

Etapas de profundización, Actividad práctica aplicada

Laura Alejandra Mendoza Prieto

Análisis y desarrollo de software, Fundación Universitaria UCompensar

Estructura de Datos

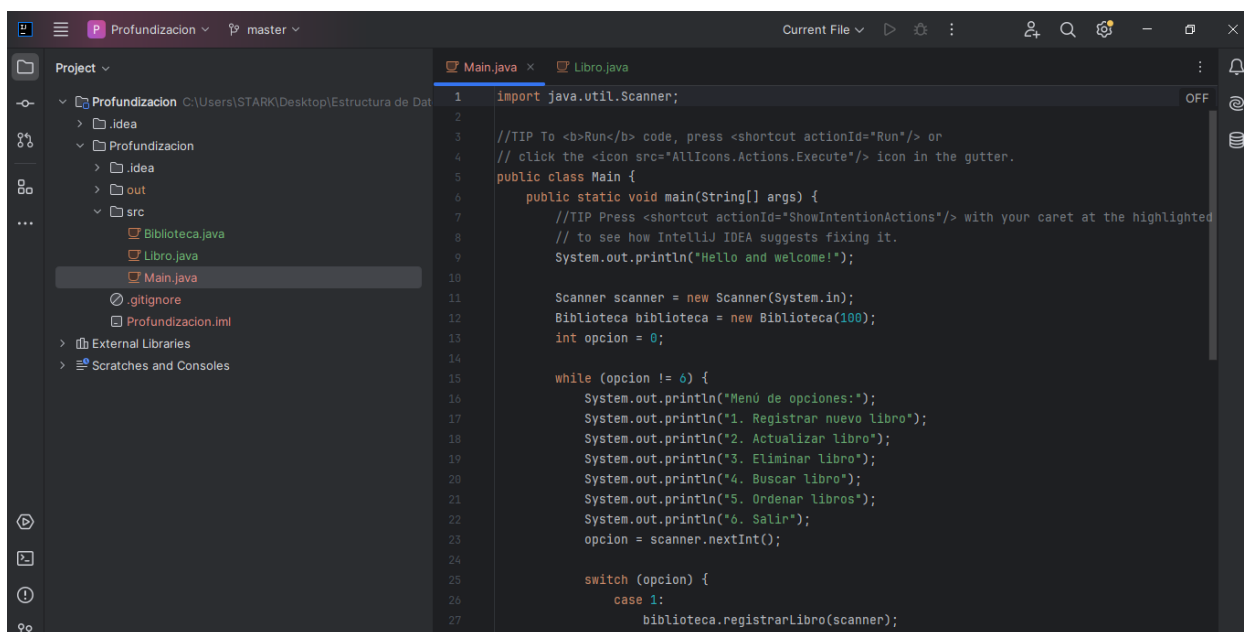
Profesor: JUAN ESTEBAN KATZ DELGADO

Noviembre 13, 2024

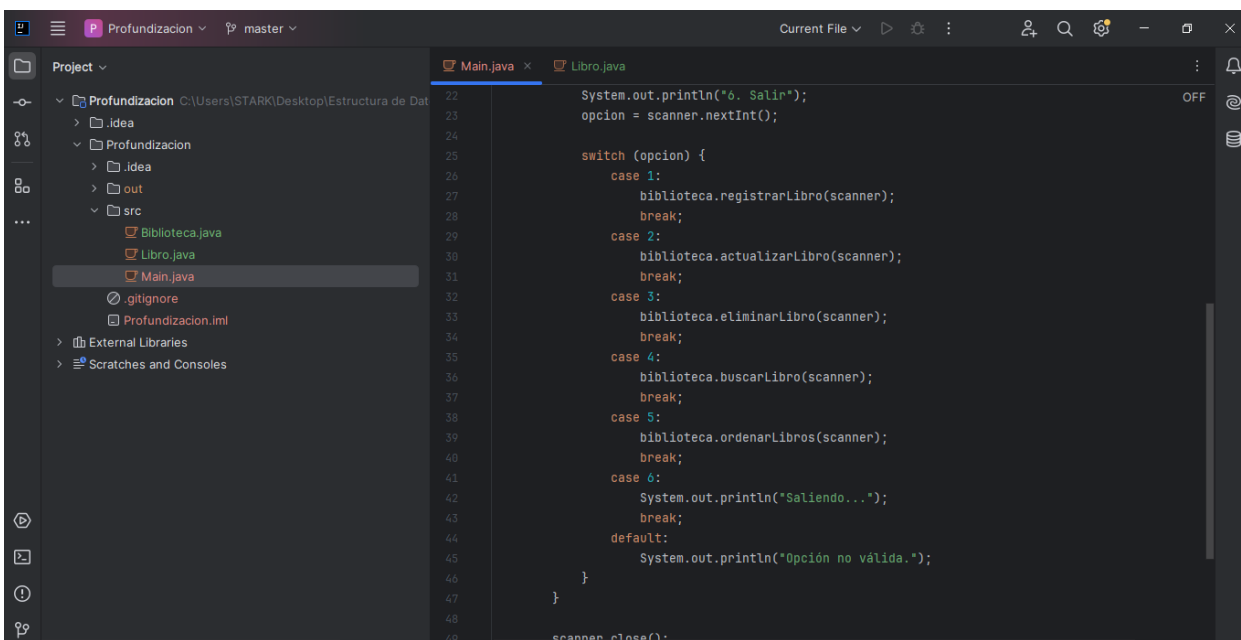
Introducción

El manejo eficiente de datos es fundamental para desarrollar aplicaciones robustas y escalables. El uso de estructura de datos como los arreglos y las listas enlazadas permite organizar y gestionar grandes cantidades de información de manera efectiva. Este trabajo tiene como objetivo implementar un sistema de gestión de biblioteca utilizando arreglos bidimensionales en JAVA, enfocado en realizar varias operaciones clave como el registro, actualización, eliminación, búsqueda y ordenación de libros. Además, se explorara la sobreexcitara de métodos y la aplicación de algoritmos de ordenamiento y búsqueda fundamentales para mejorar la eficiencia del sistema .A lo largo de este documento, se presentara la implementación del código, sus perspectivas funciones y reflexión sobre el uso de estructuras de datos, con el fin de proporcionar una solución práctica y bien estructurada a las necesidades de un sistema de gestión de biblioteca.

Clase main:

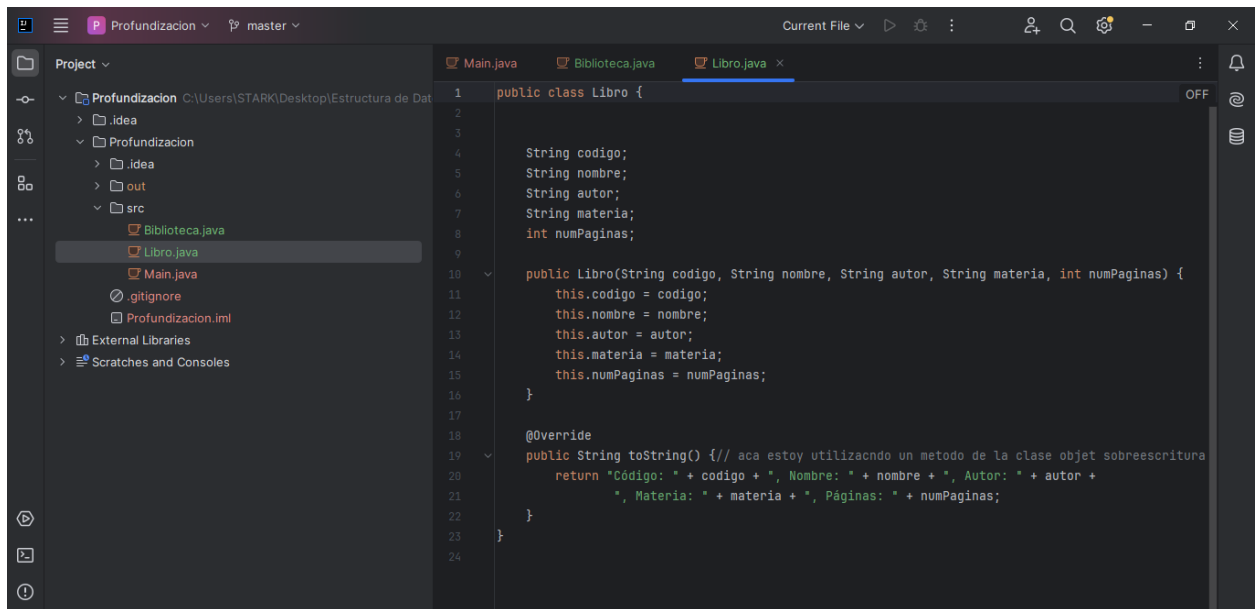


```
1 import java.util.Scanner;
2
3 //TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or
4 // click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.
5 public class Main {
6     public static void main(String[] args) {
7         //TIP Press <shortcut actionId="ShowIntentionActions"/> with your caret at the highlighted
8         // to see how IntelliJ IDEA suggests fixing it.
9         System.out.println("Hello and welcome!");
10
11         Scanner scanner = new Scanner(System.in);
12         Biblioteca biblioteca = new Biblioteca(100);
13         int opcion = 0;
14
15         while (opcion != 0) {
16             System.out.println("Menú de opciones:");
17             System.out.println("1. Registrar nuevo libro");
18             System.out.println("2. Actualizar libro");
19             System.out.println("3. Eliminar libro");
20             System.out.println("4. Buscar libro");
21             System.out.println("5. Ordenar libros");
22             System.out.println("6. Salir");
23             opcion = scanner.nextInt();
24
25             switch (opcion) {
26                 case 1:
27                     biblioteca.registrarLibro(scanner);
```



```
22         System.out.println("6. Salir");
23         opcion = scanner.nextInt();
24
25         switch (opcion) {
26             case 1:
27                 biblioteca.registrarLibro(scanner);
28                 break;
29             case 2:
30                 biblioteca.actualizarLibro(scanner);
31                 break;
32             case 3:
33                 biblioteca.eliminarLibro(scanner);
34                 break;
35             case 4:
36                 biblioteca.buscarLibro(scanner);
37                 break;
38             case 5:
39                 biblioteca.ordenarLibros(scanner);
40                 break;
41             case 6:
42                 System.out.println("Saliendo...");
43                 break;
44             default:
45                 System.out.println("Opción no válida.");
46         }
47     }
48
49     scanner.close();
```

Clase Libro:



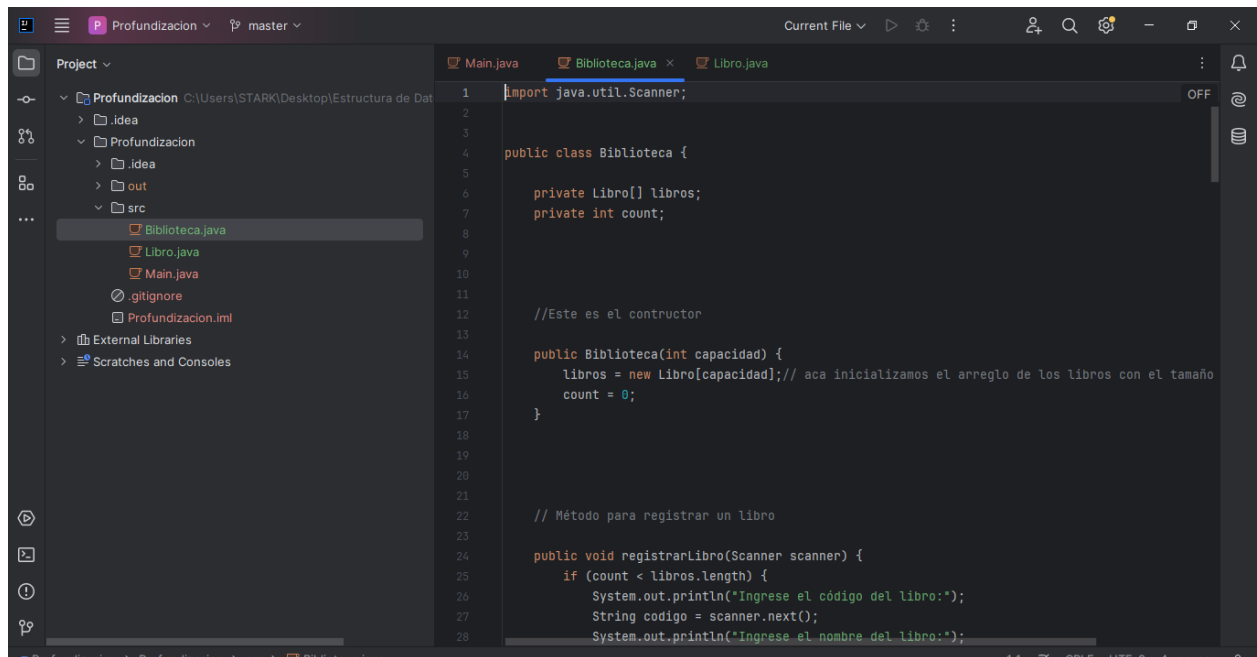
The screenshot shows an IDE with the project 'Profundizacion' open. The file explorer on the left shows the project structure, including 'Biblioteca.java', 'Libro.java', and 'Main.java'. The main editor displays the code for 'Libro.java'.

```

1 public class Libro {
2
3
4     String codigo;
5     String nombre;
6     String autor;
7     String materia;
8     int numPaginas;
9
10    public Libro(String codigo, String nombre, String autor, String materia, int numPaginas) {
11        this.codigo = codigo;
12        this.nombre = nombre;
13        this.autor = autor;
14        this.materia = materia;
15        this.numPaginas = numPaginas;
16    }
17
18    @Override
19    public String toString() { // aca estoy utilizando un metodo de la clase objet sobreescritura
20        return "Código: " + codigo + ", Nombre: " + nombre + ", Autor: " + autor +
21            ", Materia: " + materia + ", Páginas: " + numPaginas;
22    }
23 }
24

```

Clase Biblioteca:



The screenshot shows the same IDE with the project 'Profundizacion' open. The file explorer on the left shows the project structure, including 'Biblioteca.java', 'Libro.java', and 'Main.java'. The main editor displays the code for 'Biblioteca.java'.

```

1 import java.util.Scanner;
2
3
4 public class Biblioteca {
5
6     private Libro[] libros;
7     private int count;
8
9
10
11    //Este es el constructor
12
13    public Biblioteca(int capacidad) {
14        libros = new Libro[capacidad]; // aca inicializamos el arreglo de los libros con el tamaño
15        count = 0;
16    }
17
18
19
20
21
22    // Método para registrar un libro
23
24    public void registrarLibro(Scanner scanner) {
25        if (count < libros.length) {
26            System.out.println("Ingrese el código del libro:");
27            String codigo = scanner.next();
28            System.out.println("Ingrese el nombre del libro:");
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

The screenshot shows an IDE with a project named 'Profundizacion'. The file explorer on the left shows the project structure: `Profundizacion` (root) contains `.idea`, `out`, `src`, `Biblioteca.java`, `Libro.java`, `Main.java`, `.gitignore`, and `Profundizacion.iml`. The main editor displays the `Biblioteca.java` file, specifically the `actualizarLibro` method. The code is as follows:

```

40     } else {
41         System.out.println("No hay espacio para más libros.");
42     }
43 }
44
45 // Método para actualizar un libro
46 public void actualizarLibro(Scanner scanner) {
47     System.out.println("Ingrese el código del libro a actualizar:");
48     String codigo = scanner.next();
49
50     for (int i = 0; i < count; i++) {
51         if (libros[i].codigo.equals(codigo)) {
52             System.out.println("Ingrese el nuevo nombre del libro:");
53             libros[i].nombre = scanner.next();
54             System.out.println("Ingrese el nuevo autor del libro:");
55             libros[i].autor = scanner.next();
56             System.out.println("Ingrese la nueva materia del libro:");
57             libros[i].materia = scanner.next();
58             System.out.println("Ingrese el nuevo número de páginas:");
59             libros[i].numPaginas = scanner.nextInt();
60             System.out.println("Libro actualizado.");
61             return;
62         }
63     }
64     System.out.println("Libro no encontrado.");
65 }
66
67 // Método para eliminar un libro

```

The screenshot shows the same IDE with the `Biblioteca.java` file. The main editor displays the `eliminarLibro` and `buscarLibro` methods. The code is as follows:

```

64     System.out.println("Libro no encontrado.");
65 }
66
67 // Método para eliminar un libro
68 public void eliminarLibro(Scanner scanner) {
69     System.out.println("Ingrese el código del libro a eliminar:");
70     String codigo = scanner.next();
71
72     for (int i = 0; i < count; i++) {
73         if (libros[i].codigo.equals(codigo)) {
74             for (int j = i; j < count - 1; j++) {
75                 libros[j] = libros[j + 1];
76             }
77             libros[count - 1] = null;
78             count--;
79             System.out.println("Libro eliminado.");
80             return;
81         }
82     }
83     System.out.println("Libro no encontrado.");
84 }
85
86 // Método para buscar un libro
87 public void buscarLibro(Scanner scanner) {
88     System.out.println("Ingrese el código del libro a buscar:");
89     String codigo = scanner.next();
90
91     for (int i = 0; i < count; i++) {

```

El desarrollo del sistema de biblioteca utilizando arreglos ha demostrado ser una solución efectiva para organizar y manejar la información de los libros. A través de la implementación de operaciones como el registro, actualización, eliminación, búsqueda y ordenación de los libros, se ha logrado cumplir con los requerimientos establecidos de manera eficiente.

Una de las lecciones más importantes que se ha aprendido a lo largo de este trabajo es la importancia de las estructuras de datos en el diseño de software. Las estructuras de datos son fundamentales porque permiten organizar y almacenar la información de manera que las operaciones sobre ella puedan realizarse de forma eficiente.

Este proyecto ha demostrado que al combinar la programación orientada a objetos con una adecuada selección de estructuras de datos y algoritmos, se puede construir un sistema robusto, flexible y eficiente.