Food Studies



Introducción

Food Studies es una herramienta que busca brindar un lugar donde compartir su trabajo y poder acceder al de otros dentro de su misma línea de investigación, así como el de ser notificados de eventos de interés, centralizando la información relacionada en una misma plataforma.

Índice

1. Análisis ………………………………………………………………..1

1.1. Identificación del problema ……………………………………….1

1. Definición ……………………………………………………………..1

2.1. Objetivos ……………….....……………………………………….1

2.2. Requisitos ………………...……………………………………….1

1. Diseño ..………………………………………………………………..1

3.1. Inicio ………………...………………………………………….....1

3.1.1. Caso iniciar sesión ..……………………………………….1

3.1.2. Caso registrarse ......……………………………………….1

1. Planificación …………………………………………………………..1
2. Conclusiones …………………………………………………………..1
3. Bibliografía ..…………………………………………………………..1
4. Anexos …….…………………………………………………………..1

1. Análisis

1.1 Identificación del problema

Después de un análisis de las diferentes herramientas web, se llegó a la conclusión de que existe una ausencia de una herramienta de comunicación dedicada para este tipo de actividades hace tener que recurrir a canales de comunicación de uso común que no ofrecen servicios pensados para las necesidades del público objetivo. Lo más similar a lo que ofrecemos es una plataforma como Linkedin que se enfoca al mundo laboral y como ofertarse en este mismo o GitHub que hace lo propio dedicado al desarrollo de aplicaciones. Por contraparte nosotros ofrecemos un producto que permite disfrutar de funcionalidades pensadas para hacer más fácil la tarea de investigación de los usuarios.

2. Definición

2.1 Objetivos

Food Studies busca brindar una herramienta para investigadores donde compartir su trabajo y poder acceder al de otros dentro de su misma línea de investigación a través de hashtags, así como el de ser notificados de eventos de interés, centralizando la información relacionada en una misma plataforma, también se ofrecerá la opción de ver el contenido e investigadores de usuarios específicos gracias a la función de seguir.

2.2 Requisitos

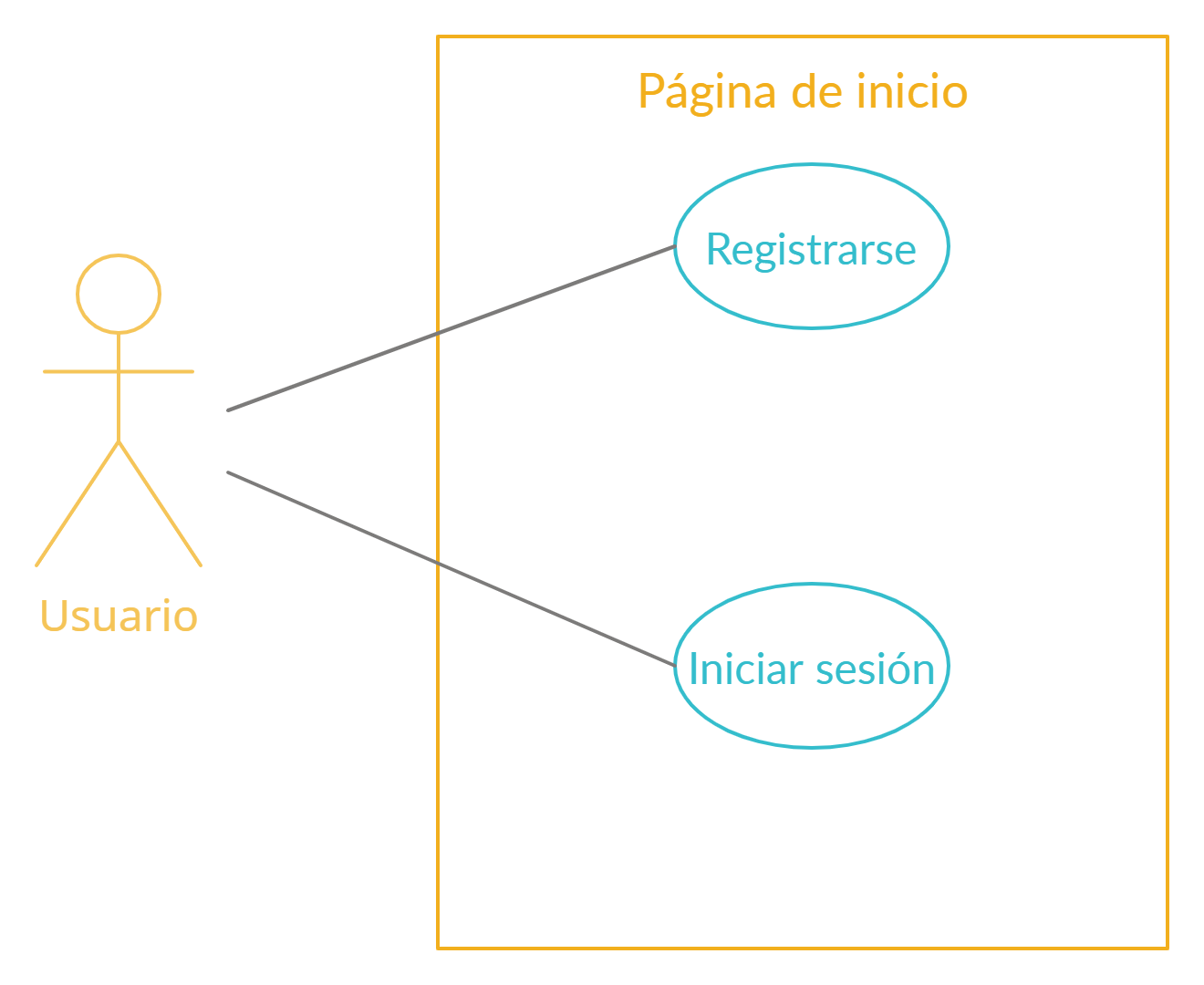
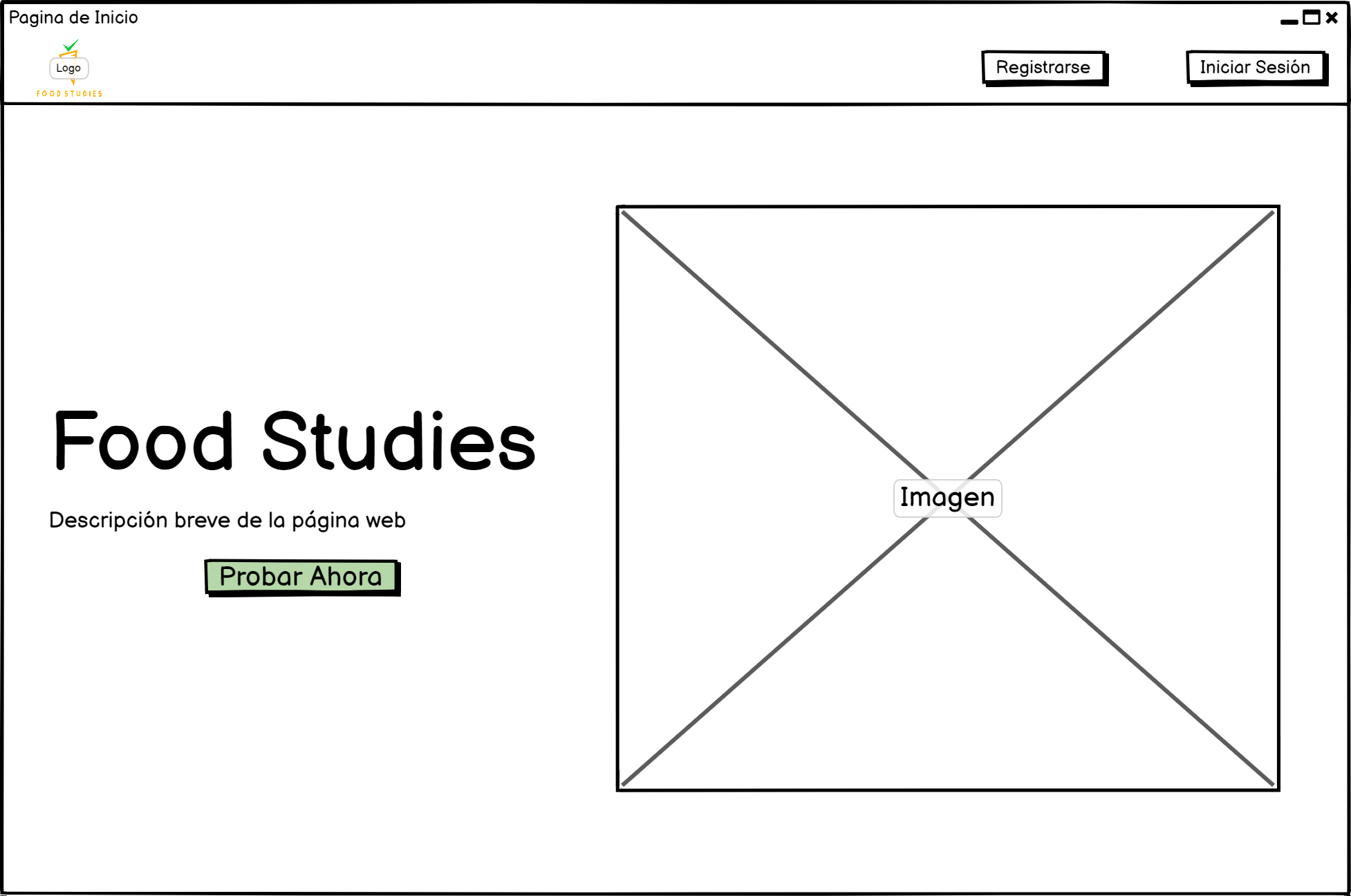
Para la realización completa del proyecto se implementará MEVN stack, es decir MongoDB, Express, VueJs, NodeJs porque permitirá crearlo como una Single Page Application (SPA), además este stack permite cubrir el frontend como el backend solo con JavaScript, también se utilizara Bootstrap 4 y CSS3 para darle estilo al frontend. Ya que el proyecto se tendrá que desplegar para su funcionamiento como red social, se contratará el servicio de hosting hostinger.

3. Diseño

La aplicación estará diseñada como una API que se encargará de toda la lógica y de la conexión con la base de datos, para poder utilizarla en otros tipos de plataformas si se desea así

3.1 Inicio

En el inicio de la aplicación solo se podrá realizar dos opciones

*Imagen 1. Diagrama de casos de uso de la página de inicio*

*Imagen 2. Boceto de la página de inicio*

3.1.1 Caso iniciar sesión:

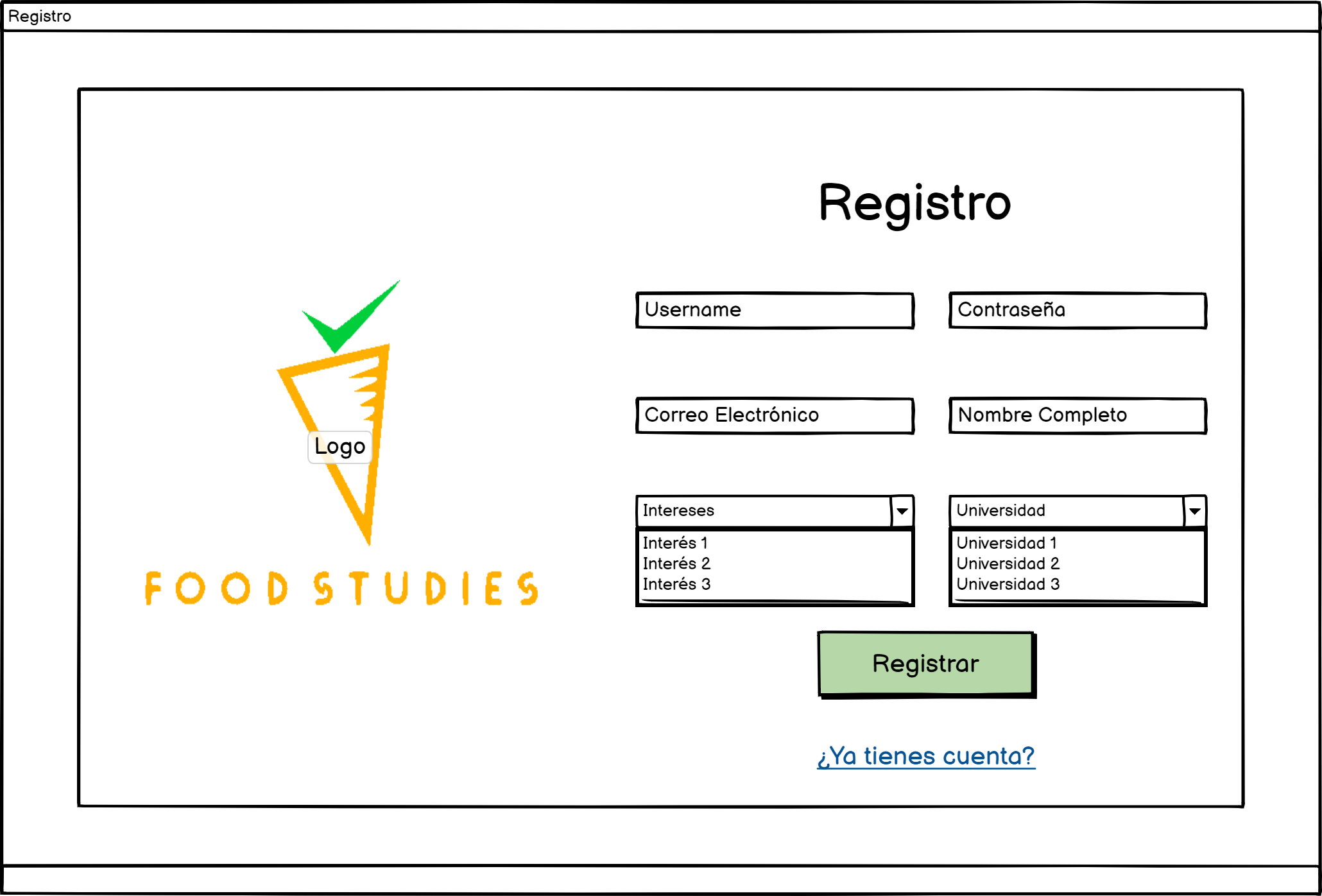


*Imagen 3. Boceto de pantalla de inicio de sesión*

Flujo base:

1. Iniciar aplicación
2. Llenar formulario de inicio de sesión: se mostrará un formulario de inicio de sesión en el que se tendrá que escribir nombre de usuario y contraseña
3. Validar: acá se comprobará si el nombre usuario y contraseña son correctos, en ese caso se seguirá el flujo base, en caso contrario se mostrará un mensaje de error y se volverá al paso dos.
4. Ingresar: una vez los datos sean validados de forma correcta, se redirigirá a la página de inicio del usuario.

3.1.2 Caso registrarse:



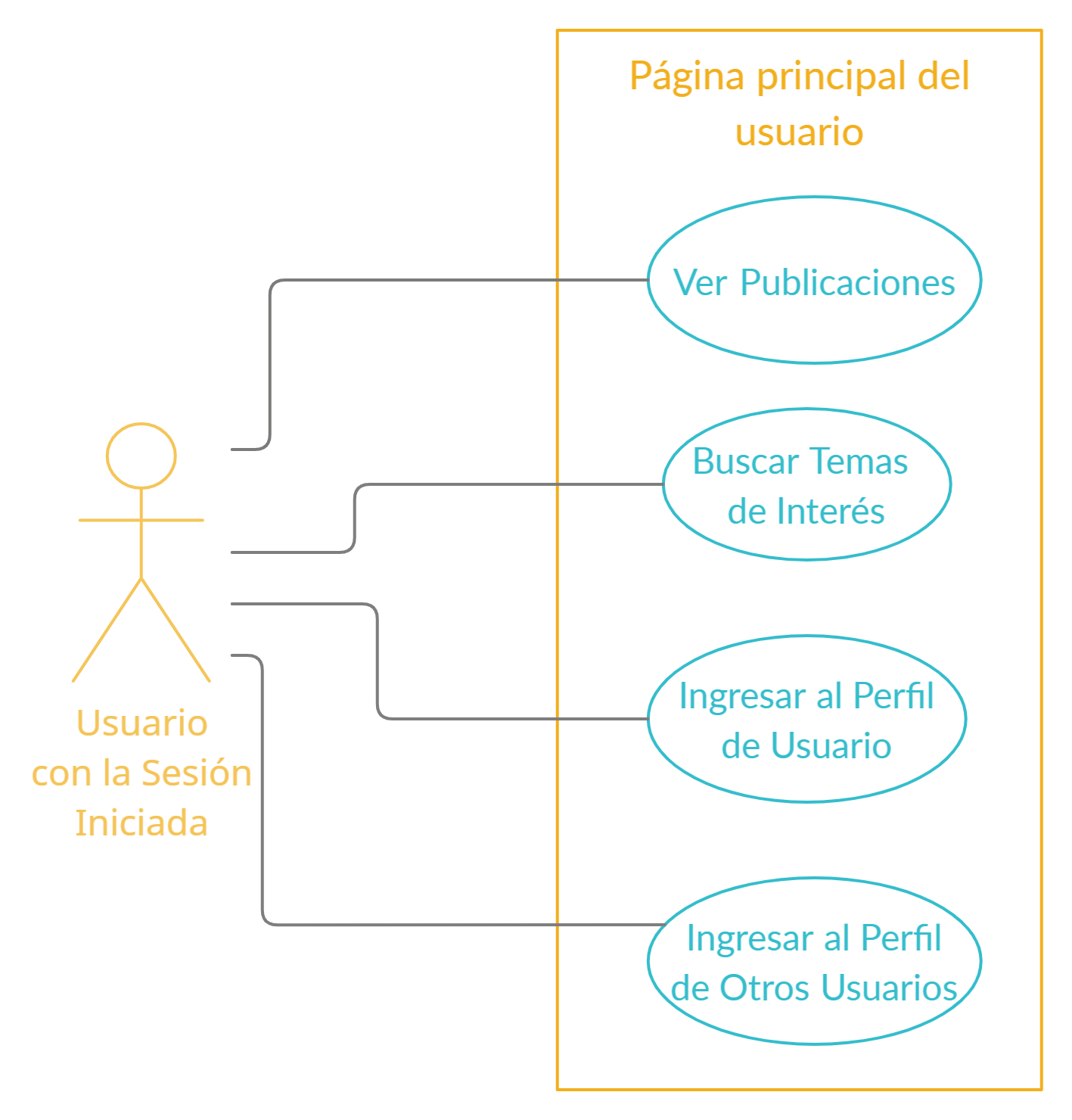
*Imagen 3. Boceto de la página de registro*

Flujo base:

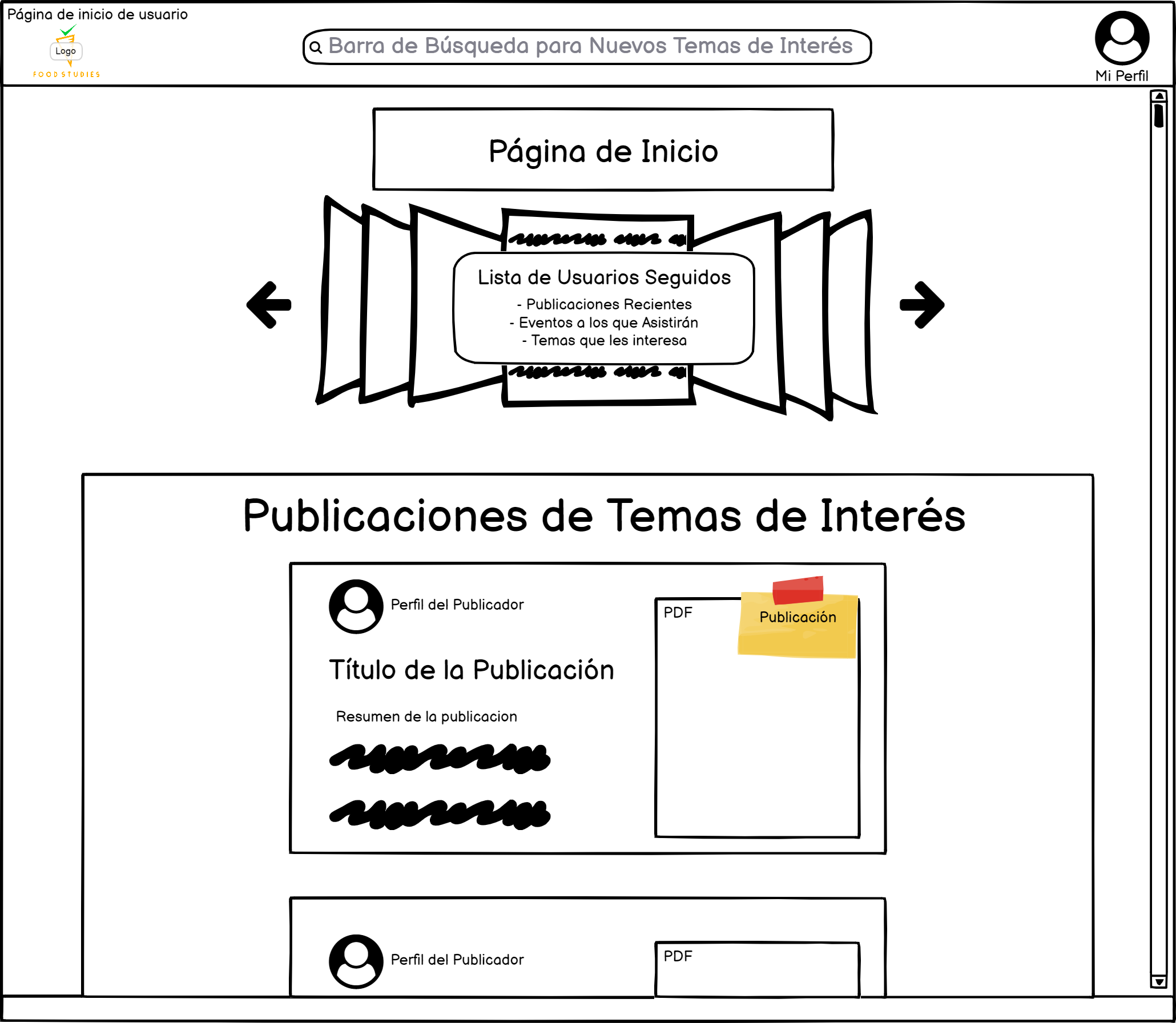
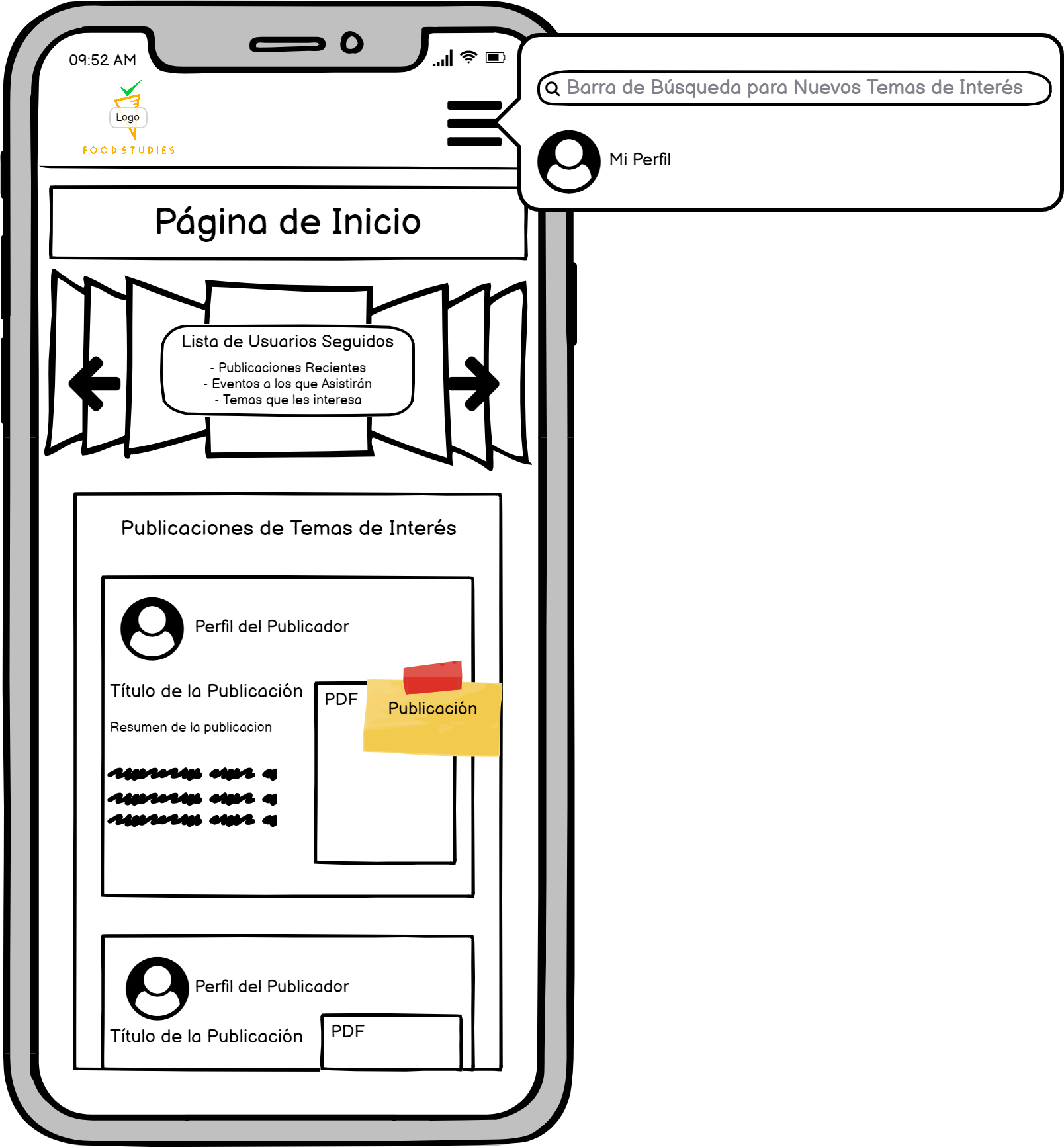
1. Iniciar aplicación
2. Llenar formulario de registro: se mostrará un formulario de inicio de registro en el que se tendrá que escribir el nombre y apellido, universidad a la que asiste, temas de interés, nombre de usuario, email y contraseña.
3. Validación: se comprobará si el nombre de usuario y el email son válidos, es decir que no haya ningún usuario registrado con esos mismos datos. En ese caso se seguirá el flujo base, en caso contrario se mostrará un mensaje de error y se volverá al paso dos.
4. Registrar: una vez los datos sean validados de forma correcta, se redirigirá a la página de inicio del usuario.

3.2 Página principal del usuario

Dentro de la página principal de un usuario con la sesion iniciada se podrán realizar distintas acciones, entre ellas están:



*Imagen 4. Diagrama de casos de uso de la página principal del usuario*



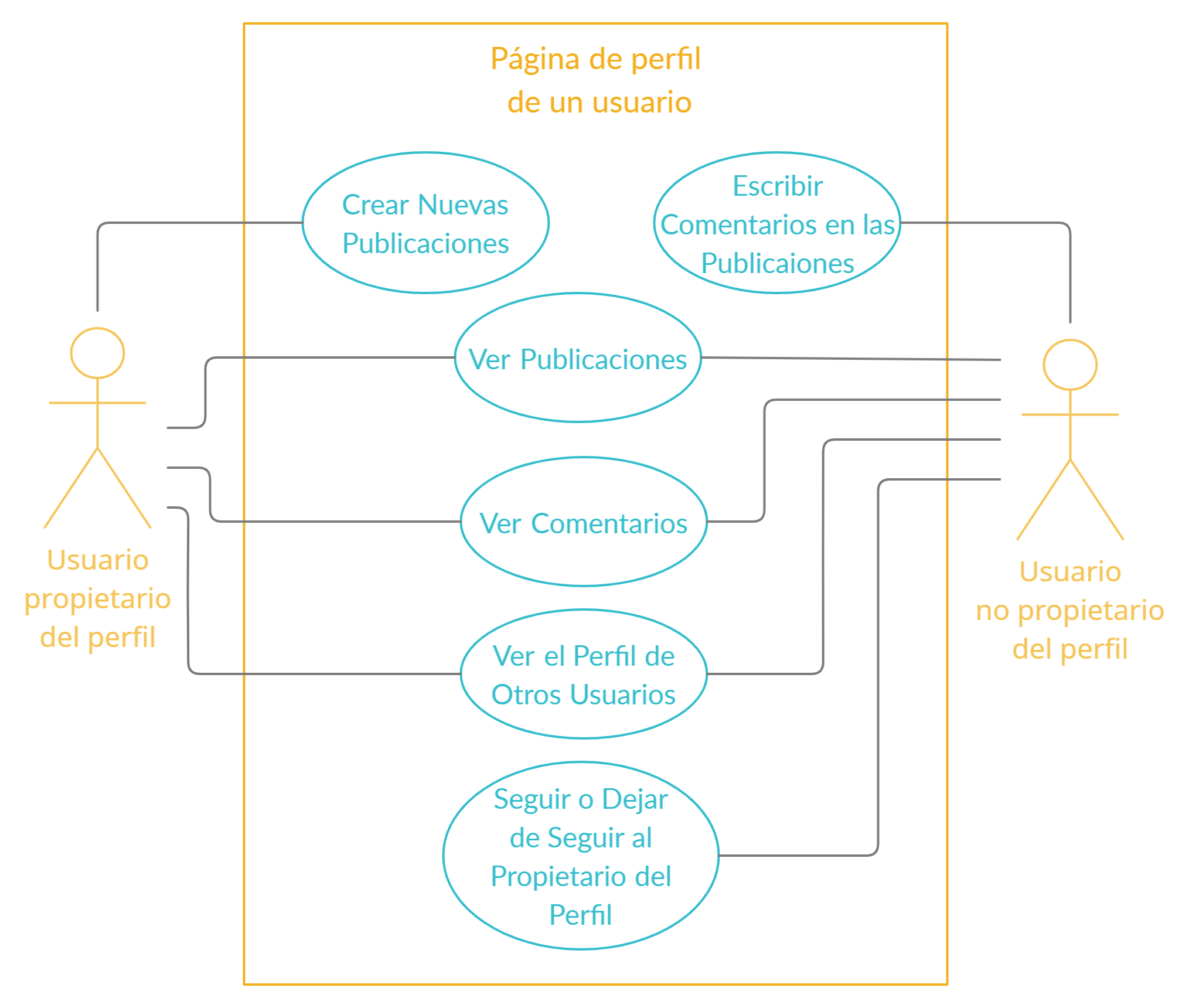
*Imagen 5. Boceto de la página de inicio un usuario con sesión iniciad*a

3.2.1 Ver publicaciones

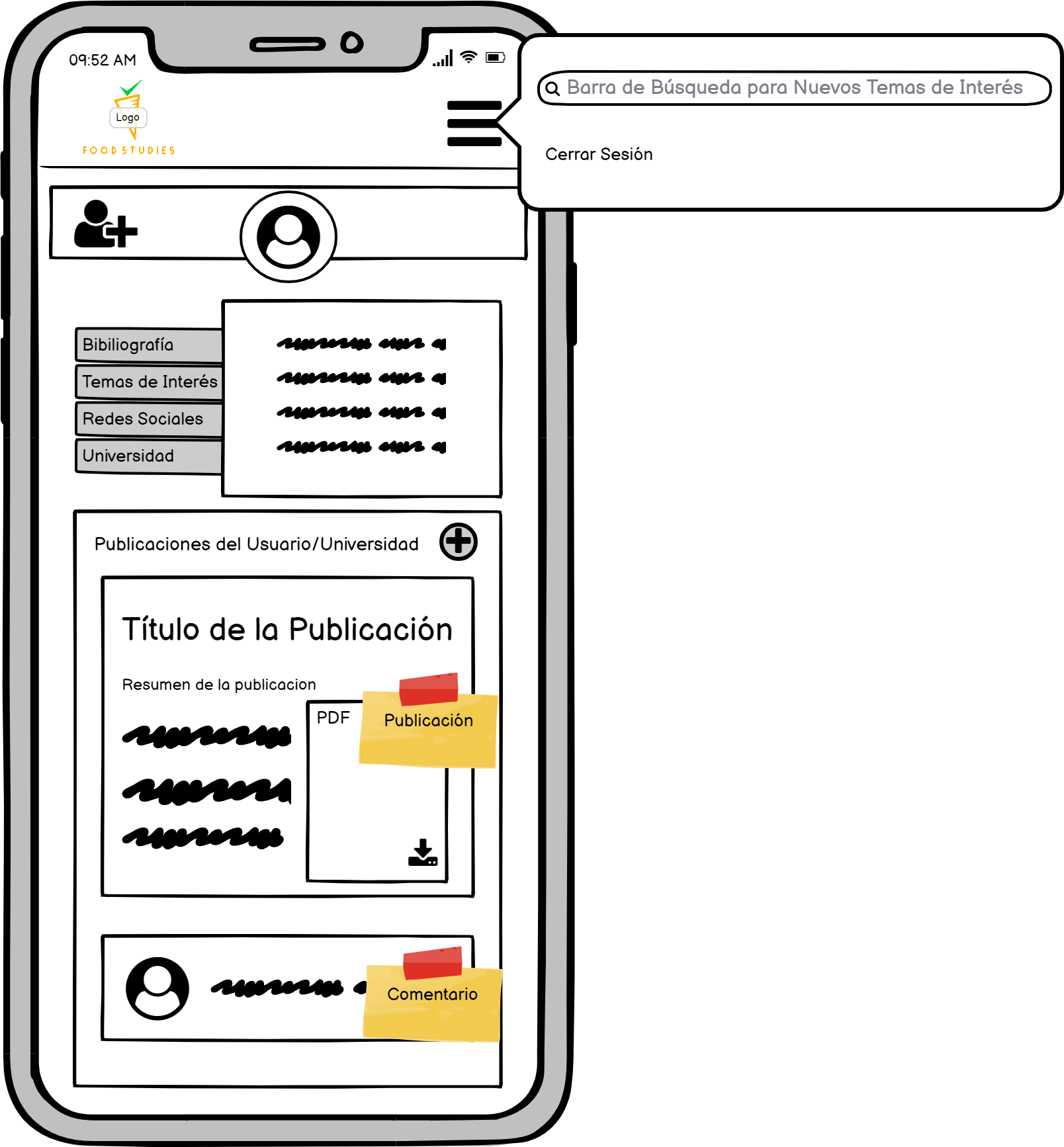
En la página principal se mostrarán dos tipos de publicaciones, estas serán las creadas por los usuarios seguidos dentro de un carrusel de donde se mostrará el título, las etiquetas relacionadas con los temas que toca, un resumen y un enlace a un archivo PDF. Más abajo se mostrarán más publicaciones que serán de usuarios no seguidos, de temas de interés seguidos por el usuario con la sesión iniciada, Al hacer click sobre cualquier publicación, se abrirá la versión completa de la misma dentro del perfil del usuario y ahí se podrá realizar la próxima acción de la aplicación.

3.2.3 Seguir otros usuarios

Para seguir a otros usuarios será necesario entrar al perfil del usuario desee seguir y hacer click en el botón.



*Imagen 6. Diagrama de casos de usos del perfil de un usuario*



*Imagen 7. Boceto de la página de perfil*

3.2.4 Ingresar al Perfil de Usuario

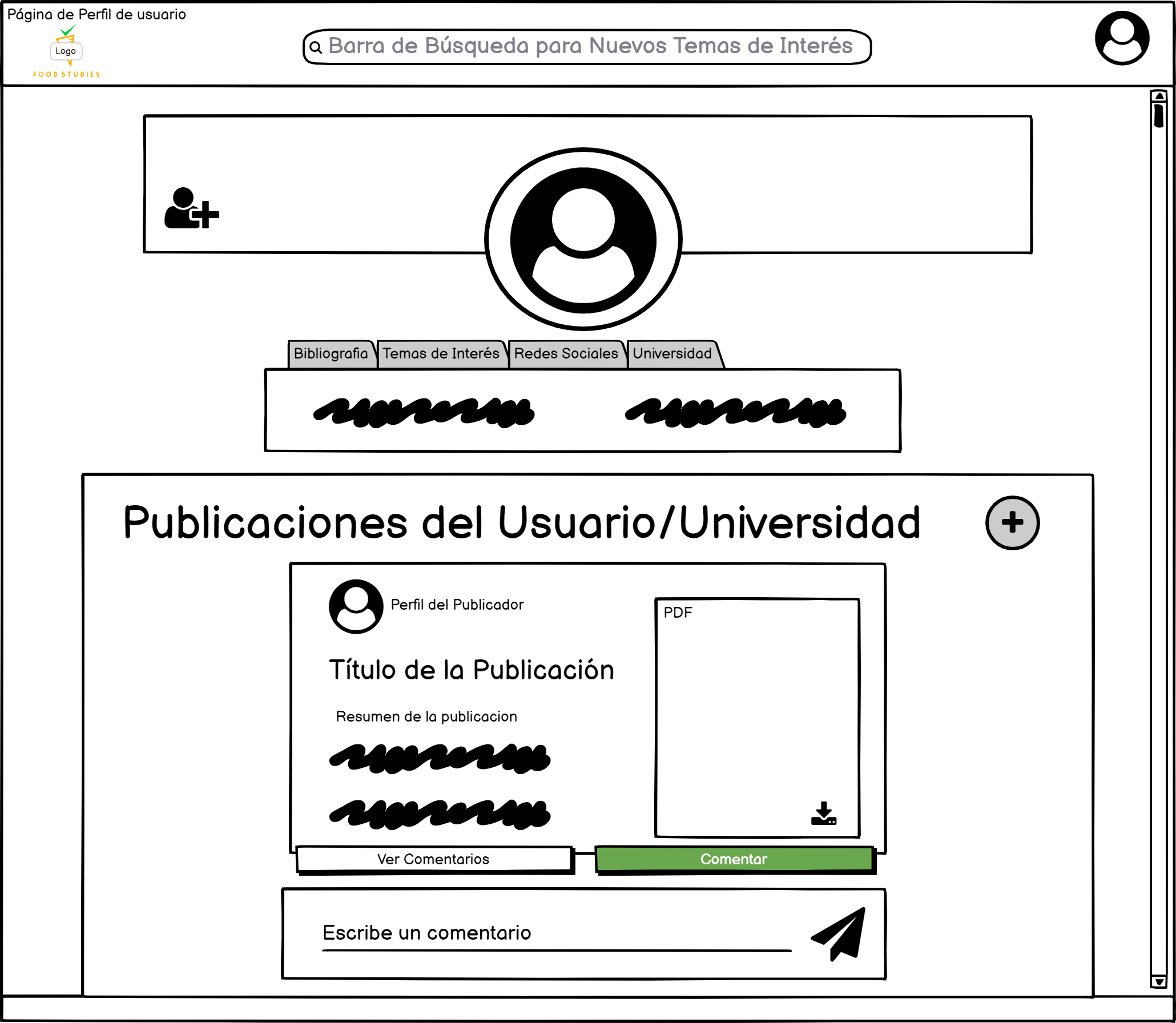
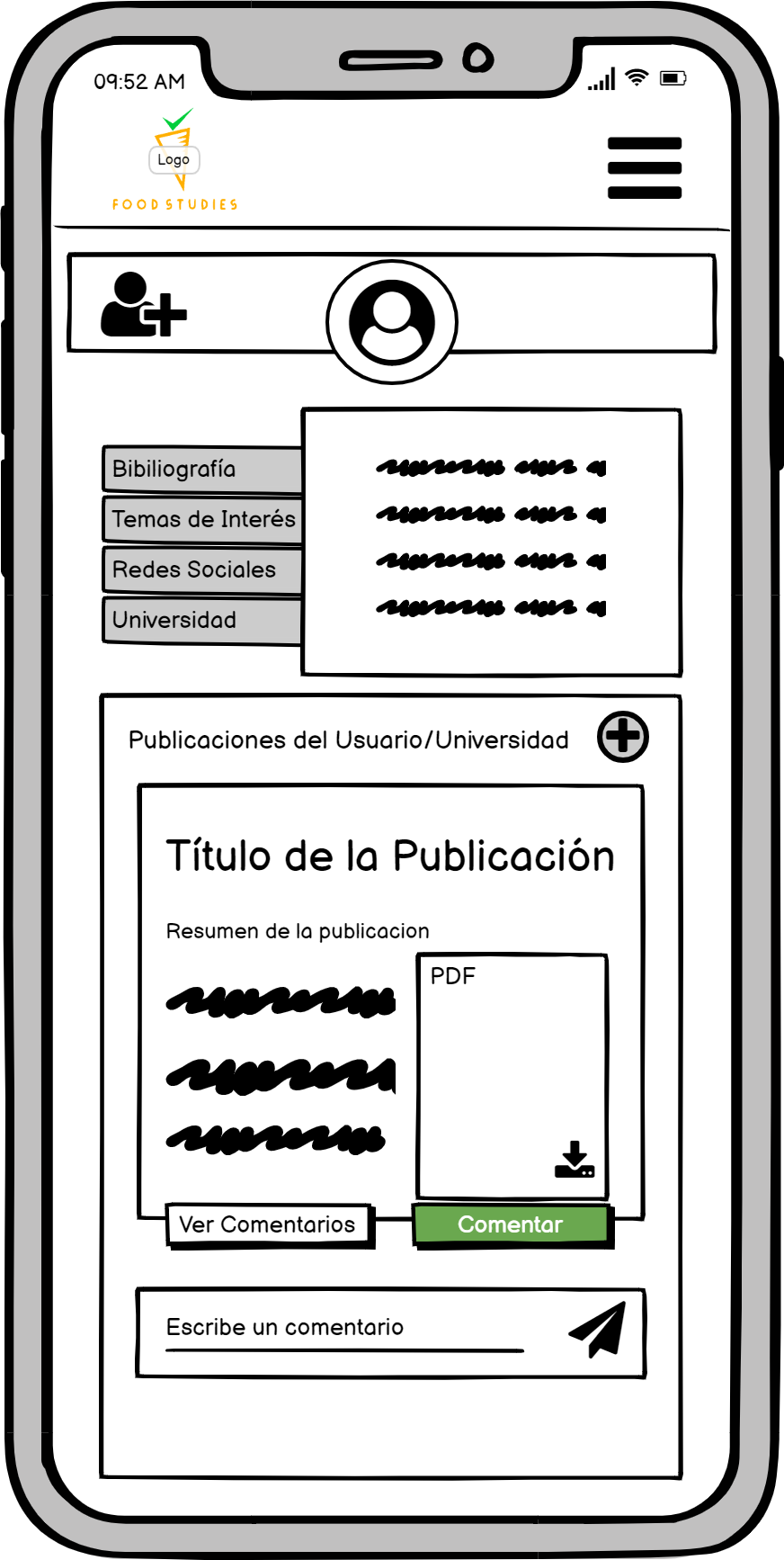
Para ingresar en el perfil de usuario con la sesión iniciada se tendrá que hacer click en el botón de perfil que se encuentra en la esquina derecha del header de la página. En el perfil se podrá revisar la bibliografía, los temas de interés, redes sociales, la universidad donde estudia, la imagen de perfil y las publicaciones creadas, además se podrá cerrar la sesión haciendo click en el botón del header a la derecha.

3.2.5 Ingresar al Perfil de Otros Usuarios

Para ver el perfil de otros usuarios bastará con hacer click sobre la imagen de perfil en cualquier usuario en la página de inicio de un usuario con la sesión iniciada. Una vez dentro de la pantalla del perfil del usuario deseado, se podrá ver las publicaciones, los temas de interés, las redes sociales, la universidad donde estudia y la opción de seguir al usuario.

3.2.6 Comentar en las Publicaciones de Otros Usuarios

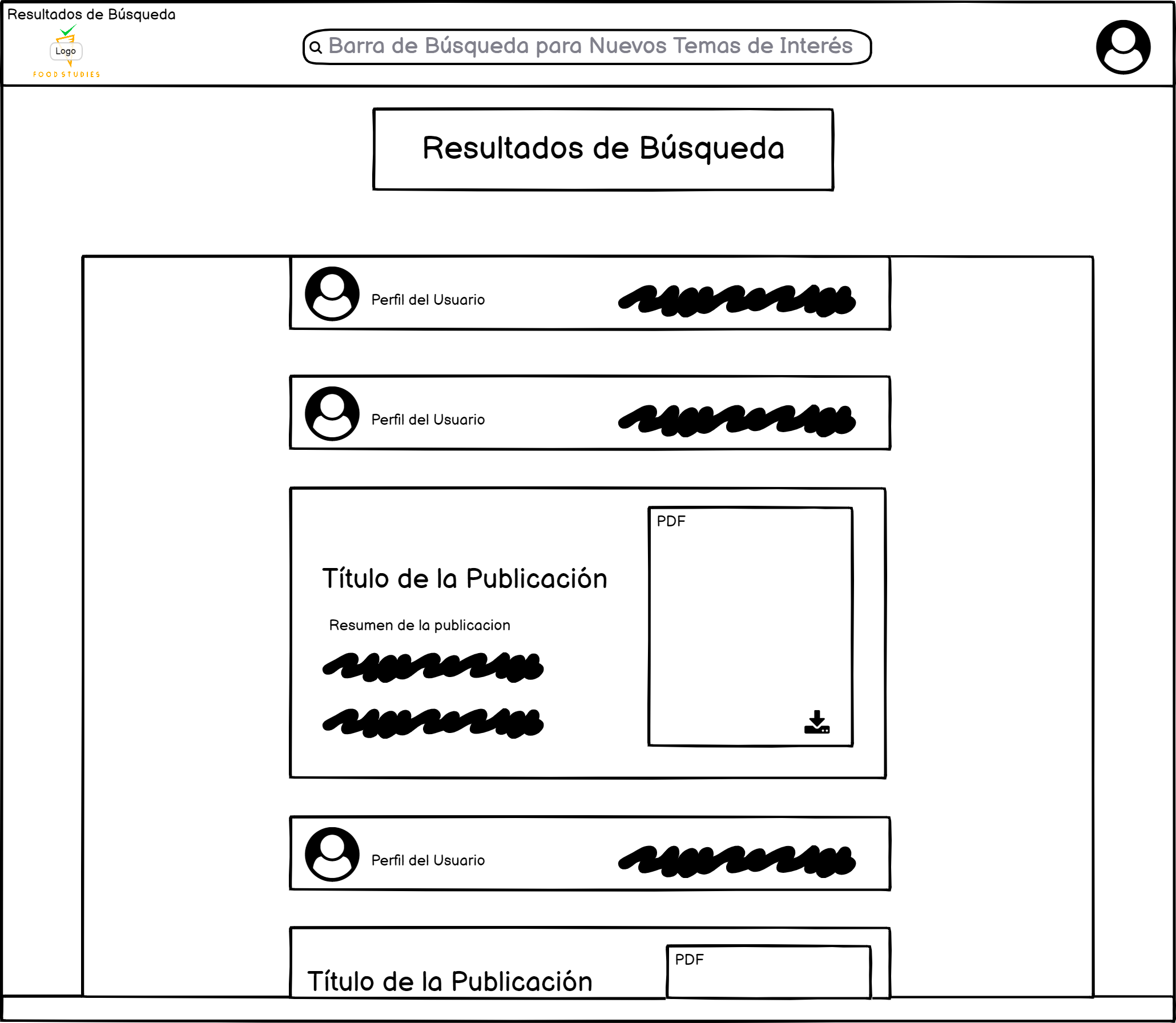
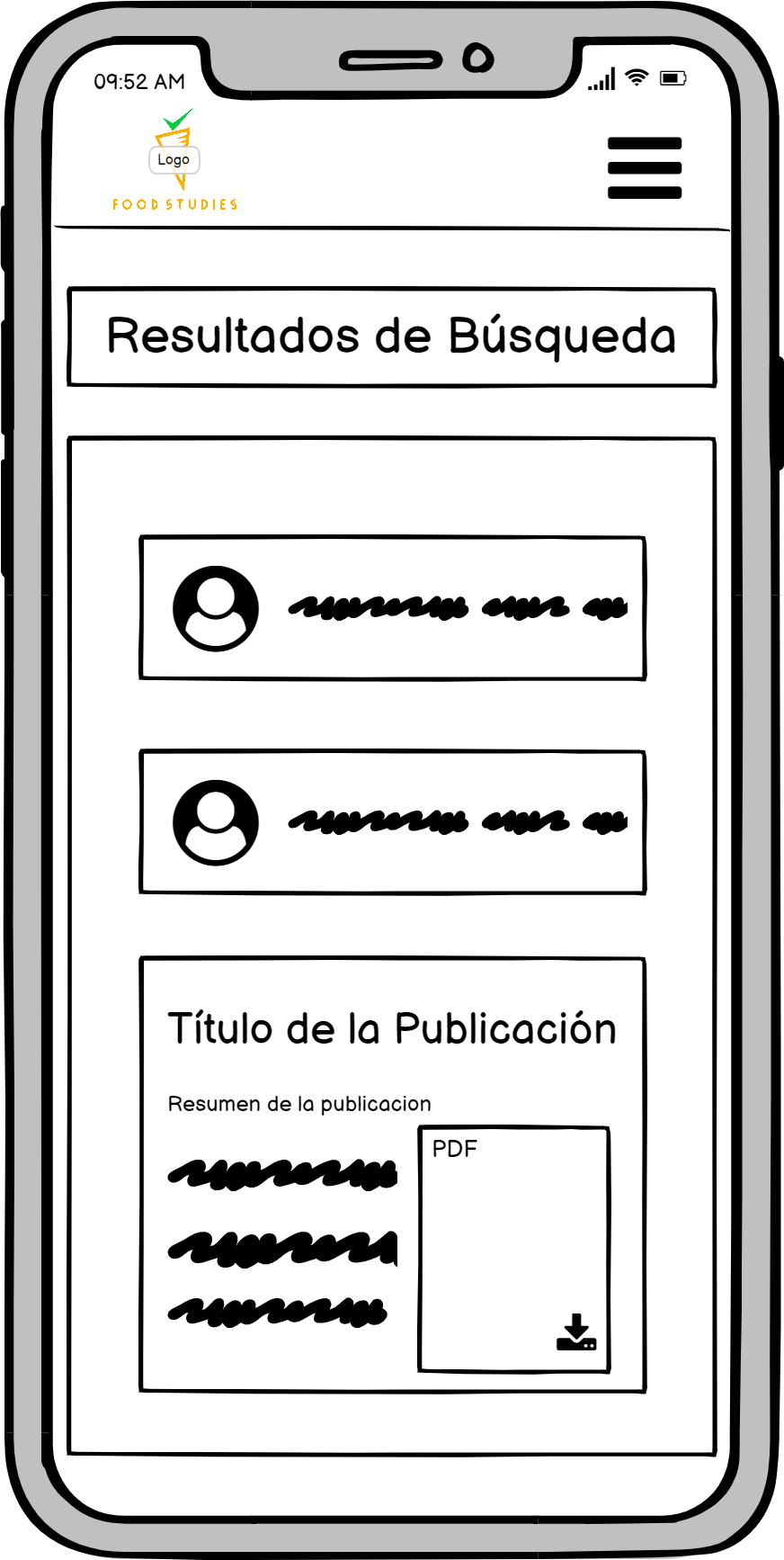
Cuando se quiera hacer un comentario en la publicación de otro usuario se deberá ir al perfil del publicador y hacer click sobre el botón de comentar situado en la esquina inferior derecha de cada publicación, esta acción mostrará un formulario para indicar el mensaje que se quiera publicar.



*Imagen 8. Boceto de la creación de comentarios*

3.2.6 Buscar Nuevos Temas de Interés o Usuarios

Para buscar nuevos temas de interés bastará con escribir en la barra de búsqueda que se encuentra siempre en parte superior de la página, una vez escrito el tema de interés se mostrarán los resultados de la búsqueda en la parte central de la página, tal y como se muestra en la siguiente ilustración.



*Imagen 9. Boceto de los resultados de búsqueda*

4. Planificación

4.1 Recursos

* Herramientas:

Para desarrollar el proyecto se usará Visual Studio Code ya que es un editor de código gratuito y bastante versátil.

* Tecnologías:

El desarrollo del proyecto completo se usará como gestor de paquetes NPM en la versión 7 y para la creación de las diferentes funcionalidades, ya sea del lado del servidor o del cliente, el MEVN stack que comprende cinco frameworks basados en javascript por lo que facilitará la programación y compatibilidad entre el servidor y el cliente. Dentro del lado del servidor se usará NodeJs con Express y MongoDB como base de datos no relacional, que al ser más rápida que una base de datos no relacional permitirá una mejor experiencia de usuario, todo esto correrá como un API a la cual hacerle las consultas a la base de datos para poder usar los resultados de las consultas en el lado del servidor, para probar la API de forma más rápida se usará Postman. En el lado del cliente se usará Vue 3 y Vue Router para crear una Single Page Application (SPA) con posibilidad de usar distintas rutas en la URL y para darle estilo a la página se usará Bootstrap 4 y CSS3 para personalizar más los diseños de las diferentes vistas del proyecto.

* Servicios:

Para los servicios de hosting del proyecto se usará Hostinger, ya que es uno de los servicios hosting más baratos que hay actualmente, además cuenta con un plan gratuito. Para sistema de control de versiones git, porque facilita bastante el trabajo en grupo y las pruebas de nuevas funcionalidades creadas, para esto se utilizará el framework jest que sirve para construir tests unitarios trabajando con matchers personalizados, crear mocks o comprobar snapshots de componentes visuales como algo sencillo y accesible.

* Temporales:

Para la realización de las funcionalidades básicas del proyecto se utilizarán aproximadamente tres meses, en promedio se estima realizar dos tareas por semana, esto incluye las pruebas con test unitarios por cada nueva funcionalidad creada

4.2 Tareas a llevar a cabo

Las tareas a llevar a cabo se dividiran en la creación de la API y el cliente:

4.2.1 Creación de la API:

Se pensó que la creación de todas las tareas de lógica y conexión con la base de datos del proyecto en una API ya que facilita la migración a otras plataformas como móviles o escritorio sin mucha dificultad, ya que solo habría que hacer peticiones a la API.

* **Crear el servidor**

Para crear el servidor se tendrá que iniciar un proyecto con NPM e instalar los paquetes de Express 4.17.1, Cors 2.8.5 para acceder a recursos seleccionados desde un servidor, en un origen distinto al que pertenece, Bcrypt 5.0.1 para la encriptación de las contraseñas de los usuarios, Passport y passport-local para facilitar la tarea de autenticar a los usuarios en la aplicación, Mongoose 5.12.2 para la conexión de la base de datos de MongoDB y Nodemon como herramienta de desarrollo.

* **Crear la base de datos**

La base de datos será creada en MongoDB y tendrá tres colecciones con sus respectivos campos:

Users:

* username
* fullName
* password
* email
* followedUsers
* followersUsers
* followedTopics

Universities:

* name
* location

Publications:

* publicatedBy
* content
* pdf
* comments
  + message
  + createdBy
* **Crear el registro de usuario**

A la hora de hacer peticiones a cualquier acción directamente relacionada con el usuario, se harán las peticiones a la URL que empiece con */api/user* asi que para iniciar sesión se hará una ruta de tipo POST llamada */register*, ahi se podra introducir varios datos como nombre de usuario, contraseña, email, nombre completo, universidad donde estudia y temas de interés relacionados con lo que estudia, luego esos datos que llegaran a la API mediante el request body y se verificará que nombre de usuario y el email sean únicos, además se encripta la contraseña por seguridad con bcrypt.

* **Crear el inicio de sesión**

Para iniciar sesión se hará una ruta de tipo POST llamada */login*, dentro de esta petición se le pasara desde el request body el nombre de usuario o email y contraseña del usuario, luego verifica si el usuario ha introducido los datos necesarios, de no ser así se devolverá una respuesta con status de error y un mensaje indicando que falta datos, si se han introducido tanto el nombre de usuario o email como la contraseña, se verificará dentro de la base de datos si el nombre de usuario o email se encuentra registrado, de no ser así se enviará un mensaje al usuario indicando que el nombre de usuario o contraseña son incorrectos. Una vez verificado que el nombre de usuario está registrado, se procede a encriptar la contraseña para poder compararla con la que se encuentra con la de la base de datos, si es correcta se envía un mensaje de éxito y si es incorrecta se enviará un mensaje de error.

* **Ver perfil del usuario**

Para ver el perfil de un usuario tiene que hacerse una ruta de tipo GET llamada */profile/:id del usuario*, entonces se buscará en la base de datos el id y si está, se enviará la información del usuario como el nombre de usuario, nombre completo, correo electrónico y temas de interés, en caso de que no se encuentre se enviará un mensaje de error.

* **Modificar perfil del usuario**

Para modificar el usuario de perfil se creará la ruta con el método PUT llamada */profile/:id del usuario*, primero se verificará si el usuario está autenticado con passport-local y passport, si el usuario que está intentando modificar no es dueño del perfil se enviará un mensaje de error, pero si la verificación es correcta, se permitirá modificar toda la información del perfil a través de los datos proporcionados por el request body, luego se actualizan las diferencias entre el perfil del usuario y los datos proporcionados.

* **Seguir o dejar de seguir a otros usuarios**

Para seguir o dejar de seguir a otros usuarios se tendrá que crear dos rutas distintas con el método GET estas son */follow/:id del usuario con sesión iniciada/to/:id del usuario seguir* y para dejar de seguir la ruta será */unfollow/:id del usuario con sesión iniciada/to/:id del usuario a dejar de seguir*. Dentro de estas rutas se leerán las id de ambos usuarios desde los parámetros de la URL, luego se buscará en la base de datos si el usuario con la sesion iniciada tiene en la lista de usuarios seguidos al que se desea seguir, si ya se encuentra en la lista, se enviará un mensaje indicando que ya se ha seguido con anterioridad al usuario, pero en caso contrario se añadirá la id del usuario a seguir en la lista de usuarios seguidos y en la lista de seguidores del usuario a seguir. En caso de querer dejar de seguir se hará el mismo procedimiento, pero con la diferencia de que se eliminará de ambas listas las id de los usuarios implicados en la petición.

* **Seguir o dejar de seguir temas de interés**

El funcionamiento de la ruta de para seguir o dejar de seguir temas de interés es parecida a la de seguir o dejar de seguir a otros usuarios, la ruta será */follow/:tema de interés* y para dejar de seguir */unfollow/:tema de interés* y se añadirá o se eliminará de la lista de intereses del usuario con la sesion iniciada.

* **Crear publicaciones**

Para crear las publicaciones se creará una una ruta con el método POST llamada */publication/publish* y recibirá dentro del request body el resumen de la publicación, los temas que trata la publicación y un fichero pdf. Una vez que se reciban los datos, estos crearán un nuevo registro dentro de la colección de publicaciones.

* **Ver publicaciones**

Para ver las publicaciones se creará una una ruta con el método GET llamada */publications/view* que devolverá todas las publicaciones con su información y para ver una específica, se creará la ruta */publications/view/:id de la publicación* esto solo mostrará la información de la publicación indicada.

* **Publicar comentarios en las publicaciones**

Para comentar en las publicaciones se creará la ruta con el método POST llamada */publications/comment/:id de la publicación* está recibirá el mensaje de la publicación y la id del usuario que lo escribió.

* **Realizar sistema de búsquedas**

Para realizar una búsqueda se creará una ruta con el método GET llamada */search/:palabra clave/:tipo de búsqueda* donde el segundo parámetro que se pasara por medio de la URL podrá ser utilizado para especificar el tipo de resultado que se quiere conseguir con la búsqueda, estos son búsqueda de usuarios, de temas de interés y de universidades, si no se pasa ningun parametro la busqueda se realizara sin filtros, por lo que pasara todos los resultados coincidentes con el primer parámetro de la URL y si no se encuentra ningún dato coincidente en la base de datos, se regresará un mensaje especificando.

4.2.1 Creación del cliente:

Para una primera versión del cliente se utilizará vue y vue router, por lo que se tendrá que hacer tres vistas y estas son:

* **Inicio**

Se creará una página de inicio que mostrará un breve resumen de la idea del proyecto y tres botones, uno que redirigirá a un formulario en la ruta /users/register, un botón que será para probar la aplicación que también redirigirá al mismo formulario y por último, un botón que redirigirá al formulario de inicio de sesión en la ruta /users/login.

* **Página principal del usuario**

La página principal mostrará la barra de búsqueda y todas las publicaciones de los temas de interés que sigue el usuario con la sesión iniciada.

* **Perfil de usuario**

En la página de perfil del usuario mostrar la información del propietario y sus publicaciones.

Los componentes necesarios para la creación y funcionamiento de la aplicación serán:

* **Formulario de inicio**

Será un formulario con el método POST con los campos username/email y password, al hacer submit hará una petición fetch a */api/users/login* y luego redirigirá a la página de inicio del usuario.

* **Formulario de registro**

Será un formulario con el método POST con los campos username, email, password, full name, interests, university y al hacer submit hará una petición fetch a */api/users/register* y luego redirigirá a la página de inicio del usuario.

* **Barra de búsqueda**

La barra de búsqueda tendrá una petición fetch para la ruta */search/:palabra clave/:tipo de búsqueda,* un botón que llevará al usuario a la página de inicio y otro que lo llevará al perfil del usuario.

* **Contenido principal del usuario**

Aca se mostraran los siguientes componentes:

* **Publicaciones de usuarios seguidos**

Para mostrar las publicaciones de los usuarios seguidos, se creará un componente que tendrá forma de carrusel, ya que estas publicaciones aparecerán con menos frecuencia que las de temas que sigue el usuario y que antes de cargar hará una petición a la API para encontrar las publicaciones de los usuarios que se encuentren en la lista de usuarios seguidos del que tenga la sesión iniciada.

* **Publicaciones**

Las publicaciones se mostrarán en un componente que hará una petición a la API que devolverá las publicaciones que puedan gustarle al usuario con la sesion iniciada, esto se conseguirá buscando publicaciones que traten temas que se encuentren en la lista de temas seguidos del usuario

* **Comentarios**

Los comentarios se cargaran consultando a la API las id de las publicaciones y cargando los comentarios asociados a estas, los comentarios solo serán visibles desde el perfil del publicador

4.3 Cronograma de actividades



*Imagen 10. Boceto de los resultados de búsqueda*

4.4 Metodología de trabajo

Para este proyecto se utilizará una metodología Agile es una gran herramienta para llevar a cabo proyectos con grupos interdisciplinarios. Es por ello que cada vez más compañías, pymes y emprendedores optan por aplicarlas en sus equipos de trabajo. En este caso se utilizara SCRUM esto permite abordar proyectos complejos desarrollados en entornos dinámicos y cambiantes de un modo flexible. Está basada en entregas parciales y regulares del producto final en base al valor que ofrecen a los clientes.

4.5 Validación

5. Conclusiones6. Bibliografía

*Herramienta creadora de diagramas de casos de uso*. (s. f.). App Creately. <https://app.creately.com/>

*Herramienta creadora de los diagramas de la estructura de la aplicación*. (s. f.). Lucidchart. <https://lucid.app/users/login>

*Qué son las metodologías ágiles en el desarrollo de Software*. (2018, 20 junio). [Vídeo]. YouTube. <https://www.youtube.com/watch?v=fHKsufzM7qQ>

*Bocetos de la aplicación*. (2016). Balsamiq Cloud. <https://balsamiq.cloud/>

*Create Infographic - Sign in*. (s. f.). Cronograma de La Aplicación. 2021, de <https://infograph.venngage.com/>