

# MANUAL TÉCNICO

# Clase Libro

Se construyeron setters y getters para sus variables.

```
package Libros;
public class Libro {
    private String isbn;
    private String autor;
    private int año;
    private String titulo;
    private String edicion;
    private String tema;
    private String desc;
    private int numCop;
    private int numCopDisp;
    private int numEst;
    private int fila;
    public Libro(String isbn,String autor,int año,String titulo,
        String edicion,String tema,String desc,
        int numCop,int numCopDisp,int numEst,int fila) {
        this.setIsbn(isbn);
        this.setAutor(autor);
        this.setAño(año);
        this.setTitulo(titulo);
        this.setEdicion(edicion);
        this.setTema(tema);
        this.setDesc(desc);
        this.setNumCop(numCop);
        this.setNumCopDisp(numCopDisp);
        this.setFila(fila);
    }
}
```

# Clase CtrlLibros

## Función “verificarIsbn”

Se encarga de verificar que el **ISBN** del libro es repetido, si lo es devuelve valor VERDADERO de lo contrario devuelve valor FALSO.

```
public static boolean verificarIsbn(String isbn) {
    for(Libro libro : Libros) {
        if(libro != null && libro.getIsbn().equals(isbn)) {
            return true;
        }
    }
    return false;
}
```

## Función “nuevoLibro”

Se encarga de agregar un nuevo libro cuando la posición del arreglo tenga un valor NULL y devolverá un valor VERDADERO, de lo contrario devuelve valor FALSO.

```
public static boolean nuevoLibro(String isbn,String autor,int año,String titulo,
    String edicion,String tema,String desc,
    int numCop,int numCopDisp,int numEst,int fila) {
    for(int i = 0; i < Libros.length; i++) {
        if(Libros[i] == null) {
            Libros[i] = new
Libro(isbn,autor,año,titulo,edicion,tema,desc,numCop,numCopDisp,numEst,fila);
            return true;
        }
    }
    return false;
}
```

## Función “bSLibro”

Se encarga de buscar un libro dentro del arreglo devolviendo un arreglo de tipo libro siempre y cuando la posición no esté vacía y que coincida el título.

```
public static Libro[] bSLibro(String titulo) {
    Libro [] bslibro = new Libro[1];
    for(Libro libro : Libros) {
        if(libro != null && libro.getTitulo().equalsIgnoreCase(titulo)) {
            bslibro[0] = libro;
            return bslibro;
        }
    }
    return null;
}
```

## Función “cantLibro”

Se encarga de contar la cantidad de libros que se tienen en un intervalo de años.

```
static int cantLibro(int añoI, int añoF) {
    int n = 0;
    for(Libro libro : Libros) {
        if(libro != null && libro.getAño() >= añoI && libro.getAño() <= añoF) {
            n ++;
        }
    }
    return n;
}
```

## Función “bLibro”

Se encarga de buscar y ordenar los libros de un intervalo de años retornando un arreglo de tipo libro.

```
public static Libro[] bLibro(int añoI, int añoF) {
    int n = cantLibro(añoI,añoF);
    if(n == 0) return null;
    Libro[] aLibro = new Libro[n];
    for(Libro libro : Libros) {
        if(libro != null && libro.getAño() >= añoI && libro.getAño() <= añoF) {
            for(int i = 0; i < aLibro.length; i++) {
                if(aLibro[i] == null) {
                    aLibro[i] = libro;
                    break;
                }
            }
        }
    }
    for(int i = 0; i < aLibro.length - 1; i++) {
        for(int j = 0; j < aLibro.length - i - 1; j++) {
            if(aLibro[j] != null && aLibro[j + 1] != null && aLibro[j].getAño()
< aLibro[j + 1].getAño()) {
                Libro tmp = aLibro[j];
                aLibro[j] = aLibro[j + 1];
                aLibro[j + 1] = tmp;
            }
        }
    }
    return aLibro;
}
```

## Función “cantLibro”

Se encarga de contar la cantidad de libros que tiene un autor.

```
static int cantLibro(String autor) {
    int n = 0;
    for(Libro libro : libros) {
        if(libro != null && libro.getAutor().equalsIgnoreCase(autor)) {
            n ++;
        }
    }
    return n;
}
```

## Función “bLibro”

Se encarga de buscar y ordenar los libros de un autor retornando un arreglo de tipo libro.

```
public static Libro[] bLibro(String autor) {
    int n = cantLibro(autor);
    if(n == 0) return null;
    Libro[] aLibro = new Libro[n];
    for(Libro libro : libros) {
        if(libro != null && libro.getAutor().equalsIgnoreCase(autor)) {
            for(int i = 0; i < aLibro.length; i++) {
                if(aLibro[i] == null) {
                    aLibro[i] = libro;
                    break;
                }
            }
        }
    }
    for(int i = 0; i < aLibro.length - 1; i++) {
        for(int j = 0; j < aLibro.length - i - 1; j++) {
            if(aLibro[j] != null && aLibro[j + 1] != null
                && aLibro[j].getAño() < aLibro[j + 1].getAño()) {
                Libro tmp = aLibro[j];
                aLibro[j] = aLibro[j + 1];
                aLibro[j + 1] = tmp;
            }
        }
    }
    return aLibro;
}
```

## Función “cantLibro”

Se encarga de contar la cantidad de libros que tiene un autor en un intervalo de años.

```
static int cantLibro(String autor, int añoI, int añoF) {
    int n = 0;
    for(Libro libro : Libros) {
        if(libro != null && libro.getAutor().equalsIgnoreCase(autor)
            && libro.getAño() >= añoI && libro.getAño() <= añoF) {
            n ++;
        }
    }
    return n;
}
```

## Función “bLibro”

Se encarga de buscar y ordenar los libros de un autor en un intervalo de años retornando un arreglo de tipo libro.

```
public static Libro[] bLibro(String autor, int añoI, int añoF) {
    int n = cantLibro(autor,añoI,añoF);
    if(n == 0) return null;
    Libro[] aLibro = new Libro[n];
    for(Libro libro : Libros) {
        if(libro != null && libro.getAutor().equalsIgnoreCase(autor)
            && libro.getAño() >= añoI && libro.getAño() <= añoF) {
            for(int i = 0; i < aLibro.length; i++) {
                if(aLibro[i] == null) {
                    aLibro[i] = libro;
                    break;
                }
            }
        }
    }
    for(int i = 0; i < aLibro.length - 1; i++) {
        for(int j = 0; j < aLibro.length - i - 1; j++) {
            if(aLibro[j] != null && aLibro[j + 1] != null
                && aLibro[j].getAño() < aLibro[j + 1].getAño()) {
                Libro tmp = aLibro[j];
                aLibro[j] = aLibro[j + 1];
                aLibro[j + 1] = tmp;
            }
        }
    }
    return aLibro;
}
```

## Función “bIsbnLibro”

Se encarga de buscar un libro mediante el ISBN, retornando un objeto de tipo Libro.

```
public static Libro bIsbnLibro(String isbn) {
    for(Libro libro : Libros) {
        if(libro != null && libro.getIsbn().equals(isbn)) {
            return libro;
        }
    }
    return null;
}
```

## Función “modCantLibro”

Se encarga de modificar la cantidad de libros.

```
public static boolean modCantLibro(String isbn, int cant) {
    for(int i = 0; i < CtrlLibros.Libros.length; i++) {
        if(CtrlLibros.Libros[i] != null && CtrlLibros.Libros[i].getIsbn().equals(isbn)) {
            Libro tmp = CtrlLibros.Libros[i];
            CtrlLibros.Libros[i] = new
Libro(isbn,tmp.getAutor(),tmp.getAño(),tmp.getTitulo(),tmp.getEdicion(),tmp.getTema(),tmp.getDesc
()),
                                tmp.getNumCop() + cant,tmp.getNumCopDisp() +
cant,tmp.getNumEst(),tmp.getFila());
            return true;
        }
    }
    return false;
}
```

# Clase Prestamista

Se construyeron setters y getters para sus variables.

```
package Prestamistas;
public class Prestamista {
    private String nombre;
    private String apellido;
    private String cui;

    Prestamista(String nombre, String apellido, String cui){
        this.setNombre(nombre);
        this.setApellido(apellido);
        this.setCui(cui);
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getApellido() {
        return apellido;
    }
    public void setApellido(String apellido) {
        this.apellido = apellido;
    }
    public String getCui() {
        return cui;
    }
    public void setCui(String cui) {
        this.cui = cui;
    }
}
```



# Clase CtrlPrestamistas

## Función “verificarPres”

Se encarga de verificar que el CUI del prestamista es repetido, si lo es devuelve valor VERDADERO de lo contrario devuelve valor FALSO.

```
public static boolean verificarPres(String cui) {
    for(Prestamista prestamista : prestamistas) {
        if(prestamista != null && prestamista.getCui().equals(cui)) {
            return true;
        }
    }
    return false;
}
```

## Función “nuevoPres”

Se encarga de agregar un nuevo prestamista cuando la posición del arreglo tenga un valor NULL y devolverá un valor VERDADERO, de lo contrario devuelve valor FALSO.

```
public static boolean nuevoPres(String nombre, String apellido, String cui) {
    for(int i = 0; i < prestamistas.length; i++) {
        if(prestamistas[i] == null) {
            prestamistas[i] = new Prestamista(nombre,apellido,cui);
            return true;
        }
    }
    return false;
}
```

## Función “busP”

Se encarga de buscar un prestamista mediante el CUI devolviendo un objeto de tipo prestamista.

```
public static Prestamista busP(String cui) {
    for(Prestamista prestamista : prestamistas) {
        if(prestamista != null && prestamista.getCui().equals(cui)) {
            return prestamista;
        }
    }
    return null;
}
```

# Clase Prestamo

Se construyeron setters y getters para sus variables.

```
package Prestamos;
public class Prestamo {
    private String isbn;
    private String cui;
    private String fechaP;
    private String fechaD;
    private boolean estado;

    public Prestamo(String isbn, String cui,String fechaP, String fechaD, boolean
estado){
        this.setIsbn(isbn);
        this.setCui(cui);
        this.setFechaP(fechaP);
        this.setFechaD(fechaD);
        this.setEstado(estado);
    }
    public String getIsbn() {
        return isbn;
    }
    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }
    public String getCui() {
        return cui;
    }
    public void setCui(String cui) {
        this.cui = cui;
    }
    public String getFechaP() {
        return fechaP;
    }
    public void setFechaP(String fechaP) {
        this.fechaP = fechaP;
    }
    public String getFechaD() {
        return fechaD;
    }
    public void setFechaD(String fechaD) {
        this.fechaD = fechaD;
    }
    public boolean getEstado() {
        return estado;
    }
    public void setEstado(boolean estado) {
        this.estado = estado;
    }
}
```

# Clase CtrlPrestamos

## Función “dispLibro”

Se encarga de verificar que la cantidad de libros disponibles sea mayor a 0.

```
public static boolean dispLibro(String isbn) {
    for(Libro libro : CtrlLibros.Libros) {
        if(libro != null && libro.getIsbn().equals(isbn) &&
            libro.getNumCopDisp() > 0) {
            return true;
        }
    }
    return false;
}
```

## Función “pendLibro”

Se encarga de verificar que el prestamista no tenga un libro pendiente de devolución.

```
public static Prestamo pendLibro(String cui) {
    for(Prestamo prestamo : prestamos) {
        if(prestamo != null && prestamo.getCui().equals(cui) &&
            !prestamo.getEstado()) {
            return prestamo;
        }
    }
    return null;
}
```

## Función “modCantLibro”

Se encarga de sumar o restar las devoluciones o prestamos a la cantidad disponible.

```
public static boolean modCantLibro(String isbn, int cant) {
    for(int i = 0; i < CtrlLibros.Libros.length; i++) {
        if(CtrlLibros.Libros[i] != null &&
            CtrlLibros.Libros[i].getIsbn().equals(isbn)) {
            Libro tmp = CtrlLibros.Libros[i];
            CtrlLibros.Libros[i] = new
                Libro(isbn,tmp.getAutor(),tmp.getAño(),tmp.getTitulo(),tmp.getEd
                    icion(),tmp.getTema(),tmp.getDesc(),
                    tmp.getNumCop(),tmp.getNumCopDisp() +
                    cant,tmp.getNumEst(),tmp.getFila());
            return true;
        }
    }
    return false;
}
```

## Función “nuevoPrestamo”

Se encarga de registrar un nuevo préstamo y reduce en 1 la cantidad de libros disponibles.

```
public static boolean nuevoPrestamo(String isbn, String cui, String fechaP, boolean estado) {
    for(int i = 0; i < prestamos.length; i++) {
        if(prestamos[i] == null) {
            prestamos[i] = new Prestamo(isbn,cui,fechaP,"",estado);
            return modCantLibro(isbn,-1);
        }
    }
    return false;
}
```

## Función “devLibro”

Se encarga de registrar una devolución y aumenta en 1 la cantidad de libros disponibles.

```
public static boolean devLibro(String isbn, String cui, String fechaD) {
    for(int i = 0; i < prestamos.length; i++) {
        if(prestamos[i] != null && prestamos[i].getIsbn().equals(isbn) &&
            prestamos[i].getCui().equals(cui) && !prestamos[i].getEstado()) {
            Prestamo tmp = prestamos[i];
            prestamos[i] = new Prestamo(isbn,cui,tmp.getFechaP(),fechaD,true);
            return modCantLibro(isbn,1);
        }
    }
    return false;
}
```

## Función “cantPrestamo”

Se encarga de contar la cantidad de préstamos realizados por un usuario.

```
static int cantPrestamo(String cui) {
    int n = 0;
    for(Prestamo prestamo : prestamos) {
        if(prestamo != null && prestamo.getCui().equals(cui)) {
            n++;
        }
    }
    return n;
}
```

## Función “historial”

Se encarga de buscar los préstamos realizados de un usuario específico retornando un arreglo de tipo Prestamo.

```
public static Prestamo[] historial(String cui) {
    int n = cantPrestamo(cui);
    if(n == 0) return null;
    Prestamo[] historial = new Prestamo[n];
    for(Prestamo prestamo : prestamos) {
        if(prestamo != null && prestamo.getCui().equals(cui)) {
            for(int i = 0; i < historial.length; i++) {
                if(historial[i] == null) {
                    historial[i] = prestamo;
                    break;
                }
            }
        }
    }
    return historial;
}
```