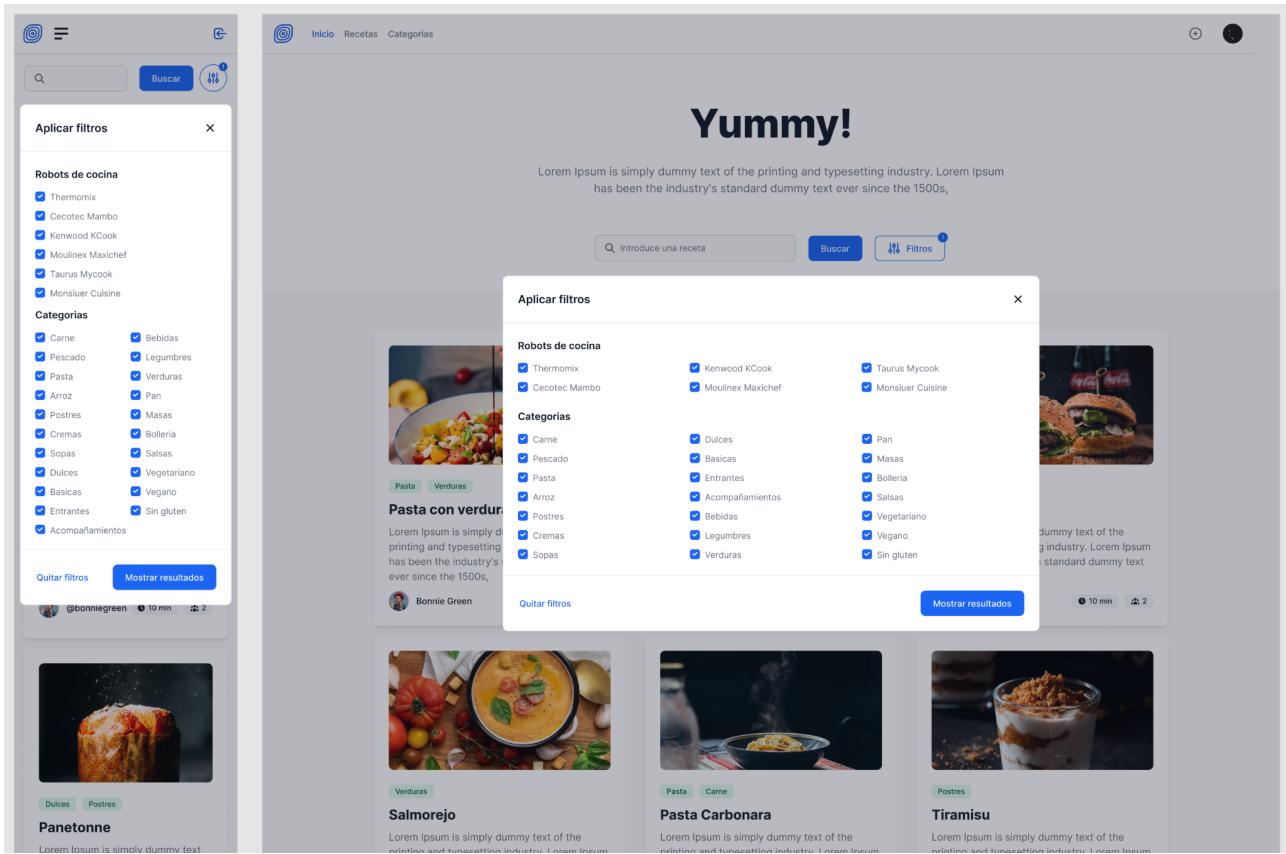


Figura 25: Recetas. [Figma](#).

En la página de recetas tendremos un buscador donde podremos introducir una palabra clave para buscar una receta y además tendremos una selección de filtros para poder aplicar si deseamos y obtener unos resultados más ajustados a nuestras preferencias.

Debajo de este buscador podremos ver las recetas obtenidas. Si el usuario no realiza ninguna búsqueda inicial veremos siempre las últimas recetas publicadas en la plataforma.

12.3.1 Receta

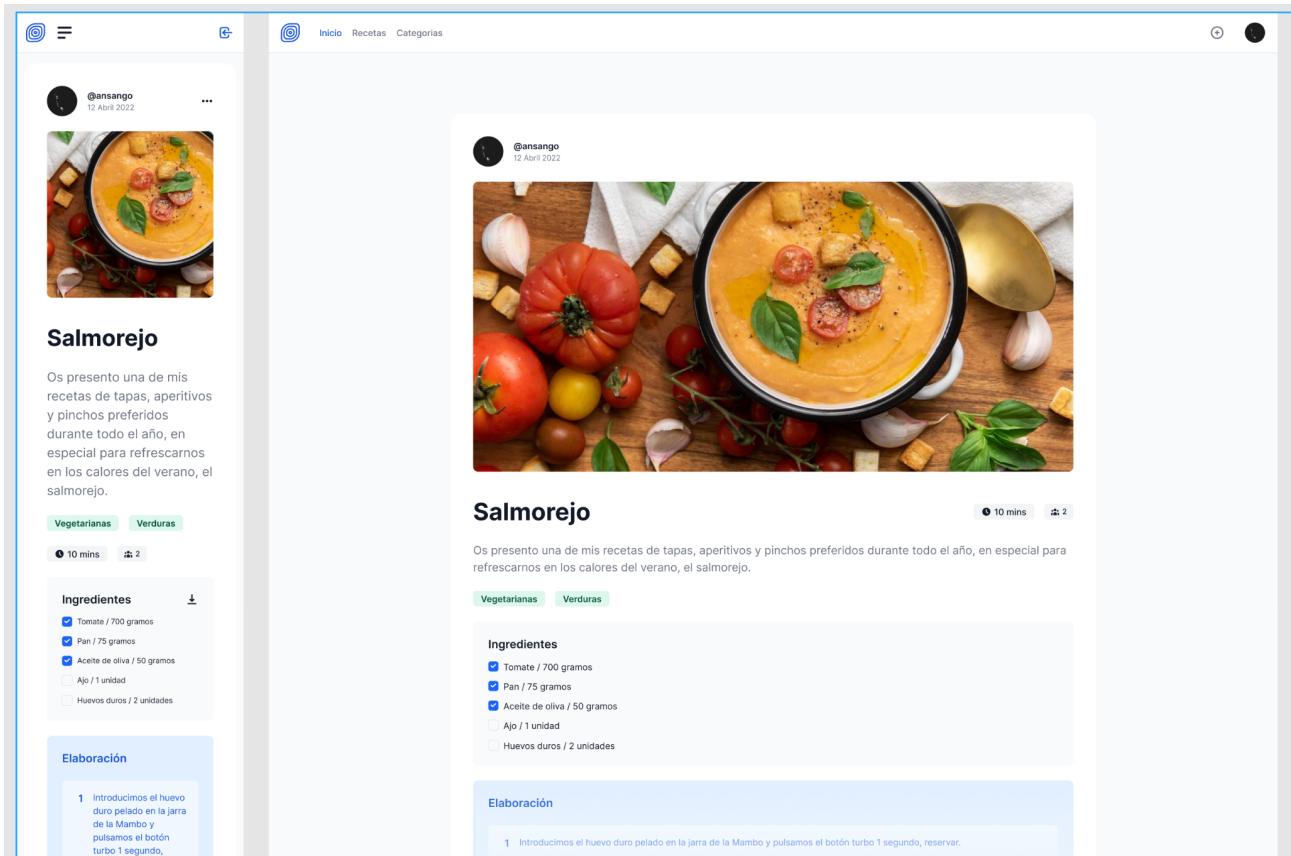


Figura 26: Receta. [Figma](#).

En la página de receta veremos los detalles de la misma, nombre, descripción, elaboración, ingredientes. Y si estamos autenticados podremos interactuar, dar “me gusta”, comentar, compartir o imprimir los ingredientes. También podremos acceder al perfil del creador de la misma.

Si es nuestra receta tendremos un botón de ajustes donde podremos o bien acceder a la edición, o eliminar directamente la receta.

12.3.1 Perfil

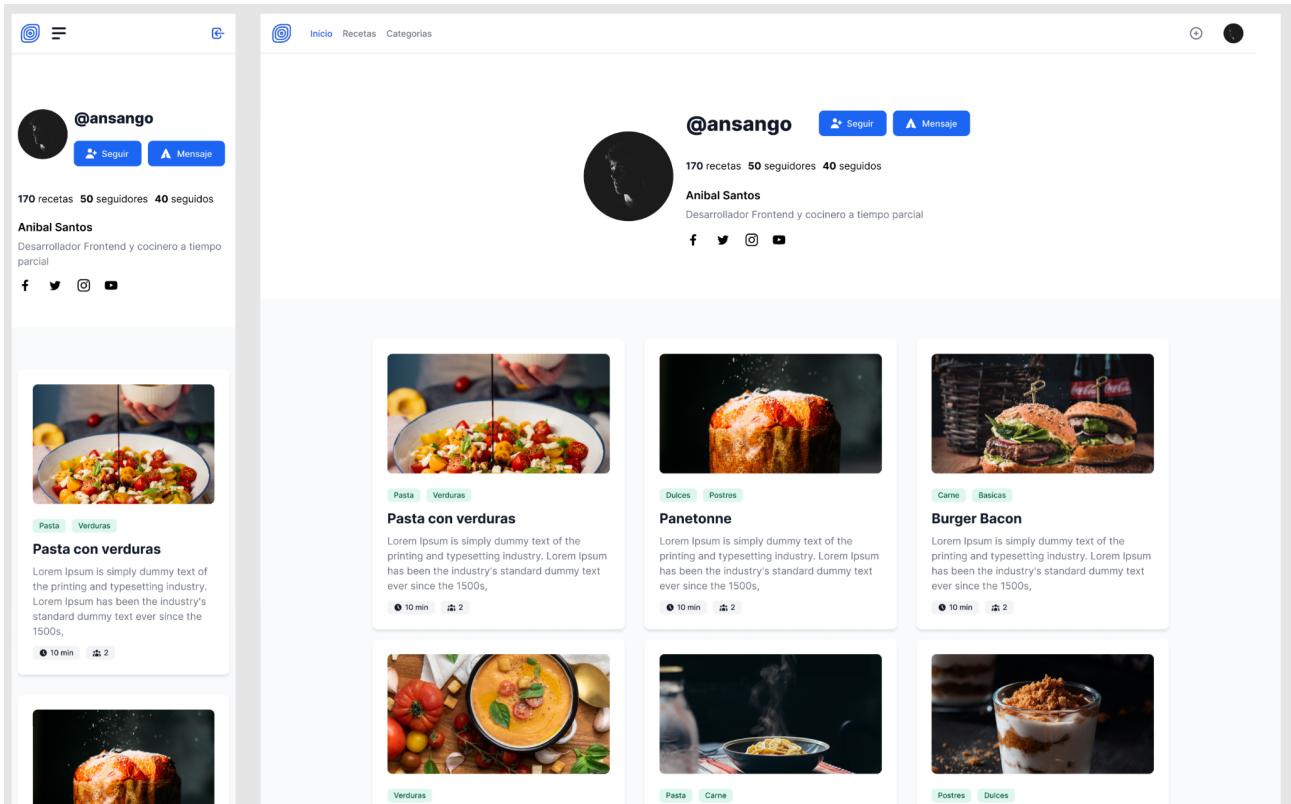


Figura 27: Perfil. [Figma](#).

En el perfil de un usuario podremos ver una mínima información personal, seguidores, seguidos, número de publicaciones, seguirle, enviarle un mensaje o ver cada una de sus publicaciones.

12.3.1 Crear / Editar / Eliminar receta

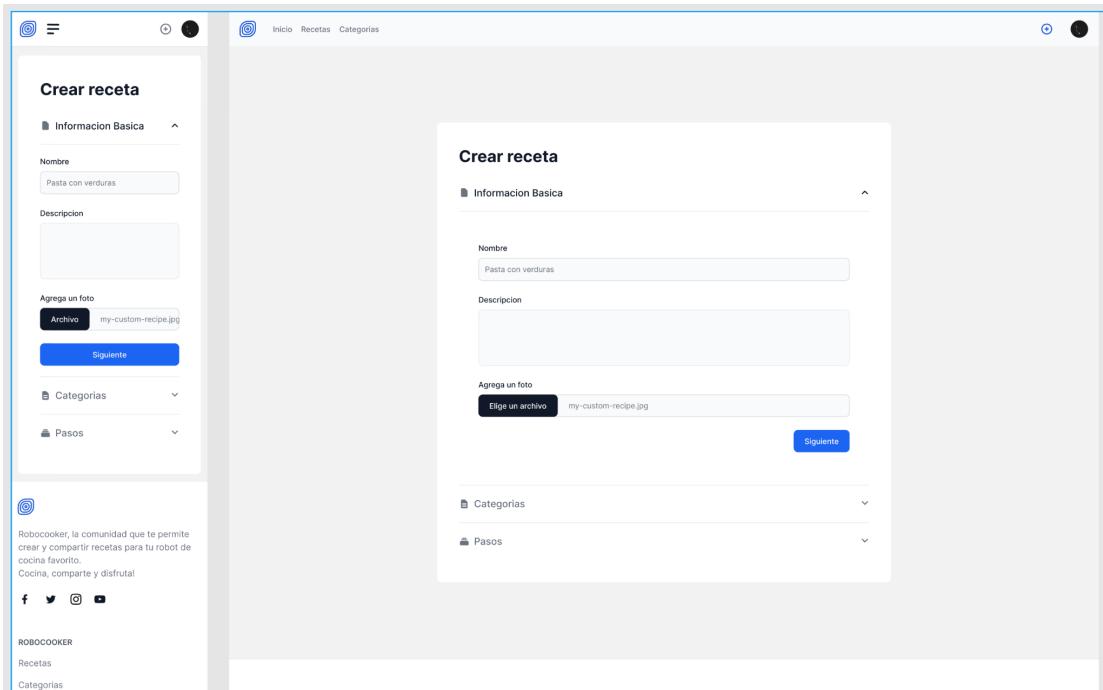


Figura 28: Crear receta. [Figma](#).

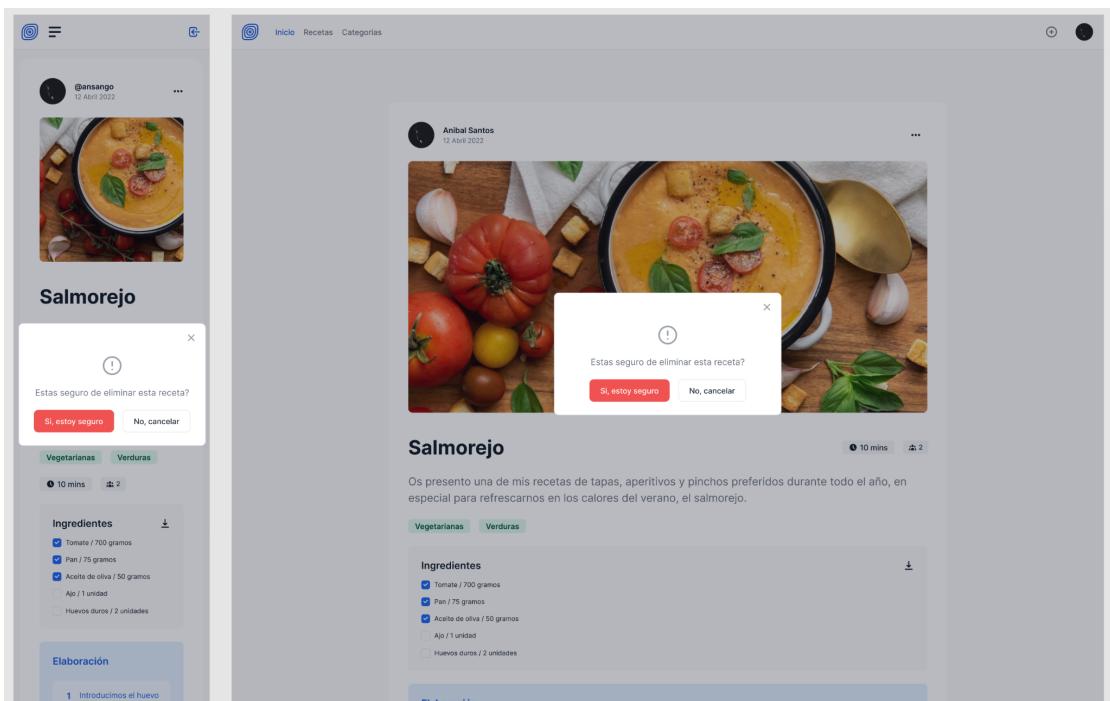


Figura 29: Eliminar receta. [Figma](#).

Podremos acceder a la creación de una receta mediante el botón en la barra de navegación, donde completamos un formulario con steps. La página de edición de la receta será accesible desde las opciones

Red social culinaria orientada a los procesadores de alimentos, Aníbal Santos Gómez

en la página de la misma. Para eliminar una receta podremos, desde las opciones en la receta, seleccionar esta opción.

12.3.1 Ajustes / Editar perfil

The figure consists of three mobile application screenshots. The leftmost screenshot shows the main 'Ajustes' (Settings) screen with a sidebar containing 'Ajustes', 'Colecciones', 'Favoritos', and 'Mensajes'. The middle screenshot shows the 'Cambio de contraseña' (Password Change) section with fields for 'Current password', 'New password', and 'Confirm new password', along with a 'Guardar' (Save) button. The rightmost screenshot shows the 'Social' section with four social media connection cards (Facebook, Twitter, Instagram, YouTube) and a 'Guardar' (Save) button. To the right of the social section is a 'Preferencias' (Preferences) sidebar with sections for 'Robots de cocina' (Kitchen Appliances) and 'Categorías' (Categories), each with a list of checked options.

Figura 29: Ajustes / Editar perfil. [Figma](#).

En ajustes podremos personalizar nuestro perfil de usuario, completar la información de registro, añadir nuestras redes sociales, y las preferencias acerca del contenido relacionado en la plataforma. En esta parte de dashboard en desktop tendremos la posibilidad de navegar con un sidebar, mientras que en móvil lo haremos con un dropdown en el avatar de la barra de navegación.

12.3.1 Colecciones

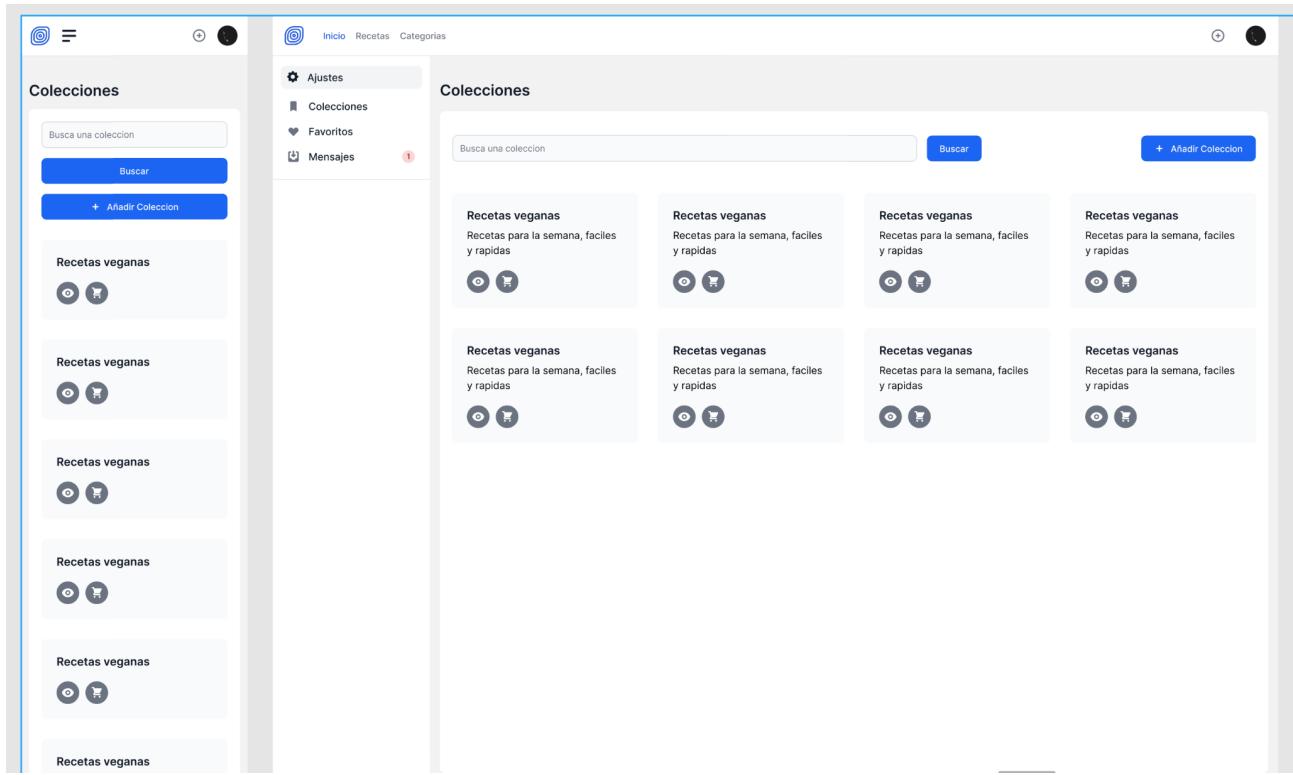


Figura 30: Colecciones. [Figma](#).

En el apartado de colecciones podremos visualizar las colecciones que el usuario ha ido agregando o crear una nueva. En cada tarjeta de colección podremos visualizar todos los ingredientes de las recetas que la componen, o bien visualizar todas las recetas añadidas a dicha colección.

Red social culinaria orientada a los procesadores de alimentos, Aníbal Santos Gómez

The image displays two side-by-side screenshots of a mobile application interface. Both screens have a header bar with icons for camera, menu, and notifications. The left screen shows a sidebar with 'Recetas veganas' and a list of ingredients: Tomates, Queso, Aceite, Vino, and Apio. It includes a search bar, a blue 'Buscar' button, and a footer with social media links (Facebook, Twitter, Instagram, YouTube) and a copyright notice. The right screen shows a similar layout with a sidebar and a main content area titled 'Recetas veganas'. The main content area features a search bar, a blue 'Buscar' button, and a table listing ingredients with their quantities and units:

INGREDIENTES	CANTIDAD	UNIDAD
Tomates	700	gramos
Queso	400	gramos
Aceite	200	gramos
Vino	200	gramos
Apio	150	gramos

Figura 31: Colecciones / Ingredientes. [Figma](#).

Al acceder a la colección de recetas mediante el botón de ingredientes, veremos la lista de ingredientes que la componen. Podremos buscar un ingrediente en concreto.

Red social culinaria orientada a los procesadores de alimentos, Aníbal Santos Gómez

The image displays two side-by-side wireframe prototypes from Figma, showing different views of a recipe collection feature.

Left Wireframe (Collection View):

- Header: Shows icons for profile, search, and more, followed by "Recetas veganas".
- Breadcrumbs: "Colecciones > Recetas veganas".
- Search Bar: "Busca una receta".
- Search Button: "Buscar".
- List of recipes:

 - Pasta con verduras
 - Tagliatella
 - Pate de mejillones
 - Mac and Cheese
 - Burger Bacon

- Each item has a small circular icon with a red "X" to its right.

Right Wireframe (List View):

- Header: Shows icons for profile, search, and more, followed by "Recetas veganas".
- Breadcrumbs: "Colecciones > Recetas veganas".
- Search Bar: "Busca una receta".
- Search Button: "Buscar".
- Table Headers: "NOMBRE", "CATEGORIAS", and "PROPIEDADES".
- Data Rows:

 - Pasta con verduras (Pasta, Verduras) - 10 min, 2 personas
 - Tagliatella (Pasta) - 10 min, 2 personas
 - Pate de mejillones (Pescado, Entrantes) - 10 min, 2 personas
 - Mac and Cheese (Pasta) - 10 min, 2 personas
 - Burger Bacon (Carne, Basicas) - 10 min, 2 personas

- Each row has a small circular icon with a red "X" to its right.

Figura 32: Colecciones / Recetas. [Figma](#).

Mientras que si accedemos a la colección de recetas podremos ver la lista de las recetas y eliminar alguna de ellas si queremos.

12.3.1 Mensajes

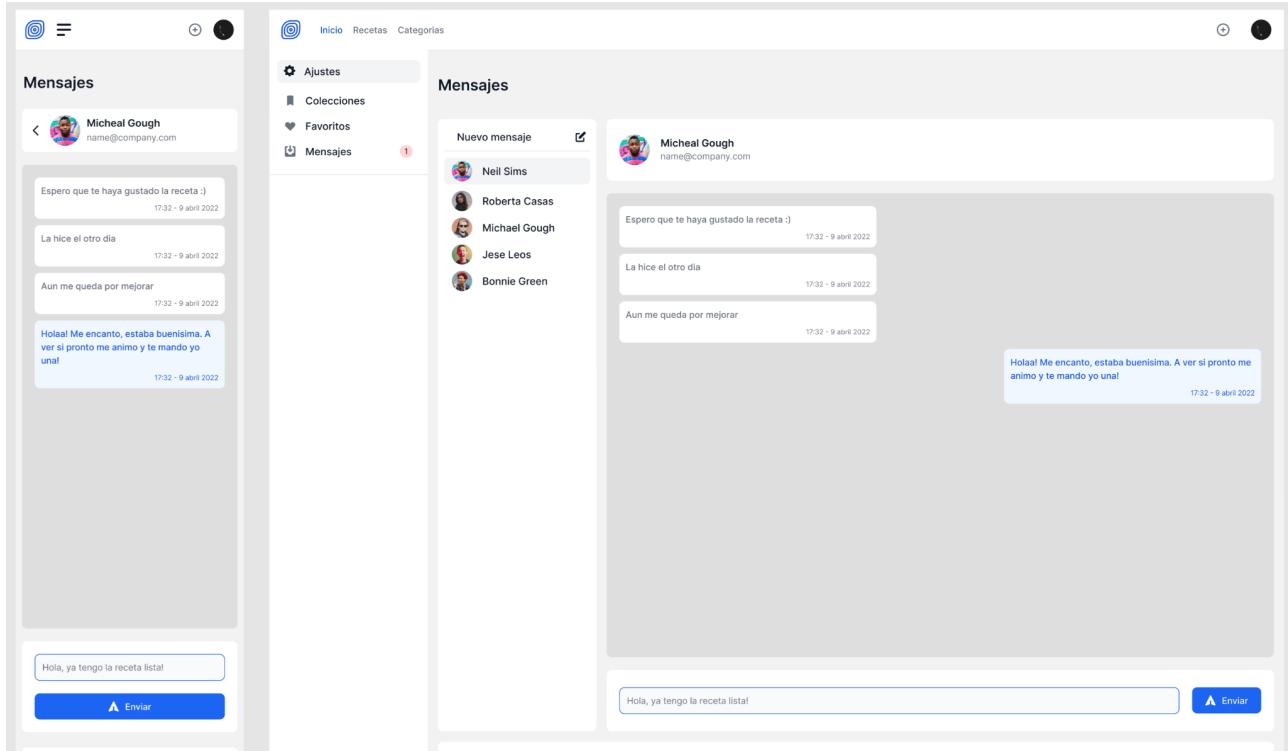


Figura 33: Mensajes. [Figma](#).

Podremos revisar nuestros mensajes con otros usuarios de la aplicación accediendo a la sección de mensajes, donde también podremos crear mensajes nuevos con usuarios de la plataforma, además de poder hacerlo sobre el perfil del usuario correspondiente.

12.3.1 Favoritos

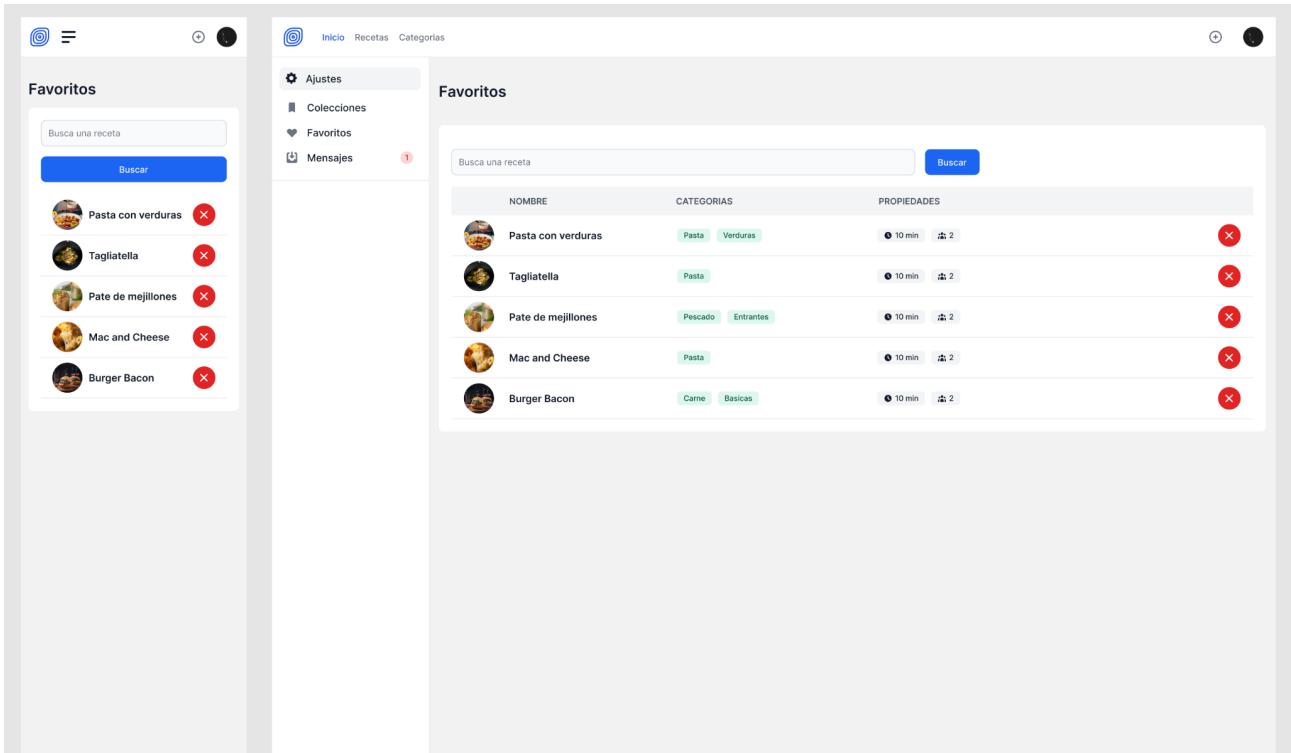


Figura 34: Favoritos. [Figma](#).

Por último en favoritos podremos ver las publicaciones a las que hemos reaccionado con un “me gusta”. Podremos buscar una receta en concreto y/o eliminarla y que desaparezca de nuestros favoritos.

13. Implementación

13.1 Desarrollo

13.2 Seguridad

13.3 Testing

13.4 Versionado

13.5 Despliegues

14. Instalación y uso

14.1 Requisitos e instalación

14.2 Configuración de entornos

14.3 Herramientas recomendadas

15. Proyección a futuro

15.1 Roadmap

15.2 Contribuciones

16. Conclusiones

Conclusiones personales acerca del proyecto realizado, el proceso de trabajo y los resultados obtenidos.

Anexo 1. Entregables del proyecto

Anexo 2. Código fuente (extractos)

Anexo 3. Librerías

- **Nextjs:**

Es un framework de JavaScript que nos permite crear fácilmente sitios web de React listos para salir a producción. Disponible en: <https://nextjs.org/>

- **React:**

Biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Disponible en: <https://es.reactjs.org/>

- **TypeScript:**

Lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. Disponible en: <https://www.typescriptlang.org/>

- **Tailwindcss:**

Es un framework de CSS de código abierto para el diseño de páginas web. Disponible en: <https://tailwindcss.com/>

- **Axios:**

Es una librería JavaScript que puede ejecutarse en el navegador y que nos permite hacer sencillas las operaciones como cliente HTTP. Disponible en: <https://github.com/axios/axios>

- **Jest:**

Es un framework desarrollado por Meta, para poder testear código Javascript. Disponible en: <https://jestjs.io/es-ES/>

- **Testing library:**

Conjunto de librerías ligeras orientadas a testing de componentes en frameworks Javascript como React. Disponible en: <https://testing-library.com/>

- **Redux Toolkit:**

Librería para el manejo de estado en aplicaciones Javascript. Disponible en: <https://redux-toolkit.js.org/>

- **Storybook:**

Herramienta para el desarrollo de interfaces aisladas de páginas y componentes. Disponible en: <https://storybook.js.org/>

- **Node.js:**

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript. Disponible en: <https://nodejs.org/es/>

- **Express.js:**

Es un framework de back-end para Node.js. Disponible en: <https://expressjs.com/es/>

- **Mongodb:**

Sistema de base de datos NoSQL, orientado a documentos y de código abierto. Disponible en: <https://www.mongodb.com/es>

- **Mongoose:**

Es una biblioteca de JavaScript que permite definir esquemas con datos fuertemente tipados. Disponible en: <https://mongoosejs.com/>

- **AWS S3 Bucket:**

Almacenamiento de objetos creado para recuperar cualquier volumen de datos desde cualquier ubicación. Disponible en: <https://aws.amazon.com/es/s3/>

- **Vercel:**

Plataforma orientada a la integración de frameworks frontend y construcción de sitios estáticos. Disponible en: <https://vercel.com/>

- **Github:**

Plataforma de desarrollo colaborativo para alojar proyectos utilizando Git. Disponible en: <https://github.com/>

- **Atlas Mongodb:**

Servicio cloud de bases de datos NoSQL basadas en MongoDB. Disponible en: <https://www.mongodb.com/atlas/database>

Anexo 4. Capturas de pantalla

Anexo 5. Bibliografía

- Sempere, P. *McLuhan en la era de Google: Memorias y profecías de la Aldea Global*. Madrid: Editorial Popular, 2007.
- Rogers, E. M. *Diffusion of Innovations*. New York: Free Press, 2003.
- Aibar, E. *La visión constructivista de la innovación tecnológica. Una introducción al modelo SCOT*. Barcelona: UOC, 2006.
- Wikipedia, *Desarrollo ágil de software*, 2022.
- A. Osman. *Introduction*, Patterns.dev, 2022.
- J.S. Fernández. *¿Por qué utilizo Clean Architecture?*, XurxoDev, 2016.
- R. C. Martin. Clean Architecture, The Clean Code Blog, 2022.
- G.Kitchen, *The home kitchen in the globalization era*, Consentino S.A, 2019

Anexo 6. Vita