

# Trabajo practico Unidad 6:

## Colecciones:

Santiago Raúl Salinas

## Caso Practico 2: Biblioteca y Libros

### 1. Clase Autor

#### Atributos

- id (string) > Identificador unico del autor.
- nombre (string) > Nombre del autor.
- nacionalidad (string) > Nacionalidad del autor.

```
public class Autor {  
    // ATRIBUTOS (Estado) - Privados por Encapsulamiento  
    private String id;  
    private String nombre;  
    private String nacionalidad;  
  
    // CONSTRUCTOR (Inicializa el estado)  
    public Autor(String id, String nombre, String nacionalidad) {  
        this.id = id;  
        this.nombre = nombre;  
        this.nacionalidad = nacionalidad;  
    }  
}
```

#### Metodos

Public String toString()

```
public String toString() {  
    // Formato para ser usado en el Libro: Nombre (Nacionalidad)  
    return nombre + " (" + nacionalidad + ")";  
}
```

#### Getters – Setters

```

    public String getNombre() {
        return nombre;
    }
}

```

## 2. Clase Libro

### Atributo

isbn (String) >	Identificación único del libro.
titulo (String) >	Título del libro.
anioPublicacion (int) >	Año de publicación.
autor (Autor) >	Autor del libro

```

public class Libro {
    // ATRIBUTOS (Estado)
    private String isbn;
    private String titulo;
    private int anioPublicacion;
    private Autor autor; // Atributo de tipo OBJETO (Asociación 1:1)

    // CONSTRUCTOR
    public Libro(String isbn, String titulo, int anioPublicacion, Autor autor) {
        this.isbn = isbn;
        this.titulo = titulo;
        this.anioPublicacion = anioPublicacion;
        this.autor = autor; // Asigna la referencia del Autor
    }
}

```

### Metodos

mostrarInfo() > Muestra título, ISBN, año y autor.

```

public void mostrarInfo() {
    System.out.printf(" ISBN: %s | Título: %s | Año: %d | Autor: %s\n",
        isbn, titulo, anioPublicacion, autor.toString());
}

```

## 3. Clase Biblioteca

private String nombre;

**private List<Libro> libros** > Colección de libros de la biblioteca.

```

public class Biblioteca {
    // ATRIBUTOS (Composición 1:N)
    private String nombre;
    private List<Libro> libros; // Usamos List para mejor práctica, implementado con ArrayList

    // CONSTRUCTOR
    public Biblioteca(String nombre) {
        this.nombre = nombre;
        // Inicialización obligatoria del ArrayList
        this.libros = new ArrayList<>();
    }
}

```

## Metodos

**agregarLibro(String isbn, String titulo, int anioPublicacion, Autor autor) >**  
 implementar la composición.

```

// MÉTODO: agregarLibro() (Implementación de la COMPOSICIÓN 1:N)
public void agregarLibro(String isbn, String titulo, int anioPublicacion, Autor autor) {
    // La Biblioteca es responsable de crear la instancia del Libro (Composición)
    Libro nuevoLibro = new Libro(isbn, titulo, anioPublicacion, autor);
    this.libros.add(nuevoLibro);
    System.out.printf(" Agregado: %s (%s)\n", titulo, isbn);
}

```

## listaLibros()

```

// MÉTODO: listarLibros()
public void listarLibros() {
    System.out.println("\n--- LISTADO DE LIBROS EN: " + this.nombre + " ---");
    if (libros.isEmpty()) {
        System.out.println("La biblioteca no tiene libros.");
        return;
    }
    for (Libro libro : libros) { // Uso del ciclo for-each
        libro.mostrarInfo();
    }
}

```

## buscarLibroPorIsbn(String isbn)

```

// MÉTODO: buscarLibroPorIsbn(String isbn)
public Libro buscarLibroPorIsbn(String isbn) {
    for (Libro libro : libros) {
        if (libro.getIsbn().equals(isbn)) {
            return libro;
        }
    }
    return null;
}

```

eliminarLibro(String isbn)

```
// MÉTODO: eliminarLibro(String isbn)
public boolean eliminarLibro(String isbn) {
    // Se usa el ciclo for tradicional para poder obtener el índice
    for (int i = 0; i < libros.size(); i++) {
        Libro libroActual = libros.get(i);

        if (libroActual.getIsbn().equals(isbn)) {
            libros.remove(i); // Elimina el elemento por índice
            System.out.printf(" Eliminado: Libro con ISBN %s\n", isbn);
            return true;
        }
    }
    System.out.printf(" Error: Libro con ISBN %s no encontrado.\n", isbn);
    return false;
}
```

obtenerCantidadLibros()

```
// MÉTODO: obtenerCantidadLibros()
public int obtenerCantidadLibros() {
    return libros.size();
}
```

filtrarLibrosPorAnio(int anio)

```
// MÉTODO: filtrarLibrosPorAnio(int anio)
public void filtrarLibrosPorAnio(int anio) {
    System.out.printf("\n--- LIBROS PUBLICADOS EN %d ---\n", anio);
    boolean encontrado = false;
    for (Libro libro : libros) {
        if (libro.getAnioPublicacion() == anio) {
            libro.mostrarInfo();
            encontrado = true;
        }
    }
    if (!encontrado) {
        System.out.println("No se encontraron libros publicados en ese año.");
    }
}
```

mostrarAutoresDisponibles()

```

// MÉTODO: mostrarAutoresDisponibles()
public void mostrarAutoresDisponibles() {
    System.out.println("\n--- AUTORES DISPONIBLES EN INVENTARIO ---");
    List<String> autoresYaListados = new ArrayList<>();
    for (Libro libro : libros) {
        String nombreAutor = libro.getAutor().getNombre();
        if (!autoresYaListados.contains(nombreAutor)) {
            System.out.println("• " + nombreAutor);
            autoresYaListados.add(nombreAutor);
        }
    }
}
}

```

## Clase Main

### 1 - Creamos la biblioteca:

```

// Tarea 1: Crear una biblioteca.
Biblioteca bibliotecaCentral = new Biblioteca("Biblioteca Central UTN");

```

### 2 - Creamos al menos 3 autores:

```

// Tarea 2: Crear al menos tres autores
Autor a1 = new Autor("A001", "Gabriel García Márquez", "Colombiana");
Autor a2 = new Autor("A002", "Jorge Luis Borges", "Argentina");
Autor a3 = new Autor("A003", "Isabel Allende", "Chilena");

```

### 3 - Agregar 5 libros asociados a alguno de los Autores de la biblioteca

```

// Tarea 3: Agregar 5 libros asociados a alguno de los Autores a la biblioteca.
// NOTA: La Biblioteca crea los objetos Libro internamente (Composición).
bibliotecaCentral.agregarLibro("978-0307474728", "Cien Años de Soledad", 1967, a1);
bibliotecaCentral.agregarLibro("978-0143105741", "El Amor en los Tiempos del Cólera", 1985, a1);
bibliotecaCentral.agregarLibro("978-0141182745", "Ficciones", 1944, a2);
bibliotecaCentral.agregarLibro("978-0141182738", "El Aleph", 1949, a2);
bibliotecaCentral.agregarLibro("978-0345803276", "La Casa de los Espíritus", 1982, a3);

```

### 4 - Listar todos los libros con su información y la del autor.

```

// Tarea 4: Listar todos los libros mostrando su información y la del autor.
bibliotecaCentral.listarLibros();

System.out.println("\n=== 2. BÚSQUEDA Y FILTRADO ===");

```

### 5 - Buscar un libro por su ISBN y mostrar su información.

```
// Tarea 5: Buscar un libro por su ISBN y mostrar su información.
String isbnBuscado = "978-0143105741";
Libro libroBuscado = bibliotecaCentral.buscarLibroPorIsbn(isbnBuscado);
if (libroBuscado != null) {
    System.out.println("✓ Libro encontrado por ISBN " + isbnBuscado + ":");
    libroBuscado.mostrarInfo();
} else {
    System.out.println("✗ Libro con ISBN " + isbnBuscado + " no encontrado.");
}
```

6 - Filtrar y mostrar los libros publicados en un año específico.

```
// Tarea 6: Filtrar y mostrar los libros publicados en un año específico.
bibliotecaCentral.filtrarLibrosPorAño(1944);
```

7 - Eliminar un libro por su ISBN y listar los libros restantes

```
// Tarea 7: Eliminar un libro por su ISBN.
String isbnEliminar = "978-0143105741"; // El Amor en los Tiempos del Cólera
System.out.println(">>> Intentando eliminar ISBN: " + isbnEliminar + " <<<");
bibliotecaCentral.eliminarLibro(isbnEliminar);
```

8 - Mostrar la cantidad total de libros en la biblioteca.

```
// Tarea 8: Mostrar la cantidad total de libros en la biblioteca.
System.out.println("\nCantidad total de libros restantes: " + bibliotecaCentral.obtenerCantidadLibros());

System.out.println("\n=== FIN DE LAS TAREAS ===");
}
```

9 - Listar todos los autores de los libros disponibles en la biblioteca

```
// Tarea 9: Listar todos los autores de los libros disponibles en la biblioteca.
bibliotecaCentral.mostrarAutoresDisponibles();
```

Resultado Consola:

```
Output - TP UNIDAD 6 (run) x
run:
=== 1. AGREGANDO LIBROS (COMPOSICIÓN 1:N) ===
Agregado: Cien Años de Soledad (978-0307474728)
Agregado: El Amor en los Tiempos del Cólera (978-0143105741)
Agregado: Ficciones (978-0141182745)
Agregado: El Aleph (978-0141182738)
Agregado: La Casa de los Espíritus (978-0345803276)

--- LISTADO DE LIBROS EN: Biblioteca Central UTN ---
ISBN: 978-0307474728 | Título: Cien Años de Soledad | Año: 1967 | Autor: Gabriel García Márquez (Colombiana)
ISBN: 978-0143105741 | Título: El Amor en los Tiempos del Cólera | Año: 1985 | Autor: Gabriel García Márquez (Colombiana)
ISBN: 978-0141182745 | Título: Ficciones | Año: 1944 | Autor: Jorge Luis Borges (Argentina)
ISBN: 978-0141182738 | Título: El Aleph | Año: 1949 | Autor: Jorge Luis Borges (Argentina)
ISBN: 978-0345803276 | Título: La Casa de los Espíritus | Año: 1982 | Autor: Isabel Allende (Chilena)

=== 2. BÚSQUEDA Y FILTRADO ===
? Libro encontrado por ISBN 978-0143105741:
  ISBN: 978-0143105741 | Título: El Amor en los Tiempos del Cólera | Año: 1985 | Autor: Gabriel García Márquez (Colombiana)

--- LIBROS PUBLICADOS EN 1944 ---
  ISBN: 978-0141182745 | Título: Ficciones | Año: 1944 | Autor: Jorge Luis Borges (Argentina)

--- AUTORES DISPONIBLES EN INVENTARIO ---
  ♦ Gabriel García Márquez
  ♦ Jorge Luis Borges
  ♦ Isabel Allende

=== 3. OPERACIONES DE BAJA Y REPORTE ===
>>> Intentando eliminar ISBN: 978-0143105741 <<<
  Eliminado: Libro con ISBN 978-0143105741

--- LISTADO DE LIBROS EN: Biblioteca Central UTN ---
  ISBN: 978-0307474728 | Título: Cien Años de Soledad | Año: 1967 | Autor: Gabriel García Márquez (Colombiana)
  ISBN: 978-0141182745 | Título: Ficciones | Año: 1944 | Autor: Jorge Luis Borges (Argentina)
  ISBN: 978-0141182738 | Título: El Aleph | Año: 1949 | Autor: Jorge Luis Borges (Argentina)
  ISBN: 978-0345803276 | Título: La Casa de los Espíritus | Año: 1982 | Autor: Isabel Allende (Chilena)

Cantidad total de libros restantes: 4

=== FIN DE LAS TAREAS ===
BUILD SUCCESSFUL (total time: 0 seconds)
```