



ACTIVIDAD 1

SERVICIOS WEB CON IONIC

EJERCICIO 1: MIS RECETAS

Desarrollar una aplicación en **Ionic** que permita gestionar un catálogo de recetas realizando un **CRUD completo** contra la API de DummyJSON (<https://dummyjson.com/recipes>) usando estos endpoints:

- GET /recipes
- GET /recipes/{id}
- POST /recipes/add
- PUT /recipes/{id}
- DELETE /recipes/{id}

LISTADO DE RECETAS (TAB O PÁGINA PRINCIPAL)

- ion-header + ion-toolbar + ion-title
- ion-refresher (pull to refresh) o botón “Recargar”
- ion-list con ion-item:
 - ion-thumbnail (imagen)
 - nombre
 - dificultad/tags
 - tiempo total (prep + cook) si existe
- Botón “+” en header (ion-fab o botón en toolbar) para crear nueva receta

DETALLE DE RECETA

- Botón volver
- Imagen grande (ion-img)
- ion-card con:
 - nombre
 - dificultad
 - ingredientes
 - instrucciones
 - tags

Botones obligatorios:

- Editar → navega a /recipes/:id/edit
- Eliminar → confirma con ion-alert y borra



CREAR RECETA

Debe ser un formulario reactivo (obligatorio):

Campos mínimos:

- name (required, minlength 3)
- difficulty (required) → ion-select
- prepTimeMinutes (min 1) → ion-input type="number"
- cookTimeMinutes (min 0) → ion-input type="number"
- ingredients (string con comas) → ion-textarea
- instructions → ion-textarea

Comportamiento:

- Botón “Guardar” deshabilitado si inválido
- Al guardar:
 - POST /recipes/add
 - mostrar confirmación con ion-toast o ion-alert
 - volver al listado

EDITAR RECETA

- Reutilizar el formulario o crear uno similar.
- Cargar receta actual (GET /recipes/:id) y pre-rellenar
- Guardar con:
 - PUT /recipes/:id
- Mostrar ion-toast “Receta actualizada”

ELIMINAR RECETA (DESDE LISTADO Y/O DETALLE)

Obligatorio:

- Confirmación con ion-alert
- Llamar a DELETE /recipes/:id
- Eliminar de la lista en memoria (signal)
- Mostrar ion-toast “Receta eliminada”



LOGIN

- Crear una opción en el menú Login que mostará el texto login si no estás logueado o logout si lo estás.
- Crear un componente login y un servicio para llamar a la api <https://dummyjson.com/auth/login>
- Esta api nos devuelve un token que guardaremos en localstorage, que será lo que marque si estamos logueados o no.
- Crear un guard para proteger rutas en función de si estás logueado o no. Proteger la ruta para crear recetas, de forma que solo se pueda acceder si estás logueado.