# Prueba Tecnica - Frontend React

**Contexto.** Esta prueba evalua tu capacidad para construir una app web en React que consume APIs REST, maneja autenticacion, estado, formularios y buenas practicas de arquitectura/UI.

**Tiempo sugerido.** 2 a 4 horas. Si no alcanzas a completar todo, prioriza los entregables "Obligatorio" y documenta lo faltante en el README.

**Referencia UI (Figma).** Se te compartira un enlace a Figma con pantallas de referencia. Puedes ajustar textos/labels si lo necesitas. El objetivo es aproximarte al diseno y mantener una UX limpia (loading, errores, exito).

**Nota sobre APIs.** Vas a consumir endpoints en dos subdominios distintos (es intencional). Maneja CORS/headers y token como corresponda.

## Objetivo

Construir una aplicacion en React que implemente: (1) Login con token, (2) Dashboard con listado paginado, y (3) un formulario para crear una nueva Accion. Se evaluara claridad, orden, manejo de estado/errores y calidad de implementacion.

## Requerimientos tecnicos generales

- React (preferiblemente React 18).
- JavaScript o TypeScript (TypeScript es un plus).
- Manejo de estado (Context API, Redux, Zustand, etc.).
- Manejo de formularios (Formik, React Hook Form o equivalente).
- Consumo de APIs (fetch o axios).
- Manejo de errores y loading states.
- Buenas practicas de componentes y separacion de responsabilidades.

## 1. Autenticacion (Login)

Endpoint:

POST /api/Authentication/Login
https://dev.apinetbo.bekindnetwork.com/api/Authentication/Login

Payload de ejemplo:

```
{
  "username": "a.berrio@yopmail.com",
  "password": "AmuFK8G4Bh64Q1uX+IxQhw=="
}
```

Comportamiento esperado (obligatorio):

- Mostrar loader mientras se autentica.
- Guardar el token y redirigir al Dashboard si es exitoso.
- Mostrar mensaje de error claro si falla.

Plus (opcional):

- Validaciones basicas del formulario.
- Proteccion de rutas privadas (si no hay token, redirigir a Login).

## 2. Dashboard - Listado paginado de Acciones

Endpoint:

GET /api/v1/actions/admin-list
https://dev.api.bekindnetwork.com/api/v1/actions/admin-list?pageNumber=1&pageSize=10

Comportamiento esperado (obligatorio):

- Mostrar una tabla/lista con la informacion devuelta por el API.
- Implementar paginacion usando pageNumber y pageSize (por defecto 10).
- Estados de UI: loading, empty state y error state.
- El token debe enviarse en cada request (segun el esquema que retorne el login).

Nota:

- El diseno en Figma puede mostrar una pantalla de "Categorias". Para esta prueba, la entidad funcional es "Acciones". Puedes adaptar el UI para que los textos digan "Acciones" o mantener la estructura visual y renombrar labels.

## 3. Crear Accion (formulario)

Endpoint:

POST /api/v1/actions/admin-add
https://dev.api.bekindnetwork.com/api/v1/actions/admin-add

Criterio minimo (obligatorio):

- Construir un formulario para crear una Accion.
- Seleccionar e implementar al menos 3 campos (a tu criterio) basados en la respuesta del API o el diseno de Figma.
- Validar lo basico (requeridos, formatos) y mostrar feedback visual (exito/error).
- Al crear exitosamente, refrescar el listado o navegar de regreso al dashboard.

Ambiguedad intencional (se evalua):

- El API no documenta aqui el payload exacto: debes inferirlo explorando la respuesta del listado o probando el endpoint. Documenta tus decisiones en el README.
- Si identificas que el endpoint requiere archivo (upload), implementalo. Si no es posible, deja el placeholder en UI y explica como lo resolverias.

## Arquitectura sugerida (referencia)

```
src/
  api/
  components/
  pages/
  hooks/
  context/
  routes/
  utils/
  styles/
```

## QA funcional (obligatorio)

Adjunta un archivo QA_CHECKLIST.md (o .txt) con al menos 10 pruebas funcionales del flujo completo (login, listado, paginacion, crear Accion).

## Entregables

- Repositorio público en GitHub (obligatorio).

- README.md con instrucciones para correr el proyecto localmente (npm install, npm start o equivalente).
- Indicar en README: decisiones tecnicas, librerias usadas y cualquier supuesto que tomaste.
- Archivo QA_CHECKLIST.md (opcional): checklist de 10 pruebas funcionales (máx. 1 página).