Sesión 07
Metodologías de Diseño de Bases de Datos.

Objetivo

Fortalecer las capacidades de los participantes, acerca del uso de metodologías y técnicas de diseño de bases de datos a partir de la descripción del caso, mediante el desarrollo de una ejercicio propuesto sobre la implementación de una base de datos observando los estándares internacionales, en una sesión práctica.

Requerimientos

- PC o Equivalente con Windows o Linux.
- Navegador web (Firefox, Chrome, Brave, Opera, etc.)
- Gestor de base de datos MySql/MariaDB/PostgreSQL
- Cliente de base de datos

Requisitos

- Haber asistido a las sesiones 3, 4, 5, 6 de la unidad didáctica.

Sesión 07
Metodologías de Diseño de Bases de Datos.

Crónica del Problema: "La Biblioteca del Saber"

Durante años, la Biblioteca Municipal de San Aurelio funcionó con métodos tradicionales: libros registrados en cuadernos, préstamos anotados a mano, y usuarios identificados por fichas de cartulina. Con el paso del tiempo, el volumen de libros creció, los usuarios se multiplicaron, y los errores también.

- Los bibliotecarios enfrentaban problemas como:
- Pérdida de registros de préstamos.
- Dificultad para saber qué libros estaban disponibles.
- Usuarios que no devolvían libros y no podían ser contactados.
- Imposibilidad de generar reportes o estadísticas.

El caos llegó a tal punto que muchos libros simplemente "desaparecían" y los usuarios preferían no usar el servicio por falta de confianza. Fue entonces cuando el municipio decidió modernizar el sistema con una base de datos que permitiera gestionar todo de forma digital.

Requerimientos para la Base de Datos

Funcionales:

- Registrar usuarios con sus datos personales.
- Registrar libros con información bibliográfica.
- Registrar préstamos de libros, incluyendo fechas de préstamo y devolución.
- Consultar qué libros están prestados y a quién.
- Generar reportes de préstamos por usuario y por libro.

No funcionales:

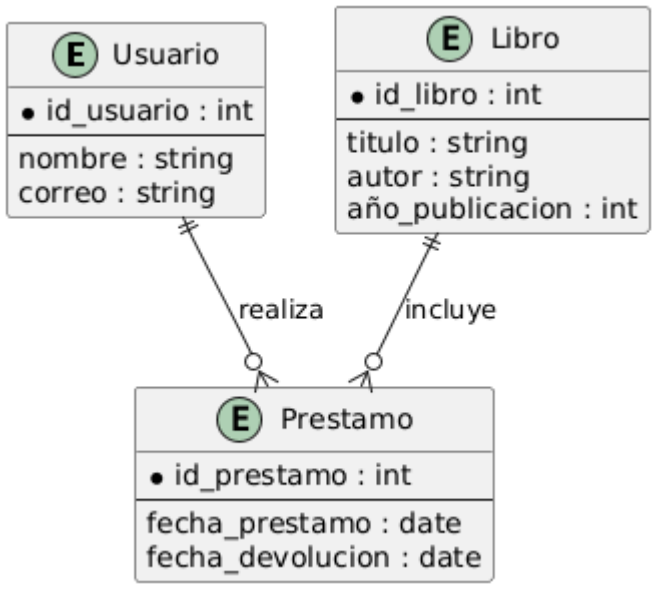
- Seguridad en el acceso a los datos.
- Escalabilidad para crecer con el tiempo.
- Facilidad de mantenimiento y respaldo.

Diseño Conceptual

Entidades:

- Usuario: persona que accede al servicio.
- Libro: ejemplar disponible en la biblioteca.
- Préstamo: registro de la entrega de un libro a un usuario.

Diagrama ER



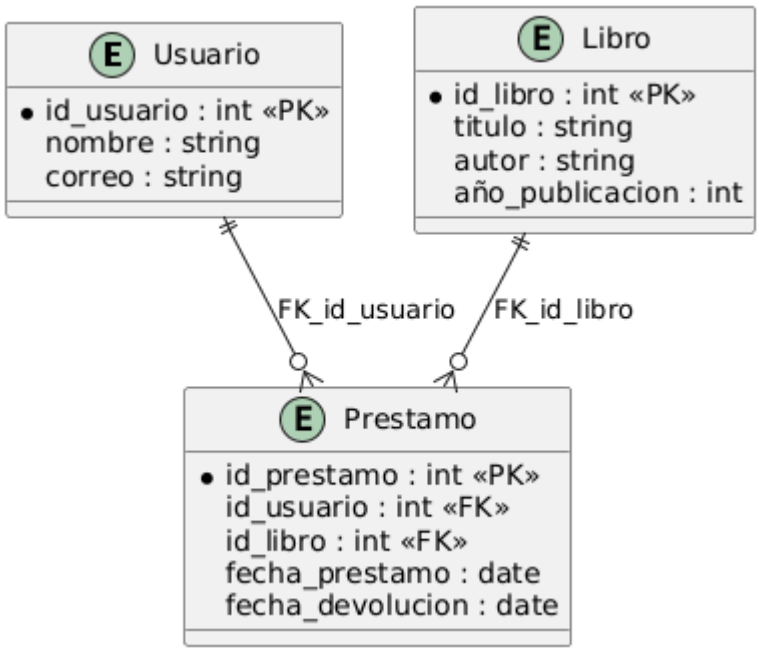
Sesión 07
Metodologías de Diseño de Bases de Datos.

Diseño Lógico

Tablas Relacionales:

- Usuario(id_usuario, nombre, correo)
- Libro(id_libro, titulo, autor, año_publicacion)
- Prestamo(id_prestamo, id_usuario, id_libro, fecha_prestamo, fecha_devolucion)

Diagrama Relacional



Sesión 07
Metodologías de Diseño de Bases de Datos.

Diseño Físico (SQL)

sql

```
CREATE TABLE Usuario (
  id_usuario SERIAL PRIMARY KEY,
  nombre VARCHAR(100),
  correo VARCHAR(100)
);

CREATE TABLE Libro (
  id_libro SERIAL PRIMARY KEY,
  titulo VARCHAR(150),
  autor VARCHAR(100),
  año_publicacion INT
);

CREATE TABLE Prestamo (
  id_prestamo SERIAL PRIMARY KEY,
  id_usuario INT REFERENCES Usuario(id_usuario),
  id_libro INT REFERENCES Libro(id_libro),
  fecha_prestamo DATE,
  fecha_devolucion DATE
);
```

Sesión 07
Metodologías de Diseño de Bases de Datos.

Extensión del caso

Vamos a extender el caso de "La Biblioteca del Saber", incorporando funcionalidades más avanzadas. Esta evolución responde a nuevas necesidades operativas y de control, y se reflejará en cada etapa del diseño: requerimientos, modelos, e implementación con procedimientos almacenados y triggers.

Requerimientos Extendidos

Funcionales adicionales:

- Controlar la disponibilidad de los libros (no permitir préstamos si ya están prestados).
- Registrar el historial de préstamos por usuario.
- Aplicar penalizaciones por retrasos en la devolución.
- Permitir reservas de libros cuando estén prestados.
- Registrar la fecha real de devolución y calcular días de retraso.

No funcionales:

- Automatización de procesos mediante triggers.
- Integridad de datos mediante restricciones y procedimientos.
- Registro de auditoría para cambios críticos.

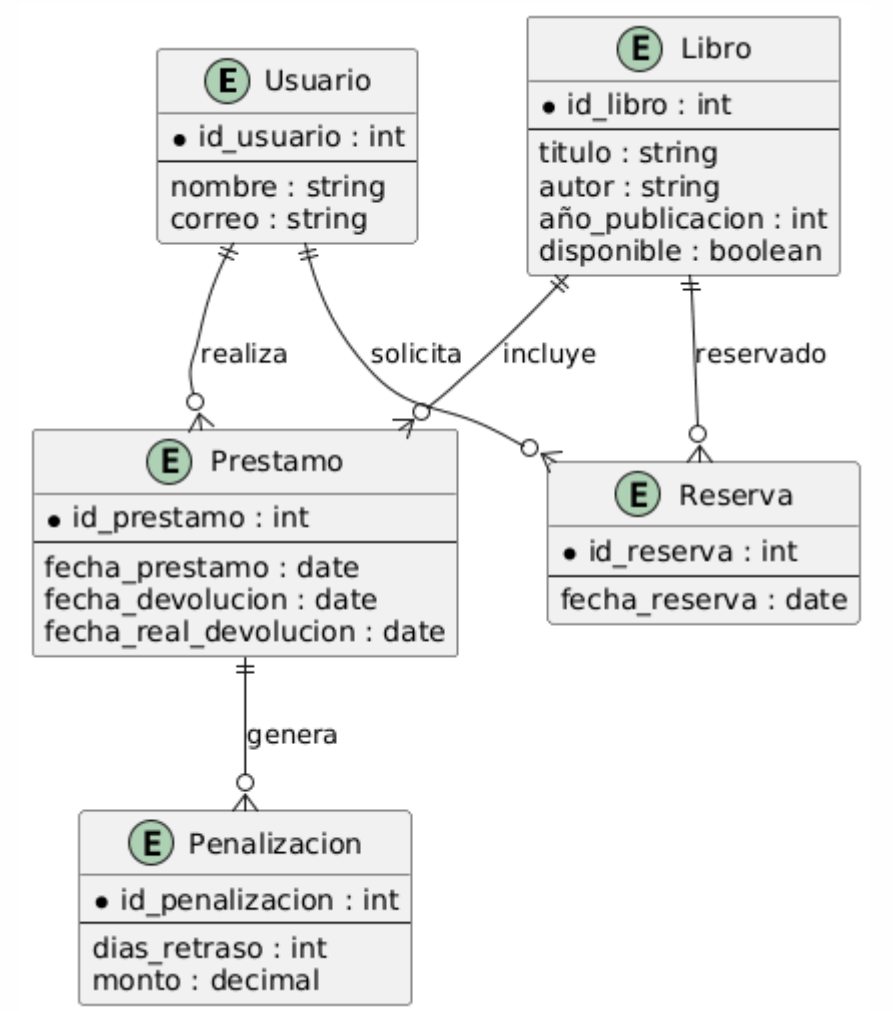
Diseño Conceptual Extendido

Nuevas entidades:

- Reserva: solicitud de un libro que está prestado.
- Penalización: registro de multa por devolución tardía.

Sesión 07
Metodologías de Diseño de Bases de Datos.

Diagrama ER extendido (PlantUML)



Sesión 07
Metodologías de Diseño de Bases de Datos.

Diseño Lógico Extendido

Nuevas tablas:

- Reserva(id_reserva, id_usuario, id_libro, fecha_reserva)
- Penalizacion(id_penalizacion, id_prestamo, dias_retraso, monto)

Se agrega fecha_real_devolucion y disponible en Libro.

Diseño Físico con SQL Avanzado

Tablas nuevas y modificadas:
sql

```
ALTER TABLE Libro ADD disponible BOOLEAN DEFAULT TRUE;
ALTER TABLE Prestamo ADD fecha_real_devolucion DATE;

CREATE TABLE Reserva (
  id_reserva SERIAL PRIMARY KEY,
  id_usuario INT REFERENCES Usuario(id_usuario),
  id_libro INT REFERENCES Libro(id_libro),
  fecha_reserva DATE
);

CREATE TABLE Penalizacion (
  id_penalizacion SERIAL PRIMARY KEY,
  id_prestamo INT REFERENCES Prestamo(id_prestamo),
  dias_retraso INT,
  monto DECIMAL(6,2)
);
```

Sesión 07
Metodologías de Diseño de Bases de Datos.

Procedimientos y Triggers

1. Trigger para controlar disponibilidad

sql

```
CREATE OR REPLACE FUNCTION actualizar_disponibilidad()
RETURNS TRIGGER AS $$
BEGIN
  UPDATE Libro SET disponible = FALSE WHERE id_libro = NEW.id_libro;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_actualizar_disponibilidad
AFTER INSERT ON Prestamo
FOR EACH ROW
EXECUTE FUNCTION actualizar_disponibilidad();
```

2. Trigger para marcar libro como disponible al devolver

sql

```
CREATE OR REPLACE FUNCTION devolver_libro()
RETURNS TRIGGER AS $$
BEGIN
  UPDATE Libro SET disponible = TRUE WHERE id_libro = NEW.id_libro;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_devolver_libro
AFTER UPDATE OF fecha_real_devolucion ON Prestamo
FOR EACH ROW
WHEN (NEW.fecha_real_devolucion IS NOT NULL)
EXECUTE FUNCTION devolver_libro();
```

Sesión 07
Metodologías de Diseño de Bases de Datos.

3. Trigger para calcular penalización

sql

```
CREATE OR REPLACE FUNCTION calcular_penalizacion()
RETURNS TRIGGER AS $$
DECLARE
  retraso INT;
BEGIN
  retraso := NEW.fecha_real_devolucion - NEW.fecha_devolucion;
  IF retraso > 0 THEN
    INSERT INTO Penalizacion(id_prestamo, dias_retraso, monto)
    VALUES (NEW.id_prestamo, retraso, retraso * 2.00); -- S/.2 por día
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_calcular_penalizacion
AFTER UPDATE OF fecha_real_devolucion ON Prestamo
FOR EACH ROW
EXECUTE FUNCTION calcular_penalizacion();
```

Sesión 07
Metodologías de Diseño de Bases de Datos.

Vamos a explorar consultas avanzadas para reportes , basadas en el modelo extendido de la Biblioteca del Saber. Estas consultas permiten extraer información útil para la gestión, auditoría y toma de decisiones.

1. Reporte de libros prestados actualmente

sql
```
SELECT l.titulo, u.nombre, p.fecha_prestamo
FROM Prestamo p
JOIN Libro l ON p.id_libro = l.id_libro
JOIN Usuario u ON p.id_usuario = u.id_usuario
WHERE p.fecha_real_devolucion IS NULL;
```
Muestra qué libros están prestados y quién los tiene.

2. Historial de préstamos por usuario

sql
```
SELECT u.nombre, l.titulo, p.fecha_prestamo, p.fecha_real_devolucion
FROM Prestamo p
JOIN Usuario u ON p.id_usuario = u.id_usuario
JOIN Libro l ON p.id_libro = l.id_libro
ORDER BY u.nombre, p.fecha_prestamo;
```
Permite ver todos los préstamos realizados por cada usuario.

3. Préstamos con retraso y penalizaciones

sql
```
SELECT u.nombre, l.titulo, p.fecha_devolucion, p.fecha_real_devolucion,
    pe.dias_retraso, pe.monto
FROM Prestamo p
JOIN Usuario u ON p.id_usuario = u.id_usuario
JOIN Libro l ON p.id_libro = l.id_libro
JOIN Penalizacion pe ON p.id_prestamo = pe.id_prestamo;
```
Identifica usuarios que devolvieron tarde y el monto de la penalización.

Sesión 07
Metodologías de Diseño de Bases de Datos.

4. Libros más prestados

sql

```
SELECT l.titulo, COUNT(*) AS veces_prestado
FROM Prestamo p
JOIN Libro l ON p.id_libro = l.id_libro
GROUP BY l.titulo
ORDER BY veces_prestado DESC
LIMIT 10;
```

Top 10 libros más populares según frecuencia de préstamo.

5. Usuarios con más préstamos

sql

```
SELECT u.nombre, COUNT(*) AS total_prestamos
FROM Prestamo p
JOIN Usuario u ON p.id_usuario = u.id_usuario
GROUP BY u.nombre
ORDER BY total_prestamos DESC
LIMIT 10;
```

Top 10 usuarios más activos en la biblioteca.

6. Libros reservados actualmente

sql

```
SELECT r.fecha_reserva, u.nombre, l.titulo
FROM Reserva r
JOIN Usuario u ON r.id_usuario = u.id_usuario
JOIN Libro l ON r.id_libro = l.id_libro
WHERE l.disponible = FALSE;
```

Muestra reservas activas de libros no disponibles.

Sesión 07
Metodologías de Diseño de Bases de Datos.

Bibliografía

- "Desarrollo de Aplicaciones Android con App Inventor 2", TOMAS DOMINGUEZ MINGUEZ, Ed. Marcombo, 2022.
- "Crea tus Aplicaciones Android con app Inventor 2", José Manuel Cabello García Ed. IC.
- "Android - Guía para desarrolladores", Ableson, F., Sen, R., King, C. Ed. Anaya Multimedia.
- "El Gran Libro de Android", Gironés, J.
  Ed. Alfaomega. Ediciones de la U, 2023;
  ISBN 9587924649, 9789587924640; N.º de páginas:316.
- https://docs.oracle.com/en/database/