



# Spring Data

//Elasticsearch

IT BOARDING

**BOOTCAMP**



**“Spring Data Elasticsearch aplica conceptos del núcleo de Spring al desarrollo de soluciones usando el motor de búsqueda Elasticsearch”.**

IT BOARDING

**BOOTCAMP**

# Índice



**01** Requisitos.

**03** Dominio

**02** Creación de clases.

**04** Repositories.

IT BOARDING

**BOOTCAMP**

# Requisitos

IT BOARDING

**BOOTCAMP**





# 1. Creación del proyecto.

Project properties

Group Id	com.relaciones.jpa
Artifact Id	jpa
Version	0.0.1-SNAPSHOT
Project type	Maven Project
Language	Java
Packaging	Jar
Java version	11
Project name	Relaciones JPA
Project description	Relaciones JPA
Package name	com.relaciones.jpa

Filter

Spring Boot version 2.5.3

Developer Tools	<input checked="" type="checkbox"/> Spring Web
Web	<input type="checkbox"/> Spring Reactive Web
Template Engines	<input type="checkbox"/> Rest Repositories
Security	<input type="checkbox"/> Spring Session
SQL	<input type="checkbox"/> Rest Repositories HAL Explorer
NoSQL	<input type="checkbox"/> Spring HATEOAS
Messaging	<input type="checkbox"/> Spring Web Services
I/O	<input type="checkbox"/> Jersey
Ops	<input type="checkbox"/> Vaadin
Observability	
Testing	

Project name: Elastic

Project location: C:\Users\fcosta\Downloads\final-challenge-group-8-develop\Elastic



## 2. Agregar dependencias

Abrimos nuestro archivo pom.xml y entre las etiquetas **<dependencies>** **</dependencies>** agregamos la siguiente dependencia:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-elasticsearch</artifactId>
</dependency>
```

# Creación de clases

IT BOARDING

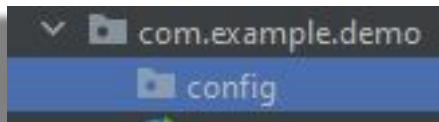
**BOOTCAMP**





## 4. Creación de Beans

Configuremos nuestro proyecto para trabajar con elasticsearch:



1.- Creamos un paquete llamado config

2.- Luego, Dentro de este paquete, creamos una clase config y la anotamos con lo siguiente:.

```
@Configuration
@EnableElasticsearchRepositories(basePackages="com.example.demo.elasticrepositories")
public class Config {
```





# Clase Config

3.- Luego, Terminamos de completar la clase Config de la siguiente manera:

Creamos un Bean de un objeto  
RestHighLevelClient  
usado para la conexión  
con al API de  
ElasticSearch

Registramos un  
ElasticsearchOperations  
en el contexto de Spring.

```
public class Config {  
    @Bean  
    public RestHighLevelClient client(){  
        ClientConfiguration clientConfiguration = ClientConfiguration.builder()  
            .connectedTo( hostAndPort: "localhost:9200")  
            .build();  
        return RestClients.create(clientConfiguration).rest();  
    }  
  
    @Bean  
    public ElasticsearchOperations elasticsearchTemplate(){  
        return new ElasticsearchRestTemplate(client());  
    }  
}
```

# Dominio

IT BOARDING

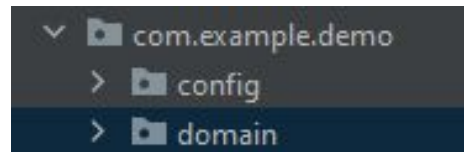
**BOOTCAMP**





# Clase Artículo

1. Vamos a crear un paquete llamado domain :



```
@Document(indexName = "blog")
public class Articulo {
    @Id
    private String id;

    private String titulo;

    @Field(type = FieldType.Nested, includeInParent = true)
    private List<Autor> autores;

    /**
     * Add setters and getters
     */
}
```

2. Creamos una clase Artículo tal como se muestra en la siguiente imagen.

# Clase Artículo



3. Verificamos los imports necesarios en nuestro código:

```
import org.springframework.data.annotation.Id;  
import org.springframework.data.elasticsearch.annotations.Document;  
import org.springframework.data.elasticsearch.annotations.Field;  
import org.springframework.data.elasticsearch.annotations.FieldType;  
  
import java.util.List;
```

# Clase Autor



4. Creamos la clase Autor necesaria para asociar al artículo:

```
public class Autor {  
  
    private String nombre;  
  
    /**  
     * Add getters and setters  
     * and custom constructors  
     */  
  
}
```

# Repositories

IT BOARDING

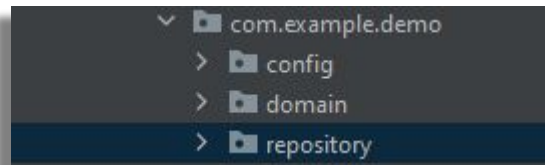
**BOOTCAMP**





# Interface `ArticuloRepository`

1. Creamos un paquete para nuestras interfaces de repositories:



2. Agregamos una interface llamada `ArticuloRepository`

```
public interface ArticuloRepository extends ElasticsearchRepository<Articulo, String> {  
  
}
```



# Gracias

IT BOARDING

**BOOTCAMP**

