

INFORME DE GUÍA PRÁCTICA

1 Portada

Tema: Uso de Arreglos Unidireccionales
Unidad de Organización Curricular: Elige un elemento.
Nivel y Paralelo: 1 - B
Alumnos participantes: Barragán Alexis

Asignatura: Fundamentos de Programación
Docente: Ing. Hernan Naranjo, Mg.

2 Informe de guía práctica

2.1 Objetivos

2.1.1 General

Desarrollar aplicaciones en Java que utilicen vectores para almacenar, procesar y manipular datos de manera eficiente, aplicando principios de programación estructurada y modular.

2.1.2 Específicos

- Implementar operaciones básicas con vectores en Java, como suma, promedio, búsqueda y ordenamiento de elementos.
- Diseñar métodos reutilizables que permitan modularizar el manejo de vectores dentro del programa.
- Desarrollar programas interactivos que permitan ingresar, procesar y mostrar datos almacenados en vectores de forma dinámica.

2.2 Modalidad

Trabajo Individual

2.3 Tiempo de duración

Presenciales: 4 No presenciales: 0

2.4 Instrucciones

Acciones previas: Ingrese al aula virtual de la asignatura en donde se halla el trabajo del tema tratado. Elabore el trabajo siguiendo las definiciones, conceptos, procesos aprendidos en clase. Leer y subrayar cada problema.

2.5 Listado de equipos, materiales y recursos

Listado de equipos y materiales generales empleados en la guía práctica:

-

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:

- X Plataformas educativas
- ☐ Simuladores y laboratorios virtuales
- ☐ Aplicaciones educativas
- X Recursos audiovisuales
- ☐ Gamificación
- X Inteligencia Artificial
- Otros (Especifique): _____

2.6 Actividades por desarrollar

Con la información propuesta en la tarea: Analice la información de cada uno de los ejercicios propuestos. Realice el Diseño, prueba, codificación y posteriormente compile, ejecute y verifique el resultado. Arribe a conclusiones. Subir a la plataforma el archivo en formato .PDF del trabajo hasta la fecha indicada.

2.7 Resultados obtenidos

De acuerdo con el tipo de trabajo, se plasmarán los resultados alcanzados en la guía práctica una vez ejecutadas las actividades. Se pueden emplear figuras y tablas las cuales deben ser numeradas.

2.8 Habilidades blandas empleadas en la práctica

- ☐ Liderazgo
- ☐ Trabajo en equipo
- ☐ Comunicación asertiva
- ☐ La empatía
- X Pensamiento crítico
- X Flexibilidad
- X La resolución de conflictos
- X Adaptabilidad
- X Responsabilidad

2.9 Conclusiones

El uso de vectores en Java facilita el almacenamiento y manejo eficiente de datos secuenciales, permitiendo optimizar procesos repetitivos.

La implementación de métodos específicos para operar con vectores promueve una programación más estructurada, clara y reutilizable.

Dominar el manejo de vectores fortalece las bases de la programación en Java y es fundamental para resolver problemas más complejos en el desarrollo de software.

2.10 Recomendaciones

Se recomienda practicar continuamente con ejercicios variados que involucren vectores en Java, enfocándose en la implementación de métodos reutilizables y el uso eficiente de estructuras de control, para fortalecer la lógica de programación y mejorar la capacidad de resolver problemas de manera estructurada. [?]

2.11 Referencias

Guia Appe Deitel, P. Deitel, H. (2015). Java: Cómo programar (10ª ed.). Pearson Educación. Deitel Deitel. Capítulo 7: Arreglos y Vectores
<https://alexis870-2.github.io/TareaVectores/>

2.12 Anexos

Ejercicios Propuestos :

Suma de elementos positivos y negativos Dado un vector de n enteros, calcular por separado la suma de los elementos positivos y la suma de los negativos.

Frecuencia de números pares e impares Contar cuántos números pares e impares hay en un vector de 50 elementos generados aleatoriamente entre 1 y 100.

Promedio de calificaciones Ingresar n calificaciones (entre 0 y 10), almacenarlas en un vector y calcular el promedio. Mostrar cuántas están sobre el promedio.

Rotación de vector hacia la izquierda Rotar los elementos de un vector una posición hacia la izquierda (el primer elemento pasa al final).

Vector palíndromo Determinar si un vector de enteros es palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda).

Elementos duplicados Identificar y mostrar los elementos duplicados dentro de un vector de n elementos.

Intercambio de extremos Intercambiar el primer y último elemento de un vector.

Vector de ocurrencias Crear un vector con números entre 1 y 5 (n=100) y contar cuántas veces aparece cada uno.

Producto punto de dos vectores Dados dos vectores de tamaño n, calcular su producto punto.

Intercalar dos vectores Dado dos vectores A y B de igual tamaño, construir un nuevo vector C intercalando sus elementos.

Buscar un elemento Buscar un número ingresado por el usuario dentro del vector. Indicar si se encuentra y en qué posición(es).

Mover ceros al final Mover todos los ceros del vector al final, sin cambiar el orden del resto de elementos.

Máximo y mínimo Encontrar el valor máximo y mínimo en un vector de n elementos, e indicar su(s) posición(es).

Fusionar vectores sin duplicados Unir dos vectores en uno solo, eliminando elementos duplicados.

Invertir vector Crear una copia del vector original pero con los elementos en orden inverso.

Comparar dos vectores Verificar si dos vectores de igual longitud son exactamente iguales (misma posición y valor).

Desplazamiento circular a la derecha Realizar un desplazamiento circular hacia la derecha k veces (ingresado por el usuario).

Eliminar elemento Eliminar un número del vector ingresado por el usuario y reorganizar el vector.

Histograma de ocurrencias Generar un vector de enteros del 1 al 6 (como dados) y mostrar un histograma textual de las frecuencias.

Promedios por categorías Ingresar edades de n personas, clasificar en rangos (¿18, 18–60, ¿60), y calcular el promedio por categoría.

2.12.1 Pasos para realizar el programa

- 1. Importación de clases necesarias

Scanner: Permite leer datos que el usuario escribe por teclado. Arrays: Ofrece utilidades para trabajar con vectores, como ordenar (sort) o copiar (copyOf).

```
import java.util.Arrays; import java.util.Scanner;
```

- 2. Estructura del programa Todo el código está dentro de una clase pública llamada App, que contiene:

Un método main() que muestra el menú y gestiona la navegación.

20 métodos llamados problema1() a problema20() que resuelven distintos ejercicios de vectores.

- 3. Menú principal

Este main() sirve como menú interactivo, permitiendo al usuario elegir qué ejercicio ejecutar. Usa un do-while para repetir el menú hasta que el usuario decida salir.

```
public static void main(String[] args) Scanner sc = new Scanner(System.in); int opcion;  
do // Muestra un menú con 20 ejercicios // Lee la opción del usuario // Ejecuta el método  
correspondiente con switch-case while (opcion != 0);  
sc.close();
```

- 4. Descripción general de los problemas Problema Funcionalidad 1 Suma los elementos de un vector ingresado. 2 Encuentra el mayor valor en un vector. 3 Cuenta cuántos valores pares hay en un vector de 10 números. 4 Muestra un vector de 5 elementos en orden inverso. 5 Calcula el promedio de un vector de 6 elementos. 6 Ordena un vector de 5 elementos de menor a mayor. 7 Busca un número en un vector y dice si está presente. 8 Copia un vector original a otro vector. 9 Suma dos vectores de 3 elementos cada uno. 10 Multiplica un vector por un escalar. 11 Muestra el menor valor y su posición. 12 Cuenta cuántas veces aparece un número dado. 13 Muestra un vector sin un elemento específico. 14 Verifica si un vector está ordenado de forma ascendente. 15 Invierte el vector (intercambia extremos). 16 Fusiona dos vectores A y B en uno solo. 17 Calcula la moda (el número más repetido). 18 Muestra los elementos únicos (sin duplicados). 19 Desplaza los elementos una posición a la derecha. 20 Verifica si un vector es un palíndromo (simétrico).

- 5. Estructura interna de cada método Todos los métodos siguen esta lógica básica:

1. Declaración de variables y vectores:

```
int[] vector = new int[n]; // según el tamaño requerido
```

2. Lectura de datos:

```
for (int i = 0; i < n; i++) System.out.print("Elemento " + i + ": "); vector[i] = sc.nextInt();
```

3. Procesamiento lógico del ejercicio:

Sumar, buscar, contar pares, verificar orden, copiar, etc.

4. Mostrar resultados:

```
System.out.println("Resultado: " + ...);
```

- 6. Técnicas destacadas utilizadas for-each: para recorrer un vector fácilmente:
for (int val : vector)
Arrays.sort(): para ordenar un vector.
Arrays.copyOf(): para copiar un vector.
Intercambios con temp para invertir o desplazar elementos.
Uso de banderas booleanas (encontrado, ordenado, etc.) para verificaciones.
- 7. Objetivo general del programa Este programa fue diseñado para practicar 20 ejercicios clásicos con vectores en Java, usando un enfoque modular:
Cada problema se resuelve en su propio método.
El menú principal permite probarlos de forma independiente.
Esto ayuda a:
Reforzar el uso de ciclos (for, do-while)
Aprender operaciones básicas con arreglos (int[])
Usar condicionales (if, switch)
Usar librerías (Arrays, Scanner)