Prueba Técnica — Frontend

React / React Native

Objetivo

Desarrollar una aplicación frontend que consuma datos desde la PokéAPI y demuestra dominio en arquitectura Front, manejo de estado, estrategias offline-first, optimización de rendimiento y experiencia de usuario.+

La prueba puede realizarse en:

- React JS
- React Native (se valora más)

Tiempo

- 24 horas a partir de la recepción.
- No es obligatorio completar el 100%
- Se evaluará calidad sobre cantidad y las decisiones técnicas tomadas

Api de referencia

https://pokeapi.co/api/v2/

Endpoints sugeridos:

- /pokemon?offset=0&limit=20
- /pokemon/{name}
- /type/{name}
- /ability/{name}

STACK TÉCNICO ESPERADO

Requisitos Obligatorios

- TypeScript
- Estado global con Zustand (o similar)
- Data fetching y caché con React Query (o equivalente)
- Arquitectura modular
- Manejo de errores, loading y estados vacíos

REACT NATIVE (MOBILE) — REQUISITOS DE ENTORNO

- React Native versión **0.74.5**
- Node **18 LTS** (o compatible con RN 0.74.5)
- **npm** como manejador de paquetes
- Navegación con @react-navigation/native
- Persistencia local con AsyncStorage
- Imágenes optimizadas con react-native-fast-image

- Android Studio con SDK y Platform Tools
- Gradle Wrapper incluido en el repositorio
- Java **JDK 17** (Temurin o Zulu recomendado)

Notas para revisión:

- El proyecto debe ejecutarse con:

```
None

        npm install

        npm run android (o npm run ios si aplica)
```

- El README debe indicar claramente:
    - versión de Node
    - versión de Java
    - cómo configurar `JAVA_HOME`

REACT JS — REQUISITOS

- Framework: Vite + React js
- TypeScript obligatorio
- Manejo de rutas:
    - React Router (rutas anidadas y query params) o
    - Next.js App Router (recomendado)

En React js es obligatorio:

- Virtualización del listado (react-window o equivalente)
- Filtros y búsqueda sincronizados con la URL
- Accesibilidad mínima:
    - Navegación por teclado
    - Labels correctos
    - Focus visible

FUNCIONALIDADES MÍNIMAS

Core:

- Listado paginado de Pokémon
- Vista de detalle con imagen, estadísticas, tipos y habilidades
- Búsqueda por nombre con debounce
- Filtro por tipo
- Filtro combinado (tipo + nombre)

Favoritos y persistencia:

- Guardar favoritos localmente
- Persistencia entre sesiones
- Vista de favoritos

Offline-first y caché:

- Cachear información para evitar refetches
- Ver listado y detalle previamente consultados sin conexión
- Historial offline de búsquedas y vistas

DISEÑO RESPONSIVO

- Soporte mobile (≥360px), tablet (≥768px), desktop (≥1024px)
- Layout adaptable
- Sin dependencia exclusiva de hover
- Sin scroll horizontal no intencional

ARQUITECTURA

La aplicación debe organizarse por **features/pantallas**, manteniendo una estructura consistente en todo el proyecto.
Esta estructura es una **referencia de organización**, no un ejemplo de implementación.

Cada feature principal debe contener los siguientes archivos:

- index.ts
  Centraliza los exports del feature.
- Layout.tsx
  Contiene únicamente la UI (presentación).
  No debe incluir lógica de negocio ni llamadas directas a servicios.
- styles.ts
  Define los estilos del feature.
- services.ts
  Define las funciones relacionadas con:
  - consumo de APIs
  - acceso a storage
  - integración con librerías de red o cache

- useFeatureController.ts
  Orquesta el estado, acciones del usuario y coordinación entre UI y servicios.

ESTRUCTURA DE REFERENCIA

Esta estructura es únicamente ilustrativa para mostrar el patrón esperado:

```
None
/src
├── features/
| ├── FeatureA/
| | ├── index.ts
| | ├── Layout.tsx
| | ├── styles.ts
| | ├── services.ts
| | └── useFeatureController.ts
| |
| ├── FeatureB/
| | ├── index.ts
| | ├── Layout.tsx
| | ├── styles.ts
| | ├── services.ts
| | └── useFeatureController.ts
|
├── shared/
| ├── components/
| ├── hooks/
| └── utils/
|
└── main.tsx (o App.tsx)
```

Entregables

1. Repositorio público (GitHub o GitLab).

2. README con instalación, ejecución y decisiones técnicas

3. Versions used:

- Plataforma
- Node
- npm
- React Native 0.74.5 (si aplica)
- Java JDK 17 (si aplica)

4. Evidencia visual