DOCUMENTACIÓN Y ACLARACIONES DEL proceso de diseño del programa para LA BIBLIOTECA LA PINGÜINERA

RESPONSABLE: **JULIO ELIAS VASQUEZ ROJAS**  
COACH: **JUAN PINEDA ANGEL**

# RESUMEN

Ese documento pretende aclarar todas las medidas y procedimientos que se tomaron para realizar el taller de Java (POO) Programación orientada a objetos número 1, el cual requiere realizar un programa para la gestión de una biblioteca llamada la pingüinera, para La cual se solicitan condiciones específicas por lo que se requiere realizar también un diseño del modelo de la base de datos, Cómo se va a manejar la información y la relación que van a existir entre las entidades.

# SUMMARY

This document aims to clarify all the steps and procedures that were taken to perform the Java workshop (OOP) Object Oriented Programming number 1, which requires making a program for the management of a library called the penguin, for which specific conditions are requested so it is also required to perform a design of the database model, how to handle the information and the relationship that will exist between entities.

Table of Contents

[RESUMEN 1](#_Toc164030532)

[SUMMARY 1](#_Toc164030533)

[Diseño del diagrama de la base de datos 3](#_Toc164030534)

[Entidad-Relación 3](#_Toc164030535)

[Relaciones: 3](#_Toc164030536)

[Aclaraciones: 3](#_Toc164030537)

[Modelo Relacional 4](#_Toc164030538)

[MR en Workbench 4](#_Toc164030539)

[Creación de la base de datos 6](#_Toc164030540)

[Comandos SQL 6](#_Toc164030541)

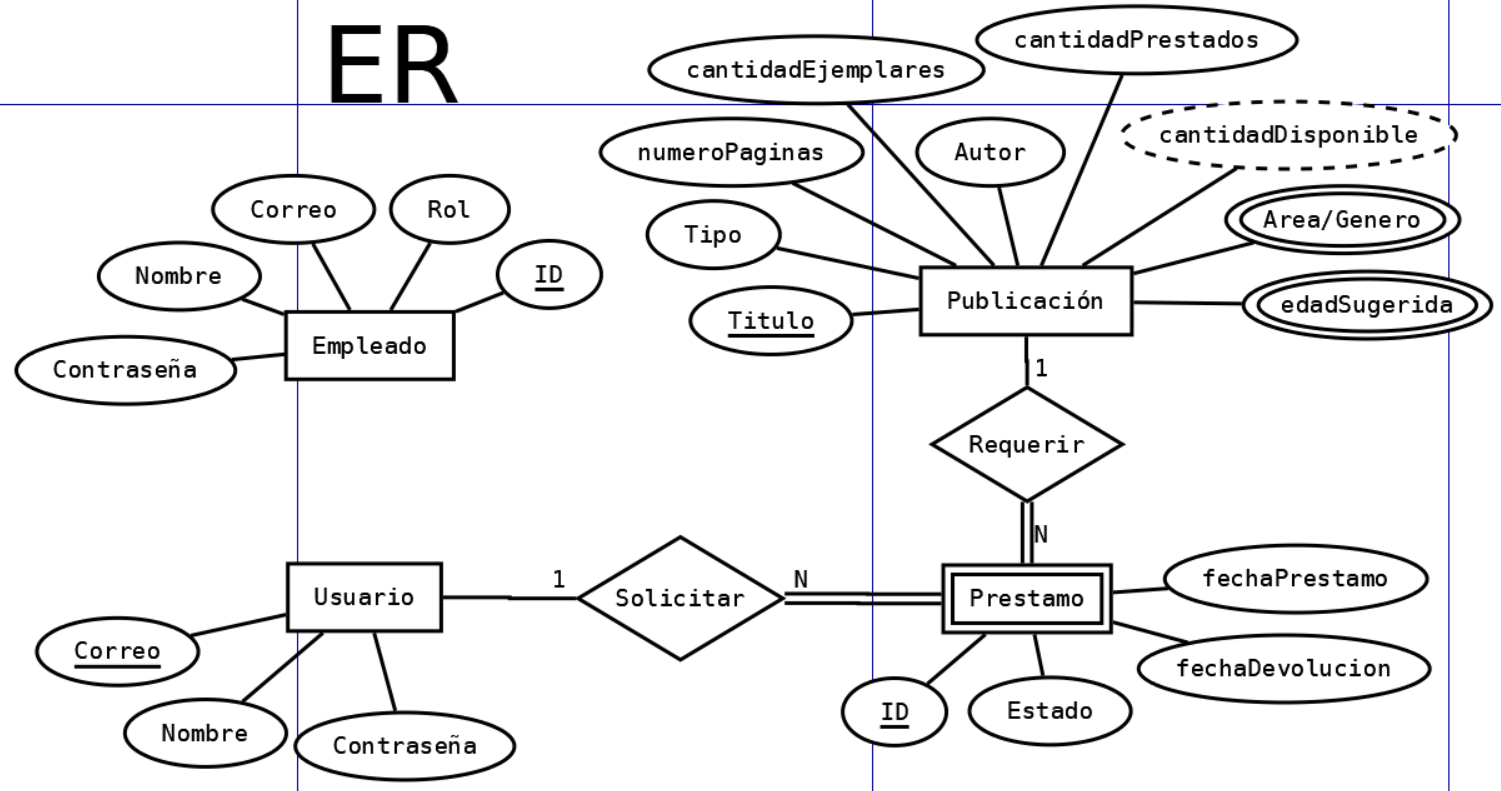
[Resultado de la ejecución 8](#_Toc164030542)

[Comandos de consulta de tablas 9](#_Toc164030543)

# Diseño del diagrama de la base de datos

Para efectos del proceso de diseño de la base de datos se parte primeramente por un diagrama entidad relación, El cual se establece las entidades necesarias para representar los datos del negocio de acuerdo sus necesidades, Además la relación entre estas entidades.

## Entidad-Relación



### Relaciones:

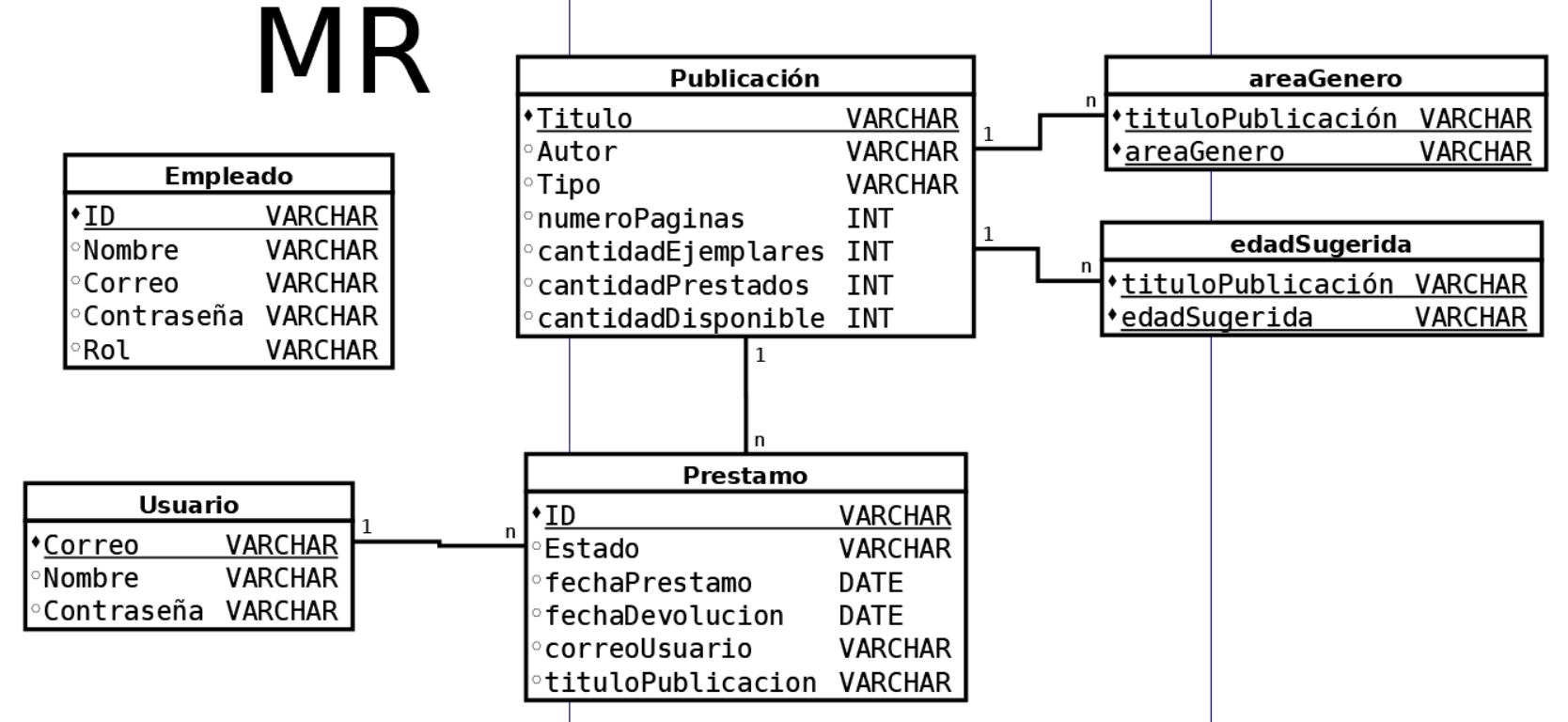
* Usuario-Solicitar-Préstamo: Un usuario puede solicitar muchos préstamos, un préstamo sólo puede ser solicitado por un usuario.
* Publicación-Requerir-Préstamo: Una publicación puede ser requerida por muchos préstamos, un préstamo solo puede requerir una publicación.

### Aclaraciones:

* La información que se va a almacenar para los empleados va a ser gestionada desde el programa diseñado en Java.
* Todas las relaciones tienen una Cardinalidad de uno a muchos o de muchos a 1, para Cumplir con los requerimientos de la manera más eficiente posible.

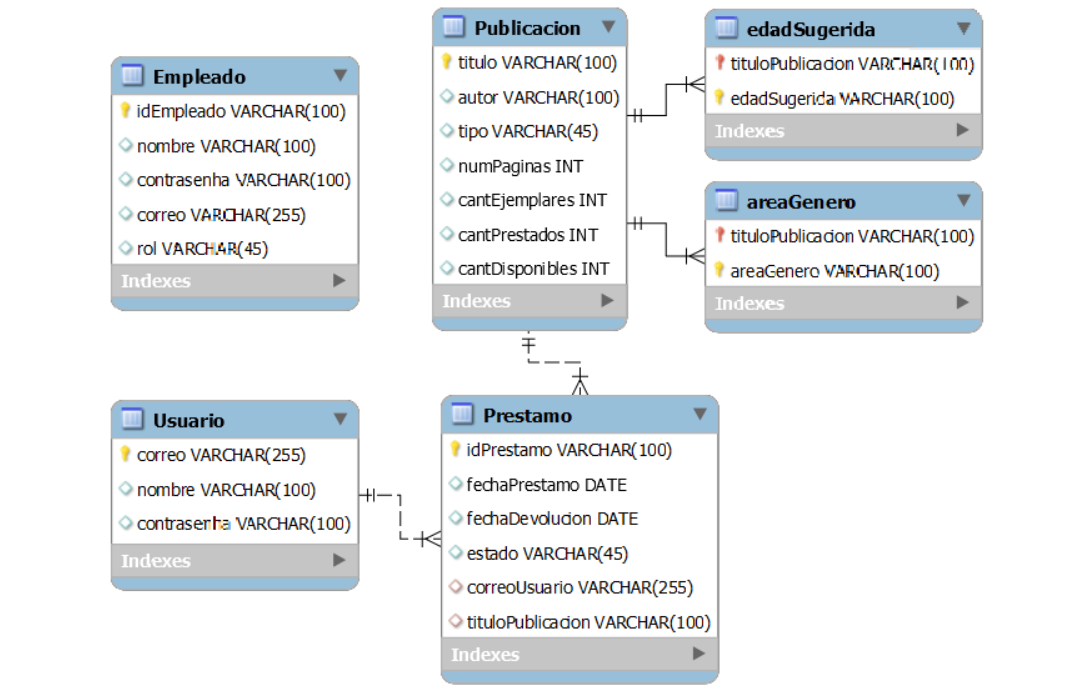
Con este modelo entidad relación se puede desarrollar o transformar en un modelo relacional, el cual nos permite representar de una forma más dinámica las tablas en las cuales se van a almacenar todos los datos.

## Modelo Relacional



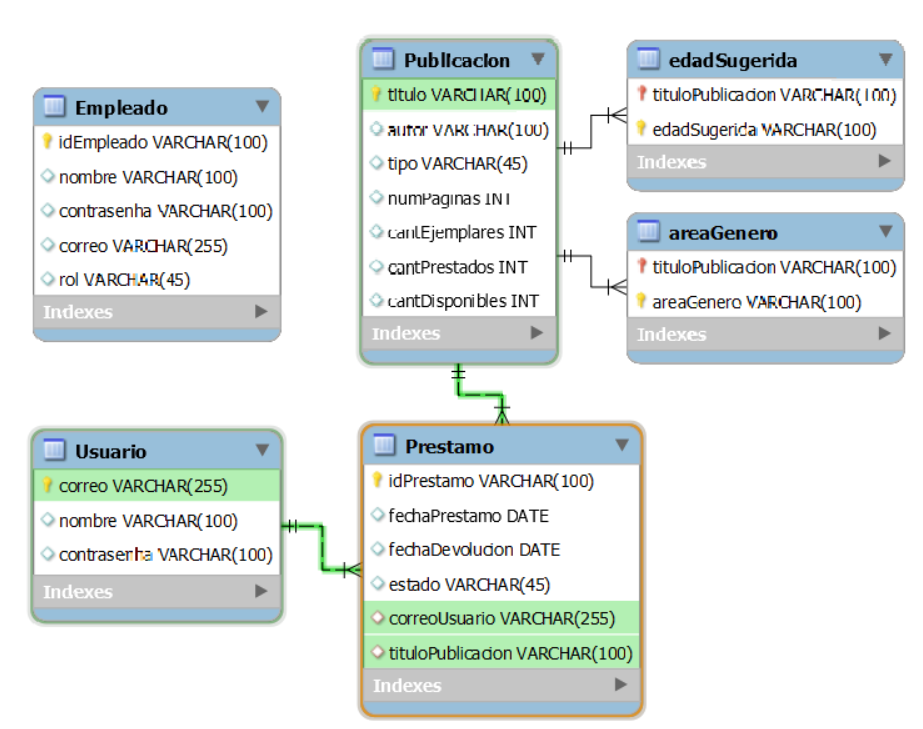
En este modelo se mantiene la misma relación entre las entidades y sus cardinalidades, Teniendo en cuenta los atributos multivaluados, que es el proceso de normalización se transformen en unas tablas que son entidades independientes.

## MR en Workbench



Se procede al desarrollo en Workbench para la creación de la base de datos.

Se establecen las relaciones de las llaves primarias y foráneas en la entidad préstamo.



Se realiza lo mismo con la entidad publicación.

A diagram of a database

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Se establece la siguiente condición en la cual:

* Cantidad Disponibles: Se estable por predeterminado en la base de datos como la resta entre la cantidad de ejemplares y la cantidad de prestados.

Todo este proceso fue enseñado por el Coach y replicado por el responsable del documento.

# Creación de la base de datos

A partir del modelo realizado en MySQL se procede a generar el código para la creación de la base de datos a través de comandos DDL abreviación que significa Lenguaje de Definición de Datos.

## Comandos SQL

-- MySQL Script generated by MySQL Workbench

-- Sat Apr 13 20:15:28 2024

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

**SET** @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=**0**;

**SET** @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=**0**;

**SET** @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema bibliotecaPingui

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema bibliotecaPingui

-- -----------------------------------------------------

**CREATE** **SCHEMA** **IF** **NOT** **EXISTS** `bibliotecaPingui` **DEFAULT** **CHARACTER** **SET** utf8 ;

**USE** `bibliotecaPingui` ;

-- -----------------------------------------------------

-- Table `bibliotecaPingui`.`Empleado`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `bibliotecaPingui`.`Empleado` (

`idEmpleado` **VARCHAR**(**100**) **NOT** **NULL**,

`nombre` **VARCHAR**(**100**) **NULL**,

`contrasenha` **VARCHAR**(**100**) **NULL**,

`correo` **VARCHAR**(**255**) **NULL**,

`rol` **VARCHAR**(**45**) **NULL**,

**PRIMARY** **KEY** (`idEmpleado`))

**ENGINE** = InnoDB;

-- -----------------------------------------------------

-- Table `bibliotecaPingui`.`Usuario`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `bibliotecaPingui`.`Usuario` (

`correo` **VARCHAR**(**255**) **NOT** **NULL**,

`nombre` **VARCHAR**(**100**) **NULL**,

`contrasenha` **VARCHAR**(**100**) **NULL**,

**PRIMARY** **KEY** (`correo`))

**ENGINE** = InnoDB;

-- -----------------------------------------------------

-- Table `bibliotecaPingui`.`Publicacion`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `bibliotecaPingui`.`Publicacion` (

`titulo` **VARCHAR**(**100**) **NOT** **NULL**,

`autor` **VARCHAR**(**100**) **NULL**,

`tipo` **VARCHAR**(**45**) **NULL**,

`numPaginas` **INT** **NULL**,

`cantEjemplares` **INT** **NULL**,

`cantPrestados` **INT** **NULL**,

`cantDisponibles` **INT** **NULL** **DEFAULT** (cantEjemplares - cantPrestados),

**PRIMARY** **KEY** (`titulo`))

**ENGINE** = InnoDB;

-- -----------------------------------------------------

-- Table `bibliotecaPingui`.`Prestamo`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `bibliotecaPingui`.`Prestamo` (

`idPrestamo` **VARCHAR**(**100**) **NOT** **NULL**,

`fechaPrestamo` **DATE** **NULL**,

`fechaDevolucion` **DATE** **NULL**,

`estado` **VARCHAR**(**45**) **NULL**,

`correoUsuario` **VARCHAR**(**255**) **NULL**,

`tituloPublicacion` **VARCHAR**(**100**) **NULL**,

**PRIMARY** **KEY** (`idPrestamo`),

**INDEX** `fkCorreo\_idx` (`correoUsuario` **ASC**) VISIBLE,

**INDEX** `fkTitulo\_idx` (`tituloPublicacion` **ASC**) VISIBLE,

**CONSTRAINT** `fkCorreo`

**FOREIGN** **KEY** (`correoUsuario`)

**REFERENCES** `bibliotecaPingui`.`Usuario` (`correo`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION,

**CONSTRAINT** `fkTitulo`

**FOREIGN** **KEY** (`tituloPublicacion`)

**REFERENCES** `bibliotecaPingui`.`Publicacion` (`titulo`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION)

**ENGINE** = InnoDB;

-- -----------------------------------------------------

-- Table `bibliotecaPingui`.`edadSugerida`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `bibliotecaPingui`.`edadSugerida` (

`tituloPublicacion` **VARCHAR**(**100**) **NOT** **NULL**,

`edadSugerida` **VARCHAR**(**100**) **NOT** **NULL**,

**PRIMARY** **KEY** (`tituloPublicacion`, `edadSugerida`),

**CONSTRAINT** `fkEdadSugerida`

**FOREIGN** **KEY** (`tituloPublicacion`)

**REFERENCES** `bibliotecaPingui`.`Publicacion` (`titulo`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION)

**ENGINE** = InnoDB;

-- -----------------------------------------------------

-- Table `bibliotecaPingui`.`areaGenero`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `bibliotecaPingui`.`areaGenero` (

`tituloPublicacion` **VARCHAR**(**100**) **NOT** **NULL**,

`areaGenero` **VARCHAR**(**100**) **NOT** **NULL**,

**PRIMARY** **KEY** (`tituloPublicacion`, `areaGenero`),

**CONSTRAINT** `fkAreaGenero`

**FOREIGN** **KEY** (`tituloPublicacion`)

**REFERENCES** `bibliotecaPingui`.`Publicacion` (`titulo`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION)

**ENGINE** = InnoDB;

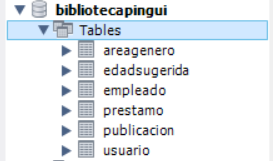
**SET** SQL\_MODE=@OLD\_SQL\_MODE;

**SET** FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

**SET** UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

## Resultado de la ejecución

Se genera la base de datos con sus tablas:

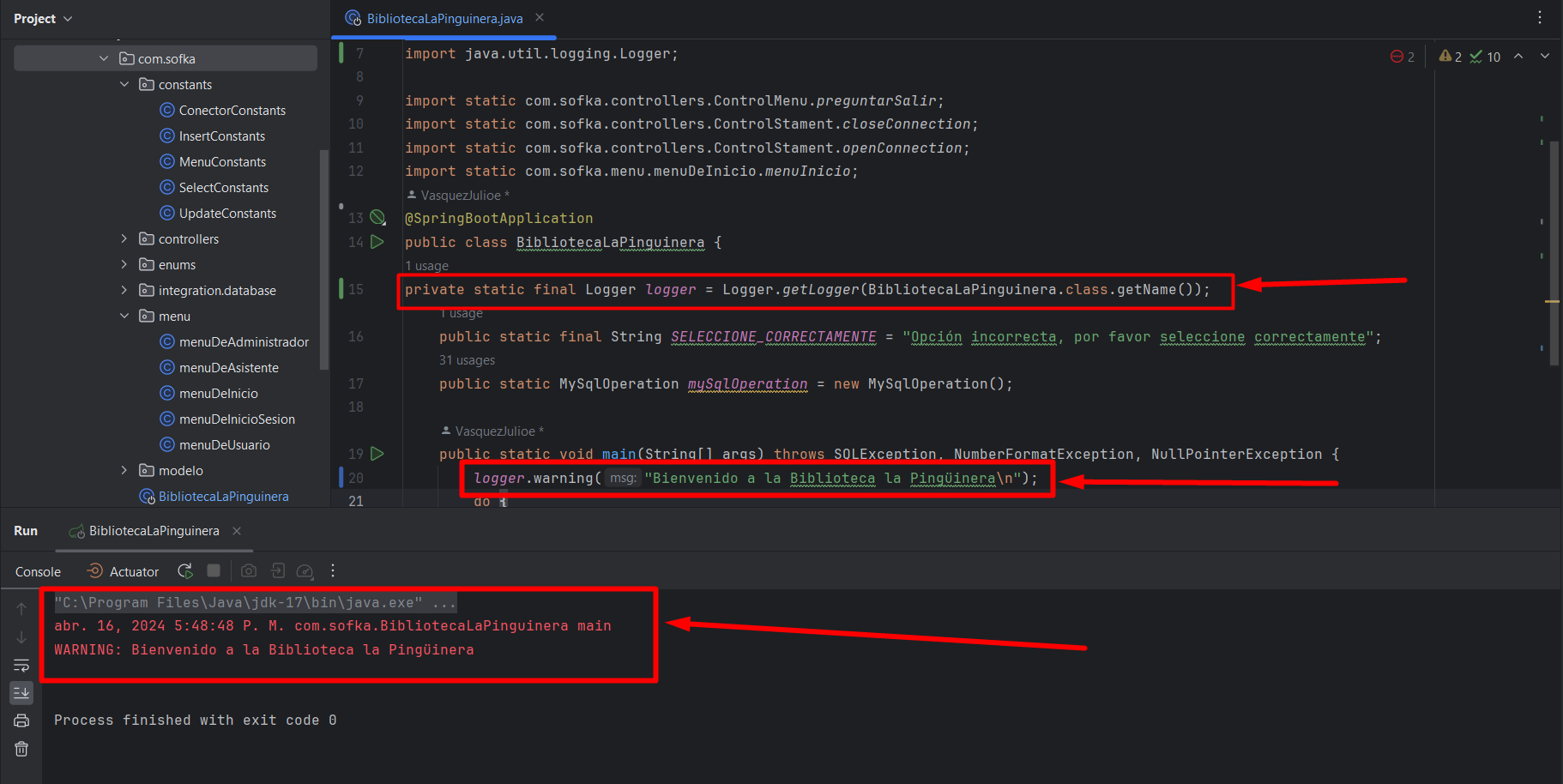


## Comandos de consulta de tablas

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **SELECT** \* **FROM** empleado;  **SELECT** \* **FROM** publicacion;  **SELECT** \* **FROM** prestamo;  **SELECT** \* **FROM** usuario;  **SELECT** \* **FROM** areagenero;  **SELECT** \* **FROM** edadsugerida; |

# Código Java

## Implementación del Logger



Para cada vez que se inicia sesión se realiza un log de inicio de la aplicación.

## Pruebas de descuento de libros disponibles en el préstamo

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A table with numbers and letters

Description automatically generated

A screenshot of a computer

Description automatically generated

