CHALLENGE FRONTEND - Angular 🚀

Objetivo

Desarrollar una aplicación para crear un equipo de superhéroes que consumirá una API externa y mostrará diferentes atributos a nivel individual de cada miembro y del equipo consolidado.

👉 Consumir los endpoints de la siguiente API para realizar las distintas operaciones. Deberás autenticarte con Facebook para realizar peticiones a la misma.

👉 Adicionalmente, las diferentes secciones que tendrá la app deberán protegerse verificando que el usuario autenticado disponga de un token que se almacenará en localStorage. El mismo, se obtendrá de una API con datos de muestra. Si un usuario intenta ingresar a cualquier ruta sin estar autenticado, deberá ser redirigido al login.

👉 Para el manejo de peticiones HTTP deberá utilizarse la librería Axios.

👉 El sitio deberá ser responsive, y utilizar Bootstrap como punto de partida para aprovechar las características de la librería.

⚠️ ¡No es indispensable hacer todo!
Mientras más completes, mayor puntaje obtendrás, pero puedes enviar la app hasta el estadío que tengas en base a tu conocimiento actual. Recuerda que el objetivo del challenge es entender tu nivel de conocimiento actual.

Requerimientos funcionales

En la pantalla de Home se deberá mostrar, además de los miembros del equipo:

- Acumulativo de powerstats, agrupados por cada uno, es decir: suma total de intelligence, strength, etc. de todos los miembros individuales del equipo.
- El powerstat que más acumulativo tenga debería aparecer arriba para categorizar el tipo de equipo (inteligencia, fuerza, etc.).
- Pesos y altura promedio del equipo.

- El equipo debe tener 6 miembros. Debe haber 3 miembros con orientación buena y 3 con orientación mala. Esto debe validarse al intentar agregar un nuevo héroe.
- Se deberá poder eliminar un miembro del equipo, lo que generará un nuevo promedio de peso, acumulativo de powerstats, etc.

Requerimientos técnicos

Aprovechando las características de Angular, deberán crearse las siguientes secciones, y modularizar las mismas en componentes reutilizables.

1. Formulario de Login

El formulario se deberá renderizar al ingresar a cualquier ruta si el usuario no está autenticado, conteniendo los campos:

- Email.
- Password.
- Botón de “Enviar”.

Al hacer click en “Enviar”, se deberá validar que ambos campos no estén vacíos, y mostrar un mensaje al usuario si lo estuviesen. Caso contrario, se deberá realizar una petición POST a la siguiente url, con los campos email y password en el BODY.

Los datos válidos para obtener un token son:

- Email: challenge@alkemy.org
- Password: angular

En el caso de obtener un error de la API, se deberá mostrar una alerta, mientras que si es satisfactorio deberá redirigir al Home y almacenar el token obtenido en localStorage.

Las validaciones del formulario deberán realizarse utilizando la librería **Angular Forms**.

2. Equipo

El Home de la aplicación mostrará a los miembros del equipo en un listado en un grid. Cada ítem del listado contendrá:

- Nombre del héroe.
- Imagen.
- Powerstats.
- Acciones para ver el detalle o eliminarlo del equipo.

3. Buscador de Héroes

Para agregar un héroe a su equipo, se deberá visualizar un formulario que realice una petición GET al endpoint de búsqueda y muestre los resultados disponibles en un grid. Esos resultados deberán mostrar:

- Nombre del héroe.
- Imagen.
- Acciones para agregarlo al equipo

Las validaciones del formulario deberán realizarse utilizando la librería **Angular Forms**.

4. Detalle de Héroe

Al hacer click en un héroe del equipo, se mostrarán los detalles que figuran en el endpoint. De ellos, mostrar:, altura, nombre completo, alias, color de ojos y cabello, y su lugar de trabajo.

- Peso.
- Altura.
- Nombre.
- Alias.
- Color de ojos.
- Color de cabello.
- Lugar de trabajo.

Tests

De forma opcional, se podrán agregar tests unitarios para validar los elementos de la app:

- Verificación de usuario autenticado al ingresar a una ruta.
- Validación de campos en submit de formulario de login o búsqueda.
- Manejo de excepciones al obtener errores de la API.

Criterios a evaluar

- Diseño responsive, moderno e intuitivo.
- Debe utilizarse Bootstrap para permitir que el proyecto sea responsive, y media queries para los elementos personalizados que se desarrollen.
- Conocimientos básicos de Angular.
- Validación de formularios utilizando la librería Angular Forms.
- Buenas prácticas de codificación.
- Buenas prácticas para el nombre de rutas.
- Código modularizado en componentes reutilizables e independientes.