

Actividad 4

Se desea diseñar una aplicación que permita a los usuarios recibir noticias sobre diferentes asuntos. La aplicación deberá ser capaz de permitir al usuario definir diferentes áreas de interés y, dentro de cada una de ellas, diferentes términos de búsqueda (por ejemplo, un área de interés podría ser Desarrollo Web y como términos de búsqueda podrían estar Angular, Spring Boot...

La aplicación constará de un **BackEnd** desarrollado en **Spring** y un **FrontEnd** desarrollado en **Angular**.

BackEnd

El BackEnd será un servicio REST desarrollado en Spring que, al menos, deberá definir los elementos necesarios para permitir:

- Crear un área de interés
- Eliminar un área de interés
- Retornar todas las áreas de interés
- Añadir un término de búsqueda a un área de interés
- Eliminar un término de búsqueda de un área de interés
- Retornar todas las noticias encontradas de todas las búsquedas de un área de interés (serán los resultados de buscar todos los términos de búsqueda y retornarlos en un bloque)

Las **áreas de interés** deberán incluir, al menos, un nombre y una descripción, así como la fecha de creación.

De los **términos de búsqueda** se almacenarán, su cadena de búsqueda, el área de interés y el número de resultados máximos deseados.

Esta información se almacenará en la BBDD y se empleará JPA para acceder a la misma.

Para localizar las noticias se va a emplear el API de <https://contextualweb.io/> al que hay que acceder a través del servicio <https://rapidapi.com/> que es el que nos proporcionará la clave para poder acceder al servicio anterior.

Este servicio retorna un objeto JSON con la siguiente información:

```
{
  _type: string,
  didUMean: string,
  totalCount: integer,
  relatedSearch: string[],
  value: noticia[]
}
```

Donde **noticia** es otro objeto JSON con la siguiente información:

```
{
    title: string,
    url: string,
    description: string,
    body: string,
    keywords: string,
    language: string,
    isSafe: boolean,
    datePublished: date,
    image: imagen
}
```

Donde **imagen** es otro objeto JSON con la siguiente información:

```
{
    url: string,
    height: entero,
    width: entero,
    thumbnail: string,
    thumbnailHeight: entero,
    thumbnailWidth: entero,
    base64Encoding: Object
}
```

Para poder asignar las noticias devueltas por el API a objetos JAVA, necesitamos definir una clase **Imagen** con un campo para cada propiedad del objeto JSON, una clase **Noticia** igual (la propiedad image será de tipo Imagen) y una última **NoticiasDTO** para el objeto devuelto (las matrices podéis definir las como matrices o como listas).

Por otro lado, para acceder al API es necesario añadir una cabecera al mismo con la clave. Dado que el método **getForObject** de la clase **RestTemplate** no permite emplear cabeceras, tenemos que emplear el método **exchange** de la misma clase. De este modo, el método para buscar las noticias quedaría de la siguiente forma:

```
HttpHeaders headers = new HttpHeaders();
headers.add("x-rapidapi-key", "333b042ea5msh873fe5be7047703p1f190ajsn6a78e3b54294");
HttpEntity<?> entity = new HttpEntity<>(headers);
ResponseEntity<NewsDTO> resp = template.exchange(
    "https://contextualwebsearch-websearch-v1.p.rapidapi.com/api/Search/NewsSearchAPI?q="
    +busqueda+"&page_size="+num,
    HttpMethod.GET,
    entity,
    NewsDTO.class);
NewsDTO datos = resp.getBody();
```

En el ejemplo se busca lo indicado en **búsqueda** y se retornan **num resultados**.

NOTA: El servicio REST deberá estar **protegido con OAuth** empleando **Client Credentials** (si alguien se anima a hacerlo con **Resource Owner Password Credentials** mejor). En el segundo caso, lo ideal sería asociar las áreas de interés y las búsquedas a los usuarios de la aplicación.

FrontEnd

El FrontEnd será una aplicación Angular que se conectará al servicio REST definido con Spring.

Deberá obtener el token desde **/oauth/token** y emplearlo para acceder a la información de la aplicación empleando HTTP_INTERCEPTOR.

La aplicación deberá mostrar, al iniciarse, las áreas de interés creadas hasta el momento en el BackEnd con mecanismos para poder eliminar y añadir nuevas áreas empleando formularios basados en plantilla y validando que exista el campo **nombre**.

Al seleccionar un área de interés, se mostrarán los resultados de las búsquedas en esa área junto con los términos de búsqueda definidos.

Además se proporcionarán mecanismos para añadir / eliminar dichos términos de búsqueda. Para añadir una nueva búsqueda se empleará un formulario reactivo y se comprobará que al menos exista la cadena de búsqueda.