

Prueba técnica frontend

Importante:

Se debe utilizar la última versión LTS de Angular, Typescript y de cualquier librería que se use en el proyecto.

Desarrollar, utilizando **Angular** y **Typescript**, una aplicación SPA que permita hacer un mantenimiento de súper héroes:

- Se deberá crear un **Servicio** que guarde la información y que permita:
 - Consultar todos los súper héroes.
 - Consultar un único súper héroe por id.
 - Consultar todos los súper héroes que contienen, en su nombre, el valor de un parámetro enviado en la petición. Por ejemplo, si enviamos "man" devolverá "Spiderman", "Superman", "Manolito el fuerte", etc.
 - Modificar un súper héroe.
 - Eliminar un súper héroe.
 - Test unitario de este servicio (opcional).
- Se deberá crear un **Componente** que, a partir del servicio anterior:
 - Mostrará una lista paginada de héroes donde aparecerán botones de *añadir*, *editar* y *borrar*.
 - Encima de esta lista paginada, se mostrará un input para filtrar por el héroe seleccionado.
 - Al pulsar el botón de *añadir* se generará un formulario vacío con las validaciones que se estimen oportunas. Después de dar de alta el nuevo héroe se volverá a la lista paginada.
 - Al pulsar el botón de *edición* se generará un formulario con los datos del héroe seleccionado y se permitirá modificar su información. Una vez editado se deberá volver a la lista paginada.
 - Al pulsar el botón de *borrar*, se preguntará si se está seguro que se desea borrar el héroe y, al confirmarlo, lo borrará.
 - Test unitario de este componente (opcional).

Puntos a tener en cuenta:

- La información de súper héroes se guardará dentro del servicio. (No hace falta un backend).
- Se valorarán las soluciones propuestas para cada punto, el modelo de datos y formato del código.
- La prueba se debe presentar en un repositorio de *Git*. No hace falta que esté publicado. Se puede pasar comprimido en un único archivo.

Puntos opcionales de mejora:

- Utilizar Angular Material como apoyo visual.
- Rutas y navegación de la página.
- Presentar la aplicación "*Dockerizada*".
- Interceptor para mostrar un elemento "*loading*" mientras se realiza alguna operación como "*borrado*" o "*edición*".
- Directiva para que al crear o editar en la caja de texto del nombre del héroe, siempre se muestre en mayúscula.
- ¿Uso de mockserver para poder implementar las llamadas HTTP sin un backend?.
- Comunicación entre componentes orientada a eventos.

Se valorará positivamente:

- Cómo se construye el modelo de datos.
- Uso de programación reactiva.
- Código legible usando lambdas.