

Prueba Técnica.

Reto de Programación

Los archivos fuente de la aplicación y de Base de datos deben ser cargados en GitHub. Dispone de 3 días a partir de la fecha de envío de la prueba.

Cree una **aplicación Web** (java preferiblemente) que tenga 2 matrices bi-dimensionales **de igual longitud**. La **Matriz 1 contendrá solo números menores a 10** en todas sus posiciones y la **Matriz 2 contendrá solo 1 letra** en cada una de sus posiciones. Así:

Matriz 1

1	3	6
2	5	2
9	3	2
4	2	1
9	6	1

Matriz 2

A	L	C
D	B	Z
T	G	T
G	M	X
N	W	A

A través de una interfaz gráfica, usando el framework front-end de su elección, debe generar un formulario Web a través del cual se inicializarán las matrices y se solicitará una operación específica sobre ellas. Dicho formulario debe componerse de:

Parámetro 1 (Dimensión de las Matrices): Se enviará a través de este parámetro las dimensiones de las matrices ALTO-ANCHO separadas por el carácter (-) guión. Ejemplo 5-3 dónde 5 es el alto y 3 es el ancho (Mirar matrices de Ejemplo)

Parámetro 2 (Contenido Matriz numérica): Se enviará el contenido por filas de acuerdo al ALTO-ANCHO definido en el parámetro 1 y la longitud de este dato debe ser suficiente para cubrir todas las posiciones de la matriz numérica. Entendiendo que para llenar la matriz de ejemplo se necesitan 15 números ya que $5 \times 3 = 15$. Ejm: 136252932421961

Parámetro 3 (Contenido Matriz letras): Se enviará el contenido por filas de acuerdo al ALTO-ANCHO definido en el parámetro 1 y la longitud de este dato debe ser suficiente para cubrir todas las posiciones de la matriz de letras. Entendiendo que para llenar la matriz de ejemplo se necesitan 15 letras ya que $5 \times 3 = 15$. Ejm: ALCDBZTGTMXNWA

Parámetro 4 (selección Matriz): Debe contar con las siguientes opciones:

- Matriz Numérica
- Matriz Letras

Parámetro 5 (Operación): Debe contar con las siguientes opciones:

- **Consultar ocurrencias:** Indica que se debe, de acuerdo a la matriz seleccionada en el paso 4, buscar las ocurrencias por elemento. Por ejemplo A(2), L(1), etc.

Prueba Técnica.

- **Eliminar repetidos:** Indica que se deben eliminar los elementos repetidos dentro de la matriz seleccionada en el paso 4 dejando solo una ocurrencia y reemplazando el resto por asterisco (*), para evidenciar esto se debe retornar el contenido completo de la matriz.
- **Ordenar ascendentemente:** Indica que se espera que se haga un ordenamiento ascendente de los elementos de la matriz seleccionada por lo tanto se debe retornar el contenido completo de la matriz ordenada.

IMPORTANTE:

- Se debe controlar que la totalidad de los parámetros sean enviados, Todos los parámetros son obligatorios. Por lo tanto, si falta alguno debe mostrar por pantalla mensajes de validación.
- Debe implementar un mensaje de validación previo al envío, de manera que se permita decidir si se desea o no desea hacer el envío.
- **Una vez se haga el envío del formulario** se deben procesar según la operación seleccionada y se deben mostrar los resultados en la interfaz Gráfica Web.
- Todas las peticiones y repuestas deben ser almacenadas en una **BASE DE DATOS (Libre escogencia)** de manera que se guarde el LOG de peticiones y su respectiva **respuesta**.
- **CREAR un Servicio Web SOAP**, que retorne el contenido de la **tabla LOG**, en dónde se han almacenado las peticiones y las respuestas realizadas.
- **CREAR un Servicio Web REST**, que retorne el contenido de la **tabla LOG**, en dónde se han almacenado las peticiones y las respuestas realizadas.

Puntos extras sí:

La Aplicación es desarrollada en Eclipse.

La Aplicación es desarrollada usando algún patrón o modelo de Arquitectura n-capas.

La capa de acceso a base de datos la debe hacer usando algún framework de persistencia de datos como JPA, Hibernate, Spring JDBC, Spring DATA, etc.