



Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA



Reporte de actividad

Programación orientada a objetos.

Práctica No. 6:

Gestión y manipulación de arreglos de objetos en Java

Alumno

Joshua Osorio Osorio - 1293271

Docente

Itzel Barriba Cazares

26 de marzo de 2025

ÍNDICE

| | |
|---|---|
| Objetivo..... | 2 |
| Introducción..... | 2 |
| Lista de materiales..... | 3 |
| Desarrollo..... | 3 |
| 1. Creación de los arreglos (no utilizar arrayList):..... | 3 |
| 2. Inicializa los arreglos:..... | 3 |
| 3. Modificación del arreglo:..... | 3 |
| 4. Procesamiento de arreglos:..... | 4 |
| Estructura del programa..... | 4 |
| Código <Principal>:..... | 4 |
| Análisis de Resultados..... | 4 |
| Conclusiones..... | 5 |
| Bibliografía..... | 5 |
| Entrega:..... | 5 |
| Criterios para evaluar:..... | 6 |

Objetivo

- Aplicar los conceptos de POO en la manipulación de arreglos de objetos en Java.
- Implementa operaciones básicas como creación, búsqueda, modificación y eliminación de objetos dentro de un arreglo.
- Desarrollar habilidades en el uso de ciclos y estructuras de control para gestionar colecciones de datos (solo arreglos)

Introducción

Los arreglos son una de las estructuras de datos más esenciales en programación, ya que permiten almacenar y manipular conjuntos de elementos de manera eficiente. En esta práctica, se exploró el manejo de arreglos en Java, aplicando conceptos como la creación, inicialización, modificación y procesamiento de arreglos de diferentes tipos de datos (enteros, decimales, caracteres, etc.). El objetivo fue desarrollar un programa interactivo que permitiera al usuario realizar operaciones como la suma de elementos, el cálculo de promedios y la identificación de valores extremos (máximo y mínimo), todo ello bajo un enfoque de programación estructurada y aplicando buenas prácticas.

Para lograrlo, se utilizaron herramientas como el JDK de Java junto con recursos de documentación oficial. El programa resultante no solo cumple con los requerimientos funcionales, sino que también fomenta habilidades prácticas como la validación de entradas, la modularización del código y la creación de interfaces de usuario sencillas mediante menús interactivos. Esta experiencia reforzó conocimientos teóricos y sentó las bases para abordar desafíos más complejos en el futuro.

Para llevar a cabo esta investigación y práctica, se requirió lo siguiente.

Lista de materiales

- Computadora con entorno de desarrollo compatible con Java.
- JDK (Java Development Kit) instalado (versión 8 o superior recomendada).
- Editor de texto.
- Documentación oficial de Java para referencia sobre la clase String.
- Recursos de apoyo como libros y artículos en línea sobre String en Java.
- Compilador de Java (incluido en el JDK) para ejecutar y probar el código.

Desarrollo

Programa 1:

Desarrolla un sistema en Java que permita gestionar y buscar libros en una biblioteca, aplicando los principios de POO, incluyendo encapsulamiento, constructores y manipulación de colección de objetos.

La clase Libro representará un libro dentro de la biblioteca y deberá contener los siguientes atributos.

- Título del libro
- Año de publicación
- Nombre de la editorial
- Genero del libro
- Número de páginas
- Autores

Cada atributo deberá contar con:

Métodos getter para obtener sus valores.

- Métodos setter para modificar sus valores.
- Al menos dos constructores:

La clase Biblioteca gestionará los libros almacenados y deberá incluir los siguientes atributos:

- Todos los libros disponibles en la biblioteca
- Nombre de la biblioteca
- Dirección de la biblioteca
- Nombre del encargado de la biblioteca

Debe contener los siguientes métodos:

- agregarLibro: Permite agregar un libro a la biblioteca.
- buscarPorAnio(int anio): Devuelve una lista de libros publicados en un año específico.
- buscarPorAutor(String autor): Devuelve una lista de libros escritos por un autor específico.
- buscarPorTitulo(String titulo): Devuelve un libro que coincida con el título especificado.
- ordenarLibrosPorAnio(): Ordena los libros por año de publicación.
- ordenarLibrosPorTitulo(): Ordena los libros alfabéticamente por título.

Java

```
public class Libro {
    // Atributos privados de la clase.

    private String tituloLibro;
    private int anioPublicacion;
    private String editorial;
    private String generos;
    private int numPaginas;
    private String autores;

    //Constructores
    public Libro() {
        this.tituloLibro = "";
        this.anioPublicacion = 0000;
        this.editorial = "";
        this.generos = "";
        this.numPaginas = 0;
        this.autores = "";
    }

    public Libro(String tituloLibro, int anioPublicacion, String editorial,
String generos, int numPaginas, String autores) {
        this.tituloLibro = tituloLibro;
        this.anioPublicacion = anioPublicacion;
        this.editorial = editorial;
        this.generos = generos;
        this.numPaginas = numPaginas;
        this.autores = autores;
    }
}
```

```
public Libro(String tituloLibro, String autores) {
    this.tituloLibro = tituloLibro;
    this.anioPublicacion = 0;
    this.editorial = "";
    this.generos = "";
    this.numPaginas = 0;
    this.autores = autores;
}

// Metodos set de la clase, se utilizan para declarar los atributos.
public void setTituloLibro(String tituloLibro) {
    this.tituloLibro = tituloLibro;
}

public void setAnioPublicacion(int anioPublicacion) {
    this.anioPublicacion = anioPublicacion;
}

public void setEditorial(String editorial) {
    this.editorial = editorial;
}

public void setGeneros(String generos) {
    this.generos = generos;
}

public void setNumPaginas(int numPaginas) {
    this.numPaginas = numPaginas;
}

public void setAutores(String autores) {
    this.autores = autores;
}

// Metodos get para traer valores de la clase.
public String getTituloLibro() {
    return tituloLibro;
}

public int getAnioPublicacion() {
    return anioPublicacion;
}
```

```

    public String getEditorial() {
        return editorial;
    }

    public String getGeneros() {
        return generos;
    }

    public int getNumPaginas() {
        return numPaginas;
    }

    public String getAutores() {
        return autores;
    }
}

```

Biblioteca.java

Java

```

package Prog1;

public class Biblioteca {

    /* Atributos privados de la clase. */
    private final int espacioLibros = 10;
    /* Se declara como final porque indica el tamaño del arreglo y ese no se
    puede modificar en el trn */
    private Libro[] librosDisponibles = new Libro[espacioLibros];

    private int cantidadLibros = 0;
    private String nombre;
    private String direccion;
    private String encargado;

    /*Constructores*/
    public Biblioteca() {
        this.cantidadLibros = 0;
        this.nombre = "";
        this.direccion = "";
        this.encargado = "";
    }
}

```

```
}

    public Biblioteca(int cantidadLibros, String nombre, String direccion,
String encargado) {
        this.nombre = nombre;
        this.cantidadLibros = cantidadLibros;
        this.direccion = direccion;
        this.encargado = encargado;
    }

    /* Metodos Set */
    public void setLibrosDisponibles(Libro[] librosDisponibles) {
        this.librosDisponibles = librosDisponibles;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }

    public void setEncargado(String encargado) {
        this.encargado = encargado;
    }

    /* Metodos Get */
    public Libro[] getLibrosDisponibles() {
        return librosDisponibles;
    }

    public String getNombre(){
        return nombre;
    }

    public String getDireccion() {
        return direccion;
    }

    public String getEncargado() {
        return encargado;
    }
}
```

```

/* Otros metodos de la clase */
public void agregarLibro(Libro nuevoLibro) {
    librosDisponibles[cantidadLibros] = nuevoLibro;
    this.cantidadLibros++;
}

public Libro[] buscarPorAnio(int anio) {
    Libro[] arr = new Libro[cantidadLibros];
    int contador = 0;
    if (cantidadLibros > 0) {
        for (int l = 0; l < cantidadLibros; l++) {
            if (librosDisponibles[l].getAnioPublicacion() == anio) {
                arr[contador] = librosDisponibles[l];
                contador++;
            }
        }
    }
    return arr;
}

public Libro[] buscarPorAutor(String autor) {
    Libro[] arr = new Libro[cantidadLibros];
    int contador = 0;
    for (int l = 0; l < cantidadLibros; l++) {
        if (librosDisponibles[l].getAutores().equalsIgnoreCase(autor)) {
            arr[contador] = librosDisponibles[l];
            contador++;
        }
    }
    return arr;
}

public Libro[] buscarPorTitulo(String titulo) {
    Libro[] arr = new Libro[cantidadLibros];
    int contador = 0;
    for (int l = 0; l < cantidadLibros; l++) {
        if (librosDisponibles[l].getAutores().equalsIgnoreCase(titulo)) {
            arr[contador] = librosDisponibles[l];
            contador++;
        }
    }
    return arr;
}

```



```

        /* Metodo quick sort para el ordenamiento de libros por anio */

        /* ordenamientoBurbuja */
        public void ordenarLibrosPorAnio(Libro[] lista) {
            for (int numPasada = cantidadLibros - 1; numPasada > 0; numPasada--) {
                for (int i = 0; i < numPasada; i++) {
                    if (lista[i].getAnioPublicacion() > lista[i +
1].getAnioPublicacion()) {
                        Libro temp = lista[i];
                        lista[i] = lista[i + 1];
                        lista[i + 1] = temp;
                    }
                }
            }
        }

        // public ordenarLibrosPorTitulo(){}
        public void ordenarLibrosPorTitulo(Libro[] lista) {
            for (int numPasada = cantidadLibros - 1; numPasada > 0; numPasada--) {
                for (int i = 0; i < numPasada; i++) {
                    if ((int) (lista[i].getTituloLibro().charAt(0)) > (int)
(lista[i + 1].getTituloLibro()).charAt(0)) {
                        Libro temp = lista[i];
                        lista[i] = lista[i + 1];
                        lista[i + 1] = temp;
                    }
                }
            }
        }
    }
}

```

Principal.java

```

Java
/*-----
Practica: 6 Gestión y manipulación de arreglos de objetos en Java.
Nombre: Joshua Osorio
Materia: Programación Orientada a Objetos
Fecha: Marzo/20/2025
-----*/

```

```

import java.util.Scanner;

public class Principal {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Creación de un objeto de la clase Libro
        Libro nuevoLibro = new Libro();
        Biblioteca biblioteca = new Biblioteca();

        // Declaración de las variables utilizadas en el programa
        String titulo, editorial, genero, autores, direccion, encargado;
        int opcion, anioPublicacion, numpaginas;

        System.out.println("Sistema para capturar información de libro");

        // Inicio el ciclo do while para estar mostrando siempre las opciones.
        do {
            // Se despliega información para mostrar las opciones.
            System.out.println("\nMenú");
            System.out.println("\t1 - Modificar nombre de la biblioteca");
            System.out.println("\t2 - Modificar dirección");
            System.out.println("\t3 - Modificar encargado");
            System.out.println("\t4 - Agregar libro");
            System.out.println("\t5 - Buscar por año");
            System.out.println("\t6 - Buscar por autor");
            System.out.println("\t7 - Buscar por título");
            System.out.println("\t8 - Ordenar libros por año");
            System.out.println("\t9 - Ordenar libros por título");
            System.out.println("\t0 - Salir");

            // Se captura la opción que desea.
            System.out.print("\tOpción: ");
            opcion = input.nextInt();

            // Limpia el búfer después de ingresar un numero
            input.nextLine();

            switch (opcion) {
                case 1:
                    System.out.println("\tModificar nombre de la biblioteca");
                    System.out.printf("\tNombre anterior: %s\n",
biblioteca.getNombre());

```

```

        System.out.print("\tIngrese nuevo nombre: ");
        titulo = input.nextLine();
        biblioteca.setNombre(titulo);
        System.out.printf("\tNombre nuevo: %s\n",
biblioteca.getNombre());
        break;

    case 2:
        System.out.println("\tModificar dirección");
        System.out.printf("\tDirección anterior: %s\n",
biblioteca.getDireccion());
        System.out.print("\tIngrese nueva dirección: ");
        direccion = input.nextLine();
        biblioteca.setDireccion(direccion);
        System.out.printf("\tDirección nueva: %s\n",
biblioteca.getDireccion());
        break;

    case 3:
        System.out.println("\tModificar encargado");
        System.out.printf("\tEncargado anterior: %s\n",
biblioteca.getEncargado());
        System.out.print("\tIngrese nuevo encargado: ");
        encargado = input.nextLine();
        biblioteca.setEncargado(encargado);
        System.out.printf("\tEncargado nuevo: %s\n",
biblioteca.getEncargado());
        break;

    case 4:
        System.out.println("\tAgregar libro");
        System.out.printf("\tIngrese título del libro: ");
        titulo = input.nextLine();
        nuevoLibro.setTituloLibro(titulo);

        System.out.printf("\tIngrese año de publicación: ");
        anioPublicacion = input.nextInt();
        input.nextLine(); // Limpiar buffer
        nuevoLibro.setAnioPublicacion(anioPublicacion);

        System.out.printf("\tIngrese editorial: ");
        editorial = input.nextLine();
        nuevoLibro.setEditorial(editorial);

        System.out.printf("\tIngrese género: ");
        genero = input.nextLine();

```

```

        nuevoLibro.setGeneros(genero);

        System.out.printf("\tIngrese número de páginas: ");
        numpaginas = input.nextInt();
        input.nextLine(); // Limpiar buffer
        nuevoLibro.setNumPaginas(numpaginas);

        System.out.printf("\tIngrese autores: ");
        autores = input.nextLine();
        nuevoLibro.setAutores(autores);

        biblioteca.agregarLibro(nuevoLibro);
        System.out.println("\tLibro agregado exitosamente.");
        break;

    case 5:
        System.out.println("\tBuscar libros por año");
        System.out.printf("\tIngrese año a buscar: ");
        anioPublicacion = input.nextInt();

        Libro[] librosPorAnio =
biblioteca.buscarPorAnio(anioPublicacion);
        mostrarLibros(librosPorAnio);
        break;

    case 6:
        System.out.println("\tBuscar libros por autor");
        System.out.printf("\tIngrese nombre del autor: ");
        autores = input.nextLine();

        Libro[] librosPorAutor =
biblioteca.buscarPorAutor(autores);
        mostrarLibros(librosPorAutor);
        break;

    case 7:
        System.out.println("\tBuscar libros por título");
        System.out.printf("\tIngrese título a buscar: ");
        titulo = input.nextLine();

        Libro[] librosPorTitulo =
biblioteca.buscarPorTitulo(titulo);
        mostrarLibros(librosPorTitulo);

```

```

        break;

        case 8:
            System.out.println("\tOrdenar libros por año");

            biblioteca.ordenarLibrosPorAnio(biblioteca.getLibrosDisponibles());
            System.out.println("\tLibros ordenados por año.");
            break;

        case 9:
            System.out.println("\tOrdenar libros por título");

            biblioteca.ordenarLibrosPorTitulo(biblioteca.getLibrosDisponibles());
            System.out.println("\tLibros ordenados por título.");
            break;

        case 0:
            System.out.println("\tSaliendo del sistema...");
            break;

        default:
            System.out.println("\tOpción no válida, intente
nuevamente.");
            break;
    }
} while (opcion != 0);

input.close();
}

// Método auxiliar para mostrar libros
public static void mostrarLibros(Libro[] libros) {
    if (libros == null || libros.length == 0) {
        System.out.println("\tNo se encontraron libros.");
        return;
    }

    System.out.println("\tLibros encontrados:");
    for (Libro libro : libros) {
        if (libro != null) {
            System.out.printf("\tTítulo: %s, Autor: %s, Año: %d\n",
                libro.getTituloLibro(),
                libro.getAutores(),
                libro.getAnioPublicacion());
        }
    }
}

```

```
}  
}  
}
```

Programa 2:

Desarrollar una aplicación en Java que gestione un arreglo de objetos de una clase Estudiante. Deberás implementar lo siguiente:

Una clase Estudiante con los siguientes atributos:

- Nombre
- Edad
- Matricula
- Carrera

Crear una clase GestorEstudiantes que contenga un arreglo de objetos Estudiantes y los siguientes métodos:

- agregarEstudiante(Estudiante e): Agrega un nuevo estudiante al arreglo.
 - mostrarEstudiantes(): Muestra todos los estudiantes en el arreglo.
 - buscarEstudiantePorMatricula(String matricula): Busca un estudiante por matrícula y devuelve sus datos.
 - modificarEstudiante(String matricula, String nuevoNombre, int nuevaEdad): Modifica los datos de un estudiante.
 - eliminarEstudiante(String matricula): Elimina un estudiante del arreglo.
- Realice pruebas con diferentes estudiantes y se ejecuten los métodos anteriores.

Estudiante.java

Java

```
public class Estudiante {  
    // Atributos privados  
  
    private String nombre;  
    private int edad;  
    private final String matricula;  
    private String carrera;
```

```

// Constructor
public Estudiante(String nombre, int edad, String matricula, String
carrera) {
    this.nombre = nombre;
    this.edad = edad;
    this.matricula = matricula;
    this.carrera = carrera;
}

// Getters
public String getNombre() {
    return nombre;
}

public int getEdad() {
    return edad;
}

public String getMatricula() {
    return matricula;
}

public String getCarrera() {
    return carrera;
}

// Setters
public void setNombre(String nombre) {
    this.nombre = nombre;
}

public void setEdad(int edad) {
    this.edad = edad;
}

public void setCarrera(String carrera) {
    this.carrera = carrera;
}

// Método toString para imprimir detalles del estudiante
@Override
public String toString() {
    return "Estudiante{"
        + "nombre='" + nombre + '\''

```

```

        + ", edad=" + edad
        + ", matricula='" + matricula + '\''
        + ", carrera='" + carrera + '\''
        + '}';
    }
}

```

GestorEstudiantes.java

Java

```

public class GestorEstudiantes {

    // Máximo número de estudiantes
    private static final int MAX_ESTUDIANTES = 100;

    // Arreglo de estudiantes
    private final Estudiante[] estudiantes;

    // Contador de estudiantes en el arreglo
    private int contadorEstudiantes;

    // Constructor
    public GestorEstudiantes() {
        estudiantes = new Estudiante[MAX_ESTUDIANTES];
        contadorEstudiantes = 0;
    }

    // Método para agregar estudiante
    public void agregarEstudiante(Estudiante e) {
        if (contadorEstudiantes < MAX_ESTUDIANTES) {
            estudiantes[contadorEstudiantes] = e;
            contadorEstudiantes++;
            System.out.println("Estudiante agregado exitosamente.");
        } else {
            System.out.println("No se pueden agregar más estudiantes. Límite alcanzado.");
        }
    }

    // Método para mostrar todos los estudiantes
    public void mostrarEstudiantes() {

```



```

        if (contadorEstudiantes == 0) {
            System.out.println("No hay estudiantes registrados.");
            return;
        }

        System.out.println("Lista de Estudiantes:");
        for (int i = 0; i < contadorEstudiantes; i++) {
            System.out.println(estudiantes[i]);
        }
    }

    // Método para buscar estudiante por matrícula
    public Estudiante buscarEstudiantePorMatricula(String matricula) {
        for (int i = 0; i < contadorEstudiantes; i++) {
            if (estudiantes[i].getMatricula().equals(matricula)) {
                return estudiantes[i];
            }
        }
        return null;
    }

    // Método para modificar estudiante
    public void modificarEstudiante(String matricula, String nuevoNombre, int
nuevaEdad) {
        Estudiante estudiante = buscarEstudiantePorMatricula(matricula);

        if (estudiante != null) {
            estudiante.setNombre(nuevoNombre);
            estudiante.setEdad(nuevaEdad);
            System.out.println("Estudiante modificado exitosamente.");
        } else {
            System.out.println("Estudiante no encontrado.");
        }
    }

    // Método para eliminar estudiante
    public void eliminarEstudiante(String matricula) {
        for (int i = 0; i < contadorEstudiantes; i++) {
            if (estudiantes[i].getMatricula().equals(matricula)) {
                // Desplazar los elementos siguientes
                for (int j = i; j < contadorEstudiantes - 1; j++) {
                    estudiantes[j] = estudiantes[j + 1];
                }
                estudiantes[contadorEstudiantes - 1] = null;
            }
        }
    }

```

```

        contadorEstudiantes--;
        System.out.println("Estudiante eliminado exitosamente.");
        return;
    }
}
System.out.println("Estudiante no encontrado.");
}
}

```

Principal.java

Java

```

/*-----
Practica: 6 Gestión y manipulación de arreglos de objetos en Java.
Nombre: Joshua Osorio
Materia: Programación Orientada a Objetos
Fecha: Marzo/26/2025
-----*/

public class Principal {

    public static void main(String[] args) {
        // Crear gestor de estudiantes
        GestorEstudiantes gestor = new GestorEstudiantes();

        // Crear algunos estudiantes
        Estudiante e1 = new Estudiante("Joshua Osorio Osorio", 22, "1293271",
        "Ingeniería en computación");
        Estudiante e2 = new Estudiante("Manuel Alejandro Cruz Lares", 23,
        "1298063", "Ingeniería en computación");
        Estudiante e3 = new Estudiante("Carlos Alberto Serrano Leyva", 21,
        "1285493", "Ingeniería en computación");

        // Agregar estudiantes
        gestor.agregarEstudiante(e1);
        gestor.agregarEstudiante(e2);
        gestor.agregarEstudiante(e3);

        // Mostrar todos los estudiantes
        System.out.println("\n--- Estudiantes Registrados ---");
        gestor.mostrarEstudiantes();
    }
}

```

```

// Buscar estudiante
System.out.println("\n--- Buscando Estudiante ---");
Estudiante encontrado = gestor.buscarEstudiantePorMatricula("1293271");
if (encontrado != null) {
    System.out.println("Estudiante encontrado: " + encontrado);
}

// Modificar estudiante
System.out.println("\n--- Modificando Estudiante ---");
gestor.modificarEstudiante("1298063", "María López", 23);
gestor.mostrarEstudiantes();

// Eliminar estudiante
System.out.println("\n--- Eliminando Estudiante ---");
gestor.eliminarEstudiante("1285493");
gestor.mostrarEstudiantes();
}
}

```

Estructura del programa

El programa se estructuró en una sola clase (Principal), dividiendo el código en métodos para cada funcionalidad específica. Esto no solo facilitó la organización del código, sino que también permitió reutilizar funciones comunes, como la validación de entradas y el cálculo de promedios.

Análisis de Resultados

El programa desarrollado cumplió con los objetivos planteados, permitiendo al usuario interactuar con arreglos de manera eficiente y segura. A través de esta práctica, se reforzaron conceptos clave de la programación orientada a objetos, como el encapsulamiento, la modularización y la validación de datos. Además, se demostró la importancia de seguir buenas prácticas de programación, como el uso de comentarios, la indentación adecuada y la documentación del código.

En conclusión, esta práctica no solo permitió aplicar conocimientos teóricos, sino también desarrollar habilidades prácticas que serán útiles en proyectos futuros. El manejo de arreglos es una competencia fundamental en la programación, y esta experiencia sentó las bases para abordar desafíos más complejos en el futuro.

Conclusiones

Bibliografía

[1] “Java a fondo,” *Google Books*. [Online]. Available:

https://books.google.com.mx/books?hl=es&lr=&id=WcL2DQAAQBAJ&oi=fnd&pg=PA1&dq=%C2%BFQu%C3%A9+es+la+clase+String+en+Java%3F&ots=iRwaAyCxI7&sig=Y9QsnqLumoraAhget_jn3QCjEE7w#v=onepage&q=%C2%BFQu%C3%A9%20es%20la%20clase%20String%20en%20Java%3F&f=false.

[2] I. González Pérez, A. J. Sánchez Martín, D. Vicente Hernández, y

Departamento de Informática y Automática Universidad de Salamanca, «Java Threads (Hilos en Java)», *Google Scholar*, p. 1, may 2002, Disponible en:

<http://lsi.vc.ehu.es/pablogn/docencia/ISO/7%20Gest%20de%20Procesos/HilosJavaUSAL.pdf>

[3] Dotnet-Bot, “String Class (Java.Lang).” [Online]. Available:

<https://learn.microsoft.com/en-us/dotnet/api/java.lang.string?view=net-android-35.0>.