

# Spring Data REST: Paginación y búsqueda

Elementos avanzados en tu API REST con Spring Boot

# Paginación

- En lugar de devolver un gran conjunto de resultados, Spring Data REST reconoce algunos parámetros de URL que influyen en el tamaño de la página y el número de la página de inicio.
- Si nuestro repositorio extiende a *PagingAndSortingRepository*<T, ID> (o a *JpaRepository*<T, ID>, que a su vez extiende al anterior), y accedemos a la lista de todas las entidades, obtendremos las 20 primeras.

# Paginación

- Podemos modificar el tamaño o cambiar el número de página con los parámetros *page* y *size* en la URL:

<http://localhost:8080/api/ciudades?page=2&size=5>

# Métodos de búsqueda

- Spring Data REST también es capaz de exponer nuestras consultas en el repositorio como *endpoints*.
- Estas, además, también puede ser paginadas.

```
@RestResource(path = "nombreComienzaPor", rel = "nombreComienzaPor")  
public Page<Ciudad> findByNombreStartsWith(  
    @Param("nombre") String nombre, Pageable p);
```

# Enlaces de navegación

- Las búsquedas paginadas suelen producir varios enlaces para navegar a las demás páginas.
- El tipo o número de enlaces dependen del tamaño y página y la página actual.

```
},  
  "_links": {  
    "first": {  
      "href": "http://localhost:8080/api/ciudades/search/nombreComienzaPor?nombre=M&page=0&size=3"  
    },  
    "self": {  
      "href": "http://localhost:8080/api/ciudades/search/nombreComienzaPor?nombre=M&page=0&size=3"  
    },  
    "next": {  
      "href": "http://localhost:8080/api/ciudades/search/nombreComienzaPor?nombre=M&page=1&size=3"  
    },  
    "last": {  
      "href": "http://localhost:8080/api/ciudades/search/nombreComienzaPor?nombre=M&page=2&size=3"  
    }  
  }  
}
```

# Ordenación

- Spring Data REST reconoce los parámetros de ordenación utilizados por los repositorios subyacentes.
- Ya hemos visto en *templates uris* de ejemplos anteriores que aparece el parámetro sort.

`http://localhost:8080/api/ciudades?sort=nombre,desc`

# Ordenación

- Para ordenar sus resultados en una propiedad en particular, agregamos el parámetro `sort` a la URL con el nombre de la propiedad sobre la que deseamos ordenar los resultados.
- Se puede controlar la dirección de la ordenación agregando una coma (,) al nombre de la propiedad más *asc* o *desc*.
- Se pueden utilizar propiedades del objeto ~~y propiedades anidadas~~.

# Reto

- Implementa algunos métodos de búsqueda más y comprueba que funcionan, incluyendo paginación y ordenación.