



UNIVERSIDAD MADERO PUEBLA
INCORPORADA A LA SEP

ALQUILER DE PELÍCULAS

Debanhi Abigail Librado Almada

Hisham Yael Rosales Morales Jesús

Elias Guerrero Corral Daniel Trejo

Galindo

José Brian Molina Nochebuena

PLATAFORMAS ABIERTAS I

ÍNDICE

1.	Requerimientos	3
1.1.	Objetivo general del proyecto	3
1.2.	Datos maestros	3
•	Cliente.....	3
1.3.	Procesos y Datos Transaccionales	3
2.	Análisis y Diseño.....	4
2.1.	Diagrama de clases	4
3.	Implementación	5
3.1	Código de entidades en Java	5
3.2	Configuración inicial para conexión a Base de datos	15
3.3	Pantallas.....	17

1. Requerimientos

1.1. Objetivo general del proyecto

Sistematizar el sistema de alquiler de películas de un blockbuster durante una semana.

1.2. Datos maestros

- **Director**
 - Id, nombre, apellido.
- **Género**
 - Id, descripción.
- **Película**
 - Id, Título, Director, Género, Año de Lanzamiento, Duración, Cantidad disponible.
- **Empleado**
 - Id, Puesto, Nombre, Apellido, Edad, Número de Teléfono.
- **Cliente**
 - Id Miembro, Correo, Dirección, Número de Teléfono, Nombre, Apellido, Edad.
- **Persona**
 - Nombre, Apellido, Id, Edad, Número de Teléfono.

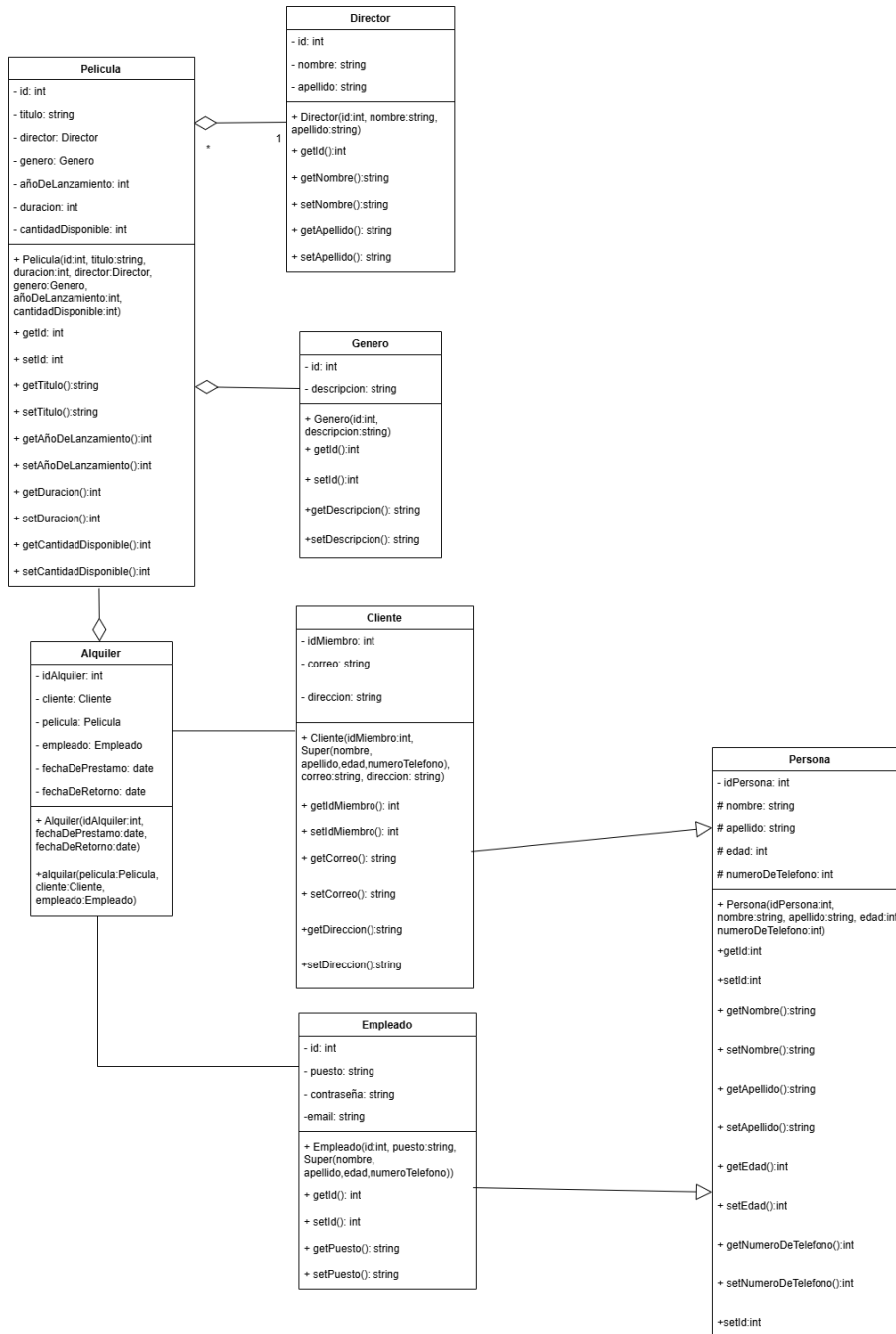
1.3. Procesos y Datos Transaccionales

- **Login y logout**
 - Acceso y cierre de sesión de un usuario al sistema
- **CRUD de datos maestros**
 - Altas, bajas, cambios y lecturas de directores, géneros, películas, empleados y clientes.
- **Alquiler de Películas**
 - **Creación de Alquiler**
 - Creación del alquiler
 - Empleado
 - Película
 - Cliente
 - (Fecha de Préstamo, Fecha de Retorno)
 - **Visualización de Alquileres**
 - Búsqueda de alquileres por cliente.
 - **Eliminación de Alquileres**
 - Eliminación de alquileres por cliente.

- **Visualización de Películas Existentes**
 - Búsqueda de películas existentes por género, nombre o director.

2. Análisis y Diseño

2.1. Diagrama de clases



3. Implementación

3.1 Código de entidades en Java

CLIENT

```
package com.umad.proyectofinalddbby.data;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Inheritance;
import jakarta.persistence.InheritanceType;
import jakarta.persistence.Table;

@Entity
@Table(name = "Clients")
public class Client extends Person {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @Column(nullable = false)
    private String email;
    private String address;

    public int getId() {
        return id;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    @Override
    public String toString() {
        return String.format("%s - %s ", this.email, this.address);
    }
}
```

```
}  
}
```

DIRECTOR

```
package com.umad.proyectofinalddbby.data;  
  
import jakarta.persistence.CascadeType;  
import jakarta.persistence.Entity;  
import jakarta.persistence.GeneratedValue;  
import jakarta.persistence.GenerationType;  
import jakarta.persistence.Id;  
import jakarta.persistence.OneToMany;  
import jakarta.persistence.Table;  
import java.util.List;  
  
/**  
 *  
 * @author deban  
 */  
@Entity  
@Table(name = "Directors")  
public class Director {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private int id;  
    private String firstName;  
    private String lastName;  
    @OneToMany(mappedBy = "director", cascade = CascadeType.PERSIST)  
    private List<Movie> movies;  
  
    public int getId() {  
        return id;  
    }  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(String firstName) {  
        this.firstName = firstName;  
    }  
  
    public String getLastName() {  
        return lastName;  
    }  
  
    public void setLastName(String lastName) {
```

```

        this.lastName = lastName;
    }

    public List<Movie> getMovies() {
        return movies;
    }

    public void setMovies(List<Movie> movies) {
        this.movies = movies;
    }

    @Override
    public String toString() {
        return String.format("%s - %s %s", this.id, this.firstName, this.lastName);
    }
}

```

EMPLOYEE

```

package com.umad.proyectorfinalddbby.data;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

/**
 *
 * @author deban
 */
@Entity
@Table(name = "Employees")
public class Employee extends Person {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @Column(nullable = false)
    private String jobPosition;
    private String password;
    private String email;

    public int getId() {
        return id;
    }

    public String getJobPosition() {
        return jobPosition;
    }
}

```

```

    }

    public void setJobPosition(String jobPosition) {
        this.jobPosition = jobPosition;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public long getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(long phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

```



```

    }

    @Override
    public String toString() {
        return String.format("%d - %s %s - %s", this.id, this.name, this.lastName, this.jobPosition);
    }
}

```

MOVIE

```
package com.umad.proyectorfinalddbby.data;
```

```

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;

```

```

@Entity
@Table(name = "Movies")
public class Movie {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @Column(nullable = false)
    private String title;
    private int releaseYear;
    private int duration;
    private int quantityAvailable;

    @ManyToOne
    @JoinColumn(name = "director_id")
    private Director director;

    @ManyToOne
    @JoinColumn(name = "genre_id")
    private Genre genre;

    public int getId() {
        return id;
    }
    public String getTitle() {
        return title;
    }

    public void setTitle(String tittle) {

```

```

        this.title = tittle;
    }

    public int getReleaseYear() {
        return releaseYear;
    }

    public void setReleaseYear(int releaseYear) {
        this.releaseYear = releaseYear;
    }

    public int getDuration() {
        return duration;
    }

    public void setDuration(int duration) {
        this.duration = duration;
    }

    public int getQuantityAvailable() {
        return quantityAvailable;
    }

    public void setQuantityAvailable(int quantityAvailable) {
        this.quantityAvailable = quantityAvailable;
    }

    public Director getDirector() {
        return director;
    }

    public void setDirector(Director director) {
        this.director = director;
    }

    public Genre getGenre() {
        return genre;
    }

    public void setGenre(Genre genre) {
        this.genre = genre;
    }

    @Override
    public String toString() {
        return String.format("%s - %s - %s - %s", title, releaseYear, duration, quantityAvailable);
    }

    public void incrementQuantity() {

```

```

        this.quantityAvailable++;
    }

    public void decrementQuantity() {
        if (this.quantityAvailable > 0) {
            this.quantityAvailable--;
        } else {
            System.out.println("No hay copias disponibles");
        }
    }
}

```

GENRE

```
package com.umad.proyectofinalddbby.data;
```

```

import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.OneToMany;
import jakarta.persistence.Table;
import java.util.List;

```

```
@Entity
```

```
@Table(name = "Genres")
```

```
public class Genre {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private int id;
```

```
    @OneToMany(mappedBy = "genre", cascade = CascadeType.PERSIST)
```

```
    private List<Movie> movies;
```

```
    @Column(nullable = false)
```

```
    private String description;
```

```
    public int getId() {
```

```
        return id;
```

```
    }
```

```
    public String getDescription() {
```

```
        return description;
```

```
    }
```

```
    public void setDescription(String description) {
```

```
        this.description = description;
```

```
    }
```

```

    public List<Movie> getMovies() {
        return movies;
    }

    public void setMovies(List<Movie> movies) {
        this.movies = movies;
    }

    @Override
    public String toString() {
        return String.format("%s", description);
    }
}

```

PERSON

```
package com.umad.proyectofinalddbby.data;
```

```

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.MappedSuperclass;

```

```

/**
 *
 * @author deban
 */
@MappedSuperclass
public class Person {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @Column(nullable = false)
    protected String name;
    protected String lastName;
    protected int age;
    protected long phoneNumber;

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }
}

```

```

    public void setName(String name) {
        this.name = name;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public long getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(long phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    @Override
    public String toString() {
        return String.format("%d - %s %s - %d", this.id, this.name, this.lastName, this.age);
    }
}

```

RENT

```

package com.umad.proyectofinalddbby.data;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;
import java.time.LocalDate;
import java.util.Date;

```

```

/**
 *
 * @author deban
 */
@Entity
@Table(name = "Rents")
public class Rent {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @Column(nullable = false)
    private LocalDate rentDay;
    private LocalDate returnDay;
    @ManyToOne
    @JoinColumn(name = "movie_id")
    private Movie movie;

    @ManyToOne
    @JoinColumn(name = "client_id")
    private Client client;

    @ManyToOne
    @JoinColumn(name = "employee_id")
    private Employee employee;

    public int getId() {
        return id;
    }

    public LocalDate getRentDay() {
        return rentDay;
    }

    public void setRentDay(LocalDate rentDay) {
        this.rentDay = rentDay;
    }

    public LocalDate getReturnDay() {
        return returnDay;
    }

    public void setReturnDay(LocalDate returnDay) {
        this.returnDay = returnDay;
    }

    public Movie getMovie() {
        return movie;
    }

```

```

    }

    public void setMovie(Movie movie) {
        this.movie = movie;
    }

    public Client getClient() {
        return client;
    }

    public void setClient(Client client) {
        this.client = client;
    }

    public Employee getEmployee() {
        return employee;
    }

    public void setEmployee(Employee employee) {
        this.employee = employee;
    }

    @Override
    public String toString() {
        return String.format("%d - Rent Day: %s - Return Day: %s - Movie: %s - Director: %s - Genre: %s",
            this.id,
            this.rentDay,
            this.returnDay,
            this.movie.getTitle(),
            this.movie.getDirector().getFirstName() + " " + this.movie.getDirector().getLastName(),
            this.movie.getGenre().getDescription());
    }
}

```

3.2 Configuración inicial para conexión a Base de datos

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration">
<hibernate-configuration>

    <session-factory>
        <!-- Database connection settings -->
        <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
        <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/TestDB</property>
        <property name="hibernate.connection.username">postgres</property>
        <property name="hibernate.connection.password">hismono08</property>
    
```

```

<!-- JDBC connection pool settings -->
<property name="hibernate.c3p0.min_size">5</property>
<property name="hibernate.c3p0.max_size">20</property>
<property name="hibernate.c3p0.timeout">300</property>
<property name="hibernate.c3p0.max_statements">50</property>
<property name="hibernate.c3p0.idle_test_period">3000</property>

<!-- SQL dialect -->
<property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>

<!-- Echo all executed SQL to stdout -->
<property name="hibernate.show_sql">true</property>

<!-- Format SQL output -->
<property name="hibernate.format_sql">true</property>

<!-- Automatically create the database schema -->
<property name="hibernate.hbm2ddl.auto">update</property>

<!-- List your annotated classes here -->
<mapping class="com.umad.proyectofinalddbby.data.Director"/>
<mapping class="com.umad.proyectofinalddbby.data.Genre"/>
<mapping class="com.umad.proyectofinalddbby.data.Movie"/>
<mapping class="com.umad.proyectofinalddbby.data.Rent"/>
<mapping class="com.umad.proyectofinalddbby.data.Person"/>
<mapping class="com.umad.proyectofinalddbby.data.Client"/>
<mapping class="com.umad.proyectofinalddbby.data.Employee"/>

</session-factory>

</hibernate-configuration>

```


3.3 Pantallas

INICIO

```
Bienvenido
Usuario:
Josué
Contraseña:
josuel234
Login Exitoso
Bienvenido Josué.
=====MENU=====
```

```
=====RENTAS=====
1. Rentar pelicula
2. Regresar pelicula
=====MAS OPCIONES=====
3. Clientes
4. Directores
5. Generos
6. Peliculas
7. Salir
```

RENTAR PELÍCULA

```
=====RENTAS=====
1. Rentar pelicula
2. Regresar pelicula
=====MAS OPCIONES=====
3. Clientes
4. Directores
5. Generos
6. Peliculas
7. Salir
1
Ingrese el correo del cliente
juan@gmail.com
```

Escribe el nombre de la película que desea rentar

Películas disponibles: Pinocho

Películas disponibles: Avengers

Pinocho

Hibernate:

```
select
    ml_0.id,
    ml_0.director_id,
    ml_0.duration,
    ml_0.genre_id,
    ml_0.quantityAvailable,
    ml_0.releaseYear,
    ml_0.title
from
    Movies ml_0
where
    ml_0.title=?
```

Hibernate:

```
insert
into
    Rents
    (client_id, employee_id, movie_id, rentDay, returnDay)
values
    (?, ?, ?, ?, ?)
returning id
```

Renta autorizada.

Fecha de entrega:2024-12-01

Fecha de retorno: 2024-12-08

REGRESAR PELICULA

=====MENU=====

=====RENTAS=====

1. Rentar película
2. Regresar película

=====MAS OPCIONES=====

3. Clientes
4. Directores
5. Generos
6. Peliculas
7. Salir

2

Ingrese el correo del cliente que devolver la película

juan@gmail.com

Pelicula rentada: Pinocho
Fecha de entrega: 2024-12-01
Fecha de retorno esperada: 2024-12-08
Desea confirmar el retorno de la pelicula s/n
s

Hibernate:

```
update
    Movies
set
    director_id=?,
    duration=?,
    genre_id=?,
    quantityAvailable=?,
    releaseYear=?,
    title=?
where
    id=?
```

Hibernate:

```
delete
from
    Rents
where
    id=?
```

La pelicula ha sido retorna con exito

AGREGAR

```

=====RENTAS=====
1. Rentar pelicula
2. Regresar pelicula
=====MAS OPCIONES=====
3. Clientes
4. Directores
5. Generos
6. Peliculas
7. Salir
5
=====GENEROS=====
1. Agregar
2. Eliminar
3. Editar
4. Consultar
1
Ingrese los siguientes datos para dar de alta un nuevo genero
Nombre:
Romance
Hibernate:
    insert
    into
        Genres
        (description)
    values
        (?)
    returning id
Nuevo genero creado

```

ELIMINAR

=====GENEROS=====

1. Agregar
2. Eliminar
3. Editar
4. Consultar

2

Escriba el genero que desea eliminar:

Hibernate:

```
select
    gl_0.id,
    gl_0.description
from
    Genres gl_0
order by
    gl_0.id
```

Horror

Action

Romance

Romance

Hibernate:

```
select
    gl_0.id,
    gl_0.description
from
    Genres gl_0
where
    gl_0.description=?
```

Hibernate:

```
delete
from
    Genres
where
    id=?
```

Genero eliminado

EDITAR

=====GENEROS=====

1. Agregar
2. Eliminar
3. Editar
4. Consultar

3

Ingrese el genero que desea editar:

Hibernate:

```
select
    gl_0.id,
    gl_0.description
from
    Genres gl_0
order by
    gl_0.id
```

Horror

Action

Action

Hibernate:

```
select
    gl_0.id,
    gl_0.description
from
    Genres gl_0
where
    gl_0.description=?
```

Nombre:

Action New

Hibernate:

```
update
    Genres
set
    description=?
where
    id=?
```

Genero actualizado

CONSULTAR

=====GENEROS=====

1. Agregar
2. Eliminar
3. Editar
4. Consultar

4

Hibernate:

```
select
    gl_0.id,
    gl_0.description
from
    Genres gl_0
order by
    gl_0.id
```

Horror

Action

DESPEDIDA

=====MENU=====

=====RENTAS=====

1. Rentar pelicula
2. Regresar pelicula

=====MAS OPCIONES=====

3. Clientes
4. Directores
5. Generos
6. Peliculas
7. Salir

7

Hasta Luego

BUILD SUCCESS

Total time: 03:46 min

Finished at: 2024-12-01T23:16:34-06:00