



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Aplicación de Listas Enlazadas (modelo MVC)

Estructura De Datos

Estudiantes: Diego Montesdeoca y Angel Rodriguez

Docente: Margoth Elisa Guaraca Moyota

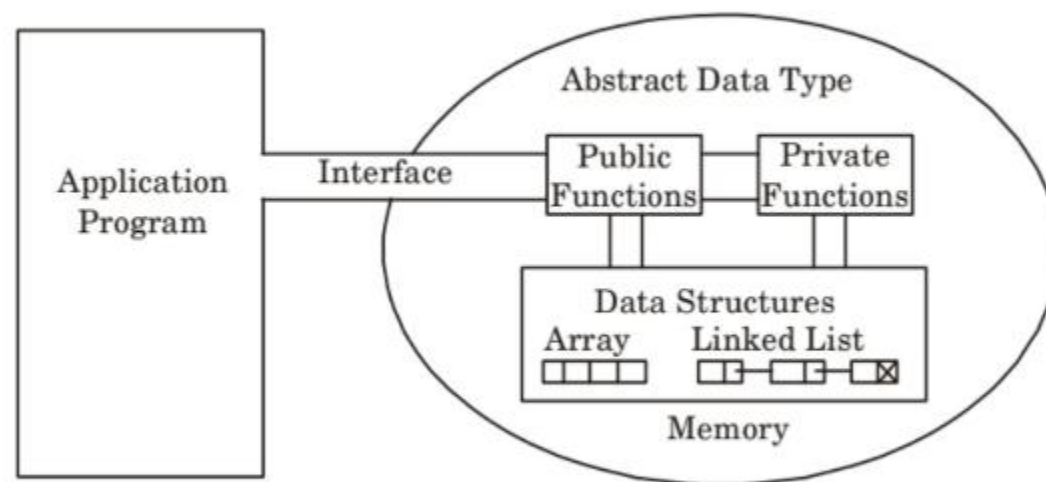
Fecha: 11/11/2025

Introducción

El presente proyecto detalla el desarrollo de una aplicación de software para la gestión de una colección de objetos "Película".

El sistema se centra en la implementación de una **Lista Enlazada Simple** para el almacenamiento, complementada con estructuras de **Pila (LIFO)** y **Cola (FIFO)**.

Para asegurar una arquitectura robusta, escalable y modular, se adoptó el patrón de diseño **Modelo-Vista-Controlador (MVC)**.






Objetivo General



Desarrollar una aplicación de gestión de información en Java, implementando una Lista Enlazada Simple con estructuras adicionales de Pila y Cola, utilizando el patrón MVC para la organización arquitectónica.

Objetivos Específicos

-  **Implementar Lista Enlazada:** Desarrollar en Java las operaciones básicas (inserción, eliminación, búsqueda) para gestionar el catálogo de películas.
-  **Desarrollar GUI con MVC:** Crear una interfaz gráfica que permita visualizar y manipular los datos, separando la lógica (Modelo), la UI (Vista) y los eventos (Controlador).
-  **Integrar Validación de Datos:** Asegurar la integridad de la información verificando campos, formatos (año) y evitando duplicados, con notificaciones al usuario.

Marco Teórico: Lista Enlazada Simple

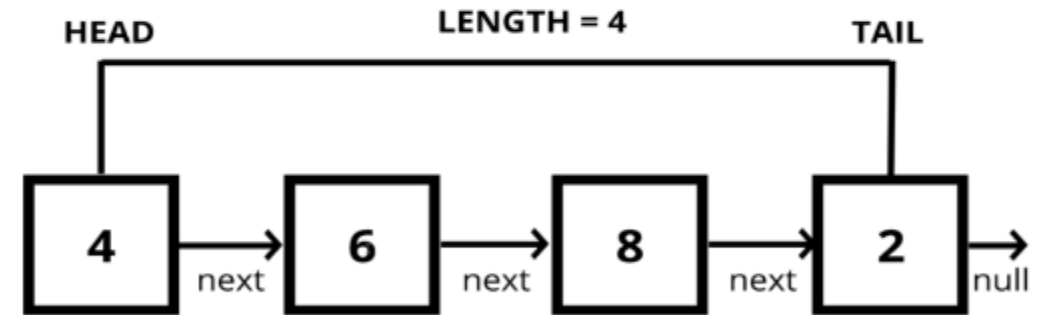
Es una colección de nodos donde el orden no es por ubicación física, sino por referencias explícitas.

Cada **Nodo** posee dos campos:

- **Dato:** Almacena la información (un objeto Pelicula).
- **Enlace (next):** Contiene la dirección del siguiente nodo.

La lista se gestiona mediante una referencia **head**. Ofrece expansión dinámica y eficiencia en inserción y eliminación.

Singly Linked Lists



Marco Teórico: Estructuras Pila y Cola

Pila (LIFO)

Last In, First Out: El último elemento en ser insertado es el primero en ser removido.

En la aplicación, se utiliza para almacenar el **historial de películas vistas**, permitiendo operaciones de **push** (apilar) y **pop** (desapilar).

Cola (FIFO)

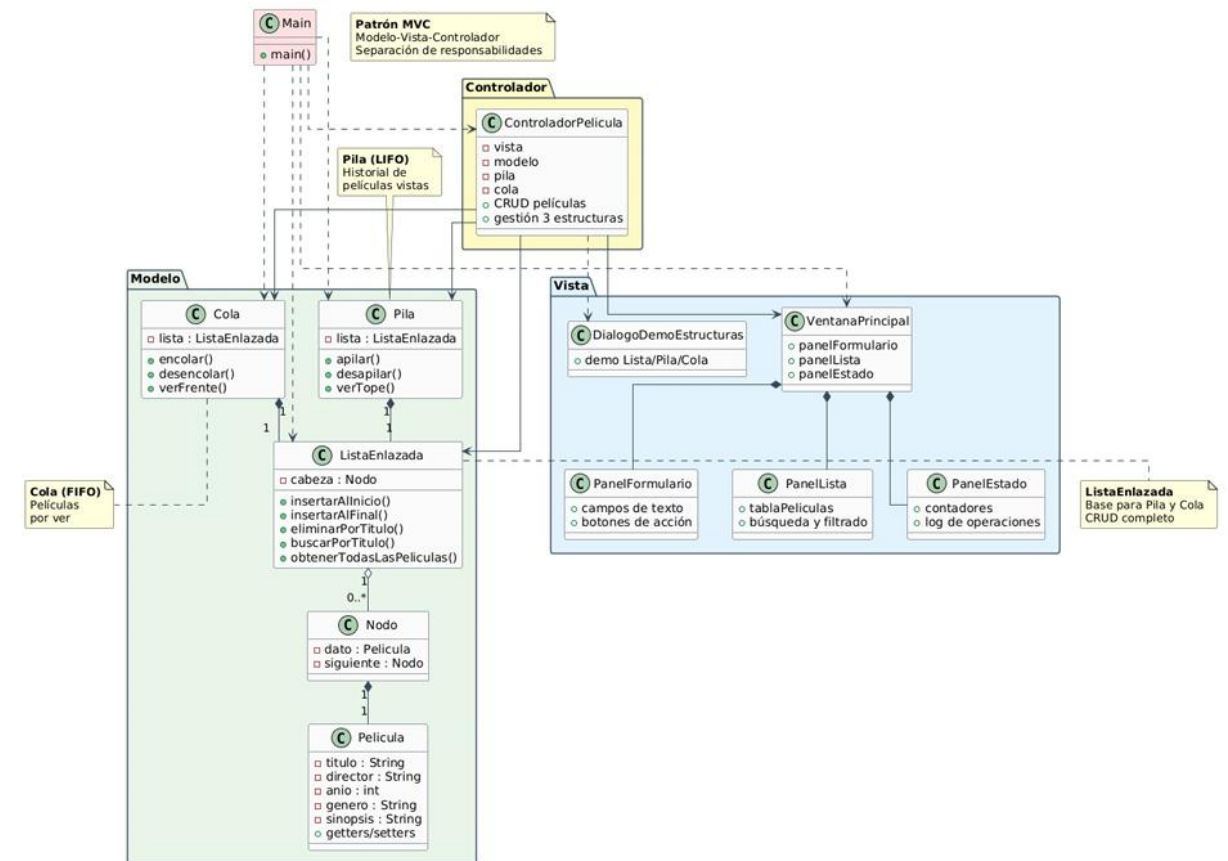
First In, First Out: El primer elemento en ser insertado es el primero en ser removido.

En la aplicación, gestiona una lista de **películas "por ver"**, implementando operaciones de **enqueue** (encolar) y **dequeue** (desencolar).

Marco Teórico: Patrón MVC

El patrón **Modelo-Vista-Controlador** es una arquitectura de diseño que separa la lógica de la aplicación en tres componentes interconectados:

- **Modelo:** Contiene la lógica del negocio y las estructuras de datos (ListaEnlazada, Pila, Cola, Pelicula).
- **Vista:** Presenta los datos al usuario y captura sus entradas (VentanaPrincipal, PanelFormulario).
- **Controlador:** Responde a los eventos del usuario y coordina las actualizaciones entre el Modelo y la Vista (ControladorPelicula).



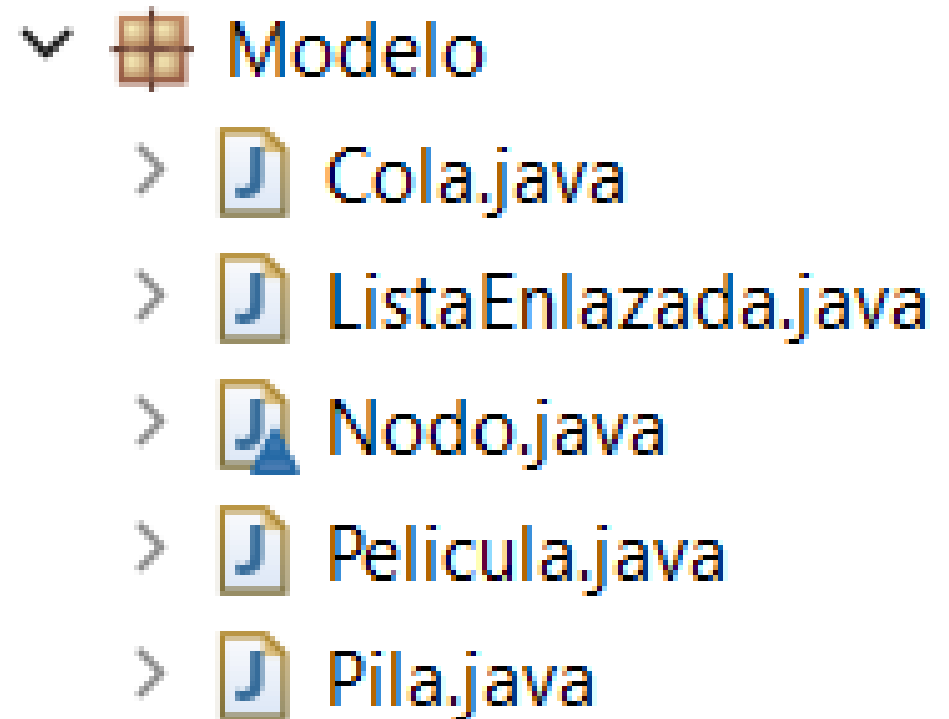
Implementación

Análisis de la Estructura del
Código

Implementación: El Modelo

El Modelo encapsula toda la lógica de datos:

- **Pelicula.java:** Entidad con atributos (título, director, año) y validaciones.
- **Nodo.java:** Almacena un `Pelicula` y una referencia `next`.
- **ListaEnlazada.java:** Corazón del modelo. Implementa:
 - `insertarAlInicio()` / `insertarAlFinal()`
 - `eliminarPorTitulo()`
 - `buscarPorTitulo()` / `obtenerTodasLasPeliculas()`
- **Pila.java y Cola.java:** Reutilizan la ListaEnlazada para implementar sus operaciones (apilar, encolar, etc.).



Implementación: Vista y Controlador

La Vista

Construida con Java Swing, se compone de múltiples `JPanel` dentro de un `JFrame` principal.

- **PanelFormulario:** Campos de texto y botones de acción.
- **PanelLista:** Contiene la `JTable` para mostrar el catálogo y la barra de búsqueda.
- **PanelEstado:** Muestra contadores y logs.
- **DialogoDemoEstructuras:** Ventana modal para la demo interactiva.

El Controlador

La clase `ControladorPelicula` actúa como el nexo:

- Implementa `ActionListener`, `ListSelectionListener` y `DocumentListener`
- Para reaccionar a los clics: ~~Para reaccionar a los clics~~ `guardarPelicula()`, `eliminarPelicula()`.
- Maneja la lógica de `moverAlInicio()` y `moverAlFinal()`.
- Actualiza la `JTable` mediante `filtrar()` y `actualizarTabla()`.

Demostración: Funcionalidad Principal

La interfaz permite un CRUD completo y una reorganización dinámica.

- **Inserción:** Se puede elegir insertar al inicio o al final.
- **Reorganización:** Los botones "Mover al Inicio" y "Mover al Final" eliminan y reinsertan el elemento en la posición deseada.
- **Búsqueda:** Filtrado en tiempo real por Título, Género o Año.
- **Validación:** Alertas para campos incompletos, año inválido (ej. '123' o '1800') o títulos duplicados.

The screenshot displays the 'Catálogo de Películas' application. On the left is the 'Formulario de Película' with fields for Title, Director, Year, Genre, and Synopsis, along with an 'Insertion Position' dropdown set to 'Insertar al Final'. Below the form are buttons for 'Guardar', 'Nuevo', 'Eliminar', 'Mover al Inicio', and 'Mover al Final', and a 'Demo Estructuras' button. On the right is the 'Catálogo de Películas' table with columns for Title, Director, Year, and Genre. The table lists four movies: Titanic, Guardians of the Galaxy, Avatar 2, and Five Nights at Freddy's. At the bottom, a 'Estado de Estructuras de Datos' section shows memory usage and a log of operations.

Título	Director	Año	Género
Titanic	James Cameron	1996	Romantica
Guardianes de la Ga...	James Gunn	2018	Acción, Superheroes
Avatar 2	James Gunn	2022	Acción, Aventura
Five Nights at freddy	Robin Piterson	2022	Miedo, Terror

Estado de Estructuras de Datos

Tamaño Lista: 4 Tamaño Pila: 0 Tamaño Cola: 0 Última: Formulario limpiado

Registro de Operaciones (más reciente primero):

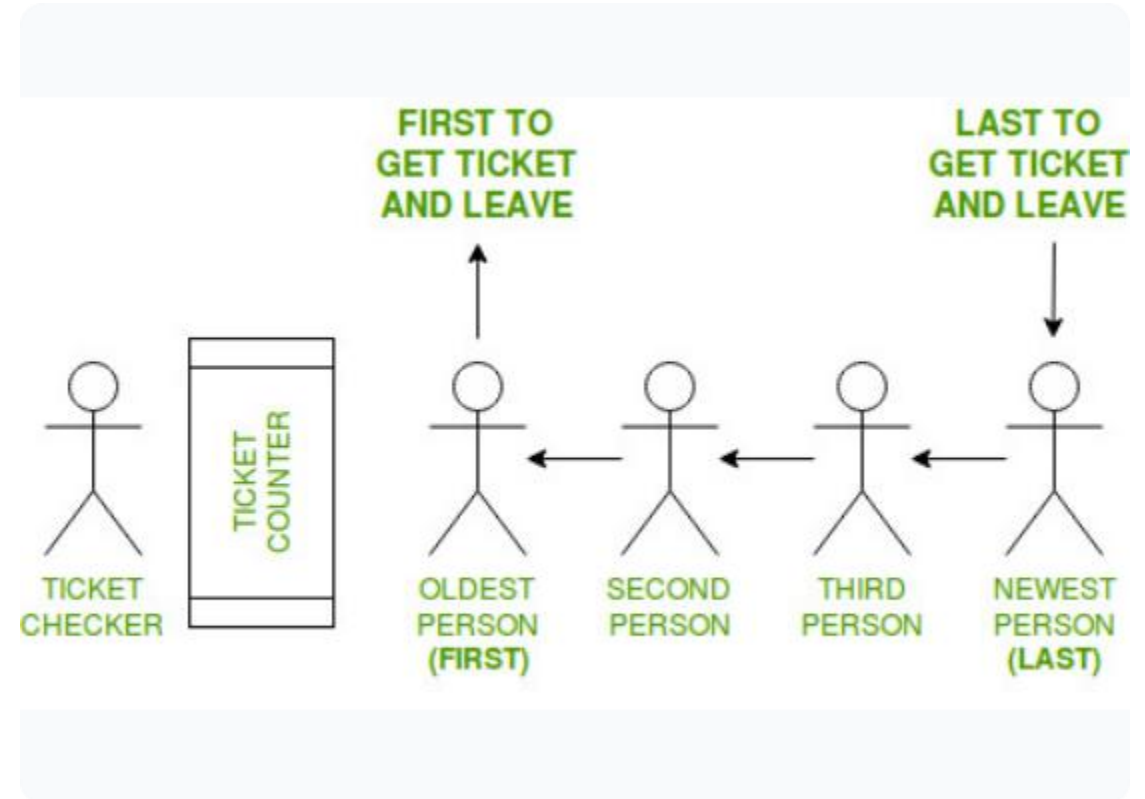
- 21:34:36 - Formulario limpiado
- 21:34:36 - Agregada: Guardianes de la Galaxia

Demostración: Demo de Estructuras

Un diálogo modal interactivo demuestra las 3 estructuras de datos en tiempo real:

- **Catálogo (ListaEnlazada):** La fuente de datos principal.
- **Vistas Recientemente (Pila LIFO):** Al "Ver" una película, se añade al historial (Push).
- **Cola Por Ver (Cola FIFO):** Se agregan películas a la "lista de espera" (Enqueue).

Las 3 estructuras comparten y manipulan los mismos datos del catálogo, demostrando su interconexión.



Conclusiones

- ✓ El patrón MVC facilitó la división de responsabilidades, mejorando la legibilidad, mantenibilidad y escalabilidad del código.
- ✓ Se demostró que la Lista Enlazada Simple es ideal para escenarios donde las inserciones y eliminaciones son frecuentes en los extremos.
- ✓ Se cumplieron los objetivos al implementar correctamente el Modelo (Lista, Pila, Cola), la Vista (GUI) y el Controlador (Eventos).
- ✓ La implementación de "mover al inicio/final" evidenció la flexibilidad de las listas para modificar el orden de elementos eficientemente.

Gracias

