**Mini proyectó**

Camilo Abel Sol Mosquera.

Septiembre 2023

Institución Universitaria Antonio José Camacho.

Valle del cauca.

Ingeniería de software ll

Patrón de diseño (ESTRUCTURAL):

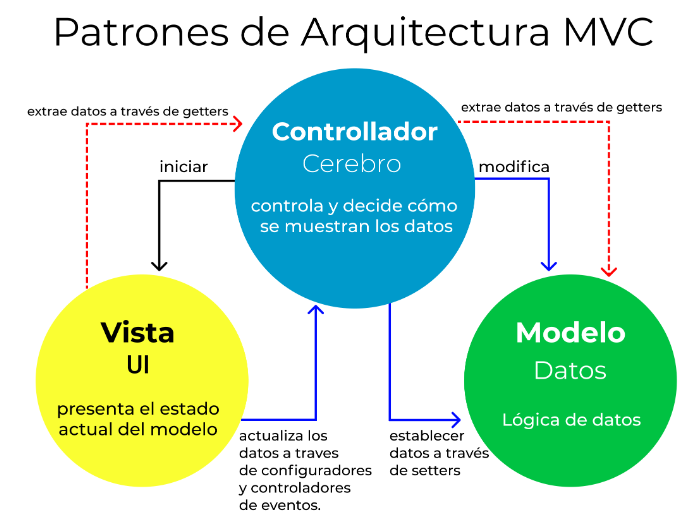
El patrón de diseño escogido por mi para la realización del miniproyecto fue el MVC (Modelo Vista Controlador)

Teniendo en cuenta que el MVC es un patrón de diseño muy conocido en el mundo de la programación web, me permite ajustarme de manera mas coda, ya que es un patrón que ya sea directa o indirectamente he usado en proyectos web anterior y actual mente, pero ¿qué es MVC (Modelo Vista Controlador)?

MVC (Modelo Vista Controlador)

MVC significa modelo (model) vista (view) controlador (controller). Esto es lo que significan cada uno de esos componentes.

* Modelo: El backend que contiene toda la lógica de datos
* Vista: El frontend o interfaz gráfica de usuario (GUI)
* Controlador: El cerebro de la aplicación que controla como se muestran los datos.



Ahora, este patrón esta aplicado de la siguiente manera:

Texto

Descripción generada automáticamente

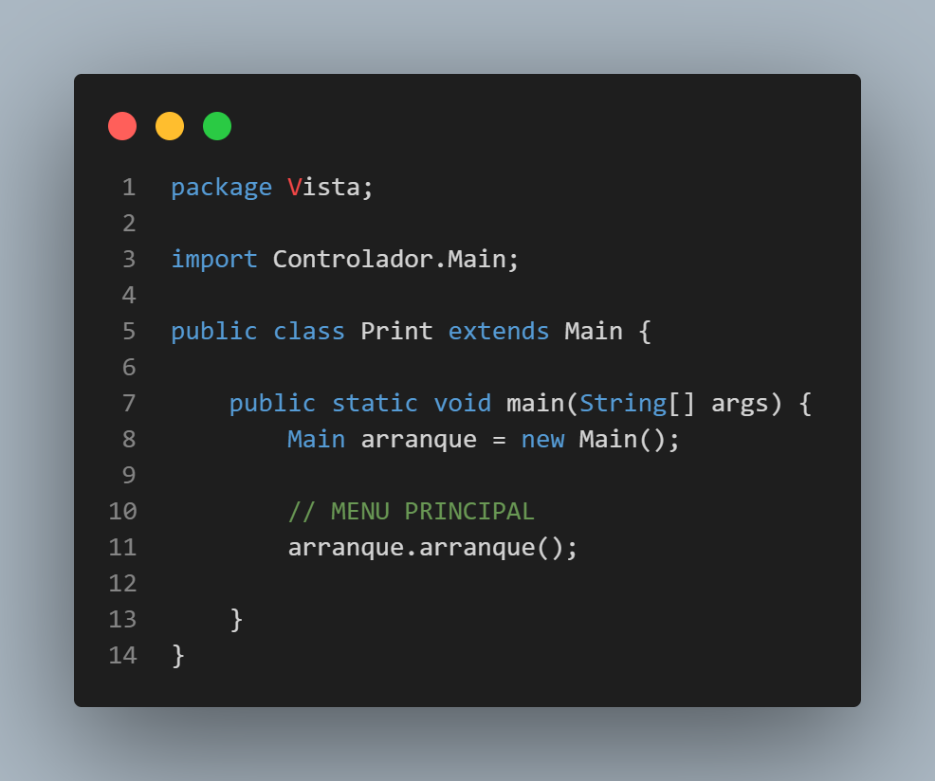
Podemos ver que contamos con carpetas separadas según el rol que cumple cada una de las clases del proyecto donde:

**Modelo:** Las clases Pelicula, Funcion, Sala, Silla y partes de la clase Cine que gestionan los datos (como la lista de películas y las funciones en las salas).

**Vista:** La vista no está claramente definida en el código proporcionado. Sin embargo, en una aplicación de escritorio basada en Java Swing, la vista podría estar compuesta por ventanas y componentes de interfaz de usuario por ende nuestra vista será la clase Print.

**Controlador:** La clase Main se encarga de la lógica de la aplicación y la interacción con el usuario, por lo que podría considerarse el controlador principal.

Código de clase de ejecución:



Patrón de diseño (CREACIONAL):

El patrón de diseño Prototype se utiliza para crear objetos duplicados o clonar objetos existentes sin depender de sus clases concretas. Esto permite crear nuevas instancias de objetos sin conocer los detalles de su implementación, se implementó el patrón Prototype en la clase Silla para permitir la clonación de objetos Silla. Para hacer esto, necesitamos agregar la interfaz Clonable a la clase y sobrescribir el método Clonable() para realizar la clonación.

La interfaz Cloneable es una interfaz marcadora que indica que la clase puede ser clonada.

Método clone() Este método se encargará de crear una copia exacta del objeto Silla.

Ahora, se puede utilizar el método clone() para crear copias exactas de objetos Silla.

Ejemplo de uso:

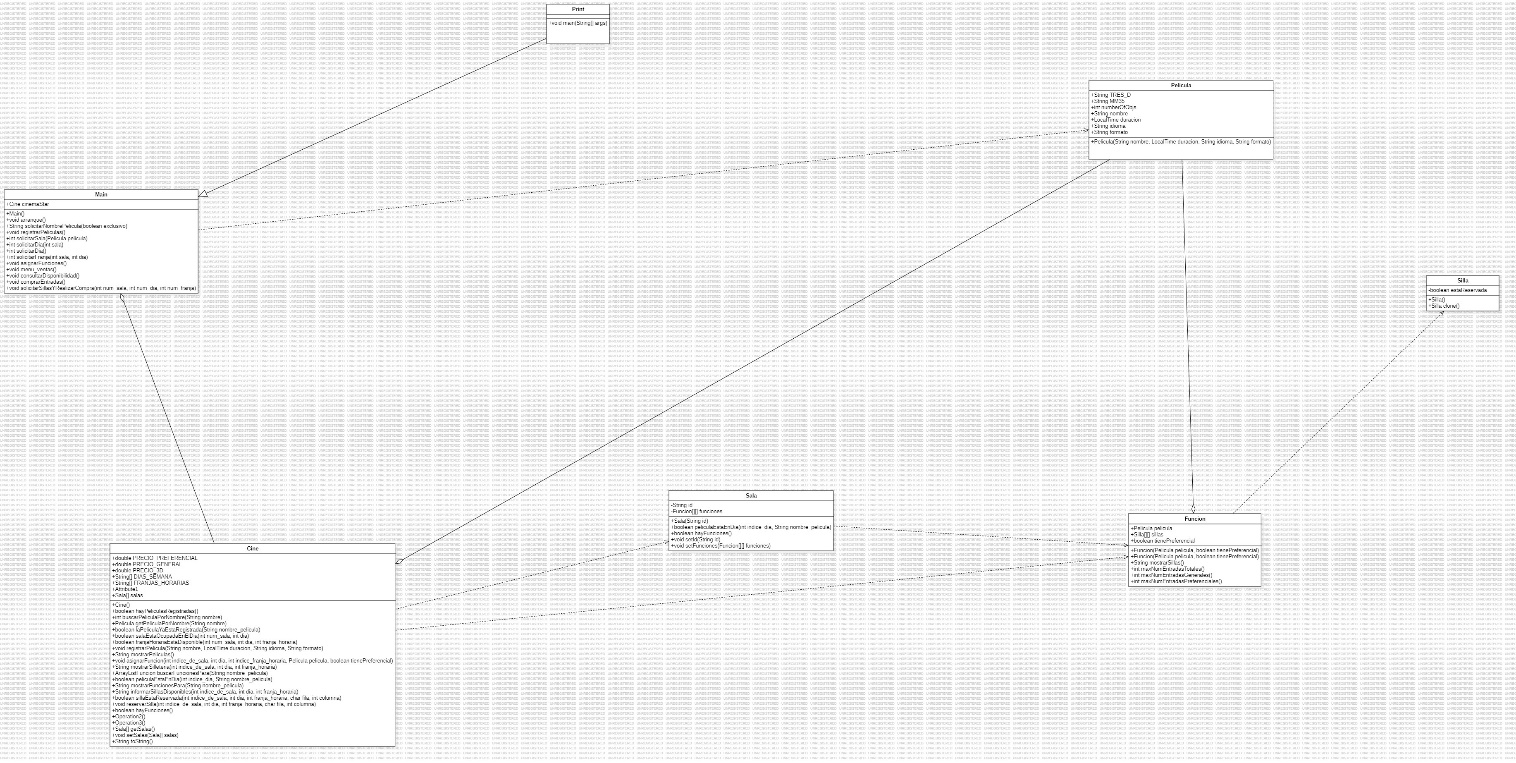
Silla sillaOriginal = new Silla();

Silla sillaClonada = sillaOriginal.clone();

Código silla:



Diagrama de clases:



Instrucciones

* Para el Miniproyecto usted deberá sustentar los patrones creacionales y estructurales que mejor se ajustan para el modelo propuesto de las Salas de Cine.
* Usted debe subir un documento con el diagrama de clases.
* La sustentación teórica de la implementación de los patrones que usó.
* El código fuente en JAVA que de soporte a los patrones.