

PROYECTO SEGUNDA EVALUACIÓN

SISTEMA DE GESTIÓN DE BIBLIOTECA CON CONECTORES JDBC EN JAVA

Contenido

1.	Descripción General	2
2.	Requisitos Funcionales	2
	Gestión de Usuarios	2
	Gestión de Libros	2
	Gestión de Préstamos	2
	Generación de Reportes	3
3.	Requisitos Técnicos	3
	Base de Datos Relacional	3
	Estructura del Código	4
	Configuración Avanzada	4
	Seguridad	4
4.	Requisitos de Entrega.....	5
	Documentación	5
	Código	5
	Extras Opcionales (puntaje adicional)	5
5.	Plazo de Entrega.....	6
6.	Criterios de Evaluación	6

1. Descripción General

El objetivo del proyecto es desarrollar un sistema de gestión de biblioteca que permita realizar operaciones como registrar usuarios, gestionar libros y préstamos, generar reportes avanzados y garantizar la seguridad de los datos. Deberás aplicar conceptos avanzados de programación en Java, diseño de bases de datos y buenas prácticas en el uso de conectores JDBC.

2. Requisitos Funcionales

Gestión de Usuarios

- Registrar nuevos usuarios con validaciones (nombre, email único, teléfono).
- Actualizar datos de un usuario existente.
- Eliminar usuarios solo si no tienen préstamos activos.
- Implementar roles (administrador y usuario regular), donde los administradores tienen permisos para gestionar libros y reportes.

Gestión de Libros

- Registrar nuevos libros en la base de datos, con información como título, autor, género y estado (disponible/no disponible).
- Actualizar información de un libro.
- Eliminar libros solo si no están prestados.
- Permitir la búsqueda de libros por título, autor o género, usando filtros dinámicos.

Gestión de Préstamos

- Registrar préstamos para un usuario (máximo 3 libros simultáneos para usuarios regulares; sin límite para administradores).
- Permitir la devolución de libros, actualizando su estado a "disponible".
- Registrar automáticamente préstamos express (devolución en menos de 7 días) en una tabla adicional.
- Validar que un libro solo pueda ser prestado si está disponible.

Generación de Reportes

- Mostrar:
 - Libros más prestados en los últimos 6 meses.
 - Usuarios con más préstamos realizados en los últimos 3 meses.
 - Porcentaje de libros por género.
- Listar todos los libros que no se han prestado en el último año.
- Mostrar usuarios con reservas pendientes o vencidas.

3. Requisitos Técnicos

Base de Datos Relacional

- Crear una base de datos llamada biblioteca con al menos las siguientes tablas:
 - usuarios (gestión de usuarios con roles).
 - libros (gestión de libros y disponibilidad).
 - prestamos (gestión de préstamos).
 - prestamos_express (registro de préstamos devueltos en menos de 7 días).
- Implementar triggers para actualizar el estado del libro automáticamente al registrar devoluciones.
- Usar consultas SQL avanzadas para reportes.

Estructura del Código

- Dividir el proyecto en paquetes, como:
 - modelo: Clases que representan entidades (Usuario, Libro, Prestamo).
 - dao: Clases para manejar el acceso a la base de datos.
 - servicio: Lógica de negocio, como validaciones y reglas.
 - util: Herramientas auxiliares, como la configuración de conexión JDBC.
 - main: Clase principal para la ejecución.
- Uso obligatorio de patrones de diseño:
 - DAO (Data Access Object) para separar la lógica de acceso a datos.
 - Observer para notificar reservas pendientes cuando un libro devuelto esté disponible.

Configuración Avanzada

- Crear un archivo de configuración (config.properties) para almacenar:
 - Datos de conexión (URL, usuario, contraseña).
 - Configuración de idioma (español/inglés).
 - Límite de transacciones y niveles de aislamiento.
- Incluir validaciones en las entradas del usuario para evitar datos inválidos.

Seguridad

- Implementar consultas parametrizadas para prevenir inyección SQL.
- Cifrar las contraseñas de los usuarios (SHA-256 o similar).
- Registrar cada operación realizada en un archivo de logs (log.txt).

4. Requisitos de Entrega

Documentación

- Diagrama Entidad-Relación de la base de datos.
- Diagrama UML representando la arquitectura del proyecto.
- Informe técnico explicando las decisiones de diseño y cómo se abordaron los desafíos.

Código

- Subir el proyecto a **Aula Virtual**, incluyendo:
 - Código fuente completo.
 - Archivo README.md con instrucciones de instalación y uso.

Extras Opcionales (puntaje adicional)

- Implementar una interfaz gráfica.
- Crear un sistema de auditoría que registre quién realizó cada operación.

5. Plazo de Entrega

- **Duración del proyecto:** hasta el 4 de marzo.
- **Forma de entrega:** el proyecto tendrá entregables parciales (diseño de la base de datos, implementación básica de DAO, funcionalidad de préstamos) y una entrega final.
- **Fechas de entrega en Aula Virtual:**
 - **Diseño de la base de datos: 30 de enero (hasta las 16:15h).**
 - El 30 de enero en clase se hará una presentación de 2 minutos/alumno.
 - **Implementación básica de DAO: 13 de febrero** (no hay que realizar ninguna presentación en clase).
 - **Funcionalidad de préstamos: 27 de febrero** (no hay que realizar ninguna presentación en clase). .
 - **Entrega Final:** 4 de marzo.
 - **Presentación en Clase:** jueves 6 de marzo

6. Criterios de Evaluación

1. Funcionalidad completa de cada módulo.
2. Correcta implementación de patrones de diseño.
3. Uso adecuado de consultas SQL avanzadas.
4. Robustez del sistema frente a errores o datos inválidos.
5. Documentación clara y bien organizada.
6. Comentarios personalizados en el código explicando qué hace cada sección. Estos comentarios deben alinearse con el informe técnico.

La nota de este Proyecto será la nota de la Evaluación 2.

La nota final del Módulo será = 10%(Nota Evaluación 1) + 90%(Evaluación 2).