

# Ejercicio entrevista Front-End - Wolox

## Introducción

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), una pantalla de login y un listado de tecnologías con filtros. El login y 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 de un usuario como del listado, tenemos la siguiente [API](#).

Esperamos que el código esté en un repositorio de git, en el cual haya al menos un commit por pantalla.

## Requerimientos y Restricciones

- Vamos a pedirte que no utilices Bootstrap ni ninguna librería con clases de CSS predefinidas.
- Utilizar flexbox como principal herramienta para definir el layout.
- Hacer responsive las pantallas hasta 300px
- No usar tablas de HTML
- Usar ES6+, Babel y agregar algún plugin a elección
- Los botones deben tener algún efecto en hover
- Utilizar al menos alguna animación
- Subir el proyecto a un ambiente similar productivo
- Configurar un linter

## Opcionales

- Utilizar SCSS y/o SCSS modules

- Utilizar CSS Grid
- Hacer tests de unidad de algunas funciones con librería a elección
- Configurar webpack
- Hacer la web “accesible”

## Landing

### Criterios de aceptación

- Respetar el diseño enviado
- Al seleccionar un link de la barra superior, se debe ir a la sección correspondiente.
- Al clicar el botón de “Síguenos” se debe redirigir al [twitter de Wolox](#)
- El botón de Login debe enviar al usuario a la pantalla de Login
- El botón de “Conocer más” debe enviar al usuario a la [página de Wolox](#)

## Login

### Criterios de aceptación

- El login debe hacerse con un email y una contraseña, además de un checkbox para preguntarle al usuario si quiere mantenerse conectado.
- Los campos del login deben tener un formato válido antes de poder enviar la request.
- En caso de seleccionar “mantenerse conectado”, el usuario debe ser guardado en LocalStorage, y la próxima vez que el usuario se conecte debe ingresar directamente al listado. El usuario no deberá poder ir a la pantalla de login, siendo redirigido al listado. En la Landing no debe verse el botón de Login.
- En caso de no seleccionar “mantenerse conectado”, al actualizar la página el usuario no será recordado, y tampoco debe poder ingresar al listado, siendo redirigido al login en caso de querer hacerlo.

## Listado

## Criterios de aceptación

- Listar los elementos retornados por el endpoint con todos sus datos
- Poder filtrar los elementos por nombre y por si son Front-end, Back-end o Mobile, cambiando el resultado a medida que se tipea. Si el filtro está vacío se deben mostrar todos los elementos
- Poder ordenar los elementos por nombre, tanto ascendente como descendentemente.
- No usar paginación
- Mostrar debajo del listado la suma total de tecnologías siendo mostrada

## Angular

### Criterios de aceptación

- Implementar una solución de routing para poder tener rutas privadas y que el usuario no conectado no pueda acceder a ellas.
- Implementar test de componente de listado con una librería de testing a elección.
- Implementar el formulario de login con reactive forms.
- Crear módulos teniendo en cuenta la potencial escalabilidad de la aplicación e implementar lazy loading.
- Utilizar el cliente HTTP que provee Angular para realizar las requests a la API.

Como requerimientos opcionales se establecen los siguientes:

- Internacionalizar la Landing page (únicamente esa vista) con una librería de tu elección, agregar un botón para cambiar de idioma (español/inglés).
- Implementar error boundary para la app entera (único para todos los errores).
- Hacer deploy la aplicación a un ambiente similar productivo.