

Asociación

Integrantes: Tomas Garcia Caffieri, Francisco Ortiz Ceballos, Francisco Lorenzo, Cona Lorenzo

Definición de Asociación:

La asociación simple es una relación entre dos clases independientes, donde una clase utiliza a la otra de alguna manera. Es decir, un objeto de una clase tiene una referencia a un objeto de otra clase, pero ninguna depende completamente de la otra para existir.

Características

- No hay una propiedad de "pertenencia" fuerte entre los objetos.
- Ambas clases pueden existir de manera independiente.
- La asociación puede ser de uno a uno, uno a muchos, muchos a uno o muchos a muchos.

Método de Asociación en el programa

```
public boolean crearPrestamo(String libroTitulo, int personaId, java.sql.Date fechaHoy, java.sql.Date fechaDevolucion) {  
    int numeroPrestamo = 1; // O tu lógica de numeración  
  
    try (Connection conn = Conexion.getConnection()) {  
        LibroDAO libroDao = new LibroDAO();  
        int libroId = libroDao.buscarLibro(conn, libroTitulo);  
  
        // Aquí obtenés el libro completo por ID  
        Libro libro = libroDao.getLibroPorId(conn, libroId); // Este método retorna titulo, clasificacion, numero  
  
        ArrayList<Libro> libros = new ArrayList<>();  
        libros.add(libro);  
  
        PersonaDAO personaDao = new PersonaDAO();  
        Persona socio = personaDao.getPersona(conn, personaId);  
  
        // Ahora si, el constructor de Prestamo lleva los datos del libro  
        Prestamo prestamo = new Prestamo(numeroPrestamo, fechaHoy, fechaDevolucion, socio, libros);  
  
        // Al insertar el préstamo usás el libroId (para la FK en la tabla prestamo)  
        PrestamoDAO presDao = new PrestamoDAO();  
        boolean respuesta = presDao.insertarPrestamo(conn, prestamo, libroId, personaId);  
  
        if (respuesta) {  
            System.out.println("Prestamo insertado con éxito");  
        } else {  
            System.out.println("Error insertando Prestamo");  
        }  
    }  
}
```

Explicación

Dentro del método:

```
"public boolean crearPrestamo(String libroTitulo, int personalId, java.sql.Date fechaHoy, java.sql.Date fechaDevolucion)"
```

- Este método crea un objeto Prestamo a partir de un objeto Persona y una lista de objetos Libro. Ninguno de estos objetos depende completamente del Prestamo para existir. Es decir, los objetos Persona y Libro pueden existir independientemente, pero se asocian temporalmente para formar un préstamo.

```
"Persona socio = personaDao.getPersona(conn, personalId);
```

```
Prestamo prestamo = new Prestamo(numeroPrestamo, fechaHoy, fechaDevolucion, socio, libros);"
```

- Prestamo tiene una referencia a un objeto de tipo Persona (socio) y a una lista de Libro.
- Esta es una asociación simple, porque Prestamo conoce a Persona, pero no hay una relación de dependencia total (ni de composición ni de agregación fuerte).

Explicación de cada tabla de la base de datos

TABLA persona

Función principal:

Almacenar los datos básicos de los usuarios que pueden realizar préstamos de libros. Es una tabla de entidad, es decir, representa un objeto principal del dominio: las personas.

Campos:

- pers_id: Clave primaria, generada automáticamente. Sirve como identificador interno único de la persona.
- pers_nombre: Nombre y apellido de la persona.
- pers_documento: DNI o número de identificación. Es único, evitando que una misma persona esté registrada más de una vez.

TABLA libro

Función principal:

- Registrar todos los libros disponibles para préstamo. También es una tabla de entidad, representando el recurso físico o digital que se puede prestar.

Campos:

- libr_id: Clave primaria única e incremental para cada libro.
- libr_titulo: Nombre del libro. Es único para evitar duplicaciones (asume que los títulos son únicos en el sistema).
- libr_clasificacion: Categoría, género o tipo del libro (ej: "Novela", "Thriller", "Fábula").
- libr_numero: Puede representar un número de inventario interno, útil para clasificar físicamente los libros o identificar copias.

TABLA préstamo

Función principal:

- Registrar cada préstamo de libros. Esta tabla es una tabla de relación (también llamada "tabla puente") que asocia personas con libros, añadiendo información específica de cada préstamo (como fechas y estado).

Campos:

- pers_id: Clave foránea a persona.pers_id. Indica quién realiza el préstamo.
- libr_id: Clave foránea a libro.libr_id. Indica qué libro se presta.
- prestamo_numero: Un número interno del préstamo (no es autogenerado, se controla desde la aplicación).
- prestamo_dia: Fecha en la que se realizó el préstamo.
- prestamo_duracion: Fecha pactada de devolución (fecha límite).
- prestamo_estado: Booleano (0 o 1) que indica si el libro fue devuelto (1) o sigue prestado (0).

Explicación de cada clase

Clase LibroDAO

Es una clase dedicada exclusivamente al acceso y manipulación de datos en la tabla libro de la base de datos.

- `getLibroPorId`: Obtiene la información completa de un libro a partir de su ID, devolviendo un objeto Libro.

Función:

- Abstraer todas las operaciones SQL sobre la tabla libro.

```
public Libro getLibroPorId(Connection conn, int libroId) {  
    String sql = "SELECT libr_titulo, libr_clasificacion, libr_numero FROM libro WHERE libr_id = ?";  
    try (PreparedStatement ps = conn.prepareStatement(sql)) {  
        ps.setInt(1, libroId);  
        ResultSet rs = ps.executeQuery();  
        if (rs.next()) {  
            String titulo = rs.getString("libr_titulo");  
            String clasificacion = rs.getString("libr_clasificacion");  
            int numero = rs.getInt("libr_numero");  
            return new Libro(titulo, clasificacion, numero);  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    return null;  
}
```

-
- Facilitar la reutilización y mantener el código limpio y separado de la lógica de negocio.

Clase PersonaDAO

Esta clase maneja la persistencia y consulta de información de personas en la base de datos.

- insertarPersona: Inserta un nuevo registro en la tabla persona con los datos del DNI y nombre.
- buscarPersona: Busca una persona según su documento (DNI) y devuelve el ID de la persona si existe.
- getPersona: Obtiene un objeto Persona a partir de un ID, con todos sus datos.

Función:

- Permite la interacción controlada con la base de datos para los usuarios.
- Provee un punto único para cambiar la lógica o estructura de la tabla sin afectar el resto de la aplicación.

Clase PrestamoDAO

Esta clase es responsable de todas las operaciones relacionadas con los préstamos de libros.

- insertarPrestamo: Inserta un nuevo préstamo relacionando un libro y una persona, con la fecha de préstamo y duración.

```
13 public class PersonaDAO{
14
15     public boolean insertarPersona(Connection conn, Persona persona) {
16         String sql = "INSERT INTO persona (pers_documento, pers_nombre) VALUES (?, ?)";
17         try (PreparedStatement ps = conn.prepareStatement(sql)) {
18             ps.setInt(1, persona.getDni());
19             ps.setString(2, persona.getNombre());
20             ps.executeUpdate();
21             return true;
22         } catch (SQLException e) {
23             e.printStackTrace();
24             return false;
25         }
26     }
27
28     public Persona buscarPersona(int id) {
29         try (PreparedStatement ps = conn.prepareStatement(sql)) {
30             ps.setInt(1, id);
31             ResultSet resultado = ps.executeQuery();
32             if (resultado.next()) {
33                 String nombre = resultado.getString("pers_nombre");
34                 int documento = resultado.getInt("pers_documento");
35                 return new Persona(documento, nombre);
36             } else {
37                 return null;
38             }
39         } catch (SQLException e) {
40             e.printStackTrace();
41             return null;
42         }
43     }
44 }
```

```

14 public class PrestamoDAO{
15
16
17     public boolean insertarPrestamo(Connection conn, Prestamo prestamo, int libroId, int personaId) {
18         String sql = "INSERT INTO prestamo (pers_id, libr_id, prestamo_numero, prestamo_dia, prestamo_duracion) VALUES (?, ?, ?, ?, ?)";
19         try (PreparedStatement ps = conn.prepareStatement(sql)) {
20             ps.setInt(1, personaId);
21             ps.setInt(2, libroId);
22             ps.setInt(3, prestamo.getNumero());
23             ps.setDate(4, prestamo.getDia_prestamo());
24             ps.setDate(5, prestamo.getDevolucion());
25             ps.executeUpdate();
26             System.out.println("Préstamo ingresado correctamente");
27             return true;
28         } catch (SQLException e) {
29             e.printStackTrace();
30             return false;
31         }
32     }
33
34
35     public ArrayList<Prestamo> getPrestamosPorPersona(Connection conn, int personaId) {
36         ArrayList<Prestamo> prestamos = new ArrayList<>();
37         String sql = ""
38 SELECT p.prestamo_numero, p.prestamo_dia, p.prestamo_duracion, p.prestamo_estado,
39         l.libr_titulo, l.libr_clasificacion, l.libr_numero
40 FROM prestamo p
41 JOIN libro l ON p.libr_id = l.libr_id
42 WHERE p.pers_id = ?

```

- **getPrestamosPorPersona:** Devuelve una lista con todos los préstamos realizados por una persona, recuperando además datos de los libros asociados para cada préstamo.
- **libroDevuelto:** Este método permite registrar de forma efectiva la devolución de un libro por parte de un usuario.

Función:

- Usa sentencias SQL con JOIN para obtener información combinada de préstamos y libros.
- Crea objetos Prestamo y Libro para devolver al controlador.

Vistas:

Vista Registro

MVC Prueba

Registro

Nombre

DNI

Aceptar

Vista Login

MVC Prueba

Bienvenido

Username

Dni

Ingresar

No tienes cuenta? [Registrarme](#)

Vista Solicitar Préstamo

MVC Prueba

Bienvenido fran

Elija la operacion que le gustaria realizar.

Solicitar Prestamo ▼

Enviar

Vista Solicitar Libro

MVC Prueba

Que libro quieres ped...

Libros

El principito ▼

Tiempo de Prestamo

1 semana ▼

Solicitar Prestamo

Volver

Vista Libros Solicitados

Hola usuario estos son tus libros solicitados hasta el momento
Si quieres devolver alguno, por favor selecciona desde la lista y luego toca el botón " Entregar"

El Principito (Fábula) - Pendiente

Entregar

Volver