

CHALLENGE - React Native

Objetivo

Desarrollar una aplicación para crear una carta de opciones de menús para un hotel que consumirá una API externa y mostrará diferentes atributos a nivel individual de cada plato y del menú finalizado.

👉 Consumir los endpoints de la siguiente API para realizar las distintas operaciones. Deberás autenticarte en la plataforma para obtener una ApiKey y poder realizar las peticiones.

Requerimientos funcionales

En la pantalla de Home se deberá mostrar, además de los platos que conforman el menú:

- Acumulativo de precio del menú.
- Promedio de tiempo de preparación entre todos los platos.
- El menú debe tener 4 platos. Debe haber 2 veganos y 2 que no lo sean. Esto debe validarse al intentar agregar un nuevo plato.
- Se deberá poder eliminar un plato del menú, lo que generará nuevamente los promedios y acumulativos (los mismos deben estar almacenados en el estado del componente utilizando Hooks)

Requerimientos técnicos

Aprovechando las características de React, deberán crearse las siguientes secciones, y modularizar las mismas en componentes reutilizables.

Además, para el manejo de peticiones HTTP deberá utilizarse la librería Axios.

1. Formulario de Login

El formulario se deberá renderizar al ingresar a cualquier ruta si el usuario no está autenticado, conteniendo los campos:

- Email.

- Password.
- Botón de “Enviar”.

Al hacer click en “Enviar”, se deberá validar que ambos campos no estén vacíos, y mostrar un mensaje al usuario si lo estuviesen. Caso contrario, se deberá realizar una petición POST a la siguiente url, con los campos email y password en el BODY.

Los datos válidos para obtener un token son:

- Email: challenge@alkemy.org
- Password: react

Se debe mostrar algún tipo de feedback al usuario mientras se está procesando la petición, no permitiendo que vuelva a accionar el botón de login hasta obtener una respuesta.

En el caso de obtener un error de la API, se deberá mostrar una alerta, mientras que si es satisfactorio deberá redirigir al Home y almacenar el token obtenido en el contextState. Para realizar las validaciones no es necesario utilizar ninguna librería.

2. Platos

El Home de la aplicación mostrará los platos del menú en un listado. Cada ítem (el cuál debe ser un componente separado) del listado contendrá:

- Nombre del plato.
- Imagen.
- Características del plato.
- Acciones para ver el detalle o eliminarlo del menú.

3. Buscador de Platos

Se debe agregar una nueva sección que permita agregar platos al menú, para esto, se deberá visualizar un formulario que realice una petición GET al endpoint de búsqueda y muestre los resultados disponibles en un flat list, utilizando el componente de ítem del punto anterior.

El formulario deberá buscar únicamente si hay más de 2 caracteres en el filtro, caso contrario no debe mostrar nada.

4. Detalle del Plato

Al hacer click en un plato del menú, se deberá mostrar los detalles de ese plato en caso de que ya este en el menú. O un botón de agregar al menú en caso de que no esté. Al agregar el plato al menú se va a guardar en el contextState.

5. Navegación entre secciones

Las diferentes secciones que tendrá la app deberán estar visibles únicamente si el usuario está autenticado y disponga de un token almacenado en el contextState. El mismo, se obtendrá de una API con datos de muestra.