Práctica Vectores

Te dejamos algunas propuestas de prácticas que puedes realizar como desafío, para que pongas en práctica lo aprendido. **Estos ejercicios no son obligatorios y no tienes que entregarlos**. Te invitamos a que los compartas en tus redes con tus compañeros y observes cómo otros lo han resuelto.

**1.**Se necesita de un vector que sea capaz de almacenar 10 números enteros entre 1 y 100. Realizar un programa que permita la carga por teclado de los 10 números solicitados.

**2.**En un vector de 15 posiciones se almacenan las edades de 15 alumnos. Se desea un programa que sea capaz de determinar cuál es la mayor edad y cuál es la menor edad que se encuentra entre los estudiantes.

**3.**Se tienen tres vectores. En el primero de ellos se guarda un número de dni, en el segundo un nombre y en el tercero un apellido. Se desea un programa que sea capaz de recorrer los tres vectores AL MISMO TIEMPO y mostrar cada uno de estos datos por pantalla. Pista: tener en cuenta que el índice de cada vector es correspondiente al índice de los demás, es decir, los datos contenidos en el índice cero del vector de dni, se corresponde con el índice cero del vector de nombres y el de apellidos.

**4.**En un vector de 23 posiciones se tienen las temperaturas máximas de las capitales de las 23 provincias argentinas en el último mes. A partir de esta información, un noticiero desea determinar el top 5 de las temperaturas más altas para poder mostrar en la pantalla de su programa, para ello se necesita un programa que sea capaz de recorrer el vector de temperaturas, determinar las 5 más altas y copiarlas en un nuevo vector de 5 posiciones

**5.**Se necesita un vector que permita cargar por teclado el nombre de 10 animales. A partir de esta carga, se desea otro vector que copie los mismos nombres pero en el orden inverso, es decir, si los nombres son: perro, gato, lagartija, el nuevo vector debe contener: lagartija, gato, perro. Una vez realizado el cambio, mostrar por pantalla ambos vectores para compararlos.

## Práctica Matrices

##### Te dejamos algunas propuestas de prácticas que puedes realizar como desafío, para que pongas en práctica lo aprendido. ****Estos ejercicios no son obligatorios y no tienes que entregarlos****. Te invitamos a que los compartas en tus redes con tus compañeros y observes cómo otros lo han resuelto.

**1.**Una matriz de 5 filas x 3 columnas almacena el total de goles de 5 jugadores de futbol en los últimos 3 partidos que jugaron, donde cada fila representa a un jugador y cada columna a la cantidad de goles que hizo. Se necesita un programa que sea capaz de permitir la carga de los goles, calcular el promedio de goles realizado por cada jugador y almacenar el resultado en un vector de 5 posiciones, donde cada posición representará a un jugador. Tener en cuenta el siguiente diagrama para llevar a cabo el planteo:

**Nota:** tener en cuenta que el promedio de goles puede dar como resultado un número que NO SEA ENTERO.

**2.**Una escuela primaria utiliza una matriz para calcular los promedios de sus alumnos. Para ello tienen una matriz de 10 filas, donde cada fila representa a un alumno y 4 columnas. Las primeras 3 columnas representan las notas de cada uno de los alumnos, mientras que la 4 es el promedio de las mismas. Se desea un programa que sea capaz de permitir la carga por teclado de las 3 notas de cada alumno, de realizar el cálculo automático del promedio, de guardar el mismo en la 4 columna y luego mostrar por pantalla cada una de las notas y el promedio obtenido. Por ejemplo.

**Nota:** tener en cuenta que generalmente las calificaciones y los promedios no suelen ser números enteros, sino que pueden tener decimales.

**3.**Se desea un programa que sea capaz de rellenar completamente con números 1 una matriz de 4 x 4 a excepción de su diagonal principal, la cual debe ser rellenada con números 0. Una vez realizada la carga, se desea mostrar la matriz de forma ordenada por pantalla. Tener como ejemplo la matriz que se presenta a continuación.

**4.**Una casa de comida rápida utiliza una matriz de 20 filas y 5 columnas para determinar cada plato del menú y sus correspondientes ingredientes que contiene. Para ello, cada fila representa un plato principal en donde en la primera columna se especifica su nombre (por ejemplo: Milanesa), y en las demás los posibles acompañamientos que puede tener el plato (por ejemplo, puré, ensalada rusa, papas fritas, ensalada de lechuga y tomate, ensalada de zanahoria, etc). En caso de que tenga el plato menos de 4 tipos de acompañamiento, en la matriz se cargará la palabra Ninguno en las posiciones que queden vacías.

Se desea un programa que sea capaz de permitir la carga de la matriz, de buscar un plato en particular que un cliente ingrese por teclado y mostrarle de forma automática los posibles acompañamientos del mismo.

**5.**Una empresa constructora desea llevar a cabo un premio para sus obreros que hayan realizado horas extras en los últimos 6 meses. Para ello, lleva una matriz de control de 15 filas x 6 columnas, donde cada fila corresponde a un empleado distinto y donde cada columna representa el total de horas trabajadas por mes. El monto de horas normal trabajado x mes es de 8 horas diarias, 5 días a la semana, 4 semanas por mes, es decir: 160 horas. Todo monto superior a este es considerado como horas extras.

Se desea un programa que permita la carga de horas trabajadas por cada obrero y que sea capaz de determinar aquellos que hayan realizado horas extras. Para registrar las horas extras realizadas se utilizará un vector de 15 posiciones en donde se deberá calcular y almacenar la cantidad de horas extras de cada obrero. En caso de que no haya realizado horas extras, deberá incluirse un número 0. Tomar como ejemplo la siguiente imagen:

Te dejamos algunas propuestas de prácticas que puedes realizar como desafío, para que pongas en práctica lo aprendido. **Estos ejercicios no son obligatorios y no tienes que entregarlos**. Te invitamos a que los compartas en tus redes con tus compañeros y observes cómo otros lo han resuelto.

**1.**Crear una clase Mascota, que tenga los siguientes atributos: nombre, especie, sexo, color, pelaje y raza (tener en cuenta todos sus atributos, constructores y métodos getters y setters). A partir de ello:

a.      Crear un ArrayList de tipo Mascota. Crear 5 mascotas y guardarlas en la lista.

b.      Recorrer la lista creada y mostrar por pantalla el nombre, especie y pelaje de las mascotas almacenadas.

c.       Durante el recorrido, cambiar el nombre de dos mascotas. Volver a recorrer la lista y mostrar los nuevos datos

d.      Agregar dos nuevas mascotas a la lista.

e.      Recorrer la lista pero solo mostrando las mascotas que sean de la especie “perro”.

**2.**Crear una clase persona que tenga los siguientes atributos: dni, nombre, apellido, dirección, celular, edad (tener en cuenta todos sus atributos, constructores y métodos getters y setters). A partir de ello:

a.      Crear una LinkedList de tipo Persona y agregar 20 personas distintas.

b.      Crear dos LinkedLists de tipo Persona diferentes a la primera, una para almacenar a los menores de edad y otra para los mayores.

c.       Recorrer la lista principal buscando a las personas que sean mayores de 18 años. En caso que una persona tenga o sea mayor a 18 años, agregarla a la lista de mayores, caso contrario agregarla a una lista de menores.

d.      Una vez realizada la separación, recorrer ambas nuevas listas y mostrar por pantalla el nombre, apellido y edad de cada una de las personas contenidas en cada lista.

**3.**Crear una stack que permita objetos de tipo Autos, donde cada objeto auto debe tener: num\_patente, marca, modelo, color, chasis. Cargar la pila con 7 autos diferentes luego hacer lo siguiente:

a.      Buscar un auto de color rojo.

b.      Mostrar el auto que se encuentra en la cima de la pila.

c.       Borrar el auto que se encuentra en la cima de la pila.

d.      Agregar dos nuevos autos a la pila.

Recorrer la pila mostrando el nuevo contenido luego de las operaciones realizadas.

## Práctica Excepciones

##### Te dejamos algunas propuestas de prácticas que puedes realizar como desafío, para que pongas en práctica lo aprendido. ****Estos ejercicios no son obligatorios y no tienes que entregarlos****. Te invitamos a que los compartas en tus redes con tus compañeros y observes cómo otros lo han resuelto.

**1.**Realizar un programa donde se provoque una excepción de tipo “error por desbordamiento”. Tratar la excepción mediante los bloques try y catch y mostrar un mensaje indicando la situación. Por ejemplo: “Se produjo un error por desbordamiento”.

**2.**Realizar un programa que permita el ingreso de dos números por teclado para ser divididos entre sí. Tomar como precaución la utilización del try y el catch por si el segundo número ingresado para dividir se trata de un cero. Mostrar un mensaje adecuado en caso de que se produzca la excepción.

**3.**Realizar una clase de Excepciones Personalizada que extienda de la clase Exception. Implementar los diferentes métodos que sean necesarios teniendo en cuenta cada uno de los mensajes personalizados que podrían arrojar. Provocar la ejecución de la excepción personalizada mediante la sentencia throw para probar la ejecución de la excepción y responder… ¿Se ejecuta correctamente? ¿Qué método de la excepción se ejecutó?

## Práctica JDBC + Collections

##### Te dejamos algunas propuestas de prácticas que puedes realizar como desafío, para que pongas en práctica lo aprendido. ****Estos ejercicios no son obligatorios y no tienes que entregarlos****. Te invitamos a que los compartas en tus redes con tus compañeros y observes cómo otros lo han resuelto.

**1.**Una despensa de un barrio desea realizar el alta de sus diferentes productos a una base de datos. Para ello, tiene una tabla creada en su base de datos, llamada productos. A partir de esto desea poder solicitar por teclado cada uno de los elementos de un producto: código, nombre, marca, precio, cantidad\_stock y guardarlos en la correspondiente tabla. Utilizar para ello la tecnología JDBC y la correspondiente consulta SQL.

**2.**Un youtuber posee almacenados en una tabla llamada videos dentro de una base de datos, todos sus videos que ya fueron subidos a Youtube. Desea un programa que sea capaz de traer todos esos videos, almacenarlos en una lista y mostrarlos luego por pantalla. Utilizar para ello la tecnología JDBC.

**3.**Una fabricadora de bijouterie desea realizar un sorteo para su página de Instagram. Para ello, había almacenado manualmente en una base de datos todas sus clientes que habían comentado la publicación. Sin embargo, se dio cuenta que varios no habían cumplido con las condiciones del sorteo, por lo cual desea eliminarlos. Cada cliente tiene los datos: nombre, apellido, edad, compartió y comento. Donde los atributos compartió y comento pueden contener los valores si o no.

Realizar un programa que sea capaz de eliminar de la base de datos todos los clientes cuyos campos compartió y/o comento contengan un no. Luego de hacer la eliminación, traer todos los clientes en una lista y mostrarlos por pantalla. Utilizar para esto la tecnología JDBC.

**4.**Una contadora encargada de la liquidación de sueldos de una empresa desea realizar la actualización de sueldos de los empleados en base a una suba de salario planificada para el nuevo mes. Para almacenar cada empleado se tiene una tabla en una base de datos que contiene: nombre, apellido, cargo y sueldo.

Realizar un programa que sea capaz de actualizar a $80.000 los sueldos de todos los empleados que tengan el cargo de “programador”.

Práctica JSP + Servlets + Bases de datos

Te dejamos esta propuesta de ejercicio que puedes realizar como desafío, para que pongas en práctica lo aprendido. **Este ejercicio no es obligatorio y no tienes que entregarlo**. Te invitamos a que los compartas en tus redes con tus compañeros y observes cómo otros lo han resuelto.

El depósito de un supermercado de la ciudad de Posadas, Misiones, Argentina necesita de un Sistema Web para la administración del stock de sus productos. Éste deberá poder trabajar con productos alimenticios perecederos (carne, verduras, huevos, queso, etc), no perecederos (Fideo, arroz, etc) o artículos de otras categorías (artículos de limpieza, bazar, etc).

El dueño del supermercado dio a conocer que necesita que el sistema pueda realizar por el momento las siguientes acciones:

●       Dar de alta nuevos productos en la base de datos a partir de la carga desde el navegador.

●       Visualizar en una tabla todos los productos existentes en la base de datos, junto con las cantidades existentes, precio de costo y precio de venta.

Por otra parte, el cliente solicitó una interfaz gráfica de usuario WEB agradable y mencionó que los colores de su empresa son verde y rojo con detalles en amarillo, por lo que desea que las pantallas/páginas/pestañas del sistema posean esa combinación de colores (recordar que para esto se puede incluir en el JSP, en conjunto con HTML y estilos CSS).

**Consideraciones a tener en cuenta**

●       Utilizar páginas JSP para el desarrollo de la IGU.

●       **Respetar la separación de capas**: JSP ---> Servlet -----> Lógica ----> Persistencia ---> Base de Datos

A partir de todas estas especificaciones, realizar el desarrollo web necesario para cumplir con lo solicitado por el dueño del supermercado

Práctica Sockets

Te dejamos esta propuesta de ejercicio que puedes realizar como desafío, para que pongas en práctica lo aprendido. **Este ejercicio no es obligatorio y no tienes que entregarlo**. Te invitamos a que los compartas en tus redes con tus compañeros y observes cómo otros lo han resuelto.

Una casa de electrodomésticos posee 2 cajas en las cuales atiende por turnos a cada uno de sus clientes para el abono de los pagos por las compras realizadas. A cada cliente se le otorga un número con el cual será atendido cuando sea su turno.

Los empleados de cada caja cuentan con una aplicación cliente que, por cada solicitud que envían al servidor, aumentan en 1 el valor de los turnos que se muestran en pantalla a los clientes que aguardan. Por ejemplo: Si el turno se encuentra en el número 15 en la pantalla y un empleado a través de la aplicación cliente envía una solicitud, éste debe aumentar a 16. El encargado de recibir las peticiones y aumentar los turnos es el servidor, el cual se encuentra ejecutándose constantemente a la espera de solicitudes de aumento de turnos.

Se necesita realizar las aplicaciones cliente y servidor necesarias que sea capaces de, a partir de las distintas solicitudes que envían los empleados de la casa de electrodomésticos, aumentar el número de turno que se muestra en la pantalla. Para ello, recordar que el encargado de recibir las solicitudes y mostrar los turnos en pantalla es el servidor, mientras que los encargados de enviar los mensajes son las aplicaciones clientes que se encuentran en la caja de cada empleado.