

# Ramen Stadium



## **La idea**

El plan es crear una aplicación web enfocada al estudio básico del ramen, con información como preparaciones de los aderezos, preparaciones de los caldos básicos etc.

La idea es atraer a los usuarios de la página creando una plataforma donde ellos mismos pueden crear su propio ramen, escogiendo desde el tipo de caldo hasta el último de los aderezos. Cada uno de los componentes que elijan tendrán su propia receta incluida, de forma que haga agradable y sencilla su preparación en casa.

Toda la información irá basada en un libro en formato cómic con diferentes recetas de los componentes de un plato de ramen.

## **Casos de usos dentro de la página**

Estos serán los usuarios de los que se compone mi aplicación:

- **Administrador**

- Control total sobre los usuarios registrados
- Visualizar una gráfica con datos de cuantas personas hay registradas y cuantas suscritas al premium
- Acceso de edición de algunos de los tutoriales

- **Usuario anónimo**

- Posibilidad de registrarse o loguearse
- Podrá ver el home pero para empezar tendrá que estar registrado
- Acceso a la historia sobre el ramen

- **Usuario registrado ( Plan gratuito )**

- Podrá crear ramen pero de forma limitada y no tendrá todos los tutoriales disponibles
- Podrá postear comentarios ( No ofensivos ) de su experiencia
- Todas las funcionalidades del usuario anónimo
- Podrá ver y postear las combinaciones que más le han gustado y a su vez valorar la de los demás
- Tiene anuncios.

- **Usuario registrado ( Con subscripción )**

- Podrá ver al completo todos los tutoriales además de que contará con combinaciones de ramen prediseñadas por expertos
- Todas las funcionalidades del usuario registrado
- No tiene anuncios
- Tendrá acceso a tutoriales de acompañamiento

## **Tecnologías pensadas para el proyecto**

El lenguaje a usar serán JavaScript para ello tengo pensado usar el stack mern que se compone de un conjunto de tecnologías que son:

- MongoDB:
  - Mongo Atlas
- ExpressJS
- ReactJs
  - NgPrime ( Librería de estilos )
  - Uso de Hooks
- NodeJs:
  - AUTH0 ( Registro y Login con google )
  - Bcrypt para encriptar contraseñas
  - Multer + Cloudinary ( U otro servidor de imágenes )

## **Diseño**

Ajustaré el diseño a lo que son las imágenes del libro que voy a coger de referencia.

También usaré varios factores visuales a la hora de navegar dentro de la aplicación como pueden ser Barras de carga, spinners de carga, paginación con animación de transición, ventanas emergentes con consejos etc.

## **Alcance**

Tras analizar y haber pensado en todas las funcionalidades que se me han podido ocurrir para la aplicación, en un principio el mínimo que me impongo es todo lo descrito y como extra para un mayor alcance sería implementar una audio guía, teniendo que usar otras librerías para leer archivos en formato imagen y finalmente convertirlo en pdf para que pueda ser extraído la información en forma de objeto.

## **Esquema base de datos no relacional**

### **Usuarios**

Atributos: ID de usuario (clave primaria), nombre de usuario, dirección de correo electrónico, contraseña, otros atributos específicos de los clientes premium (por ejemplo, nivel de membresía, beneficios adicionales, etc.)

### **Admin**

Atributos: ID de admin (clave primaria), nombre de admin, dirección de correo electrónico, contraseña, permisos (un array de permisos )

### **Tutoriales**

Atributos: ID de tutorial (clave primaria), título del tutorial, descripción del tutorial, pasos o instrucciones para hacer el ramen( una serie de imagenes que funcionarán como el tutorial ) , ID de usuario que creó el tutorial ( una referencia al ID de usuario), nivel de dificultad, tiempo de preparación, puntuación media de los otros usuario (calculada con la puntuación de los comentarios )

### **Platos**

Atributos: ID de plato (clave primaria), nombre del plato, partes del plato (un array de id para relacionarlos con los tutoriales ) , ID de usuario que creó el plato (una referencia al ID de usuario), nivel de picante

## **Comentarios**

Atributos: ID de comentario (clave primaria), contenido del comentario, fecha y hora del comentario, calificación del plato, ID de usuario que realizó el comentario ( referencia al ID de usuario), ID de plato al que se refiere el comentario ( una referencia al ID de plato ).

## **TABLA DE INFORMACIÓN DE LA API REST**

Servicio	Descripción	Método HTTP	Ruta
Registro de un usuario	Registra a un usuario con su email o por su cuenta de Google o Facebook	POST	/register
Login de un usuario	Loguea a un usuario mediante su email y contraseña, o con la cuenta de Google o Facebook, y *obtener el token de sesión. Si al hacer login se detecta que es un admin se le redireccionará a una pantalla diferente.	POST *GET	/login
Obtener todos los tutoriales	Obtiene todos los tutoriales, estos serán una recopilación de imágenes	GET	/tutorial
Obtener un tutorial en concreto para editarlo	Obtiene un tutorial en concreto para editarlo por el admin	PUT	/admin/tutorial:{id}
Obtener	Obtiene todos los	GET	/usuario:{id}

comentarios por usuario	comentarios de un usuario		
Obtener comentarios por plato	Obtiene todos los comentarios de un plato	GET	/platos:{id}/comentarios
Subir un comentario de un plato	Sube un comentario pasandole un filtro de palabras sensibles	POST	/platos:{id}/comentarios
Obtener todos los platos de un usuario	Obtiene todos los platos hechos o cogidos de otros usuarios por el usuario	GET	/usuario:{id}/platos
Obtener todos los platos de la base de datos	Obtiene todos los platos en la base de datos	GET	/platos
Obtener plato	Obtiene un plato en concreto con toda su información	GET	/platos:{id}
Eliminar plato	Elimina un plato creado por el usuario de la base de datos de un usuario o de la lista de platos guardados por el usuario. Es decir si el plato es suyo se eliminará para siempre de la base de datos pero si es de otra persona lo eliminará simplemente de su lista.	DELETE	/usuario/platos:{id}
Eliminar usuario	Elimina el registro de una cuenta en la página	DELETE	/admin/usuario:{id}
Eliminar comentario	Elimina un comentario de un plato	DELETE	/admin/plato:{id}/comentario:{id}

Actualizar la información de un usuario	Actualizar información personal del usuario	PUT	/usuario:{id}
Añadir un tutorial a la aplicación	Añadir un nuevo tutorial siendo admin	POST	/admin/tutoriales:{id}