

*ESCUELAS SALESIANAS MARÍA AUXILIADORA*

**CICLO FORMATIVO DE GRADO SUPERIOR DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**PROYECTO FINAL |1ºDAM**

**Marina Dominguez Pascual**

**INDICE**

1. **Análisis y diseño del sistema de gestión de películas y actores**
   1. **Introducción**
   2. **Objetivo**
   3. **Requisitos funcionales**
2. **Implementación**
   1. **Estructura de paquetes**
   2. **Diagrama de clases**
   3. **Base de datos**
      1. **Diagrama E-R**
      2. **Modelo Relacional**
      3. **Script SQL**
   4. **Procedimientos almacenados**
3. **Desarrollo de la aplicación**
   1. **Implementación DAO**
   2. **Funcionalidades principales**
4. **Manual de usuario y administradores**
   1. **Inicio de la aplicación**
   2. **Menú según rol**

**Repositorio GitHub - Enlace**

**1. Análisis y diseño del sistema**

**Introducción**

Este proyecto consiste en una aplicación para organizar información sobre películas y los actores que participan en ellas.

**Objetivo:**

Su objetivo es facilitar la búsqueda y gestión de datos como:

* **Actores y actrices**: Nombre, apellidos, año de nacimiento y nacionalidad.
* **Películas**: Título, año de estreno, duración, resumen y género (acción, comedia, terror, etc.).
* **Relaciones**: Qué actores trabajaron en cada película.

La aplicación está diseñada para ser útil para administradores y para usuarios que solo quieran buscar información.

**Requisitos funcionales**

**Actores/Actrices:**

* Añadir nuevos actores.
* Editar o eliminar datos si hay errores.
* Ver en qué películas ha trabajado un actor.

**Películas:**

* Registrar nuevas películas.
* Corregir información si cambia la duración o el género.
* Eliminar una participación si un actor fue añadido por error a una película.

**Género:**

* Añadir nuevos géneros.
* Ver todas las películas de un género

**Usuarios**

* Buscar actores por nacionalidad o edad.
* Filtrar películas por año o género

**2. Implementación**

**Arquitectura del sistema**

**Estructura de paquetes:**

**src/**

**aplicación/**

* + GestorCine.java

**conexión/**

* + conexionBaseDatos.java

**dao**/

* + ActorDAO.java
  + AdminDAO.java
  + PeliculaDAO.java
  + RepartoDAO.java
  + UsuarioDAO.java

**dto/**

* + Actor.java
  + Admin.java
  + Pelicula.java
  + Reparto.java
  + Usuario.java

**vista/**

* + Como\_Administrador.java
  + Como\_Usuario.java
  + IniciarSesion.java

Diagrama

El contenido generado por IA puede ser incorrecto.

**Diagrama de clases**

**Base de datos**

Para el diseño de la base de datos se ha creado el siguiente Diagrama Entidad-Relación y Modelo relacional:

**Diagrama E-R**

**Actor**

* actor\_id (PK)
* nombre
* apellidos
* año\_nacimiento
* nacionalidad

**Película**

* pelicula\_id (PK)
* titulo
* año\_estreno
* duración

**Género**

* id\_genero (PK)
* nombre
* película\_id

**Relaciones**

* **Participa** (Muchos-muchos **Actor y Película**)
* **Calificada** (Uno-muchos entre **Género y Película**)
* Cada película pertenece a un género y cada género puede tener muchas película

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Modelo relacional**

--Creación de la base de datos--

**CREATE DATABASE gestionCine;**

**USE gestionCine;**

**CREATE TABLE actor (**

**actor\_id INT PRIMARY KEY,**

**nombre VARCHAR(50) NOT NULL,**

**apellidos VARCHAR(100) NOT NULL,**

**año\_nacimiento INT,**

**nacionalidad VARCHAR(50)**

**);**

**CREATE TABLE genero (**

**genero\_id INT PRIMARY KEY,**

**nombre VARCHAR(50) NOT NULL**

**);**

**CREATE TABLE pelicula (**

**pelicula\_id INT PRIMARY KEY,**

**titulo VARCHAR(100) NOT NULL,**

**año\_estreno INT,**

**duracion INT,**

**genero\_id INT,**

**FOREIGN KEY (genero\_id) REFERENCES genero(genero\_id)**

**);**

**CREATE TABLE reparto (**

**pelicula\_id INT NOT NULL,**

**actor\_id INT NOT NULL,**

**personaje VARCHAR(100),**

**PRIMARY KEY (pelicula\_id, actor\_id),**

**FOREIGN KEY (pelicula\_id) REFERENCES pelicula(pelicula\_id),**

**FOREIGN KEY (actor\_id) REFERENCES actor(actor\_id)**

**);**

*--Filtros--*

**ALTER TABLE** actor

**ADD PRIMARY KEY** (actor\_id);

**ALTER TABLE** genero

**ADD PRIMARY KEY** (genero\_id),

**ADD UNIQUE KEY** nombre (nombre);

**ALTER TABLE** pelicula

**ADD PRIMARY KEY** (pelicula\_id),

**ADD KEY** genero\_id (genero\_id);

**ALTER TABLE** reparto

**ADD PRIMARY KEY** (pelicula\_id, actor\_id),

**ADD KEY** actor\_id (actor\_id);

**ALTER TABLE** actor

**MODIFY** actor\_id **int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=11;**

**ALTER TABLE** genero

**MODIFY** genero\_id **int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;**

**ALTER TABLE** pelicula

**MODIFY** pelicula\_id **int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;**

**ALTER TABLE** pelicula

**ADD CONSTRAINT** pelicula\_ibfk\_1 **FOREIGN KEY** (genero\_id) **REFERENCES** genero (genero\_id);

**ALTER TABLE** reparto

**ADD CONSTRAINT** reparto\_ibfk\_1 **FOREIGN KEY** (pelicula\_id) **REFERENCES** pelicula (pelicula\_id) **ON** **DELETE CASCADE,**

**ADD CONSTRAINT** reparto\_ibfk\_2 **FOREIGN KEY** (actor\_id) REFERENCES actor (actor\_id) **ON DELETE** **CASCADE;**

Datos insertados en las tablas

**INSERT INTO genero (genero\_id, nombre) VALUES**

**(1, 'Terror'),**

**(2, 'Drama'),**

**(3, 'Romance'),**

**(4, 'Ciencia Ficción'),**

**(5, 'Aventura'),**

**(6, 'Fantasía');**

**INSERT INTO actor (actor\_id, nombre, apellidos, año\_nacimiento, nacionalidad) VALUES**

**(1, 'Robert', 'Downey Jr', 1965, 'Estadounidense'),**

**(2, 'Chris', 'Evans', 1981, 'Estadounidense'),**

**(3, 'Christian', 'Bale', 1974, 'Británico'),**

**(4, 'Vera', 'Farmiga', 1973, 'Estadounidense'),**

**(5, 'Patrick', 'Wilson', 1973, 'Estadounidense'),**

**(6, 'Bill', 'Skarsgård', 1990, 'Sueco'),**

**(7, 'Tom', 'Hanks', 1956, 'Estadounidense'),**

**(8, 'Viggo', 'Mortensen', 1958, 'Estadounidense-Danés'),**

**(9, 'Leonardo', 'DiCaprio', 1974, 'Estadounidense'),**

**(10, 'Matthew', 'McConaughey', 1969, 'Estadounidense'),**

**(11, 'Toni', 'Collette', 1972, 'Australiana'),**

**(12, 'Alex', 'Wolff', 1997, 'Estadounidense'),**

**(13, 'Tim', 'Robbins', 1958, 'Estadounidense'),**

**(14, 'Morgan', 'Freeman', 1937, 'Estadounidense'),**

**(15, 'Emma', 'Stone', 1988, 'Estadounidense'),**

**(16, 'Ryan', 'Gosling', 1980, 'Canadiense'),**

**(17, 'Kate', 'Winslet', 1975, 'Británica'),**

**(18, 'Harrison', 'Ford', 1942, 'Estadounidense'),**

**(19, 'Karen', 'Allen', 1951, 'Estadounidense'),**

**(20, 'Sam', 'Neill', 1947, 'Neozelandés'),**

**(21, 'Laura', 'Dern', 1967, 'Estadounidense'),**

**(22, 'Elijah', 'Wood', 1981, 'Estadounidense'),**

**(23, 'Ian', 'McKellen', 1939, 'Británico'),**

**(24, 'Daniel', 'Radcliffe', 1989, 'Británico'),**

**(25, 'Rupert', 'Grint', 1988, 'Británico'),**

**(26, 'Ana', 'de Armas', 1988, 'Española'),**

**(27, 'Mark', 'Ruffalo', 1967, 'Estadounidense'),**

**(28, 'Scarlett', 'Johansson', 1984, 'Estadounidense');**

**INSERT INTO pelicula (pelicula\_id, titulo, año\_estreno, duracion, genero\_id) VALUES**

**(1, 'El Conjuro', 2013, 112, 1),**

**(2, 'Hereditary', 2018, 127, 1),**

**(3, 'Forrest Gump', 1994, 142, 2),**

**(4, 'The Shawshank Redemption', 1994, 142, 2),**

**(5, 'Titanic', 1997, 195, 3),**

**(6, 'La La Land', 2016, 128, 3),**

**(7, 'Avengers', 2012, 143, 4),**

**(8, 'Blade Runner 2049', 2017, 164, 4),**

**(9, 'Indiana Jones: Raiders of the Lost Ark', 1981, 115, 5),**

**(10, 'Jurassic Park', 1993, 127, 5),**

**(11, 'El Señor de los Anillos: La Comunidad del Anillo', 2001, 178, 6),**

**(12, 'Harry Potter y la Piedra Filosofal', 2001, 152, 6);**

**INSERT INTO reparto (pelicula\_id, actor\_id, personaje) VALUES**

**(1, 4, 'Lorraine Warren'),**

**(1, 5, 'Ed Warren'),**

**(2, 11, 'Annie Graham'),**

**(2, 12, 'Peter Graham'),**

**(3, 7, 'Forrest Gump'),**

**(3, 17, 'Jenny Curran'),**

**(4, 13, 'Andy Dufresne'),**

**(4, 14, 'Ellis Boyd "Red" Redding'),**

**(5, 9, 'Jack Dawson'),**

**(5, 17, 'Rose DeWitt Bukater'),**

**(6, 16, 'Sebastian Wilder'),**

**(6, 15, 'Mia Dolan'),**

**(7, 1, 'Tony Stark/Iron Man'),**

**(7, 28, 'Natasha Romanoff/Black Widow'),**

**(8, 16, 'K'),**

**(8, 26, 'Joi'),**

**(9, 18, 'Indiana Jones'),**

**(9, 19, 'Marion Ravenwood'),**

**(10, 20, 'Dr. Alan Grant'),**

**(10, 21, 'Dr. Ellie Sattler'),**

**(11, 22, 'Frodo Bolsón'),**

**(11, 23, 'Gandalf'),**

**(12, 24, 'Harry Potter'),**

**(12, 25, 'Ron Weasley');**

**Procedimiento: Obtener Filmografía de un Actor**

**DELIMITER //**

**CREATE PROCEDURE** obtener\_filmografia\_actor(**IN** actor\_id\_param **INT**)

**BEGIN**

**SELECT**

p.titulo **AS** Pelicula,

p.año\_estreno **AS** Año,

p.duracion **AS** 'Duración (min)',

g.nombre **AS** Genero,

r.personaje **AS** Personaje

**FROM**

reparto r

**JOIN**

pelicula p **ON** r.pelicula\_id = p.pelicula\_id

**JOIN**

genero g **ON** p.genero\_id = g.genero\_id

**WHERE**

r.actor\_id = actor\_id\_param

**ORDER** **BY**

p.año\_estreno **DESC**;

**END //**

**DELIMITER ;**

**Disparador: Validar Año de Nacimiento del Actor**

**DELIMITER //**

**CREATE TRIGGER** validar\_edad\_actor

**BEFORE INSERT** **ON** actor FOR EACH ROW

**BEGIN**

**DECLARE** error\_edad **CONDITION FOR SQLSTATE '70001';**

**DECLARE** año\_actual **INT;**

**SET** año\_actual **= YEAR(CURDATE());**

-- Verificar que el actor tenga entre 5 y 120 años

**IF NEW**.año\_nacimiento **IS NOT NULL AND**

(NEW.año\_nacimiento **< (**año\_actual **- 120)** **OR NEW**.año\_nacimiento **> (**año\_actual **- 5)) THEN**

**SIGNAL** error\_edad

**SET MESSAGE\_TEXT =** 'El año de nacimiento no es válido. El actor debe tener entre 5 y 120 años',

**MYSQL\_ERRNO = 7001;**

**END IF;**

-- Validar la nacionalidad usando la función creada

**IF NOT** validar\_nacionalidad(NEW.nacionalidad) **THEN**

**SIGNAL SQLSTATE '70002'**

**SET MESSAGE\_TEXT =** 'Nacionalidad no válida. Use uno de los valores permitidos',

**MYSQL\_ERRNO = 7002;**

**END IF;**

**END //**

**DELIMITER ;**

**3. Desarrollo de la aplicación**

**Implementación DAO**

public class PeliculaDAO {

private static List<Pelicula> peliculas = new ArrayList<>();

public void crearPelicula(Pelicula pelicula) {

if (pelicula == null || pelicula.getGenero() == null) {

throw new IllegalArgumentException("Película o género no válidos");

}

peliculas.add(pelicula);

}

public List<Pelicula> filtrarPorGenero(Genero genero) {

List<Pelicula> resultado = new ArrayList<>();

for (Pelicula p : peliculas) {

if (p.getGenero() == genero) {

resultado.add(p);

}

}

return resultado;

}

public Pelicula buscarPorTitulo(String titulo) {

return peliculas.stream()

.filter(p -> p.getTitulo().equalsIgnoreCase(titulo))

.findFirst()

.orElse(null);

}

public List<Pelicula> listarPeliculas() {

return new ArrayList<>(peliculas);

}

}

**Funcionalidades clave**

- Registro de películas con validación de géneros

- Búsqueda de actores por nacionalidad

- Consulta de filmografías completo

**Manual de usuario**

Esta aplicación está diseñada para organizar información sobre películas y actores, permitiendo a los usuarios buscar datos y a los administradores gestionar el contenido.

**Inicio de la Aplicación**

* Al abrir la aplicación, se mostrará una pantalla de inicio de sesión donde:
* **Administradores:** Ingresan con su usuario y contraseña para acceder a funciones avanzadas.
* **Usuarios normales:** Pueden explorar la información sin necesidad de iniciar sesión.

**Menú según Rol:**

* **Para Administradores**

El menú del administrador incluye las siguientes opciones:

**Gestión de Actores/Actrices**

* **Añadir nuevo actor:** Clic en "Nuevo" → Llenar formulario (nombre, apellidos, año de nacimiento, nacionalidad).
* **Editar actor:** Seleccionar actor en la lista → Clic en "Modificar" → Editar datos → Guardar.
* **Eliminar actor:** Seleccionar actor → Clic en "Borrar" → Confirmar eliminación.
* **Ver películas del actor:** Seleccionar actor → Clic en "Ver filmografía".

**Gestión de Películas**

* **Añadir nueva película:** Clic en "Nueva Película" → Completar datos (título, año, duración, resumen, género).
* **Editar película:** Seleccionar película → Clic en "Modificar" → Actualizar información.
* **Eliminar película:** Seleccionar película → Clic en "Borrar" → Confirmar.
* **Añadir/Eliminar actor de una película:** Abrir detalles de la película → "Añadir Actor" (seleccionar de la lista) o "Quitar Actor".

**Gestión de Géneros**

* **Añadir nuevo género:** Ir a "Géneros" → Clic en "Nuevo Género" → Ingresar nombre.
* **Ver películas por género:** Seleccionar género en la lista → Mostrará todas las películas asociadas.

**Cerrar Sesión:** Clic en "Cerrar Sesión" para volver al inicio.

**Para Usuarios sin Inicio de sesión**

Los usuarios pueden:

**Búsqueda de Actores**

**Por nacionalidad:** Ir a "Actores" → Filtrar por país.

**Por edad:** Usar el filtro "Rango de años" para buscar actores por edad.

**Búsqueda de Películas**

**Por año de estreno:** Ir a "Películas" → Filtrar por año.

**Por género:** Seleccionar un género en el menú "Géneros" para ver películas relacionadas.

**Repositorio GitHub:** <https://github.com/MarinaDP2006/ProyectoDam1/tree/main>