

Prueba técnica Front End position Acsendo

Tiempo de entrega: 3 días

Medio de entrega: almacene el proyecto en cualquier repositorio público (Gitlab, github, etc) y comparta el proyecto al email indicado por su gestor de proceso.

1. Github API

Cree una aplicación web simple (usando Angular +12) con un campo de búsqueda que filtre usuarios de la API de usuarios de Github y luego muestre los resultados en la interfaz (diseño libre)

(API de Github:

<https://developer.github.com/v3/search/#searchusers>)

Cuando un usuario ingrese al menos 3 caracteres en el campo de texto, realice una solicitud al API de github, analice el JSON resultante y muestre los siguientes datos para los primeros cien elementos:

- avatar_url: El avatar del usuario de Github (representado como una imagen)
- login: el nombre de usuario del usuario de github
- type: El tipo de usuario.
- score: un valor decimal que representa la precisión del resultado frente al término de búsqueda.

Cuando una solicitud HTTP activa está en curso, coloque un indicador de progreso en cualquier parte de la página para indicar la actividad de la red.

Una vez que se realiza la búsqueda, borrar la entrada, también debería borrar la lista de usuarios mostrada antes.

NOTA: La API de Github tiene una tasa limitada. Realizar demasiadas solicitudes demasiado rápido terminará con las

solicitudes de API del usuario bloqueadas por un corto período de tiempo.

2. Pipes

En el listado de usuarios anterior agregue una nueva propiedad en el listado basado en el **score** del usuario que indique el nivel de precisión, si el usuario tiene **menos de 0.4** debe mostrar el texto **"bajo"**, si tiene **menos de 0.7** debe mostrar **"medio"** y si tiene **más de 0.7** debe mostrar **"alto"**

NOTA: Recuerde usar los Angular Pipes para lograr esta funcionalidad

3. Prueba lógica

Dado un arreglo de ítems:

```
let unavailableItems = [
  { startPx: 10, endPx: 30 },
  { startPx: 55, endPx: 65 },
  { startPx: 35, endPx: 50 },
  { startPx: 20, endPx: 40 },
  { startPx: 60, endPx: 70 },
]
```

Escriba una función (usando ES6) que una todos los elementos superpuestos y produzca una matriz de elementos que no se superponen.

por ejemplo:

```
{ startPx: 10, endPx: 30 }
{ startPx: 20, endPx: 40 }
```

Debe retornar:

```
{ startPx: 10, endPx: 40 }
```