**Code challenge**

El presente code challenge está compuesto por 2 ejercicios bajo las siguientes pautas:

* **Enunciado 1**: Lógica de programación, sintaxis y conocimiento de las estructuras básicas de Java y manejo de Springboot - API Rest.

**Puntaje máximo -> 40 puntos.**

* **Enunciado 2**: Comprensión funcional de historias de usuario, manejo de Hibernate y BBDD, diseño de pantallas web.  
  **Puntaje máximo -> 60 puntos.**

Resolver ambos enunciados preferentemente utilizando las siguientes tecnologías:

* Java.
* Springboot.
* Hibernate.
* MySql – SQL Server – H2.
* React o Primefaces – JSF para el Front.

**Entregar la resolución de lo pedido con un link a un repositorio en github y enviar el link de acceso al email** [**ldagostino@mbsoft.com.ar**](mailto:ldagostino@mbsoft.com.ar) **con copia a los mails** [**pmujico@mbsoft.com.ar**](mailto:pmujico@mbsoft.com.ar) **y** [**vemiranda@mbsoft.com.ar**](mailto:vemiranda@mbsoft.com.ar)

**Enunciado 1**

Tengo una lista de códigos de producto, el formato de estos códigos es: "XXX-NNNNN-Y"

Donde:

XXX son letras (A-Z) ---> código alfabético del producto

NNNNN son números (0-9) ---> código numérico de la región geográfica

Y es un dígito verificador (0-9)

Se pide:

**a)** Realizar la función:

boolean **esPrioritario**(codigo), donde devuelve true o false si el código de producto comienza con las letras 'P' o 'W'

**b)** Realizar la función:

boolean **verificar**(codigo), donde chequea si el dígito verificador es correcto. Y devuelve true o false si el dígito verificador es correcto.

El dígito verificador se calcula de la siguiente manera: se suma cada cifra del código de la región geográfica => si obtengo un número de más de 2 dígitos, repito (vuelvo a sumar cada dígito) así, hasta llegar a 1 sola cifra, ese será mi dígito verificador.

Ejemplo: producto "DCR-88578-9", se realiza la suma 8+8+5+7+8 = 36. Luego se suma 3+6 = 9. Paro. 9 es mi dígito verificador.

**c)** Realizar una función que recibe una lista de productos (lista/array de los códigos), y las ordena alfabéticamente según su código alfabético (XXX), de menor a mayor.

**d)** Realizar una función que recibe 2 listas de productos (listas/arrays de los códigos) y devuelve una lista/array que representa la Unión de los elementos.

**e)** Realizar una función que recibe 2 listas de productos (listas/arrays de los códigos) y devuelve una lista/array que representa la Intersección de los elementos.

**IMPORTANTE: Mostrar los resultados de todas las funciones mediante el consumo de API Rest indicando con ejemplos cómo serían las llamadas a los endpoints (tener en cuenta los parámetros enviados).**

**Enunciado 2**

Como laboratorio farmacéutico, somos fabricantes de medicamentos, los cuales son de distintos tipos: aerosoles, cremas, colirios, cápsulas, polvo en suspensión, jarabes, entre otros. (EJ: Medicamento llamado comercialmente “amoxol” con código 12345 cuya droga es “amoxicilina” es del tipo “cápsulas”).  
Conociendo cuál es la operación principal de nuestra empresa, se nos indica el siguiente backlog de tareas:

* Como usuario, quiero dar de alta un nuevo tipo de medicamento.
* Como usuario, quiero dar de baja un tipo de medicamento (baja lógica).
* Como usuario, quiero crear un nuevo medicamento ingresando un código numérico, un nombre comercial, un nombre para describir qué droga es y un tipo de medicamento (aerosol, crema, colirio, etc).
* Como usuario, quiero listar todos los medicamentos cuyo tipo sea “aerosol”.
* Como usuario, quiero listar todos los medicamentos cuyo nombre comercial comience con la letra “a”.

**IMPORTANTE: Desarrollar las 5 historias de usuario generando las pantallas web necesarias para cubrir las funcionalidades pedidas.   
Se puede utilizar una base de datos en memoria como H2 o utilizar algún servidor de base de datos. Adjuntar los scripts de creación de la base de datos así como los scripts para poblar las tablas.**