



EVALUACIÓN CONTINUA/PROCESUAL FINAL

HITO 5

Estudiante:

Yahir Adrian Rios Quispe
Jhon Deyvid Ajoururo Condori
Cristian Mamani Velasquez
Rodnny Giovani Altamirano Flores

Docente: William Roddy Barra Paredes

Materia: Base de datos 1

Carrera: INGENIERÍA DE SISTEMAS

Paralelo: BDA (1)

fecha: 04/12/2023

1. Introducción

El presente proyecto tiene como objetivo el desarrollo de una aplicación de gestión integral para una biblioteca, con el propósito de optimizar y facilitar el manejo de información relacionada con libros, autores, usuarios, préstamos, categorías, empleados, eventos y compras. La aplicación se concibe como una herramienta centralizada que permitirá a los usuarios gestionar de manera eficiente los recursos bibliotecarios, mejorar el control de préstamos, y proporcionar una visión detallada de la actividad y eventos relacionados con la biblioteca.

En el contexto del proyecto, se ha diseñado una base de datos relacional que refleja la estructura lógica de la información necesaria para el funcionamiento de la aplicación. La base de datos abarca tablas que representan entidades clave, como libros, autores, editoriales, usuarios, préstamos, categorías, empleados, eventos, proveedores y compras. La interrelación entre estas tablas se establece a través de claves primarias y foráneas, garantizando la integridad referencial y facilitando consultas complejas.

El sistema de gestión de biblioteca se concibe como una herramienta versátil que permitirá a los usuarios, tanto empleados como clientes, acceder a información actualizada, realizar búsquedas eficientes, gestionar préstamos, participar en eventos y mantener un control detallado de las compras realizadas a proveedores. La implementación de esta aplicación busca mejorar la experiencia tanto para el personal de la biblioteca como para los usuarios finales, contribuyendo así a la eficacia y eficiencia en la administración y uso de los recursos bibliotecarios.

Objetivos.

- Mostrar el manejo y gestión de bases de datos relacionales.
- Consultas relacionales INNER JOIN.
- Mostrar el uso y creación consultas SQL, manejo de funciones de la base de Datos.
- Manejo de funciones UDF (Funciones definidas por el usuario)

Mockup:

Pagina de búsqueda

Biblioteca

BúsquedaDisponibilidadAdministración

Nombre de Libro

Nombre de Autor

Categorías

Editoriales

Q Ingresar Datos

Buscar

Nombre Del libro	Autor	Cantidad	Categoría	Editorial	Agotado
Psicología Oscura	Rodolfo T.	3	Psicología	Founder & CEO	<input type="checkbox"/>
Hábitos Atómicos	Marcos	2	Superacion Personal	Tuttofare	<input type="checkbox"/>
Física II	Mariah M.	0	Matemáticas	Better Half	<input checked="" type="checkbox"/>

Imagen creada en <https://balsamiq.com/>

Página de disponibilidad

Biblioteca

BúsquedaDisponibilidadAdministración

Nombre Libro ▼
Autor

Q Ingrese Datos

Buscar

Nombre del libro ▲	Cantidad ◆	Precio Venta	Precio Renta	Vendido ▼
Psicologia oscura	3	40	20	<input type="checkbox"/>
Habitos Atomicos	2	45	22	<input checked="" type="checkbox"/>
Fisica II	0	30	15	<input type="checkbox"/>

Imagen creada en <https://balsamiq.com/>

Página de administración

Biblioteca

BúsquedaDisponibilidadAdministración

Prestamos ▼
Empleados
Sueldo
Gastos
Ingresos

Q Ingrese Datos

Buscar

Nombre del libro ▲	Rentado a: ◆	Desde:	Hasta:
Psicologia oscura	Luis Perez	4/11	12/11
Habitos Atomicos	Adrian Torres	25/11	2/12

Imagen creada en <https://balsamiq.com/>

2. Análisis y diseño de la base de datos

a. Contexto de la Base de Datos: El nombre de la base de datos es **DB_Biblio_Gest** el cual es una base de datos que ayuda a la gestión de una biblioteca.

b. Entidades/tablas de sistema:

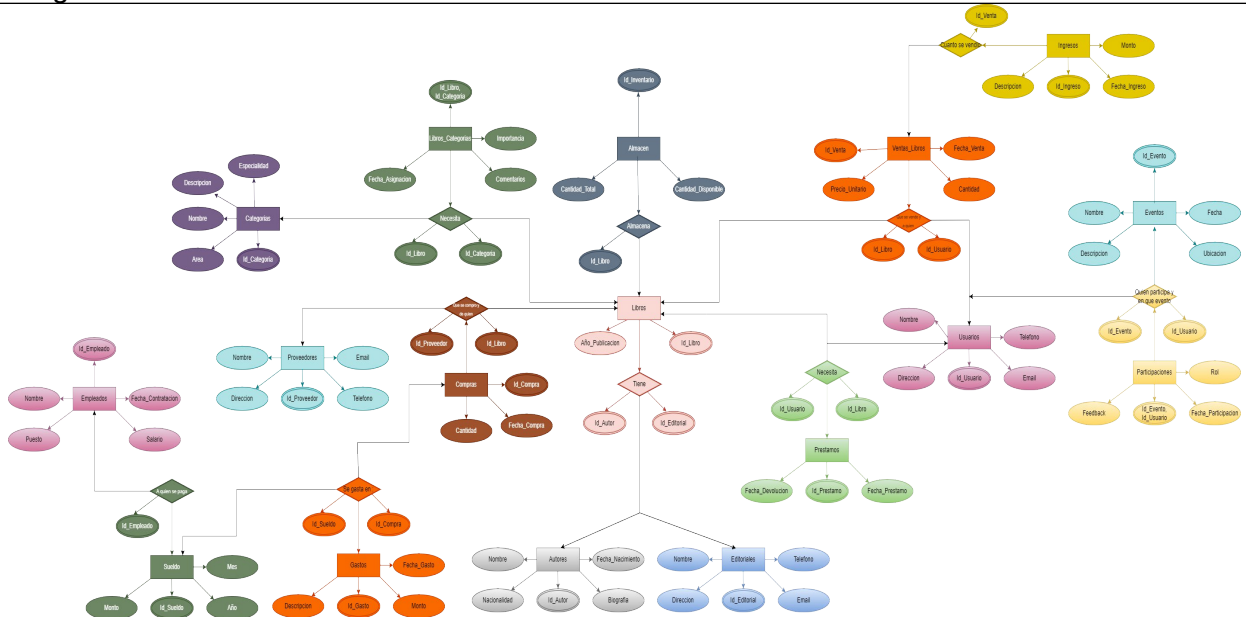
Tablas	Descripción
Libros	<p>Descripción: Almacena la información relacionada con cada libro en la biblioteca.</p> <p>Propósito: Permite gestionar detalles clave de los libros, como el título, el autor, la editorial, el año de publicación, y establece relaciones con las tablas de Autores y Editoriales mediante claves foráneas.</p>
Autores	<p>Descripción: Contiene información sobre los autores de los libros en la biblioteca.</p> <p>Propósito: Permite gestionar datos específicos de los autores, como su nombre, nacionalidad, fecha de nacimiento y biografía. La tabla se relaciona con la tabla de Libros mediante claves foráneas.</p>
Editoriales	<p>Descripción: Almacena datos sobre las editoriales de los libros en la biblioteca.</p> <p>Propósito: Facilita la gestión de información relacionada con las editoriales, incluyendo el nombre, dirección, teléfono y correo electrónico. La tabla se relaciona con la tabla de Libros mediante claves foráneas.</p>
Usuarios	<p>Descripción: Guarda información sobre los usuarios que interactúan con la biblioteca.</p> <p>Propósito: Permite gestionar detalles de los usuarios, como su nombre, dirección, teléfono y correo electrónico. Utilizada en la tabla de Prestamos para registrar quién realiza un préstamo.</p>
Prestamos	<p>Descripción: Registra la información relacionada con los préstamos de libros.</p> <p>Propósito: Permite realizar un seguimiento de los préstamos, incluyendo la fecha de préstamo, la fecha de devolución y las relaciones con los libros y usuarios mediante claves foráneas.</p>
Categorías	<p>Descripción: Contiene información sobre las categorías asignadas a los libros.</p> <p>Propósito: Facilita la organización y clasificación de los libros en la biblioteca mediante categorías específicas. Utilizada para establecer relaciones con la tabla de Libros Categorías.</p>
Libros_Categorías	<p>Descripción: Establece las relaciones entre los libros y las categorías.</p> <p>Propósito: Permite asignar múltiples categorías a un libro y viceversa. Además de las claves foráneas, incluye detalles adicionales como la fecha de asignación, comentarios y la importancia de la categoría para un libro específico.</p>
Empleados	<p>Descripción: Guarda información sobre los empleados de la biblioteca.</p> <p>Propósito: Facilita la gestión de datos relacionados con los empleados, incluyendo su nombre, puesto, fecha de contratación y salario.</p>
Eventos	<p>Descripción: Establece las relaciones entre los usuarios y los eventos en los que participan.</p> <p>Propósito: Permite registrar la participación de usuarios en eventos, incluyendo el rol, el feedback y la fecha de participación.</p>
Proveedores	<p>Descripción: Contiene información sobre los proveedores de la biblioteca.</p> <p>Propósito: Facilita la gestión de datos relacionados con los</p>

	proveedores de libros, incluyendo su nombre, dirección, teléfono y correo electrónico.
Compras	<p>Descripción: Registra información sobre las compras de libros a proveedores.</p> <p>Propósito: Permite realizar un seguimiento de las compras de libros, incluyendo la cantidad, la fecha de compra y las relaciones con los proveedores y libros mediante claves foráneas.</p>
Ventas_Libros	<p>Descripción: Registra información detallada sobre las ventas de libros realizadas a usuarios.</p> <p>Propósito: Permite realizar un seguimiento pormenorizado de las ventas de libros, incluyendo la cantidad de ejemplares vendidos, el precio unitario, la fecha de la venta, y las relaciones con las tablas de "Libros" y "Usuarios" mediante claves foráneas. Facilita el análisis de las transacciones comerciales relacionadas con la venta de libros en la biblioteca.</p>
Almacen	<p>Descripción: Gestiona el inventario de libros en la biblioteca, indicando la cantidad total de ejemplares y la cantidad disponible para préstamo.</p> <p>Propósito: Permite mantener un registro actualizado del inventario, especificando cuántos ejemplares están disponibles y cuántos están en total. La tabla se conecta a la tabla de "Libros" mediante una clave foránea, proporcionando un vínculo directo con la información detallada de cada libro.</p>
Ingresos	<p>Descripción: Registra información sobre los ingresos asociados con la operación de la biblioteca.</p> <p>Propósito: Permite llevar un registro detallado de los ingresos, incluyendo una descripción, el monto del ingreso y la fecha en que se registró. Esta tabla es fundamental para comprender la situación financiera de la biblioteca y evaluar su sostenibilidad económica.</p>
Gastos	<p>Descripción: Registra información sobre los gastos asociados con la operación de la biblioteca.</p> <p>Propósito: Permite realizar un seguimiento detallado de los gastos, incluyendo una descripción, el monto del gasto y la fecha en que se realizó. Esta tabla es útil para gestionar el presupuesto y evaluar los costos operativos de la biblioteca.</p>
Sueldo	<p>Descripción: Registra los pagos de sueldos a los empleados de la biblioteca.</p> <p>Propósito: Permite el seguimiento detallado de los sueldos pagados, incluyendo el monto, mes, año y relaciones con los empleados a través de claves foráneas. Esencial para la gestión financiera y análisis de costos de la nómina bibliotecaria.</p>
Participaciones	<p>Descripción: Registra la participación de usuarios en eventos de la biblioteca.</p> <p>Propósito: Permite documentar de manera detallada la participación de usuarios en eventos, incluyendo el rol desempeñado, comentarios y la fecha de participación. Esta tabla facilita un seguimiento minucioso de la interacción de los usuarios con los eventos, estableciendo relaciones con las tablas de "Eventos" y "Usuarios" mediante claves foráneas. Su diseño posibilita un análisis exhaustivo de la participación de usuarios en diversas actividades y eventos organizados por la biblioteca.</p>

c. Diseño de la base de datos.

i. Diseño E-R

Diagrama E-R



Fuente: Imagen creada en <https://app.diagrams.net/>

ii. Archivo .sql

Creacion de la base datos y las tablas:

Create database DB_Biblio_Gest

use DB_Biblio_Gest

CREATE TABLE Libros (

Id_Libro INT PRIMARY KEY IDENTITY(1,1),

Titulo VARCHAR(255) NOT NULL,

Id_Autor INT NOT NULL,

Id_Editorial INT NOT NULL,

Año_Publicacion INT NOT NULL,

FOREIGN KEY (Id_Autor) REFERENCES Autores(Id_Autor),

FOREIGN KEY (Id_Editorial) REFERENCES Editoriales(Id_Editorial)

);

CREATE TABLE Autores (

Id_Autor INT PRIMARY KEY IDENTITY(1,1),

Nombre VARCHAR(255) NOT NULL,

```
Nacionalidad VARCHAR(100) NOT NULL,  
Fecha_Nacimiento DATE NOT NULL,  
Biografia TEXT NOT NULL  
);
```

```
CREATE TABLE Editoriales (  
    Id_Editorial INT PRIMARY KEY IDENTITY(1,1),  
    Nombre VARCHAR(255) NOT NULL,  
    Direccion VARCHAR(255) NOT NULL,  
    Telefono VARCHAR(20) NOT NULL,  
    Email VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Usuarios (  
    Id_Usuario INT PRIMARY KEY IDENTITY(1,1),  
    Nombre VARCHAR(255) NOT NULL,  
    Direccion VARCHAR(255) NOT NULL,  
    Telefono VARCHAR(20) NOT NULL,  
    Email VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Prestamos (  
    Id_Prestamo INT PRIMARY KEY IDENTITY(1,1),  
    Id_Usuario INT NOT NULL,  
    Id_Libro INT NOT NULL,  
    Fecha_Prestamo DATE NOT NULL,  
    Fecha_Devolucion DATE NOT NULL,  
    FOREIGN KEY (Id_Usuario) REFERENCES Usuarios(Id_Usuario),  
    FOREIGN KEY (Id_Libro) REFERENCES Libros(Id_Libro)  
);
```

```
CREATE TABLE Categorias (  
    Id_Categoria INT PRIMARY KEY IDENTITY(1,1),  
    Nombre VARCHAR(255) NOT NULL,  
    Descripcion TEXT NOT NULL,  
    Area VARCHAR(100) NOT NULL,  
    Especialidad VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Libros_Categorias (  
    Id_Libro INT NOT NULL,  
    Id_Categoria INT NOT NULL,  
    Fecha_Asignacion DATE NOT NULL,  
    Comentarios TEXT NOT NULL,  
    Importancia INT NOT NULL,  
    FOREIGN KEY (Id_Libro) REFERENCES Libros(Id_Libro),  
    FOREIGN KEY (Id_Categoria) REFERENCES Categorias(Id_Categoria),  
    PRIMARY KEY (Id_Libro, Id_Categoria)  
);
```

```
CREATE TABLE Empleados (  
    Id_Empleado INT PRIMARY KEY IDENTITY(1,1),  
    Nombre VARCHAR(255) NOT NULL,  
    Puesto VARCHAR(100) NOT NULL,  
    Fecha_Contratacion DATE NOT NULL,  
    Salario DECIMAL(10, 2) NOT NULL  
);
```

```
CREATE TABLE Eventos (  
    Id_Evento INT PRIMARY KEY IDENTITY(1,1),
```



```
Nombre VARCHAR(255) NOT NULL,  
Descripcion TEXT NOT NULL,  
Fecha DATE NOT NULL,  
Ubicacion VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Participaciones (  
    Id_Evento INT NOT NULL,  
    Id_Usuario INT NOT NULL,  
    Rol VARCHAR(100) NOT NULL,  
    Feedback TEXT NOT NULL,  
    Fecha_Participacion DATE NOT NULL,  
    FOREIGN KEY (Id_Evento) REFERENCES Eventos(Id_Evento),  
    FOREIGN KEY (Id_Usuario) REFERENCES Usuarios(Id_Usuario),  
    PRIMARY KEY (Id_Evento, Id_Usuario)  
);
```

```
CREATE TABLE Proveedores (  
    Id_Proveedor INT PRIMARY KEY IDENTITY(1,1),  
    Nombre VARCHAR(255) NOT NULL,  
    Direccion VARCHAR(255) NOT NULL,  
    Telefono VARCHAR(20) NOT NULL,  
    Email VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Compras (  
    Id_Compra INT PRIMARY KEY IDENTITY(1,1),  
    Id_Proveedor INT NOT NULL,  
    Id_Libro INT NOT NULL,  
    Cantidad INT NOT NULL,
```

```
Fecha_Compra DATE NOT NULL,  
FOREIGN KEY (Id_Proveedor) REFERENCES Proveedores(Id_Proveedor),  
FOREIGN KEY (Id_Libro) REFERENCES Libros(Id_Libro)  
);
```

```
CREATE TABLE Ventas_Libros (  
    Id_Venta INT PRIMARY KEY IDENTITY(1,1),  
    Id_Libro INT NOT NULL,  
    Id_Usuario INT NOT NULL,  
    Cantidad INT NOT NULL,  
    Precio_Unitario DECIMAL(10, 2) NOT NULL,  
    Fecha_Venta DATE NOT NULL,  
    FOREIGN KEY (Id_Libro) REFERENCES Libros(Id_Libro),  
    FOREIGN KEY (Id_Usuario) REFERENCES Usuarios(Id_Usuario)  
);
```

```
CREATE TABLE Almacen (  
    Id_Inventario INT PRIMARY KEY IDENTITY(1,1),  
    Id_Libro INT NOT NULL,  
    Cantidad_Disponible INT NOT NULL,  
    Cantidad_Total INT NOT NULL,  
    FOREIGN KEY (Id_Libro) REFERENCES Libros(Id_Libro)  
);
```

```
CREATE TABLE Ingresos (  
    Id_Ingreso INT PRIMARY KEY IDENTITY(1,1),  
    Descripcion VARCHAR(255) NOT NULL,  
    Monto DECIMAL(10, 2) NOT NULL,  
    Fecha_Ingreso DATE NOT NULL,  
    Id_Venta INT,
```

```
FOREIGN KEY (Id_Venta) REFERENCES Ventas_Libros(Id_Venta)
);
```

```
CREATE TABLE Sueldo (
    Id_Sueldo INT PRIMARY KEY IDENTITY(1,1),
    Id_Empleado INT NOT NULL,
    Monto DECIMAL(10, 2) NOT NULL,
    Mes INT NOT NULL,
    Año INT NOT NULL,
    FOREIGN KEY (Id_Empleado) REFERENCES Empleados(Id_Empleado)
);
```

```
CREATE TABLE Gastos (
    Id_Gasto INT PRIMARY KEY IDENTITY(1,1),
    Descripcion VARCHAR(255) NOT NULL,
    Monto DECIMAL(10, 2) NOT NULL,
    Fecha_Gasto DATE NOT NULL,
    Id_Sueldo INT,
    Id_Compra INT,
    FOREIGN KEY (Id_Sueldo) REFERENCES Sueldo(Id_Sueldo),
    FOREIGN KEY (Id_Compra) REFERENCES Compras(Id_Compra)
);
```

Registros de todas las tablas:

--Registros

```
INSERT INTO Autores (Nombre, Nacionalidad, Fecha_Nacimiento, Biografia)
VALUES
('Gabriel García Márquez', 'Colombiana', '1927-03-06', 'Autor de Cien Años de Soledad.'),
('J.K. Rowling', 'Británica', '1965-07-31', 'Autora de Harry Potter.'),
```

```
('Haruki Murakami', 'Japonesa', '1949-01-12', 'Autor de Kafka en la Orilla.'),
('Isabel Allende', 'Chilena', '1942-08-02', 'Autora de La Casa de los Espíritus.'),
('George Orwell', 'Británica', '1903-06-25', 'Autor de 1984.');
```

```
INSERT INTO Editoriales (Nombre, Direccion, Telefono, Email)
```

```
VALUES
```

```
('Editorial Planeta', 'Calle Falsa 123', '555-1234', 'contacto@planeta.com'),
('Penguin Random House', 'Avenida Siempre Viva 456', '555-5678',
'contacto@penguinrandom.com'),
('Alfaguara', 'Diagonal 789', '555-9012', 'contacto@alfaguara.com'),
('Anagrama', 'Ruta 10 km 11', '555-3456', 'contacto@anagrama.com'),
('Acantilado', 'Bulevar de los Sueños 987', '555-7890', 'contacto@acantilado.com');
```

```
INSERT INTO Libros (Titulo, Id_Autor, Id_Editorial, Año_Publicacion)
```

```
VALUES
```

```
('Cien años de soledad', 1, 1, 1967),
('Harry Potter y la piedra filosofal', 2, 2, 1997),
('Kafka en la orilla', 3, 3, 2002),
('La casa de los espíritus', 4, 4, 1982),
('1984', 5, 5, 1949);
```

```
INSERT INTO Usuarios (Nombre, Direccion, Telefono, Email)
```

```
VALUES
```

```
('Juan Pérez', 'Calle Luna 111', '555-6543', 'juanperez@mail.com'),
('Ana López', 'Avenida Sol 222', '555-6789', 'analopez@mail.com'),
('Carlos Gómez', 'Calle Estrella 333', '555-9876', 'carlosgomez@mail.com'),
('Lucía Rodríguez', 'Boulevard Cometa 444', '555-1230', 'luciarodriguez@mail.com'),
('Miguel Ángel Torres', 'Calle Planeta 555', '555-4567', 'miguelangel@mail.com');
```

```
INSERT INTO Prestamos (Id_Usuario, Id_Libro, Fecha_Prestamo, Fecha_Devolucion)
```

VALUES

```
(1, 1, '2023-01-10', '2023-01-20'),  
(2, 2, '2023-02-15', '2023-02-25'),  
(3, 3, '2023-03-12', '2023-03-22'),  
(4, 4, '2023-04-08', '2023-04-18'),  
(5, 5, '2023-05-05', '2023-05-15');
```

INSERT INTO Categorías (Nombre, Descripción, Área, Especialidad)

VALUES

```
('Literatura', 'Libros de ficción y no ficción.', 'Artes', 'Literatura General'),  
( 'Ciencia Ficción', 'Libros de ciencia ficción y fantasía.', 'Artes', 'Narrativa Fantástica'),  
( 'Historia', 'Libros de historia y biografías.', 'Ciencias Sociales', 'Historia Universal'),  
( 'Ciencia', 'Libros de ciencia y tecnología.', 'Ciencias', 'Divulgación Científica'),  
( 'Filosofía', 'Libros de filosofía y pensamiento.', 'Humanidades', 'Filosofía Contemporánea');
```

INSERT INTO Libros_Categorías (Id_Libro, Id_Categoría, Fecha_Asignación, Comentarios, Importancia)

VALUES

```
(1, 1, '2023-03-01', 'Clásico de la literatura latinoamericana', 5),  
(2, 2, '2023-03-02', 'Bestseller internacional', 4),  
(3, 1, '2023-03-03', 'Obra maestra de Murakami', 5),  
(4, 1, '2023-03-04', 'Importante obra de Isabel Allende', 4),  
(5, 2, '2023-03-05', 'Clásico de la literatura distópica', 5);
```

INSERT INTO Empleados (Nombre, Puesto, Fecha_Contratación, Salario)

VALUES

```
('Luisa Fernández', 'Gerente', '2019-05-01', 3500.00),  
( 'Marco Antonio Solís', 'Bibliotecario', '2020-06-15', 2500.00),  
( 'Rosa Martínez', 'Administrativa', '2018-07-10', 2200.00),
```

('Pedro Jiménez', 'Mantenimiento', '2021-01-20', 1800.00),

('Ana Maria López', 'Cajera', '2022-03-05', 2000.00);

INSERT INTO Eventos (Nombre, Descripcion, Fecha, Ubicacion)

VALUES

('Feria del Libro', 'Evento anual de libros y autores', '2023-04-15', 'Centro de Convenciones'),

('Encuentro de Jóvenes Escritores', 'Reunión de nuevos talentos literarios', '2023-06-20', 'Biblioteca Municipal'),

('Charla sobre Literatura Contemporánea', 'Conferencia sobre tendencias literarias actuales', '2023-08-10', 'Universidad Nacional'),

('Exposición de Libros Antiguos', 'Muestra de libros raros y antiguos', '2023-10-05', 'Museo de la Ciudad'),

('Taller de Escritura Creativa', 'Curso de escritura para aficionados', '2023-11-15', 'Centro Cultural');

INSERT INTO Participaciones (Id_Evento, Id_Usuario, Rol, Feedback, Fecha_Participacion)

VALUES

(1, 1, 'Asistente', 'Muy interesante', '2023-04-15'),

(2, 2, 'Ponente', 'Excelente experiencia', '2023-06-20'),

(3, 3, 'Organizador', 'Gran participación', '2023-08-10'),

(4, 4, 'Expositor', 'Evento muy bien organizado', '2023-10-05'),

(5, 5, 'Instructor', 'Taller muy interactivo', '2023-11-15');

INSERT INTO Proveedores (Nombre, Direccion, Telefono, Email)

VALUES

('Libros y Libros S.A.', 'Calle de los Libreros 123', '555-1122', 'contacto@libroslibros.com'),

('Distribuidora Nacional', 'Avenida Central 456', '555-3344', 'ventas@distribuidoranacional.com'),

('Papelería del Norte', 'Diagonal 78', '555-5566', 'info@papeleranorte.com'),

('Impresiones Rápidas Ltda.', 'Ruta 5 km 12', '555-7788', 'servicios@impresionesrapidas.com'),

```
('Editorial Digital', 'Bulevar Virtual 101', '555-9900', 'ebooks@editorialdigital.com');
```

```
INSERT INTO Compras (Id_Proveedor, Id_Libro, Cantidad, Fecha_Compra)
```

```
VALUES
```

```
(1, 1, 50, '2023-03-01'),  
(2, 2, 30, '2023-03-10'),  
(3, 3, 20, '2023-04-15'),  
(4, 4, 25, '2023-05-20'),  
(5, 5, 15, '2023-06-30');
```

```
INSERT INTO Ventas_Libros (Id_Libro, Id_Usuario, Cantidad, Precio_Unitario,  
Fecha_Venta)
```

```
VALUES
```

```
(1, 1, 1, 20.00, '2023-01-15'),  
(2, 2, 2, 15.00, '2023-02-20'),  
(3, 3, 1, 25.00, '2023-03-25'),  
(4, 4, 1, 30.00, '2023-04-30'),  
(5, 5, 2, 10.00, '2023-05-05');
```

```
INSERT INTO Almacen (Id_Libro, Cantidad_Disponible, Cantidad_Total)
```

```
VALUES
```

```
(1, 20, 50),  
(2, 10, 30),  
(3, 15, 20),  
(4, 10, 25),  
(5, 5, 15);
```

```
INSERT INTO Ingresos (Descripcion, Monto, Fecha_Ingreso, Id_Venta)
```

```
VALUES
```

```
('Venta de Cien años de soledad', 20.00, '2023-01-15', 1),
```

```
('Venta de Harry Potter y la piedra filosofal', 30.00, '2023-02-20', 2),  
( 'Venta de Kafka en la orilla', 25.00, '2023-03-25', 3),  
( 'Venta de La casa de los espíritus', 30.00, '2023-04-30', 4),  
( 'Venta de 1984', 20.00, '2023-05-05', 5);
```

```
INSERT INTO Sueldo (Id_Empleado, Monto, Mes, Año)
```

```
VALUES
```

```
(1, 3500.00, 1, 2023),  
(2, 2500.00, 2, 2023),  
(3, 2200.00, 3, 2023),  
(4, 1800.00, 4, 2023),  
(5, 2000.00, 5, 2023);
```

Truncate table Gastos

```
INSERT INTO Gastos (Descripcion, Monto, Fecha_Gasto, Id_Sueldo, Id_Compra)
```

```
VALUES
```

```
('Pago de sueldo a Luisa Fernández', 3500.00, '2023-01-31', 1, NULL),  
( 'Compra de libros a Libros y Libros S.A.', 3500.00, '2023-03-01', NULL, 1),  
( 'Pago de sueldo a Marco Antonio Solís', 2500.00, '2023-02-28', 2, NULL),  
( 'Compra de libros a Distribuidora Nacional', 2500.00, '2023-03-10', NULL, 2),  
( 'Pago de sueldo a Rosa Martínez', 2200.00, '2023-03-31', 3, NULL),  
( 'Compra de libros a Libros y Libros S.A.', 1500.00, '2023-03-05', NULL, 3),  
( 'Pago de sueldo a Pedro Jiménez', 1800.00, '2023-04-30', 4, NULL),  
( 'Compra de libros a Editorial Digital', 2000.00, '2023-06-15', NULL, 5),  
( 'Pago de sueldo a Ana Maria López', 2000.00, '2023-05-31', 5, NULL),  
( 'Compra de libros a Papelera del Norte', 1200.00, '2023-06-20', NULL, 4);
```

Consultas:

```
SELECT L.Titulo, A.Nombre
```

```
FROM Libros L
```



```
JOIN Autores A ON L.Id_Autor = A.Id_Autor;
```

```
SELECT L.Titulo, A.Nombre AS Autor, E.Nombre AS Editorial
```

```
FROM Libros L
```

```
JOIN Autores A ON L.Id_Autor = A.Id_Autor
```

```
JOIN Editoriales E ON L.Id_Editorial = E.Id_Editorial;
```

```
SELECT P.Fecha_Prestamo, P.Fecha_Devolucion, U.Nombre AS Usuario, L.Titulo AS  
Libro
```

```
FROM Prestamos P
```

```
JOIN Usuarios U ON P.Id_Usuario = U.Id_Usuario
```

```
JOIN Libros L ON P.Id_Libro = L.Id_Libro;
```

```
SELECT V.Fecha_Venta, U.Nombre AS Comprador, L.Titulo AS Libro
```

```
FROM Ventas_Libros V
```

```
JOIN Usuarios U ON V.Id_Usuario = U.Id_Usuario
```

```
JOIN Libros L ON V.Id_Libro = L.Id_Libro;
```

```
SELECT E.Nombre, S.Monto, G.Descripcion, G.Fecha_Gasto
```

```
FROM Empleados E
```

```
JOIN Sueldo S ON E.Id_Empleado = S.Id_Empleado
```

```
JOIN Gastos G ON S.Id_Sueldo = G.Id_Sueldo;
```

Funciones de la base de Datos SQL Server

```
SELECT COUNT(*) as NumEmpleados FROM Empleados;
```

```
SELECT AVG(Salario) FROM Empleados;
```

```
SELECT MAX(Precio_Unitario) as PrecioMAX FROM Ventas_Libros ;
```

```
SELECT SUM(Monto) Total_Ingresos FROM Ingresos;
```

```
SELECT MIN(Cantidad_Disponible) as MenorCant FROM Almacen
```

--Funciones de Agregación con Manejo con 3 tablas

-- Cuantas ganancias genero cada autor

```
SELECT A.Nombre AS Autor, L.Titulo AS Libro, V.Cantidad, SUM(V.Precio_Unitario *  
V.Cantidad) AS Total_Ventas
```

```
FROM Ventas_Libros V
```

```
JOIN Libros L ON V.Id_Libro = L.Id_Libro
```

```
JOIN Autores A ON L.Id_Autor = A.Id_Autor
```

```
GROUP BY A.Nombre, L.Titulo, V.Cantidad;
```

--Cuantas ganancias se genero por mes

```
SELECT YEAR(I.Fecha_Ingreso) AS Año, MONTH(I.Fecha_Ingreso) AS Mes,  
SUM(I.Monto) As Monto_Mensual
```

```
FROM Ingresos I
```

```
JOIN Ventas_Libros V ON I.Id_Venta = V.Id_Venta
```

```
JOIN Libros L ON V.Id_Libro = L.Id_Libro
```

```
GROUP BY YEAR(I.Fecha_Ingreso), MONTH(I.Fecha_Ingreso);
```

-- Muestra los libros adquiridos, el proveedor, la editorial y la cantidad de libros adquiridos

```
SELECT L.Titulo, PR.Nombre AS Proveedor, E.Nombre AS Editorial, SUM(C.Cantidad) AS  
Total_Compras
```

```
FROM Compras C
```

```
JOIN Libros L ON C.Id_Libro = L.Id_Libro
```

```
JOIN Proveedores PR ON C.Id_Proveedor = PR.Id_Proveedor
```

```
JOIN Editoriales E ON L.Id_Editorial = E.Id_Editorial
```

```
GROUP BY L.Titulo, PR.Nombre, E.Nombre;
```

--Muestra cuanto vendio las categorias de los libros

```
SELECT CAT.Nombre AS Categoria, SUM(I.Monto) AS Total_Ingresos
```

```
FROM Ingresos I
```

```
JOIN Ventas_Libros V ON I.Id_Venta = V.Id_Venta
```

```
JOIN Libros L ON V.Id_Libro = L.Id_Libro
JOIN Libros_Categorias LC ON L.Id_Libro = LC.Id_Libro
JOIN Categorías CAT ON LC.Id_Categoría = CAT.Id_Categoría
GROUP BY CAT.Nombre;
```

--Muestra cuanto gana cada empleado y en que mes

```
SELECT YEAR(G.Fecha_Gasto) AS Año, MONTH(G.Fecha_Gasto) AS Mes, E.Puesto,
SUM(G.Monto) AS Total_Sueldos
FROM Gastos G
JOIN Sueldo S ON G.Id_Sueldo = S.Id_Sueldo
JOIN Empleados E ON S.Id_Empleado = E.Id_Empleado
WHERE G.Id_Sueldo IS NOT NULL
GROUP BY YEAR(G.Fecha_Gasto), MONTH(G.Fecha_Gasto), E.Puesto;
```

Funciones UDF:

--Funciones

--1ra Calcula el Total de Ingresos

```
CREATE FUNCTION CalcularTotalIngresos ()
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @TotalIngresos DECIMAL(10, 2);
    SELECT @TotalIngresos = SUM(Monto)
    FROM Ingresos;
    RETURN @TotalIngresos;
END;
```

-- Declara una variable para almacenar el resultado

```
DECLARE @ResultadoTotalIngresos DECIMAL(10, 2);
```

```
-- Ejecuta la función y almacena el resultado en la variable @ResultadoTotalIngresos  
SET @ResultadoTotalIngresos = dbo.CalcularTotalIngresos();
```

```
-- Imprime el resultado
```

```
PRINT 'El total de ingresos es: $' + CAST(@ResultadoTotalIngresos AS VARCHAR);
```

```
--2da Calcula el Total de Gastos
```

```
CREATE FUNCTION CalcularTotalGastos ()
```

```
RETURNS DECIMAL(10, 2)
```

```
AS
```

```
BEGIN
```

```
    DECLARE @TotalGastos DECIMAL(10, 2);
```

```
    SELECT @TotalGastos = SUM(Monto)
```

```
    FROM Gastos;
```

```
    RETURN @TotalGastos;
```

```
END;
```

```
-- Declara una variable para almacenar el resultado
```

```
DECLARE @ResultadoTotalGastos DECIMAL(10, 2);
```

```
-- Ejecuta la función y almacena el resultado en la variable @ResultadoTotalGastos
```

```
SET @ResultadoTotalGastos = dbo.CalcularTotalGastos();
```

```
-- Imprime el resultado
```

```
PRINT 'El total de gastos es: $' + CAST(@ResultadoTotalGastos AS VARCHAR);
```

```
--3ra Calcula el Balance
```

```
CREATE FUNCTION CalcularBalance ()
```

```
RETURNS DECIMAL(10, 2)
```

AS

BEGIN

```
DECLARE @Balance DECIMAL(10, 2);
```

```
SET @Balance = dbo.CalcularTotalIngresos() - dbo.CalcularTotalGastos();
```

```
RETURN @Balance;
```

END;

-- Declara una variable para almacenar el resultado

```
DECLARE @ResultadoBalance DECIMAL(10, 2);
```

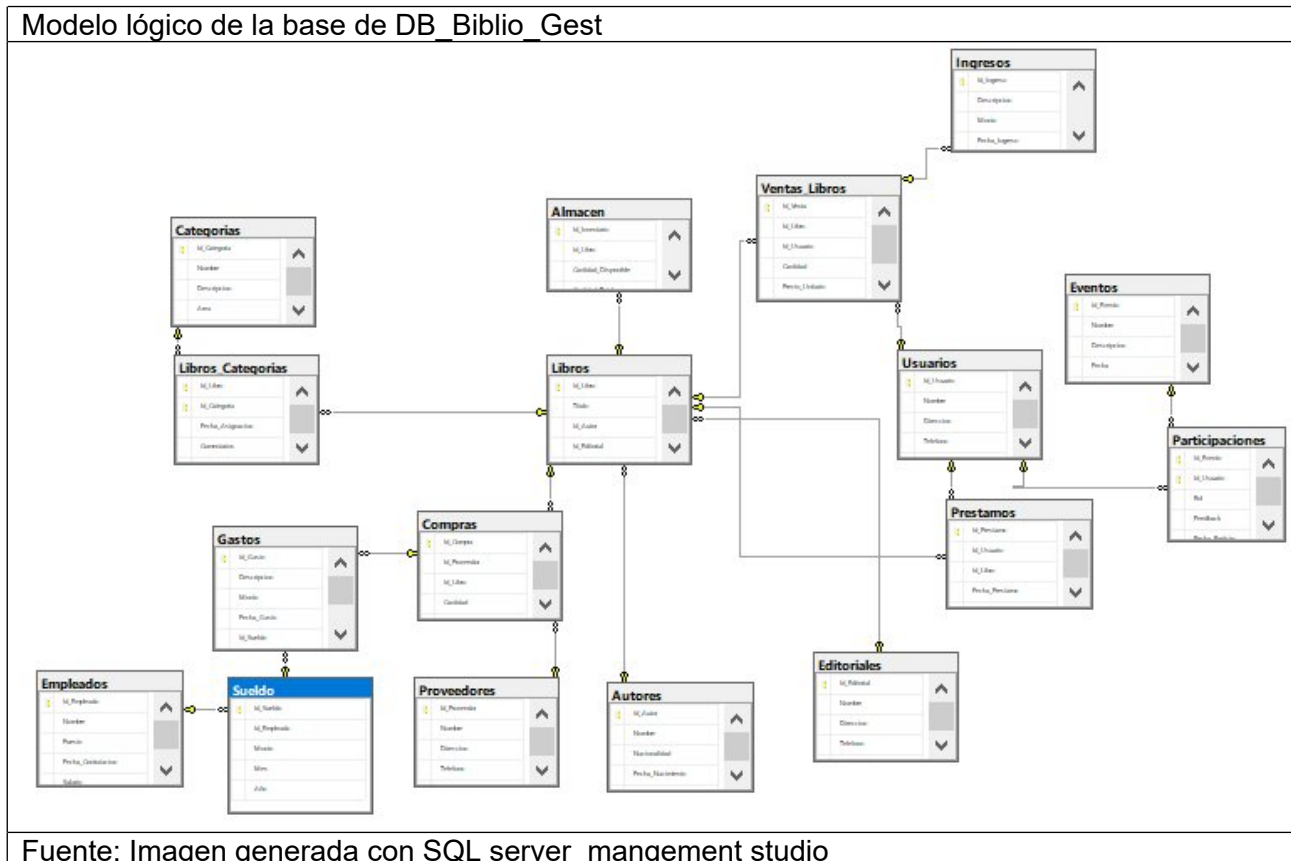
-- Ejecuta la función y almacena el resultado en la variable @ResultadoBalance

```
SET @ResultadoBalance = dbo.CalcularBalance();
```

-- Imprime el resultado

```
PRINT 'El balance es: $' + CAST(@ResultadoBalance AS VARCHAR);
```

iii. Modelo Lógico



3. Usabilidad

a. Imágenes del sistema.

Creación de tablas

```
-- Create database DB_Biblio_Gest
use DB_Biblio_Gest

-- CREATE TABLE Libros (
    Id_Libro INT PRIMARY KEY IDENTITY(1,1),
    Titulo VARCHAR(255) NOT NULL,
    Id_Autor INT NOT NULL,
    Id_Editorial INT NOT NULL,
    Año_Publicacion INT NOT NULL,
    FOREIGN KEY (Id_Autor) REFERENCES Autores(Id_Autor),
    FOREIGN KEY (Id_Editorial) REFERENCES Editoriales(Id_Editorial)
);

-- CREATE TABLE Autores (
    Id_Autor INT PRIMARY KEY IDENTITY(1,1),
    Nombre VARCHAR(255) NOT NULL,
    Nacionalidad VARCHAR(100) NOT NULL,
    Fecha_Nacimiento DATE NOT NULL,
    Biografia TEXT NOT NULL
);

-- CREATE TABLE Editoriales (
    Id_Editorial INT PRIMARY KEY IDENTITY(1,1),
    Nombre VARCHAR(255) NOT NULL,
    Direccion VARCHAR(255) NOT NULL,
    Telefono VARCHAR(20) NOT NULL,
    Email VARCHAR(100) NOT NULL
);
```

Fuente: Captura tomada de la creacion de tablas en sql server

Registros de las tablas

Results		Messages	
Id_Inventario	Id_Libro	Cantidad_Disponible	Cantidad_Total
1	1	20	50
2	2	10	30
3	3	15	20
4	4	10	25
5	5	5	15

Id_Autor	Nombre	Nacionalidad	Fecha_Nacimiento	Biografia
1	Gabriel García Márquez	Colombiana	1927-03-06	Autor de Cien Años de Soledad.
2	J.K. Rowling	Británica	1965-07-31	Autora de Harry Potter.
3	Haruki Murakami	Japonesa	1949-01-12	Autor de Kafka en la Orilla.
4	Isabel Allende	Chilena	1942-08-02	Autora de La Casa de los Esp...
5	George Orwell	Británica	1903-06-25	Autor de 1984.

Id_Categoria	Nombre	Descripcion	Area	Especialidad
1	Literatura	Libros de ficción y no ficción.	Artes	Literatura General
2	Ciencia Ficción	Libros de ciencia ficción y fantasía.	Artes	Narrativa Fantástica
3	Historia	Libros de historia y biografías.	Cie...	Historia Universal
4	Ciencia	Libros de ciencia y tecnología.	Cie...	Divulgación Cientí...
5	Filosofía	Libros de filosofía y pensamiento.	Hu...	Filosofía Contemp...

Id_Compra	Id_Proveedor	Id_Libro	Cantidad	Fecha_Compra
1	1	1	50	2023-03-01
2	2	2	30	2023-03-10
3	3	3	20	2023-04-15
4	4	4	25	2023-05-20
5	5	5	15	2023-06-30

Fuente: Captura tomada en sql server

Creación de función

```
--Funciones
--1ra Calcula el Total de Ingresos
CREATE FUNCTION CalcularTotalIngresos ()
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @TotalIngresos DECIMAL(10, 2);
    SELECT @TotalIngresos = SUM(Monto)
    FROM Ingresos;
    RETURN @TotalIngresos;
END;

-- Declara una variable para almacenar el resultado
DECLARE @ResultadoTotalIngresos DECIMAL(10, 2);

-- Ejecuta la función y almacena el resultado en la variable @ResultadoTotalIngresos
SET @ResultadoTotalIngresos = dbo.CalcularTotalIngresos();

-- Imprime el resultado
PRINT 'El total de ingresos es: $' + CAST(@ResultadoTotalIngresos AS VARCHAR);
```

89 %

Messages

El total de ingresos es: \$125.00

Completion time: 2023-12-03T22:02:14.0211523-04:00

Fuente: Captura tomada en sql server

Funciones de agregación

```
FROM Ventas_Libros V
JOIN Usuarios U ON V.Id_Usuario = U.Id_Usuario
JOIN Libros L ON V.Id_Libro = L.Id_Libro;

SELECT E.Nombre, S.Monto, G.Descripcion, G.Fecha_Gasto
FROM Empleados E
JOIN Sueldo S ON E.Id_Empleado = S.Id_Empleado
JOIN Gastos G ON S.Id_Sueldo = G.Id_Sueldo;

--Funciones de Agregación
SELECT COUNT(*) as NumEmpleados FROM Empleados;
SELECT AVG(Salario) FROM Empleados;
SELECT MAX(Precio_Unitario) as PrecioMAX FROM Ventas_Libros ;
SELECT SUM(Monto) Total_Ingresos FROM Ingresos;
SELECT MIN(Cantidad_Disponible) as MenorCant FROM Almacen

--Funciones de Agregación con Manejo con 3 tablas
-- Cuantas ganancias genero cada autor
SELECT A.Nombre AS Autor, L.Titulo AS Libro, V.Cantidad, SUM(V.Pre
FROM Ventas_Libros V
```

9 %

Results Messages

	NumEmpleados
1	5

	(No column name)
1	2400.000000

	PrecioMAX
1	30.00

	Total_Ingresos
--	----------------

Query executed successfully.

Fuente: Captura tomada en sql server

Funciones de agregacion con inner joins

```
GROUP BY L.Titulo, PK.Nombre, E.Nombre;

--Muestra cuanto vendio las categorias de los libros
SELECT CAT.Nombre AS Categoria, SUM(I.Monto) AS Total_Ingresos
FROM Ingresos I
JOIN Ventas_Libros V ON I.Id_Venta = V.Id_Venta
JOIN Libros L ON V.Id_Libro = L.Id_Libro
JOIN Libros_Categorias LC ON L.Id_Libro = LC.Id_Libro
JOIN Categorias CAT ON LC.Id_Categoria = CAT.Id_Categoria
GROUP BY CAT.Nombre;

--Muestra cuanto gana cada empleado y en que mes
SELECT YEAR(G.Fecha_Gasto) AS Año, MONTH(G.Fecha_Gasto) AS Mes, E.Puesto, SUM(G.Monto) AS Total_Sueldos
FROM Gastos G
JOIN Sueldo S ON G.Id_Sueldo = S.Id_Sueldo
JOIN Empleados E ON S.Id_Empleado = E.Id_Empleado
WHERE G.Id_Sueldo IS NOT NULL
GROUP BY YEAR(G.Fecha_Gasto), MONTH(G.Fecha_Gasto), E.Puesto;

--Funciones
--1ra Calcula el Total de Ingresos
CREATE FUNCTION CalcularTotalIngresos ()
```

89 %

Results Messages

	Año	Mes	Puesto	Total_Sueldos
1	2023	1	Gerente	3500.00
2	2023	2	Bibliotecario	2500.00
3	2023	3	Administrativa	2200.00
4	2023	4	Mantenimiento	1800.00
5	2023	5	Cajera	2000.00

Fuente: Captura tomada en sql server

b. Vídeo de la funcionalidad del sistema.

https://drive.google.com/drive/folders/1bXtvlNJYLR-6V7eFXI58_AzWY2YSshL?usp=drive_link

4. Conclusiones

1. Éxito en el Manejo y Gestión de Bases de Datos Relacionales:

El proyecto ha logrado de manera efectiva el objetivo de demostrar el manejo y gestión de bases de datos relacionales. La implementación de consultas y operaciones sobre la base de datos ha demostrado un conocimiento sólido de los principios fundamentales de las bases de datos relacionales.

2. Eficiencia en Consultas Relacionales INNER JOIN:

La utilización de consultas INNER JOIN ha sido abordada con eficiencia, mostrando la capacidad para manejar de manera efectiva la relación entre tablas en la base de datos. Esto sugiere un buen entendimiento de cómo realizar consultas complejas para extraer información específica de varias tablas relacionadas.

3. Dominio en Creación y Uso de Consultas SQL y Funciones de Base de Datos:

Se ha alcanzado exitosamente el objetivo de mostrar el uso y creación de consultas SQL, así como el manejo de funciones de la base de datos. El proyecto demuestra la capacidad para estructurar y ejecutar consultas SQL de manera competente, así como la aplicación efectiva de funciones que enriquecen las operaciones en la base de datos.

En general, el proyecto ha cumplido con los objetivos establecidos, demostrando un sólido entendimiento y aplicación de los conceptos relacionados con bases de datos relacionales y consultas SQL. Sin embargo, siempre hay espacio para la mejora continua, y consideramos la posibilidad de ampliar la complejidad de las consultas y operaciones, así como explorar nuevas funcionalidades y técnicas en futuros proyectos para fortalecer aún más las habilidades en este ámbito.