

Licenciatura en Sistemas de Información
Desarrollo y Arquitectura en aplicaciones móviles – 2025

ALUMNO: NICOLAS CARDINAUX

Link GitHub Repo:

<https://github.com/NicolasCardinaux/ProyectosProgMovil.git>

Ejercicios

02 - a Fetch a API pública + Búsqueda + Pull-to-refresh

Consigna

- Consumir <https://pokeapi.co/api/v2/pokemon?limit=50>.
- Mostrar estado cargando (spinner) o skeleton y errores de red.sedwqwef
- Agregar un [TextInput](#) para filtrar por nombre (client-side).
- Implementar pull-to-refresh que re-fetchée la lista.
- Mantener la búsqueda aplicada tras el refresh.

Criterios de aceptación

- Al abrir la app: se ve [ActivityIndicator](#) hasta que termina el fetch.
- Error de red muestra un mensaje visible y permite reintentar con pull-to-refresh.
- El filtro es case-insensitive y filtra en tiempo real.

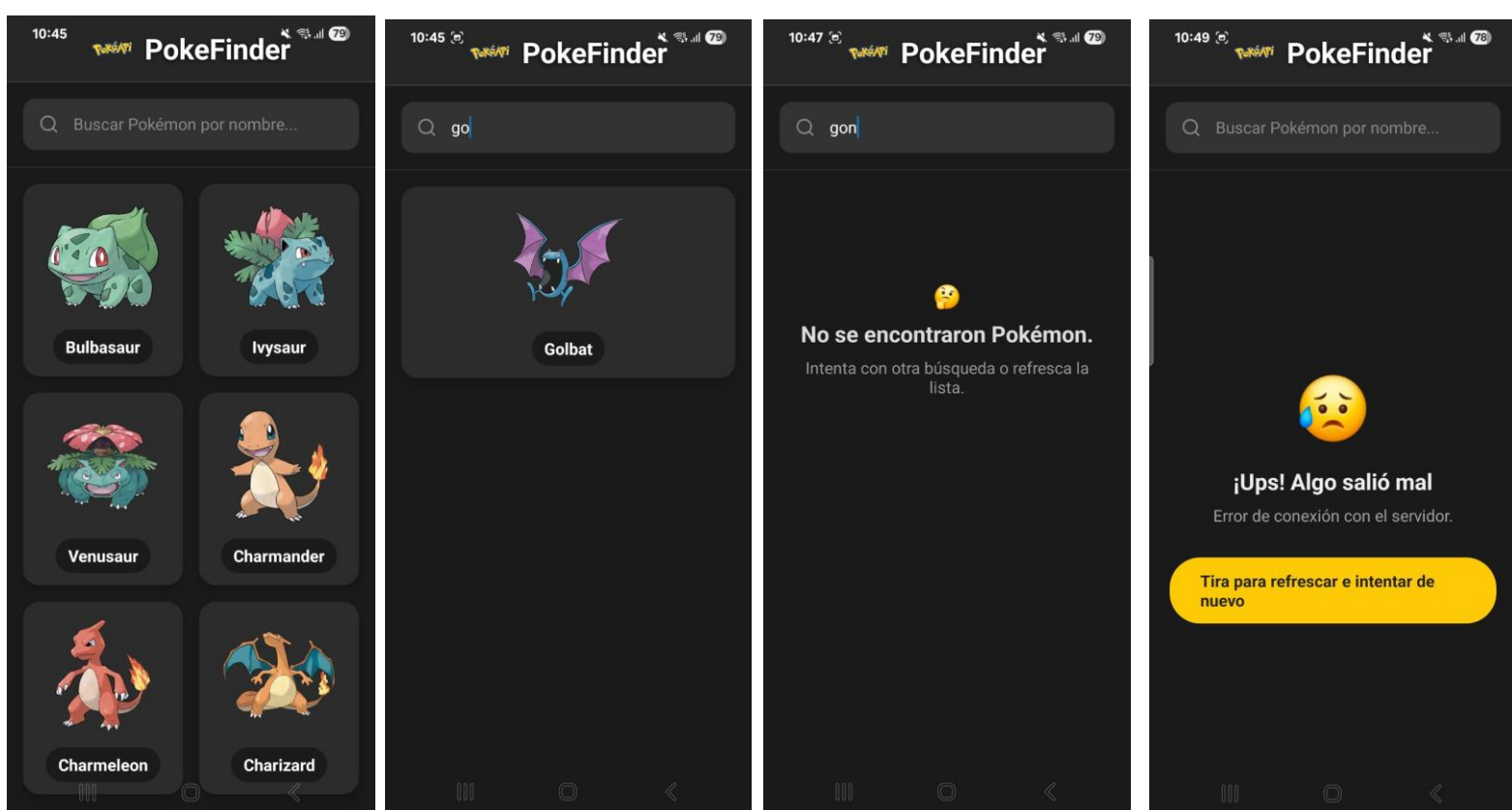
- La lista refresca y vuelve a estado no refrescando al completar.

Restricciones

- Usar `fetch` nativo y `useEffect` para la carga inicial.
- Optimizar filtrado con `useMemo` o similar (lista de 50 ítems mínima).

Bonus (opcional)

- Paginación (`next/previous`) usando `next` de la API.
- Indicador de “sin resultados” cuando el filtro no matchea nada.
- Cacheo simple en memoria para evitar flicker de loading entre páginas.



02 - b Aplicación de accesos por QR

Consigna

- Campo de texto para ingresar el valor a codificar.
- Render de QR con `react-native-qrcode-svg`.
- Botón Escanear (Expo) que solicita permisos y abre el scanner (`expo-barcode-scanner`).
- Al escanear, mostrar el valor leído y cerrar el scanner.

Criterios de aceptación

- El QR se actualiza en tiempo real al cambiar el texto.
- Si la cámara no tiene permisos, se muestra un mensaje claro.
- Al iniciar el scanner, ocupa un área visible y vuelve a la vista normal al detectar un código.
- El último valor escaneado permanece visible en la pantalla.

Restricciones

- Dependencies: `react-native-qrcode-svg` y (si usás Expo) `expo-barcode-scanner`.
- Manejar estado de permisos y errores sin crashear.

Bonus (opcional)

- Botón Copiar el último valor escaneado al portapapeles.
- Validar formato de “pago” (por ejemplo `PAY:<id>|<monto>|ARS`) y mostrarlo parseado.

- Guardar historial de escaneos con [AsyncStorage](#).

