

# Ejercicio entrevista Front-End - Wolox

La finalidad del ejercicio es desarrollar una aplicación web. La web consiste en una página inicial o landing page (cuyo diseño y assets se te enviarán en conjunto con este ejercicio) y un listado de Pokémon con filtros. El listado no tienen un diseño definido, por lo cual esperamos que lo hagas por tu cuenta, y si bien no esperamos un diseño superador, si es deseable que sean pantallas usables. Para obtener los datos tanto del listado.

## Requerimientos y Restricciones

- No utilizar Bootstrap ni ninguna librería con clases de CSS predefinidas.
- Utilizar flexbox y Grid como principales herramientas para definir el layout.
- Las vistas deben ser responsive en mobile y tablet.
- Configurar un linter.
- Si agrega botones deben tener algún efecto en hover.
- Utilizar al menos una animación además del efecto de hover en botones.

## Landing

### Criterios de aceptación

- Respetar el diseño enviado.
- Al seleccionar un link de la barra superior, se debe scrollear a la sección correspondiente. Por ejemplo: En caso del link de beneficios, scrollear hasta la sección de beneficios de la página.
- La Navbar deberá estar fixed.
- Al clickear el botón de “Síguenos” debe enviar el usuario al twitter de Wolox.
- El botón de “Conocer más” debe enviar al usuario a la página de Wolox.
- El botón de Registrarse debe enviar al usuario a la pantalla de Registro. (En el diseño este botón aparece como Login, no es necesario hacer el login de la aplicación)

### Criterios de aceptación

- Listar los Pokémon obtenidos de la siguiente [API](#).
- El listado deberá estar paginado en páginas de 20 elementos cada una por default.
- Se deberá poder cambiar la cantidad de elementos de cada página. Se requiere un selector con las opciones: 5, 10, 20, 50, 100
- Poder buscar un pokémon por su nombre o id.

- En el listado se deberá poder ver el nombre e imagen de un Pokémon.

## **React**

### **Criterios de aceptación**

- Implementar una solución de routing
- Usar un state management a elección propia (Redux/Context/Recoil/etc).
- Implementar test de componente de listado con Jest y react testing library.
- Utilizar react hooks.
- Hacer deploy la aplicación a un ambiente similar productivo.

### **Como requerimientos opcionales se establecen los siguientes:**

- Utilizar i18n para todos los textos de la landing y agregar un botón para cambiar de idioma (español/inglés).
- Utilizar Error Boundaries.
- Hacer el deploy a un bucket de AWS.