

# INDITEX

## Caso Práctico Backend Tools

Dado un listado de productos que se muestra en una categoría de camisetas se necesita implementar un algoritmo que permita ordenar ese listado en base a una serie de criterios de ordenación. Cada criterio de ordenación tendrá un peso asociado de manera que la puntuación de cada producto a ordenar vendrá dada por la suma ponderada de los criterios.

Los criterios de ordenación definidos serán el criterio de ventas por unidades y el criterio de ratio de stock, puede que a futuro se añadan nuevos criterios.

El criterio de ventas por unidades dará una puntuación a cada producto basado en el número de unidades vendidas.

El criterio de ratio de stock dará una puntuación en función de las tallas que contengan stock en ese momento.

El listado de productos es el siguiente:

id	name	sales_units	stock
1	V-NECH BASIC SHIRT	100	S: 4 / M:9 / L:0
2	CONTRASTING FABRIC T-SHIRT	50	S: 35 / M:9 / L:9
3	RAISED PRINT T-SHIRT	80	S: 20 / M:2 / L:20
4	PLEATED T-SHIRT	3	S: 25 / M:30 / L:10
5	CONTRASTING LACE T-SHIRT	650	S: 0 / M:1 / L:0
6	SLOGAN T-SHIRT	20	S: 9 / M:2 / L:5

La funcionalidad debe exponerse a través de un servicio REST de manera que recibirá los pesos para cada criterio.

Requisitos:

- Implementado en Java 8 o superior
- Usar Springboot
- Evitar la anemia de código
- Implementar siguiendo Arquitectura Hexagonal y los principios tácticos de DDD.
- Tener bien en cuenta la separación de capas.
- Incluir tests. Si es posible en sus 3 formas (unitarios, integración E2E con rest assured)
- Añadir persistencia de datos con MongoDB

Lo que más se tendrá en cuenta será la calidad de código y la organización del mismo. Intenta crear un código limpio y sostenible.