Logotipo

El contenido generado por IA puede ser incorrecto.

Universidad Fidélitas, Costa Rica

Lenguaje de Base de Datos

Sistema de Gestión para una Biblioteca

**Profesor**

Leitón Jiménez Randall Alonso

**Estudiantes**

Aguilar Aguilar José Daniel

Salas Cerdas Darling Dayana

**II Avance**

I cuatrimestre

18 de marzo de 2025

**Objetivo General**

Desarrollar un sistema de gestión para una biblioteca que ayude a automatizar el control de libros como préstamos, devoluciones y registros esenciales como tipo de usuarios, categorías y autores.

**Objetivos Específicos**

* Desarrollar una interfaz en Java que le permita a los usuarios poder reservar libros e interactuar con el sistema.
* Crear una base de datos por medio de Oracle 19c para almacenar de manera segura y eficiente la información de la biblioteca.
* Registrar reportes e información sobre libros, usuarios, autores y categorías.

**Introducción**

Hoy en día, la automatización de procesos se ha vuelto una exigencia esencial para la optimización del tiempo y la optimización en la administración de la información. En este escenario, la creación de un sistema de administración para una biblioteca constituye una respuesta eficaz para simplificar la gestión de libros, que abarca los procedimientos de préstamos, devoluciones y la estructuración de los registros de usuarios, autores y categorías.

El propósito principal de este proyecto es diseñar e instaurar un sistema que emplee Java para la interfaz de usuario y Oracle 19c como administrador de bases de datos. Java ha sido elegido por su compatibilidad con Oracle, su habilidad para escalar y su enfoque en objetos, lo que simplifica el mantenimiento del sistema. En Oracle 19c, la base de datos asegurará un almacenamiento seguro y eficaz de los datos de la biblioteca. La implementación de este sistema potenciará la experiencia de los usuarios al ofrecer una plataforma fácil de usar para la reserva de libros y la administración de los registros de la biblioteca. Además, se producirán informes exhaustivos acerca de libros, usuarios y categorías, lo que favorecerá la toma de decisiones y el perfeccionamiento constante de la administración de bibliotecas.

**Justificación de la Interfaz**

Para este proyecto se tomó la decisión de trabajar con Java para el desarrollo de la interfaz esto con varios motivos, entre ellos, uno de los más importante sería lo compatible que es con Oracle 19c. No obstante, también se consideró este lenguaje de programación debido a la capacidad que tiene y lo escalable que puede llegar a ser.

La idea de utilizar Java se debe al conocimiento adquirido durante los cuatrimestres de carrera, pero sobre todo con la intención de llegar a profundizar en las herramientas de aprendizaje. Además, se sabe perfectamente que Java es un Lenguaje de Programación en el cual se pueden llegar a desarrollar diferentes sistemas con la posibilidad de ejecutarlos en diferentes plataformas.

Otro punto favorable, o en este caso las ventajas que se quieren destacar en esta decisión es el hecho de que Java está orientada a objetos, pero ¿qué quiere decir esto? Bueno, esto significa que permite estructurar el código de una manera organizada y eficiente donde los desarrolladores podrán darle mantenimiento al sistema en caso de ser necesario. Otro aspecto importante es que al momento de investigar se pueden encontrar grandes fuentes de información o documentación sobre este lenguaje que van desde revistas hasta videos lo cual facilitará el proceso de desarrollo cuando comiencen a surgir dudas o inquietudes referente al sistema de gestión que se desarrollará para una biblioteca.

A pesar de las ventajas mencionadas el grupo de este proyecto tiene claro que se podrían presentar algunos retos entre los cuales podría destacarse el conocimiento de los miembros. Posiblemente algunos van a estar más adelantados que otros o quizá entiendan de manera más sencilla Oracle 19c, mientras que los otros tendrán que dedicar más tiempo de estudio para poder alcanzar o nivelar esos conocimientos. No obstante, siendo este uno de los posibles retos a enfrentarse, cada uno de los integrantes tiene claro que esto es un equipo en el cual se van a apoyar y ayudar en todo lo que sea necesario no solo para avanzar con el desarrollo del sistema y la base de datos, sino también para cumplir con cada uno de los objetivos propuestos de manera rápida y eficiente de acuerdo con el cronograma propuesto.

**Desarrollo**

El desarrollo del Sistema de Gestión para una Biblioteca se realizará siguiendo un enfoque que permita la flexibilidad, la adaptación y sobre todo que logre ser eficiente y seguro siempre buscando alinearse a las necesidades del usuario, en este caso busca basarse y seguir por completo los requerimientos funcionales para poder cumplir con los objetivos establecidos. Gracias a este enfoque se considera que tanto el grupo de desarrolladores como los usuarios van a establecer una conexión que les permitirá desarrollar un sistema no solo funcional, sino que también eficiente.

En cada proyecto que se va a desarrollar existen diferentes etapas o fases que se deben seguir para poder cumplir totalmente con el objetivo, por lo tanto, en este también hay una serie de pasos que se deben seguir según el cronograma propuesto, entre ellos:

* **Planificar y Analizar Requisitos del Usuario**

El primer paso que un desarrollador debe seguir es preguntarle al cliente cuáles son las necesidades del proyecto o bien deducir con base en la conversación establecida los requerimientos funcionales que desea para su sistemas. Con base en esto los desarrolladores van a recopilar toda la información necesaria o en este caso clave para poder ejecutar el sistema de manera eficiente.

Tomando en cuenta lo anterior es importante destacar o recordar que para este sistema es necesario elaborar o desarrollar un registro y la gestión de libros, donde además de esto se puedan administrar usuarios como por ejemplo los profesores, los estudiantes y los funcionarios. También se busca generar dentro del mismo sistema la gestión de libros prestados y las respectivas devoluciones para tener una base de datos organizada y segura. Además la idea propuesta es crear un reporte sobre los libros disponibles, sus respectivos autores y las categorías que existen.

* **Diseñar un Modelo de Base de Datos**

Uno de los requisitos de este proyecto es generar un modelo de entidad relación por medio de un esquema, por lo tanto la base de datos se va a desarrollar por medio de Oracle 19c. Es importante mencionar que este modelo o este esquema incluirá una cierta cantidad de entidades principales como usuario, libro, autor, categoría, préstamo, reserva y reporte. Para cada una de estas tablas es completamente necesario establecer una llave primaria por medio de un atributo para poder representarla.

Para poder comprender de manera más sencilla estas entidades y sus respectivos atributos se puede decir que usuario es la tabla que va a contener información sobre los usuarios por ejemplo el nombre, el correo, número de teléfonos y si es administrador o usuario. Además se establece como primary key un id.

Por otra parte la tabla o entidad libro va a representar los libros disponibles en esa biblioteca, de igual manera contará con un identificador único, pero además tomará el id de otras entidades como foreign key, en este caso sería la relación que existe entre el libro y el autor y la relación entre el libro y la categoría. También tendrá un atributo llamado disponible como indicador de si se encuentra o no en la biblioteca.

Abarcando otras entidades se puede decir que autor va a almacenar los datos de los autores y como cada una de las otras tablas tendrá un identificador único (id\_autor). En cuanto a la entidad categoría va a ser quien define la clasificación de cada uno de los libros según su género y su PK será (id\_categoria).

Además de esas entidades existirá la entidad préstamos que se encargará de almacenar el registro de los libros que se le han prestado a los usuarios. Para ello se va a establecer un PK llamado “id\_prestamo” y también tendrá dos llaves foráneas para hacer relación con el usuario y el libro. Adicionalmente se van a establecer dos atributos más para indicar la fecha en que se prestó y la fecha en que se devolvió.

Por otra parte la entidad reserva permitirá a los usuarios reservar libros antes de solicitar el préstamo, es decir; tendrá la oportunidad de hacer una reserva, para ello se creará un PK llamado “id\_reserva” y se hará uso de dos llaves foráneas con el fin de establecer una relación entre el libro, el usuario y la reserva. Además se va a agregar otro atributo llamado “fecha\_reserva” que como su nombre lo indica registrará la fecha en que se realizó la reserva.

Finalmente, la entidad reporte va a generar todos los informes sobre la biblioteca. Así como las demás contará con un identificador único que en este caso será “id\_reporte”, a la tabla se le agregarán también otros atributos como por ejemplo el tipo de reporte que en este caso podrían ser el historial de préstamos, las categorías que más se han consultado y uno de los más importantes; los libros disponible. El otro atributo sería “fecha\_generacion” que se utilizará con el fin de tener un registro sobre cuándo se generó ese reporte.

* **Crear la Base de Datos en Oracle 19c**

Como se mencionaba anteriormente, la base de datos se va a crear por medio de Oracle 19c con la intención de poner en práctica las habilidades desarrolladas durante el curso lectivo. En esta base se van a crear tablas de acuerdo con las entidad principales establecidas en el punto anterior, además se realizarán relaciones de acuerdo con el modelo de entidad relación esto con el fin de poder asegurar un sistema eficiente. Para poder cumplir con este punto se deben crear o establecer llaves primaria (PK) y foráneas (FK). Además de implementar constrainsts y otras funciones importantes como crear procedimientos y consultas, entre otros.

* **Desarrollar el Sistema en Java**

Para poder llevar a cabo este proyecto es importante hacer uso de otra herramienta, en el caso de este grupo, se optó por trabajar con Java creando el proyecto en NetBeans, donde se va a configurar o establecer una conexión con la base de datos creada en Oracle 19c. En esta misma plataforma se va a desarrollar la implementación de la gestión de libros, tanto de los registros como de las consultas de los libros disponibles y todo el registro de los libros prestados y devueltos. En resumen, se podría decir que Java es una herramienta clave en el desarrollo de este Sistema ya que permitirá realizar distintas funciones y almacenar la lógica también.

* **Realizar Pruebas**

Finalmente, con el objetivo de asegurarse de que todo esté funcionando correctamente, los integrantes del grupo deberán realizar pruebas, estas no deben ser complejas, basta con demostrar e intentar insertar usuarios, registrar libros o incluso consultar las categorías o autores, entre otras otras pruebas de este tipo que se puedan realizar. A este tipo de pruebas se le conocen como pruebas unitarias ya que el usuario o el desarrollador va probando una a una estas funciones. No obstante, además de estas existen las pruebas de integración, las cuales buscan probar que exista una comunicación segura y eficiente entre programas utilizados. En este caso, los miembros realizarán pruebas para asegurarse de que exista una conexión entre Java y la base de datos que van a desarrollar por medio de Oracle 19c, básicamente lo único que tienen que verificar es que cada uno de los datos se registren de manera correcta y que cuando se realice algún tipo de consulta se vea reflejada la información que claramente se está esperando.

Otro aspecto importante que se ha considerado como equipo es la idea de ir documentando todo el código que se escriba en NetBeans con la intención de dejar un registro en caso de que alguien más necesite acceder. Además se considera que esta es una buena práctica que debe realizarse siempre que sea posible.

**Requerimientos Funcionales**

1. ***Gestión de Usuarios*:**
   * Registro, modificación y eliminación de usuarios.
   * Asignación de roles (administrador, usuario regular).
   * Consulta de información de usuarios.
2. ***Gestión de Libros*:**
   * Registro, modificación y eliminación de libros.
   * Clasificación por categorías y autores.
   * Consulta de disponibilidad de libros.
3. ***Préstamos y Devoluciones*:**
   * Registro de préstamos de libros.
   * Control de fechas de devolución.
   * Notificaciones de vencimiento de préstamo.
4. ***Reservas*:**
   * Permitir a los usuarios realizar reservas de libros.
   * Consulta y cancelación de reservas.
   * Notificación cuando un libro reservado está disponible.
5. ***Gestión de Reportes*:**
   * Generación de reportes sobre libros disponibles y prestados.
   * Informes sobre usuarios y su historial de préstamos.
   * Reportes de categorías y autores más consultados.
6. ***Seguridad y Control de Acceso*:**
   * Inicio de sesión con autenticación.
   * Control de acceso según el rol del usuario.
   * Registro de actividad de los usuarios.
7. ***Interfaz de Usuario:***
   * Desarrollo en Java con una interfaz amigable e intuitiva.
   * Fácil navegación entre las opciones del sistema.
   * Compatibilidad con distintos dispositivos.

**Requerimientos No Funcionales**

#### 1. **Seguridad**

* Implementación de autenticación de usuarios con credenciales encriptadas.
* Control de acceso basado en roles (administrador, usuario regular).
* Registro y auditoría de actividades de los usuarios en el sistema.
* Protección contra inyecciones SQL y ataques de seguridad en la base de datos.
* Respaldo y recuperación de la base de datos para evitar pérdida de información.

#### 2. **Rendimiento**

* La base de datos debe ser capaz de gestionar al menos 100.000 registros sin afectar el rendimiento.
* El tiempo de respuesta para la consulta de disponibilidad de libros debe ser menor a 3 segundos.
* Los reportes deben generarse en menos de 5 segundos.
* Capacidad para manejar múltiples sesiones concurrentes sin afectar el rendimiento.

#### **3. Usabilidad**

* La interfaz debe ser intuitiva y fácil de usar tanto para administradores como para usuarios regulares.
* El sistema debe proporcionar mensajes de error claros y comprensibles.
* Diseño de interfaz amigable y responsive para facilitar el uso en diferentes dispositivos.

#### **4. Disponibilidad**

* El sistema debe estar disponible al menos el 99% del tiempo.
* Implementación de un sistema de respaldo y recuperación ante fallos.
* El sistema debe soportar mantenimiento programado sin afectar la disponibilidad de las funciones esenciales.

#### **5. Escalabilidad**

* El sistema debe poder escalarse para soportar un mayor número de usuarios sin necesidad de modificaciones estructurales drásticas.
* Capacidad de expansión para futuras funcionalidades como integración con catálogos externos o acceso remoto a los libros digitales.

#### **6. Mantenimiento**

* El código debe estar documentado para facilitar su mantenimiento y futuras mejoras.
* Uso de buenas prácticas de programación orientada a objetos en Java.
* La base de datos debe permitir actualizaciones sin afectar la operatividad del sistema.

#### **7. Compatibilidad**

* Compatible con sistemas operativos Windows, MacOS y Linux.
* Integración con Oracle 19c como gestor de base de datos.
* Soporte para diferentes navegadores en caso de futuras versiones web del sistema.

#### **8. Confiabilidad**

* El sistema debe garantizar la integridad de los datos, evitando duplicaciones o pérdida de información.
* Se deben realizar pruebas de estrés y carga para asegurar la estabilidad del sistema en condiciones de alto tráfico.

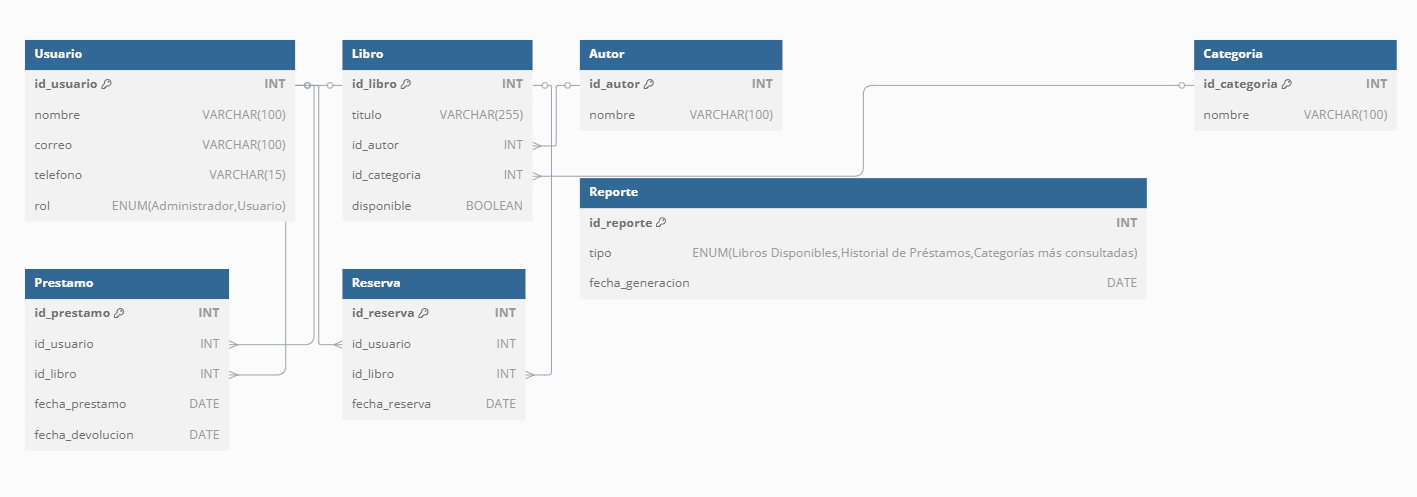
#### **9. Legalidad**

* Cumplir con normativas de protección de datos y privacidad aplicables en Costa Rica.
* Garantizar la correcta gestión de información personal de los usuarios conforme a las leyes de seguridad informática.

**Cronograma**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actividad** | **14 de enero 2025-15 de abril 2025** | | |
| **28 enero-17 febrero**  **Semana 3 a 6** | **19 febrero-17 marzo**  **Semana 6 a 10** | **19 marzo-14 abril**  **Semana 10 a 14** |
| Marco conceptual |  |  |  |
| Requerimientos (funcionales no funcionales |  |  |  |
| Objetivos generales y específicos |  |  |  |
| Cronograma |  |  |  |
| ER |  |  |  |
| Requerimientos |  |  |  |
| Propuesta de interfaz |  |  |  |
| Definición objetos PL/SQL (diccionario de datos) |  |  |  |
| Conexión a base de datos (código o capturas de pantalla) |  |  |  |
| Avance de interfaz (capturas de pantalla) |  |  |  |
| Definición objetos PL/SQL (diccionario de datos) |  |  |  |
| Documento final |  |  |  |
| Aplicación |  |  |  |

**Esquema Entidad Relación:**

****

**Scripts y Esquema de la Base de Datos**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Procedimientos Almacenados**

Para los procedimientos almacenados, como en el segundo avance, la primera instrucción indicaba que no se podían realizar consultas directas como Select \* From o inclusive Insert Into, se decidió hacer una investigación sobre un tema no visto en clase para poder implementar los procedimientos con un código más profundo. En este caso, gracias a lo aprendido en otros cursos y a herramientas de Internet o sitios web como “IBM” se pudo comprender que una manera eficiente, clara y sencilla de hacer consultas cumpliendo esta regla eran los cursores. En la biografía se puede ver la página que se estudió para poder realizar los procedimientos almacenados con base en los requisitos de este entregable. También se vieron video con el fin de comprender mejor el tema, el mismo está ubicado al final del entregable.

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

PROCEDIMIENTOS ALMACENADOS

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

--1. AGREGAR USUARIO

CREATE OR REPLACE PROCEDURE SP\_AGREGAR\_USUARIO(

SP\_NOMBRE IN VARCHAR2,

SP\_CORREO IN VARCHAR2,

SP\_TELEFONO IN VARCHAR2,

SP\_ROL IN VARCHAR2

)

AS

CONTADOR NUMBER := 1;

VSUM NUMBER := 0;

BEGIN

INSERT INTO USUARIOS(NOMBRE, CORREO, TELEFONO, ROL)

VALUES(SP\_NOMBRE, SP\_CORREO, SP\_TELEFONO, SP\_ROL);

DBMS\_OUTPUT.PUT\_LINE('Usuario agregado: ' || SP\_NOMBRE);

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: No se pudo agregar el usuario. Por favor, intente de nuevo');

END;

SET SERVEROUTPUT ON;

EXEC SP\_AGREGAR\_USUARIO('Carlos López', 'carlos.lopez@email.com', '5552345678', 'USUARIO');

--2. ELIMINAR USUARIO POR ID

CREATE OR REPLACE PROCEDURE SP\_ELIMINAR\_USUARIO(

SP\_ID\_USUARIO IN NUMBER

)

AS

VCOUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO VCOUNT

FROM USUARIOS

WHERE ID\_USUARIO = SP\_ID\_USUARIO;

IF VCOUNT > 0 THEN

DELETE FROM USUARIOS

WHERE ID\_USUARIO = SP\_ID\_USUARIO;

DBMS\_OUTPUT.PUT\_LINE('Usuario eliminado correctamente: ' || SP\_ID\_USUARIO);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Error: Usuario no encontrado.');

END IF;

END;

EXEC SP\_ELIMINAR\_USUARIO(1);

--3. ACTUALIZAR DATOS USUARIO

CREATE OR REPLACE PROCEDURE SP\_ACTUALIZAR\_USUARIO(

SP\_ID IN NUMBER,

SP\_NOMBRE IN VARCHAR2,

SP\_CORREO IN VARCHAR2,

SP\_TELEFONO IN VARCHAR2,

SP\_ROL IN VARCHAR2

)

AS

VCOUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO VCOUNT

FROM USUARIOS

WHERE ID\_USUARIO = SP\_ID;

IF VCOUNT > 0 THEN

UPDATE USUARIOS

SET NOMBRE = SP\_NOMBRE,

CORREO = SP\_CORREO,

TELEFONO = SP\_TELEFONO,

ROL = SP\_ROL

WHERE ID\_USUARIO = SP\_ID;

DBMS\_OUTPUT.PUT\_LINE('Datos del usuario actualizados correctamente: ' || SP\_ID);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Error: Usuario no encontrado.');

END IF;

END;

BEGIN

SP\_ACTUALIZAR\_USUARIO(10, 'Sofía', 'sofia.fernandez@email.com', '6677889999', 'USUARIO');

END;

-- 4. AGREGAR NUEVO LIBRO

CREATE OR REPLACE PROCEDURE SP\_AGREGAR\_LIBRO(

SP\_ID IN NUMBER,

SP\_TITULO IN VARCHAR2,

SP\_AUTOR IN VARCHAR2,

SP\_CATEGORIA IN VARCHAR2,

SP\_DISPONIBLE IN CHAR

)

AS

VCOUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO VCOUNT

FROM LIBROS

WHERE ID\_LIBRO = SP\_ID;

IF VCOUNT = 0 THEN

INSERT INTO LIBROS(ID\_LIBRO, TITULO, AUTOR, CATEGORIA, DISPONIBLE)

VALUES(SP\_ID, SP\_TITULO, SP\_AUTOR, SP\_CATEGORIA, SP\_DISPONIBLE);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Libro agregado correctamente: ' || SP\_TITULO);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Error: El libro con ID ' || SP\_ID || ' ya existe.');

END IF;

END;

EXEC SP\_AGREGAR\_LIBRO(1, 'El Gran Gatsby', 'F. Scott Fitzgerald', 'Ficción', 'S');

EXEC SP\_AGREGAR\_LIBRO(125, 'El conde de Montecristo', 'Alexandre Dumas', 'Aventura', 'S');

--5. ACTUALIZAR LIBRO

CREATE OR REPLACE PROCEDURE SP\_ACTUALIZAR\_LIBRO(

SP\_ID IN NUMBER,

SP\_TITULO IN VARCHAR2,

SP\_AUTOR IN VARCHAR2,

SP\_CATEGORIA IN VARCHAR2,

SP\_DISPONIBLE IN CHAR

)

AS

VCOUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO VCOUNT

FROM LIBROS

WHERE ID\_LIBRO = SP\_ID;

IF VCOUNT > 0 THEN

UPDATE LIBROS

SET TITULO = SP\_TITULO,

AUTOR = SP\_AUTOR,

CATEGORIA = SP\_CATEGORIA,

DISPONIBLE = SP\_DISPONIBLE

WHERE ID\_LIBRO = SP\_ID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Libro actualizado correctamente: ' || SP\_TITULO);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Error: El libro con ID ' || SP\_ID || ' no existe.');

END IF;

END;

EXEC SP\_ACTUALIZAR\_LIBRO(122, 'Alicia en el país de las maravillas', 'Lewis Carroll', 'Fantasía', 'N');

--cuando no existe el libro

EXEC SP\_ACTUALIZAR\_LIBRO(127, 'Alicia a través del espejo', 'Lewis Carroll', 'Fantasía', 'N');

-- 6. ELIMINAR LIBRO POR ID

CREATE OR REPLACE PROCEDURE SP\_ELIMINAR\_LIBRO(

SP\_ID\_LIBRO IN NUMBER

)

AS

VCOUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO VCOUNT

FROM LIBROS

WHERE ID\_LIBRO = SP\_ID\_LIBRO;

IF VCOUNT > 0 THEN

DELETE FROM LIBROS

WHERE ID\_LIBRO = SP\_ID\_LIBRO;

DBMS\_OUTPUT.PUT\_LINE('El libro ha sido eliminado correctamente: ' || SP\_ID\_LIBRO);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Error: Libro no encontrado.');

END IF;

END;

EXEC SP\_ELIMINAR\_LIBRO(115);

EXEC SP\_ELIMINAR\_LIBRO(190); -- No existe

--7. FECHA DE DEVOLUCION ACTUALIZADA

CREATE OR REPLACE PROCEDURE SP\_ACTUALIZA\_FECHA\_DEVOLUCION(

SP\_ID\_PRESTAMO IN NUMBER,

SP\_FECHA\_ACTUALIZADA IN DATE

)

AS

BEGIN

UPDATE PRESTAMOS

SET FECHA\_DEVOLUCION = SP\_FECHA\_ACTUALIZADA

WHERE ID\_PRESTAMO = SP\_ID\_PRESTAMO;

END;

--- Probamos:

INSERT INTO PRESTAMOS (ID\_PRESTAMO, FECHA\_DEVOLUCION)

VALUES (190, TO\_DATE('2025-04-01', 'YYYY-MM-DD'));

EXEC SP\_ACTUALIZA\_FECHA\_DEVOLUCION(190, TO\_DATE('2025-05-01', 'YYYY-MM-DD'));

--8. RESERVAR LIBRO

CREATE OR REPLACE PROCEDURE SP\_RESERVAR\_LIBRO(

SP\_ID\_RESERVA IN NUMBER,

SP\_ID\_USUARIO IN NUMBER,

SP\_ID\_LIBRO IN NUMBER,

SP\_FECHA\_RESERVACION IN DATE

)

AS

V\_COUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO V\_COUNT

FROM RESERVAS

WHERE ID\_USUARIO = SP\_ID\_USUARIO

AND ID\_LIBRO = SP\_ID\_LIBRO

AND FECHA\_RESERVA = SP\_FECHA\_RESERVACION;

IF V\_COUNT > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Lo siento. El libro ya se encuentra reservado.');

ELSE

INSERT INTO RESERVAS (ID\_RESERVA, ID\_USUARIO, ID\_LIBRO, FECHA\_RESERVA)

VALUES (SP\_ID\_RESERVA, SP\_ID\_USUARIO, SP\_ID\_LIBRO, SP\_FECHA\_RESERVACION);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Reserva realizada con éxito: ' || SP\_ID\_RESERVA);

END IF;

END;

INSERT INTO LIBROS (ID\_LIBRO, TITULO, AUTOR, CATEGORIA, DISPONIBLE)

VALUES (203, 'Sobreviviendo a las Sombras', 'Darling Salas', 'Poemario', 'N');

INTO USUARIOS (ID\_USUARIO, NOMBRE, APELLIDO, CORREO, TELEFONO, ROL)

VALUES (12, 'Miguel', 'Azofeifa', 'miguel.azofeifa@email.com', '9876543210', 'USUARIO')

EXEC SP\_RESERVAR\_LIBRO(1, 12, 203, TO\_DATE('2025-03-20', 'YYYY-MM-DD'));

-- 9. PRESTAMO LIBRO

CREATE OR REPLACE PROCEDURE SP\_PRESTAMO(

SP\_ID\_PRESTAMO IN NUMBER,

SP\_ID\_USUARIO IN NUMBER,

SP\_ID\_LIBRO IN NUMBER,

SP\_FECHA\_PRESTAMO IN DATE,

SP\_FECHA\_DEVOLUCION IN DATE

)

AS

RESERVADO NUMBER;

BEGIN

SELECT COUNT(\*) INTO RESERVADO

FROM RESERVAS

WHERE ID\_LIBRO = SP\_ID\_LIBRO

AND FECHA\_RESERVA = SP\_FECHA\_PRESTAMO;

IF RESERVADO > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Lo siento. El libro no está disponible para préstamo porque está reservado.');

RETURN;

END IF;

INSERT INTO PRESTAMOS (ID\_PRESTAMO, ID\_USUARIO, ID\_LIBRO, FECHA\_PRESTAMO, FECHA\_DEVOLUCION)

VALUES (SP\_ID\_PRESTAMO, SP\_ID\_USUARIO, SP\_ID\_LIBRO, SP\_FECHA\_PRESTAMO, SP\_FECHA\_DEVOLUCION);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Préstamo realizado con éxito: ' || SP\_ID\_PRESTAMO);

END;

EXEC SP\_PRESTAMO(1, 12, 203, TO\_DATE('2025-03-20', 'YYYY-MM-DD'), TO\_DATE('2025-04-20', 'YYYY-MM-DD'));

--10. ELIMINAR PRESTAMO LIBRO

CREATE OR REPLACE PROCEDURE SP\_ELIMINAR\_PRESTAMO(SP\_ID IN NUMBER)

AS

BEGIN

DELETE FROM PRESTAMOS

WHERE ID\_PRESTAMO = SP\_ID;

END;

--hacemos la prueba:

INSERT INTO LIBROS (ID\_LIBRO, TITULO, AUTOR)

VALUES (178, 'Lágrimas Sigilosas', 'Darling');

INSERT INTO PRESTAMOS (ID\_PRESTAMO, ID\_LIBRO, FECHA\_PRESTAMO)

VALUES (1005, 178, TO\_DATE('2025-03-18', 'YYYY-MM-DD'));

EXEC SP\_ELIMINAR\_PRESTAMO(1005);

--11. PRESTAMOS ACTIVOS DE UN SUUARIO

CREATE OR REPLACE PROCEDURE SP\_PRESTAMOS\_ACTIVOS (SP\_ID\_USUARIO IN NUMBER)

AS

CURSOR C\_PRESTAMOS IS

SELECT ID\_PRESTAMO

FROM PRESTAMOS

WHERE ID\_USUARIO = SP\_ID\_USUARIO

AND FECHA\_DEVOLUCION IS NULL;

BEGIN

FOR PRESTAMO IN C\_PRESTAMOS LOOP

DBMS\_OUTPUT.PUT\_LINE('El prestamo: ' || prestamo.ID\_PRESTAMO || ' se encuentra activo');

END LOOP;

END;

---probamos:

INSERT INTO PRESTAMOS (ID\_PRESTAMO, ID\_USUARIO, ID\_LIBRO, FECHA\_PRESTAMO, FECHA\_DEVOLUCION)

VALUES (1008, 14, 178, TO\_DATE('2025-03-18', 'YYYY-MM-DD'), NULL);

INSERT INTO PRESTAMOS (ID\_PRESTAMO, ID\_USUARIO, ID\_LIBRO, FECHA\_PRESTAMO, FECHA\_DEVOLUCION)

VALUES (1009, 14, 178, TO\_DATE('2025-03-18', 'YYYY-MM-DD'), NULL);

EXEC SP\_PRESTAMOS\_ACTIVOS(14);

--12. VERIFICAR DISPONIBILIDAD

CREATE OR REPLACE PROCEDURE SP\_DISPONIBILIDAD (

SP\_ID\_LIBRO IN NUMBER,

SP\_DISPONIBLE OUT CHAR

)

AS

CURSOR C\_LIBRO IS

SELECT DISPONIBLE

FROM LIBROS

WHERE ID\_LIBRO = SP\_ID\_LIBRO;

VDIS CHAR(1);

BEGIN

OPEN C\_LIBRO;

FETCH C\_LIBRO

INTO VDIS;

IF C\_LIBRO%FOUND THEN

SP\_DISPONIBLE := VDIS;

ELSE

SP\_DISPONIBLE := 'N';

END IF;

CLOSE C\_LIBRO;

END;

--PRUEBA:

DECLARE

VDISP CHAR(1);

BEGIN

SP\_DISPONIBILIDAD(SP\_ID\_LIBRO => 122, SP\_DISPONIBLE => VDISP);

DBMS\_OUTPUT.PUT\_LINE('La disponibilidad del libro es: ' || VDISP);

END;

/\*

La disponibilidad del libro es: N

\*/

--13. NOMBRE DE USUARIO POR ID

CREATE OR REPLACE PROCEDURE SP\_NOMBRE\_USUARIO(

SP\_ID\_USUARIO IN NUMBER,

SP\_NOMBRE OUT VARCHAR2

)

AS

CURSOR C\_USUARIO IS

SELECT NOMBRE

FROM USUARIOS

WHERE ID\_USUARIO = SP\_ID\_USUARIO;

VNOM VARCHAR2(50);

BEGIN

FOR USUARIO IN C\_USUARIO LOOP

SP\_NOMBRE := USUARIO.NOMBRE;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

SP\_NOMBRE := NULL;

END;

DECLARE

V\_NOMBRE VARCHAR2(50);

BEGIN

SP\_NOMBRE\_USUARIO(SP\_ID\_USUARIO => 14, SP\_NOMBRE => V\_NOMBRE);

DBMS\_OUTPUT.PUT\_LINE('El nombre del usuario es: ' || V\_NOMBRE);

END;

/\*

El nombre del usuario es: Andrés

\*/

--14. ELIMINAR RESERVA POR ID

CREATE OR REPLACE PROCEDURE SP\_ELIMINAR\_RESERVA (SP\_ID IN NUMBER)

AS

VCOUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO VCOUNT

FROM RESERVAS

WHERE ID\_RESERVA = SP\_ID;

IF VCOUNT > 0 THEN

DELETE FROM RESERVAS

WHERE ID\_RESERVA = SP\_ID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Reserva con ID ' || SP\_ID || ' eliminada exitosamente.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('La reserva con ID ' || SP\_ID || ' no existe.');

END IF;

END;

--- PRUEBA:

INSERT INTO RESERVAS (ID\_RESERVA, ID\_USUARIO, ID\_LIBRO, FECHA\_RESERVA)

VALUES (2005, 13, 110, TO\_DATE('2024-03-10', 'YYYY-MM-DD'));

EXEC SP\_ELIMINAR\_RESERVA(2005);

/\*

Reserva con ID 2005 eliminada exitosamente.

\*/

-- 15. LIBROS DISPONIBLES POR CATEGORIA

CREATE OR REPLACE PROCEDURE SP\_LIBRO\_DIS\_CATE(SP\_CATE IN VARCHAR2)

AS

BEGIN

FOR LIBRO IN (SELECT TITULO FROM LIBROS

WHERE CATEGORIA = SP\_CATE

AND DISPONIBLE = 'S')

LOOP

DBMS\_OUTPUT.PUT\_LINE('Libro disponible: ' || LIBRO.TITULO);

END LOOP;

END;

EXEC SP\_LIBRO\_DIS\_CATE('Novela');

/\*

Libro disponible: Cien años de soledad

Libro disponible: El último recuerdo que dejó tu muerte

\*/

--16. TOTAL LIBROS PRESTADOS

CREATE OR REPLACE PROCEDURE SP\_LIBROS\_PRESTADOS(SP\_TOTAL OUT NUMBER)

AS

CURSOR C\_PRESTAMOS IS

SELECT COUNT(\*) AS TOTAL

FROM PRESTAMOS

WHERE FECHA\_DEVOLUCION IS NULL;

BEGIN

OPEN C\_PRESTAMOS;

FETCH C\_PRESTAMOS INTO SP\_TOTAL;

CLOSE C\_PRESTAMOS;

END;

--PROBAMOS

DECLARE

VTOTAL NUMBER;

BEGIN

SP\_LIBROS\_PRESTADOS(VTOTAL);

DBMS\_OUTPUT.PUT\_LINE('Total de libros prestados: ' || V\_TOTAL);

END;

/\*

Total de libros prestados: 2

\*/

--17. ELIMINAR REPORTE POR ID

CREATE OR REPLACE PROCEDURE ELIMINAR\_REPORTE(SP\_ID IN NUMBER)

AS

VCOUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO VCOUNT FROM REPORTES WHERE ID\_REPORTE = SP\_ID;

IF VCOUNT > 0 THEN

DELETE FROM REPORTES WHERE ID\_REPORTE = SP\_ID;

DBMS\_OUTPUT.PUT\_LINE('Reporte con ID ' || SP\_ID || ' eliminado correctamente.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('No se encontró reporte con ID ' || SP\_ID || '.');

END IF;

END;

--- PRUEBA

INSERT INTO REPORTES (ID\_REPORTE, TIPO\_REPORTE, FECHA\_GENERACION)

VALUES (3004, 'Reporte de Préstamos', TO\_DATE('2024-03-18', 'YYYY-MM-DD'));

EXEC ELIMINAR\_REPORTE(3004);

/\*

Reporte con ID 3004 eliminado correctamente.

\*/

EXEC ELIMINAR\_REPORTE(3005);

/\*

No se encontró reporte con ID 3005.

\*/

--18. PRESTAMOS POR LIBRO

CREATE OR REPLACE PROCEDURE SP\_PRESTAMOS\_POR\_LIBRO(SP\_ID\_LIBRO IN NUMBER)

AS

CURSOR C\_PRESTAMOS IS

SELECT ID\_PRESTAMO, ID\_USUARIO, FECHA\_PRESTAMO, FECHA\_DEVOLUCION

FROM PRESTAMOS

WHERE ID\_LIBRO = SP\_ID\_LIBRO

AND FECHA\_DEVOLUCION IS NULL;

BEGIN

FOR PRESTAMO IN C\_PRESTAMOS LOOP

DBMS\_OUTPUT.PUT\_LINE('Préstamo ID: ' || PRESTAMO.ID\_PRESTAMO || ' - Usuario ID: ' || PRESTAMO.ID\_USUARIO || ' - Fecha de préstamo: ' || TO\_CHAR(PRESTAMO.FECHA\_PRESTAMO, 'DD-MM-YYYY'));

END LOOP;

END;

--- PRUEBA:

INSERT INTO PRESTAMOS (ID\_PRESTAMO, ID\_USUARIO, ID\_LIBRO, FECHA\_PRESTAMO, FECHA\_DEVOLUCION)

VALUES (1010, 10, 112, TO\_DATE('2024-03-01', 'YYYY-MM-DD'), NULL);

EXEC SP\_PRESTAMOS\_POR\_LIBRO(112);

/\*

Préstamo ID: 1010 - Usuario ID: 10 - Fecha de préstamo: 01-03-2024

\*/

--19. USUARIOS POR ROL

CREATE OR REPLACE PROCEDURE SP\_USER\_POR\_ROL(SP\_ROL IN VARCHAR2)

AS

BEGIN

FOR USUARIO IN (SELECT \* FROM USUARIOS WHERE ROL = SP\_ROL)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Usuario: ' || USUARIO.NOMBRE);

END LOOP;

END;

EXEC SP\_USER\_POR\_ROL('USUARIO');

/\*

Usuario: Luisa

Usuario: Pedro

Usuario: Laura

Usuario: Sofía

Usuario: Ricardo

Usuario: Sofía

Usuario: Donatien

Usuario: Miguel

Usuario: Martha

Usuario: Andrés

Usuario: Sara

Usuario: Katthy

Usuario: Carlos

Usuario: Mónica

\*/

-- 20. TODOS LOS LIBROS

CREATE OR REPLACE PROCEDURE SP\_LISTA\_LIBROS

AS

VCOUNT NUMBER := 1;

VTOTAL NUMBER;

BEGIN

SELECT COUNT(\*) INTO VTOTAL FROM LIBROS;

DBMS\_OUTPUT.PUT\_LINE('Total de libros disponibles: ' || VTOTAL || ' ');

FOR LIBRO IN(SELECT \* FROM LIBROS) LOOP

DBMS\_OUTPUT.PUT\_LINE('Libro ' || VCOUNT || ': ' || LIBRO.TITULO || ' - ' || LIBRO.AUTOR);

VCOUNT := VCOUNT + 1;

END LOOP;

END;

EXEC SP\_LISTA\_LIBROS;

/\*

Total de libros disponibles: 28

Libro 1: La sombra del viento - Carlos Ruiz Zafón

Libro 2: 1984 - George Orwell

Libro 3: Fahrenheit 451 - Ray Bradbury

Libro 4: El nombre de la rosa - Umberto Eco

Libro 5: La tregua - Mario Benedetti

Libro 6: Crónica de una muerte anunciada - Gabriel García Márquez

Libro 7: Rayuela - Julio Cortázar

Libro 8: Cumbres Borrascosas - Emily Brontë

Libro 9: El Gran Gatsby - F. Scott Fitzgerald

Libro 10: Orgullo y prejuicio - Jane Austen

Libro 11: Sobreviviendo a las Sombras - Darling Salas

Libro 12: Los hermanos Karamazov - Fiódor Dostoyevski

Libro 13: Ulises - James Joyce

Libro 14: El retrato de Dorian Gray - Oscar Wilde

Libro 15: El alquimista - Paulo Coelho

Libro 16: Los miserables - Victor Hugo

Libro 17: El viento en los sauces - Kenneth Grahame

Libro 18: Alicia en el país de las maravillas - Lewis Carroll

Libro 19: La metamorfosis - Franz Kafka

Libro 20: Moby Dick - Herman Melville

Libro 21: El conde de Montecristo - Alexandre Dumas

Libro 22: El Gran Gatsby - F. Scott Fitzgerald

Libro 23: Título del libro - Autor del libro

Libro 24: Lágrimas Sigilosas - Darling

Libro 25: Cien años de soledad - Gabriel García Márquez

Libro 26: El principito - Antoine de Saint-Exupéry

Libro 27: Don Quijote de la Mancha - Miguel de Cervantes

Libro 28: El último recuerdo que dejó tu muerte - Darling Salas

\*/

**Funciones**

**Bibliografía**

De Roer, D. D., & De Roer, D. D. (2022, 12 junio). Conectar Java a una base de datos Oracle. *Disco Duro de Roer -*. <https://www.discoduroderoer.es/conectar-java-a-una-base-de-datos-oracle/>

*Qué es un diagrama entidad-relación*. (s. f.). Lucidchart. <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>

*Oracle Database 19c*. (s. f.). <https://info.quistor.com/migraci%C3%B3n/>

Equipo editorial, Etecé. (2024, 24 diciembre). *Base de datos - Qué es, tipos y ejemplos*. Concepto. <https://concepto.de/base-de-datos/>

Solutions, V., & Jain, A. (2023, 2 enero). *Qué son los requisitos funcionales: ejemplos, definición, guía completa*. Visure Solutions. <https://visuresolutions.com/es/blog/functional-requirements/#:~:text=Un%20requisito%20funcional%20es%20una,caracter%C3%ADsticas%20que%20el%20usuario%20detecta>.

IBM. (n.d.). *Cursors in procedures*. IBM Documentation. <https://www.ibm.com/docs/es/db2/11.5?topic=procedures-cursors>

Villavicencio, G. (2022, Agosto 13). Procedimientos almacenados uso de cursor en SQLServer Parte 2[Video]. YouTube. <https://www.youtube.com/watch?v=Lztys1PcMV4>