

PROGRAMACIÓN

LibrosLibres APP

ÍNDICE

ÍNDICE.....	1
EXPLICACIÓN.....	2
INTRODUCCIÓN.....	2
ENTRANDO EN MATERIA.....	2
CÓDIGO.....	3
CLASE LIBRO.....	3
CLASE USUARIO.....	10
CLASE CONNECT.....	15
CLASE AUTORES.....	16
MAIN (TESTEO):.....	22
TESTEO.....	26
DESCARGA.....	26
MANUAL DE USO PARA TRABAJADORES.....	27
Introducción:.....	27
Menú principal:.....	27
Sección de Usuarios:.....	27
Sección de Libros:.....	27
Sección de Autores:.....	28
Conclusión:.....	28
TABLAS NECESARIAS DE CONOCER PARA TRABAJAR CON LA APP:.....	29

EXPLICACIÓN

INTRODUCCIÓN

LibrosLibres se está empezando a extender y a ganar popularidad, gracias a ello tenemos nuevos empleados, hemos construido una aplicación que facilita en gran cantidad la subida, modificación y eliminación de usuarios y libros de nuestra Base de Datos.

ENTRANDO EN MATERIA

a. Preparación

Hemos necesitado un entorno que nos facilita la edición de nuestros archivos, nuestra aplicación está construida en java y hemos dividido el proyecto en dos partes:

1. Implementación de clases (VSCODE)
 - a. Hemos implementado las clases en una interfaz de consola para mejorar la efectividad en la creación de ellas.
2. Testeo
 - a. Testeamos las clases y métodos que hemos comprado.

b. Clases

Hemos implementado cuatro clases:

- Usuarios (Contiene todas las funciones relacionadas con la gestión de usuarios)
- Libros (Contiene todas las funciones relacionadas con la gestión de libros)
- Autores (Contiene todas las funciones relacionadas con la gestión de autores)
- Connect (Contiene las funciones de conexión con la Base de datos, a parte de la implementación de los drivers necesarios)

También hemos creado una nueva carpeta llamada lib en la que hemos implementado el driver para la conexión con MySQL..



CÓDIGO

CLASE LIBRO

La clase se llama "Libro" y se encarga de modelar un objeto libro. Contiene varios atributos que representan las propiedades de un libro, como ISBN, nombre, sinopsis, fecha de publicación, idioma, número de páginas, código de autor, código de género, código de editorial y un enlace al archivo PDF del libro. Además, tiene métodos para crear, eliminar y modificar un libro en una base de datos.

El constructor de la clase toma varios parámetros que representan los atributos del libro y los asigna a las variables correspondientes. También crea una conexión a la base de datos a través de un objeto de la clase "connect".

La clase tiene varios métodos "getters" y "setters" para obtener y establecer los valores de los atributos del libro.

El método "crearLibro" se encarga de insertar un nuevo libro en la base de datos. Toma varios parámetros que representan los atributos del libro y ejecuta una consulta SQL para insertarlos en la tabla "libros".

El método "eliminarLibro" elimina un libro de la base de datos. Toma un parámetro que representa el ISBN del libro y ejecuta una consulta SQL para eliminarlo de la tabla "libros".

El método "modificarlibro" actualiza los valores de un libro existente en la base de datos. Toma varios parámetros que representan los nuevos valores de los atributos del libro y ejecuta una consulta SQL para actualizarlos en la tabla "libros".

La clase "Libro" utiliza un objeto de la clase "connect" para conectarse a la base de datos y ejecutar las consultas SQL necesarias.

```
import java.sql.*;

public class Libro {
    connect con;
    private int ISBN;
    private String nombre;
    private String sinopsis;
    private Date fechaPublicacion;
    private String idioma;
    private String numeroPag;
    private int codAutor;
    private int codGenero;
    private int codEditorial;
    private String linkPdf;

    // Constructor
    public Libro(int ISBN, String nombre, String sinopsis, Date
fechaPublicacion,
                String idioma, String numeroPag, int codAutor, int
codGenero,
                int codEditorial, String linkPdf) {
        con = new connect();
        this.ISBN = ISBN;
        this.nombre = nombre;
        this.sinopsis = sinopsis;
        this.fechaPublicacion = fechaPublicacion;
        this.idioma = idioma;
        this.numeroPag = numeroPag;
        this.codAutor = codAutor;
        this.codGenero = codGenero;
        this.codEditorial = codEditorial;
        this.linkPdf = linkPdf;
    }

    // Getters y Setters
    public int getISBN() {
        return ISBN;
    }

    public void setISBN(int ISBN) {
```

```
        this.ISBN = ISBN;
    }

    public String getNombre() {
        return nombre;
    }

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

    public String getSinopsis() {
        return sinopsis;
    }

    public void setSinopsis(String sinopsis) {
        this.sinopsis = sinopsis;
    }

    public Date getFechaPublicacion() {
        return fechaPublicacion;
    }

    public void setFechaPublicacion(Date fechaPublicacion) {
        this.fechaPublicacion = fechaPublicacion;
    }

    public String getIdioma() {
        return idioma;
    }

    public void setIdioma(String idioma) {
        this.idioma = idioma;
    }

    public String getNumeroPag() {
        return numeroPag;
    }

    public void setNumeroPag(String numeroPag) {
        this.numeroPag = numeroPag;
    }
}
```

```

    }

    public int getCodAutor() {
        return codAutor;
    }

    public void setCodAutor(int codAutor) {
        this.codAutor = codAutor;
    }

    public int getCodGenero() {
        return codGenero;
    }

    public void setCodGenero(int codGenero) {
        this.codGenero = codGenero;
    }

    public int getCodEditorial() {
        return codEditorial;
    }

    public void setCodEditorial(int codEditorial) {
        this.codEditorial = codEditorial;
    }

    public String getLinkPdf() {
        return linkPdf;
    }

    public void setLinkPdf(String linkPdf) {
        this.linkPdf = linkPdf;
    }

    public void crearLibro(String isbn, String nombre, String sinopsis,
String fecha_publicacion, String idioma, int numero_pag, int cod_autor, int
cod_genero, int cod_editorial, String linkpdf) {
        con.Conectar();
        try {
            PreparedStatement consulta =
con.conexion.prepareStatement("INSERT INTO libros(ISBN, nombre, sinopsis,
Fecha_publicacion, idioma, numero_pag, cod_autor, cod_genero,

```

```

cod_editorial, linkpdf) VALUES(?,?,?,?,?,?,?,?,?,?,?));
        consulta.setString(1, isbn);
        consulta.setString(2, nombre);
        consulta.setString(3, sinopsis);
        consulta.setString(4, fecha_publicacion);
        consulta.setString(5, idioma);
        consulta.setInt(6, numero_pag);
        consulta.setInt(7, cod_autor);
        consulta.setInt(8, cod_genero);
        consulta.setInt(9, cod_editorial);
        consulta.setString(10, linkpdf);
        consulta.executeUpdate();
        System.out.println("Libro creado exitosamente");
        consulta.close();
    } catch (SQLException e) {
        System.out.println("Error al crear el libro: " +
e.getMessage());
    }
    con.CerrarConexion();
}

public void eliminarLibro(String isbn) {
    con.Conectar();
    try {
        PreparedStatement consulta =
con.conexion.prepareStatement("DELETE FROM libros WHERE ISBN = ?");
        consulta.setString(1, isbn);
        consulta.executeUpdate();
        System.out.println("Libro eliminado exitosamente");
        consulta.close();
    } catch (SQLException e) {
        System.out.println("Error al eliminar el libro: " +
e.getMessage());
    }
    con.CerrarConexion();
}

public void modificarlibro(String isbn, String nombre, String sinopsis,
String fecha_publicacion, String idioma, int numero_pag, int cod_autor, int
cod_genero, int cod_editorial, String linkpdf) {
    con.Conectar();

```

```

        try {
            PreparedStatement ps = con.conexion.prepareStatement("UPDATE
libros SET nombre=?, sinopsis=?, Fecha_publicacion=?, idioma=?,
numero_pag=?, cod_autor=?, cod_genero=?, cod_editorial=?, linkpdf=? WHERE
ISBN=?");

            ps.setString(1, nombre);
            ps.setString(2, sinopsis);
            ps.setString(3, fecha_publicacion);
            ps.setString(4, idioma);
            ps.setInt(5, numero_pag);
            ps.setInt(6, cod_autor);
            ps.setInt(7, cod_genero);
            ps.setInt(8, cod_editorial);
            ps.setString(9, linkpdf);
            ps.setString(10, isbn);
            ps.executeUpdate();
            System.out.println("Libro modificado exitosamente");
        } catch (SQLException ex) {
            System.out.println("Error al modificar libro: " +
ex.getMessage());
        }
        con.CerrarConexion();
    }

    public void visualizarLibros() {
        con.Conectar();
        try {
            Statement consulta = con.conexion.createStatement();
            ResultSet resultado = consulta.executeQuery("SELECT * FROM
libros");

            while (resultado.next()) {
                System.out.println("ISBN: " + resultado.getString("ISBN"));
                System.out.println("Nombre: " +
resultado.getString("nombre"));
                System.out.println("Sinopsis: " +
resultado.getString("sinopsis"));
                System.out.println("Fecha de publicación: " +
resultado.getString("Fecha_publicacion"));
                System.out.println("Idioma: " +
resultado.getString("idioma"));
                System.out.println("Número de páginas: " +
resultado.getInt("numero_pag"));
            }
        }
    }
}

```



```

        System.out.println("Código de autor: " +
resultado.getInt("cod_autor"));
        System.out.println("Código de género: " +
resultado.getInt("cod_genero"));
        System.out.println("Código de editorial: " +
resultado.getInt("cod_editorial"));
        System.out.println("Link PDF: " +
resultado.getString("linkpdf"));
        System.out.println("-----");
    }
    resultado.close();
    consulta.close();
} catch (SQLException e) {
    System.out.println("Error al visualizar los libros: " +
e.getMessage());
}
con.CerrarConexion();
}

}

```



CLASE USUARIO

La clase "Usuario" es una clase Java que representa un objeto Usuario y se utiliza para crear, eliminar, modificar y visualizar los usuarios de una base de datos. Esta clase tiene los siguientes atributos privados:

- `cod_usuario`: un entero que representa el código único del usuario.
- `usuario`: una cadena de caracteres que representa el nombre de usuario del usuario.
- `nombre`: una cadena de caracteres que representa el nombre del usuario.
- `apellido`: una cadena de caracteres que representa el apellido del usuario.
- `contrasena`: una cadena de caracteres que representa la contraseña del usuario.
- `email`: una cadena de caracteres que representa la dirección de correo electrónico del usuario.

La clase también tiene un objeto "connect" que se utiliza para conectar y desconectar la base de datos. La clase tiene un constructor que recibe los valores de los atributos como parámetros y se utiliza para inicializar los valores de los atributos. La clase tiene métodos getter y setter para cada atributo para acceder y modificar los valores de los atributos.

La clase tiene los siguientes métodos públicos:

- `CrearUsuario`: un método que recibe los valores de los atributos como parámetros y se utiliza para crear un nuevo usuario en la base de datos.
- `EliminarUsuario`: un método que recibe el código de usuario como parámetro y se utiliza para eliminar el usuario correspondiente de la base de datos.
- `modificarUsuario`: un método que recibe los valores de los atributos como parámetros y se utiliza para modificar los datos del usuario correspondiente en la base de datos.
- `visualizarUsuarios`: un método que se utiliza para recuperar y mostrar todos los usuarios de la base de datos.

```
import java.sql.*;

public class Usuario {
    connect con;

    private int cod_usuario;
    private String usuario;
    private String nombre;
    private String apellido;
    private String contrasena;
    private String email;

    public Usuario(int cod_usuario, String usuario, String nombre, String
apellido, String contrasena, String email) {
        this.cod_usuario = cod_usuario;
        con = new connect();
        con.Conectar();
        this.usuario = usuario;
        this.nombre = nombre;
        this.apellido = apellido;
        this.contrasena = contrasena;
        this.email = email;
    }

    // Métodos getters y setters
    public int getCodUsuario() {
        return cod_usuario;
    }

    public void setCodUsuario(int cod_usuario) {
        this.cod_usuario = cod_usuario;
    }

    public String getUsuario() {
        return usuario;
    }

    public void setUsuario(String usuario) {
        this.usuario = usuario;
    }

    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 getContrasena() {
        return contrasena;
    }

    public void setContrasena(String contrasena) {
        this.contrasena = contrasena;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public void CrearUsuario(String usuario, String nombre, String apellido,
String contrasena, String email){
        con.Conectar();
        try{
            PreparedStatement consulta = con.conexion.prepareStatement("INSERT
INTO usuarios(usuario, nombre, apellido, contrasena, email)
VALUES(?,?,?,?,?)");
            consulta.setString(1, usuario);
            consulta.setString(2, nombre);
            consulta.setString(3, apellido);
            consulta.setString(4, contrasena);
            consulta.setString(5, email);
            consulta.executeUpdate();

```

```

        System.out.println("Usuario creado exitosamente");
        consulta.close();
    }
    catch(SQLException e){
        System.out.println("Error al crear el usuario: " +
e.getMessage());
    }
    con.CerrarConexion();
}

public void EliminarUsuario(int xid){
    con.Conectar();
    try {
        PreparedStatement consulta = con.conexion.prepareStatement("DELETE
FROM usuarios WHERE cod_usuario = ?");
        consulta.setInt(1, xid);
        consulta.executeUpdate();
        System.out.println("Usuario eliminado exitosamente");
        consulta.close();
    } catch (SQLException e) {
        System.out.println("Error al eliminar el usuario: " +
e.getMessage());
    }
    con.CerrarConexion();
}

public void modificarUsuario(int id, String nombre, String usuario, String
apellido, String correo, String password) {
    con.Conectar();
    try {
        PreparedStatement ps = con.conexion.prepareStatement("UPDATE
usuarios SET nombre=?, usuario=?, apellido=?, email=?, contrasena=? WHERE
cod_usuario=?");
        ps.setString(1, nombre);
        ps.setString(2, usuario);
        ps.setString(3, apellido);
        ps.setString(4, correo);
        ps.setString(5, password);
        ps.setInt(6, id);
        ps.executeUpdate();
    } catch (SQLException ex) {
        System.out.println("Error al modificar usuario: " +
ex.getMessage());
    }
}

```

```

        con.CerrarConexion();

    }

    public void visualizarUsuarios(){
        con.Conectar();
        try{
            PreparedStatement consulta = con.conexion.prepareStatement("SELECT
* FROM usuarios");
            ResultSet resultado = consulta.executeQuery();
            while(resultado.next()){
                System.out.println("Codigo de Usuario: " +
resultado.getInt("cod_usuario"));
                System.out.println("Usuario: " +
resultado.getString("usuario"));
                System.out.println("Nombre: " + resultado.getString("nombre"));
                System.out.println("Apellido: " +
resultado.getString("apellido"));
                System.out.println("Email: " + resultado.getString("email"));
                System.out.println("-----");
            }
            resultado.close();
            consulta.close();
        }
        catch(SQLException e){
            System.out.println("Error al visualizar los usuarios: " +
e.getMessage());
        }
        con.CerrarConexion();
    }

}

```

CLASE CONNECT

```
public class connect {
    Connection conexion;
    String url="jdbc:mysql://localhost:3306/libroslibresbd";
    String usuario="root";
    String contrasena= "";
    boolean bandera=true;
    public void Conectar(){
        //-----BBDD USUARIOS-----
        try{
            conexion = DriverManager.getConnection(url, usuario,
contrasena);
            System.out.println("Conexión exitosa a la base de datos");
        }
        catch (SQLException e){
            System.out.print("Imposible conectar" + e.getMessage());
        }
    }

    public void CerrarConexion(){
        try {
            conexion.close();
            System.out.println("Conexión cerrada exitosamente");
        }
        catch (SQLException e) {
            System.out.print("Error al cerrar la conexión" +
e.getMessage());
        }
    }
}
```

CLASE AUTORES

es una clase Java llamada "autor" que representa un autor de libros. La clase contiene varios métodos que permiten la inserción, actualización y eliminación de autores en una base de datos SQL, así como la visualización de todos los autores existentes en la base de datos y la búsqueda de un autor por nombre.

La clase "autor" tiene los siguientes atributos:

- `cod_autor`: entero que representa el código del autor
- `nombre`: cadena que representa el nombre del autor
- `apellido`: cadena que representa el apellido del autor
- `fecha_nacimiento`: cadena que representa la fecha de nacimiento del autor
- `nacionalidad`: cadena que representa la nacionalidad del autor

El constructor de la clase acepta todos los atributos de un autor y los inicializa.

La clase tiene varios métodos para interactuar con una base de datos SQL. El método "anadirautor" permite agregar un nuevo autor a la base de datos. El método "modificarautor" permite actualizar los datos de un autor existente en la base de datos. El método "eliminarautor" permite eliminar un autor existente de la base de datos. El método "verAutores" permite mostrar todos los autores existentes en la base de datos en una tabla. Finalmente, el método "buscarAutor" busca un autor en la base de datos por nombre y devuelve el código del autor si lo encuentra.

La clase también tiene varios métodos de acceso y modificación para los atributos "cod_autor", "nombre" y "apellido".

Además, la clase contiene una instancia de la clase "connect", que se utiliza para conectarse a la base de datos SQL y ejecutar las consultas necesarias.


```

import java.sql.*;

public class autor {
    private int cod_autor;
    private String nombre;
    private String apellido;
    private String fecha_nacimiento;
    private String nacionalidad;
    connect con = new connect();

    public autor(int cod_autor, String nombre, String apellido, String
fecha_nacimiento, String nacionalidad) {
        this.cod_autor = cod_autor;
        this.nombre = nombre;
        this.apellido = apellido;
        this.fecha_nacimiento = fecha_nacimiento;
        this.nacionalidad = nacionalidad;
    }

    public void anadirautor( String nombre, String apellido, String
fecha_nacimiento, String nacionalidad) {
        try {
            con.Conectar();
            String query = "INSERT INTO autores ( nombre, apellido,
fecha_nacimiento, nacionalidad) " +
                "VALUES (?, ?, ?, ?, ?)";
            PreparedStatement stmt = con.conexion.prepareStatement(query);
            stmt.setInt(1, cod_autor);
            stmt.setString(2, nombre);
            stmt.setString(3, apellido);
            stmt.setString(4, fecha_nacimiento);
            stmt.setString(5, nacionalidad);
            stmt.executeUpdate();
            System.out.println("autor añadido correctamente.");
            con.CerrarConexion();
        } catch (SQLException e) {
            System.out.println("Error al añadir autor: " + e.getMessage());
        }
    }

    public void modificarautor(int cod_autor, String nombre, String apellido,
String fecha_nacimiento, String nacionalidad) {

```

```

try {
    con.Conectar();
    String query = "UPDATE autores SET nombre=?, apellido=?,
fecha_nacimiento=?, nacionalidad=? WHERE cod_autor=?";
    PreparedStatement stmt = con.conexion.prepareStatement(query);
    stmt.setString(1, nombre);
    stmt.setString(2, apellido);
    stmt.setString(3, fecha_nacimiento);
    stmt.setString(4, nacionalidad);
    stmt.setInt(5, cod_autor);
    int rowsUpdated = stmt.executeUpdate();
    if (rowsUpdated > 0) {
        System.out.println("autor modificado correctamente.");
    } else {
        System.out.println("No se encontró el autor con código " +
cod_autor);
    }
    con.CerrarConexion();
} catch (SQLException e) {
    System.out.println("Error al modificar autor: " + e.getMessage());
}

}

public void eliminarautor(int cod_autor) {
    try {
        con.Conectar();
        String query = "DELETE FROM autores WHERE cod_autor=?";

        PreparedStatement stmt = con.conexion.prepareStatement(query);
        stmt.setInt(1, cod_autor);
        int rowsDeleted = stmt.executeUpdate();
        if (rowsDeleted > 0) {
            System.out.println("autor eliminado correctamente.");
        } else {
            System.out.println("No se encontró el autor con código " +
cod_autor);
        }
        con.CerrarConexion();
    } catch (SQLException e) {
        System.out.println("Error al eliminar autor: " + e.getMessage());
    }
}
}

```

```

    public void verAutores() {
        con.Conectar();
        String query = "SELECT * FROM autores";
        try {

System.out.println("+-----+-----+-----+-----+");
--+-----+");
            System.out.printf("| %-4s | %-14s | %-12s | %-19s | %-13s |\n",
"ID", "Nombre", "Apellido", "Fecha de nacimiento", "Nacionalidad");

System.out.println("+-----+-----+-----+-----+");
--+-----+");
            Statement stmt = con.conexion.createStatement();
            ResultSet rs = stmt.executeQuery(query);
            while (rs.next()) {
                int cod_autor = rs.getInt("cod_autor");
                String nombre = rs.getString("nombre");
                String apellido = rs.getString("apellido");
                Date fecha_nacimiento = rs.getDate("fecha_nacimiento");
                String nacionalidad = rs.getString("nacionalidad");
                System.out.printf("| %-4d | %-14s | %-12s | %-19s | %-13s |\n",
cod_autor, nombre, apellido, fecha_nacimiento, nacionalidad);
                System.out.println();
            }

System.out.println("+-----+-----+-----+-----+");
--+-----+");

            rs.close();
            stmt.close();
            con.CerrarConexion();
        } catch (SQLException e) {
            System.out.println("Error al ejecutar la consulta: " +
e.getMessage());
        }

System.out.println("+-----+-----+-----+-----+");
--+-----+");

        con.CerrarConexion();
    }
    public int buscarAutor(String nombre){

```

```

int codigoAutor = -1; // valor por defecto si no se encuentra el autor
try{
    con.Conectar();
    String query = "SELECT cod_autor FROM autores WHERE nombre=?";
    PreparedStatement stmt = con.conexion.prepareStatement(query);
    stmt.setString(1, nombre);

    ResultSet rs = stmt.executeQuery(); // ejecutar la consulta y
obtener el resultado

    if (rs.next()) { // si hay al menos una fila en el resultado
        codigoAutor = rs.getInt("cod_autor"); // obtener el valor de la
columna cod_autor
    }

    con.CerrarConexion();
} catch (SQLException e) {
    System.out.println("Error al buscar autor: " + e.getMessage());
}
return codigoAutor;
}

public int getCod_autor() {
    return cod_autor;
}

public void setCod_autor(int cod_autor) {
    this.cod_autor = cod_autor;
}

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 getFecha_nacimiento() {
    return fecha_nacimiento;
}

public void setFecha_nacimiento(String fecha_nacimiento) {
    this.fecha_nacimiento = fecha_nacimiento;
}

public String getNacionalidad() {
    return nacionalidad;
}

public void setNacionalidad(String nacionalidad) {
    this.nacionalidad = nacionalidad;
}

public connect getCon() {
    return con;
}

public void setCon(connect con) {
    this.con = con;
}
}
```

MAIN (TESTEO):

```
import java.util.Scanner;

public class main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Usuario usuario = new Usuario(0, "", "", "", "", "");
        Libro libros = new Libro(0, null, null, null, null, null, 0, 0, 0,
null);
        boolean bandera=true;

        while (bandera=true) {
            System.out.println("#          ### #####          #####          #
### #####          #####          #####          " );
            System.out.println("#          # #          # #          # #          # #
# #          # #          # #          # #          ");
            System.out.println("#          # #          # #          # #          #
# #          # #          # #          # #          ");
            System.out.println("#          # #####          #####          #          # #####          #
# #####          #####          #####          ");
            System.out.println("#          # #          # #          # #          # #
# #          # #          # #          # #          ");
            System.out.println("#          # #          # #          # #          # #
# #          # #          # #          # #          ");
            System.out.println("##### ### #####          #          # #####          #####
#####          #####          #          # #####          #####          ");

            System.out.println("MENU");
            System.out.println("1. Crear usuario");
            System.out.println("2. Eliminar usuario");
            System.out.println("3. Modificar usuario");
            System.out.println("4. Añadir libro");
            System.out.println("5. Eliminar libro");
            System.out.println("6. Modificar libro");
            System.out.println("7. Ver libros");
            System.out.println("8. Ver usuarios");
            System.out.println("9. Salir");
            System.out.print("Elija una opción: ");
            int opcion = scanner.nextInt();

            switch (opcion) {
```

```

case 1:
    System.out.print("Ingrese el nombre de usuario: ");
    String nombreUsuario = scanner.next();
    System.out.print("Ingrese el nombre: ");
    String nombre = scanner.next();
    System.out.print("Ingrese el apellido: ");
    String apellido = scanner.next();
    System.out.print("Ingrese la contraseña: ");
    String contrasena = scanner.next();
    System.out.print("Ingrese el correo electrónico: ");
    String correo = scanner.next();
    usuario.CrearUsuario(nombreUsuario, nombre, apellido,
contrasena, correo);
    break;
case 2:
    System.out.print("Ingrese el código de usuario a eliminar:
");

    int codigoEliminar = scanner.nextInt();
    usuario.EliminarUsuario(codigoEliminar);
    break;
case 3:
    System.out.print("Ingrese el código de usuario a modificar:
");

    int codigoModificar = scanner.nextInt();
    System.out.print("Ingrese el nuevo nombre: ");
    String nuevoNombre = scanner.next();
    System.out.print("Ingrese el nuevo nombre de usuario: ");
    String nuevoNombreUsuario = scanner.next();
    System.out.print("Ingrese el nuevo apellido: ");
    String nuevoApellido = scanner.next();
    System.out.print("Ingrese el nuevo correo electrónico: ");
    String nuevoCorreo = scanner.next();
    System.out.print("Ingrese la nueva contraseña: ");
    String nuevaContrasena = scanner.next();
    usuario.modificarUsuario(codigoModificar, nuevoNombre,
nuevoNombreUsuario, nuevoApellido, nuevoCorreo, nuevaContrasena);
    break;
case 4:
    System.out.print("Ingrese el ISBN del libro: ");
    String isbn = scanner.next();
    System.out.print("Ingrese el nombre del libro: ");
    String nombreLibro = scanner.next();
    System.out.print("Ingrese la sinopsis del libro: ");

```

```

        String sinopsis = scanner.next();
        System.out.print("Ingrese la fecha de publicación del libro (en
formato yyyy-mm-dd): ");
        String fechaPublicacion = scanner.next();
        System.out.print("Ingrese el idioma del libro: ");
        String idioma = scanner.next();
        System.out.print("Ingrese el número de páginas del libro: ");
        int numeroPaginas = scanner.nextInt();
        System.out.print("Ingrese el código del autor del libro: ");
        int codAutor = scanner.nextInt();
        System.out.print("Ingrese el código del género del libro: ");
        int codGenero = scanner.nextInt();
        System.out.print("Ingrese el código de la editorial del libro:
");

        int codEditorial = scanner.nextInt();
        System.out.print("Ingrese el link del pdf del libro: ");
        String linkpdf = scanner.next();
        libros.crearLibro(isbn, nombreLibro, sinopsis,
fechaPublicacion, idioma, numeroPaginas, codAutor, codGenero, codEditorial,
linkpdf);

        break;
    case 5:
        System.out.print("Ingrese el ISBN del libro a eliminar: ");
        String isbnEliminar = scanner.next();
        libros.eliminarLibro(isbnEliminar);
        break;
    case 6:
        System.out.print("Ingrese el ISBN del libro a modificar: ");
        String isbnModificar = scanner.next();
        System.out.print("Ingrese el nuevo nombre del libro: ");
        String nuevoNombreLibro = scanner.next();
        System.out.print("Ingrese la nueva sinopsis del libro: ");
        String nuevaSinopsis = scanner.next();
        System.out.print("Ingrese la nueva fecha de publicación del
libro (en formato yyyy-mm-dd): ");
        String nuevaFechaPublicacion = scanner.next();
        System.out.print("Ingrese el nuevo idioma del libro: ");
        String nuevoIdioma = scanner.next();
        System.out.print("Ingrese el nuevo número de páginas del libro:
");

        int nuevoNumeroPaginas = scanner.nextInt();
        System.out.print("Ingrese el nuevo código del autor del libro:
");

```



```

        int nuevoCodAutor = scanner.nextInt();
        System.out.print("Ingrese el nuevo código del género del libro: ");
    });

    int nuevoCodGenero = scanner.nextInt();
    System.out.print("Ingrese el nuevo código de la editorial del
libro: ");

    int nuevoCodEditorial = scanner.nextInt();
    System.out.print("Ingrese el nuevo link del pdf del libro: ");
    String nuevoLinkpdf = scanner.next();
    libros.modificarlibro(isbnModificar, nuevoNombreLibro,
nuevaSinopsis, nuevaFechaPublicacion, nuevoIdioma, nuevoNumeroPaginas,
nuevoCodAutor, nuevoCodGenero, nuevoCodEditorial, nuevoLinkpdf);
    break;
    case 7:
        libros.visualizarLibros();break;
    case 8:
        usuario.visualizarUsuarios();break;
    case 9:
        System.out.println("Saliendo del programa...");
        System.exit(0);
    default:
        System.out.println("Opción inválida");
        break;
    }
}
}
}
}

```



TESTEO

VIDEO DE DEMOSTRACIÓN:

<https://icecreamapps.com/v/cusutq3>

La demostración está hecha con usuarios, sería lo mismo con libros y autores, la app pide la información y lo reemplaza en la bbdd, para más profundización descargar y usar el main.

DESCARGA

https://drive.google.com/drive/folders/1kTzFGY4lj6hmUeJ5999On-k92JSNXRPB?usp=share_link



MANUAL DE USO PARA TRABAJADORES

Introducción:

Este programa es una herramienta de gestión de libros en la cual puedes llevar un registro de los usuarios, libros y autores.

Menú principal:

Al iniciar el programa, te encontrarás con un menú principal que te brinda varias opciones. Las opciones te permiten crear, modificar, eliminar y ver información relacionada con los usuarios, libros y autores registrados en el sistema.

Sección de Usuarios:

En el apartado de Usuarios, tendrás la posibilidad de:

- Crear un nuevo usuario ingresando su información personal como su nombre, apellido, correo electrónico, contraseña y nombre de usuario.
- Eliminar un usuario específico ingresando su código.
- Modificar los datos de un usuario ya existente en el sistema.
- Ver una lista de todos los usuarios registrados en el sistema.

Sección de Libros:

Si decides seleccionar la opción de Libros, podrás:

- Agregar un nuevo libro al sistema ingresando información detallada sobre el mismo, como su ISBN, nombre, sinopsis, fecha de publicación, idioma, número de páginas, código de autor, código de género, código de editorial y link del PDF del libro.
- Eliminar o modificar un libro específico.
- Ver una lista de todos los libros registrados en el sistema.



Sección de Autores:

En el apartado de Autores, podrás:


- Agregar nuevos autores ingresando su información personal como su nombre, apellido, nacionalidad y fecha de nacimiento.
- Eliminar o modificar un autor específico.
- Ver una lista de todos los autores registrados en el sistema.

Conclusión:

En resumen, este programa es una herramienta muy útil para llevar un control detallado y organizado de los usuarios, libros y autores. Con este programa, podrás gestionar y mantener actualizado el inventario de libros, así como llevar un registro de los usuarios y autores que interactúan con el sistema.

TABLAS NECESARIAS DE CONOCER PARA TRABAJAR CON LA APP:

cod_genero	tipo	genero
1	prueba	prueba
2	Novela	Fantasia
3	Novela	Accion
4	Novela	Terror
5	Novela	Comedia
6	Novela	Drama
7	Novela	Infantil
8	Novela	Educativo
9	Novela	Deporte
10	Novela	Biografia
11	Novela	Erotico
20	Fabula	Fantasia
21	Fabula	Accion
22	Fabula	Drama
23	Fabula	Infantil
24	Fabula	Educativo
25	Fabula	Deporte
26	Fabula	Biografia
27	Fabula	Erotico
28	Fabula	Terror
29	Fabula	Comedia
30	Cuento	Fantasia
31	Cuento	Accion
32	Cuento	Drama
33	Cuento	Infantil
34	Cuento	Educativo
35	Cuento	Deporte
36	Cuento	Biografia
37	Cuento	Erotico
38	Cuento	Terror
39	Cuento	Comedia



LAS EDITORIALES DEBEN ESTAR REGISTRADAS PRIMERO EN LA PÁGINA WEB, LOS TRABAJADORES DISPONDRÁN DE UNA PANTALLA CERCA DE SU SITIO DE TRABAJO PARA TENER ACCESO DIRECTO A LA VISUALIZACIÓN DE ESA TABLA.