

Prueba Técnica — React Native + PokéAPI

Objetivo

Desarrollar una **aplicación móvil** con **React Native (0.74+ y TypeScript)** que consuma datos desde la **PokéAPI** (https://pokeapi.co/).

La aplicación debe demostrar dominio en arquitectura front-end, manejo de estado, optimización de rendimiento y experiencia de usuario.

Requerimientos técnicos

- Framework: **React Native 0.74+**
- Lenguaje: **TypeScript**
- Estado global: **Zustand** (o similar)
- Manejo de red y caché: **React Query** o equivalente
- Persistencia local: **AsyncStorage** o **MMKV**
- Navegación: **@react-navigation/native**
- Imágenes optimizadas: **react-native-fast-image**
- Testing (opcional): **Jest + Testing Library**

API de referencia

Base URL: ``https://pokeapi.co/api/v2/``

Endpoints sugeridos:

- ``/pokemon?offset=0&limit=20`` → Listado con paginación
- ``/pokemon/{name}`` → Detalle del Pokémon
- ``/type/{name}`` → Información de tipos
- ``/ability/{name}`` → Información de habilidades

Puntos a cumplir

- ☐ Listado paginado de Pokémons (infinite scroll o paginación clásica)
- ☐ Ficha de detalle al tocar un Pokémon con sprite, estadísticas, tipos y habilidades
- ☐ Búsqueda por nombre (con auto-complete o debounce para optimizar peticiones)
- ☐ Filtros por tipo (agua, fuego, planta, etc.)
- ☐ Filtro adicional por nombre
- ☐ Guardar ****favoritos localmente**** (debe existir persistencia al cerrar y reabrir la app)
- ☐ Implementar ****caché de datos**** para evitar refetches innecesarios (React Query u otra estrategia)
- ☐ Toda ****paginación consultada**** debe seguir mostrando contenido **offline**
- ☐ Cada ****búsqueda realizada o tarjeta consultada**** debe almacenarse y mostrarse **offline**

Arquitectura

Cada componente principal debe de constar de 5 partes, su TS de fetchs, controlador, estilos, index donde se centralizan los imports y finalmente el layout

Entregables

- Código fuente en un repositorio público o privado (GitHub / GitLab)
- Archivo `README.md` con:
- Descripción general del proyecto
- Pasos de instalación y ejecución (`yarn install`, `yarn android`/`yarn ios`)
- Notas adicionales o decisiones técnicas relevantes

NOTA:

1. Una vez que recibas la prueba tendrás 24 hrs para desarrollarla y entregarla.
2. Si no concluyes todos los puntos de la prueba igual puedes enviarla para cumplir con la entrega en tiempo y forma, se valorará los puntos logrados en el desarrollo