

Informe - Cine Distrital

Docente

Miguel Alfonso Feijóo

Integrantes

Miguel Nicolás Díaz Vargas 20202020054

Jose Miguel Londoño 20202020064

Daniel Felipe Páez Mosquera 20202020070



BOGOTÁ D.C

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Facultad de Ingeniería

Ingeniería de Sistemas

2022 - 1

Requisitos Funcionales

Código	Nombre		Fecha	Grado de necesidad
RQF-001	Inicio de Sesión		17/08/2022	Alto
Descripción	El sistema debe permitir a los usuarios ingresar por medio de su usuario y contraseña			
Entradas	Fuente	Salida	Destino	Regla de negocio
Usuario Contraseña	Vista "inicio sesión"	De ser incorrectas las credenciales ingresadas da un mensaje "Usuario no encontrado" o "usuario encontrado satisfactoriamente"	Menú Cliente / Empleado / Admin	El sistema hace una verificación de las credenciales con la base de datos y da paso al menú y funcionalidades correspondientes, en el caso de que el usuario no se encuentre, imprime un mensaje.

Código	Nombre		Fecha	Grado de necesidad
RQF-002	Registrar Usuarios		24/07/2022	Alto
Descripción	El sistema debe permitir al usuario cliente registrarse por sí mismo y al usuario empleado registrar un cliente en el sistema.			
Entradas	Fuente	Salida	Destino	Regla de negocio
Cédula, nombre y contraseña.	Vista "Registrar Usuarios"	Mensaje "Ingrese todos los datos correctamente" o "Usuario registrado correctamente"	Base de datos y vuelta a la vista "Inicio Sesión"	Los usuarios registrados aquí tienen un rol de cliente, los cuales tienen la posibilidad de comprar boletas de cine, snacks y calificar películas que hayan visto. Para esto tienen que ingresar por el login y es por

				esto que deben pasar por una etapa de registro para poseer las credenciales y así tener acceso a estas funcionalidades. Este registro le envía los datos a la base de datos por medio de una inserción.
--	--	--	--	---

Código	Nombre		Fecha	Grado de necesidad
RQF-003	Menú Cliente		27/07/2022	Alto
Descripción	El sistema debe mostrarle al usuario cliente las funcionalidades que posea, como lo son comprar boletos y snacks así como calificar películas y la posibilidad de cerrar su sesión.			
Entradas	Fuente	Salida	Destino	Proceso
Selección de multiplex y botón presionado	Vista "Menú Cliente"		Vista Selección de Función, Visa Snacks o Vista Calificar	Dependiendo del botón que se elija se direcciona a la ventana correspondiente, en el caso de que sea Boletos o Tienda, toma el Multiplex seleccionado para mostrar en la siguiente vista datos relacionados con este multiplex.

Código	Nombre		Fecha	Grado de necesidad
RQF-004	Selección de función		01/08/2022	Media
Descripción	El sistema permite al usuario seleccionar la película, hora, fecha, el tipo de silla (preferencial o general), y cantidad de boletas a comprar según disponibilidad			
Entradas	Fuente	Salida	Destino	Regla de negocio
Películas, fecha, hora, tipo silla y cantidad de sillas	Vista "Vista Función"		Vista de Selección de Sillas	Con las entradas se configurará la siguiente ventana donde se seleccionaran las sillas según la función elegida.

Código	Nombre		Fecha	Grado de necesidad
RQF-005	Selección de Sillas		02/08/2022	Alto
Descripción	El sistema debe permitir al usuario seleccionar las sillas según la cantidad elegida en la ventana anterior y las sillas disponibles y no disponibles.			
Entradas	Fuente	Salida	Destino	Regla de negocio
Sillas seleccionadas	Vista "Selección de Sillas"	Mensaje "No puedes seleccionar más asientos"	Vista de "Recibo Boletas"	Se traen datos de las sillas no disponibles desde la base de datos y según la cantidad de sillas a comprar seleccionadas en la vista anterior, el programa permite al cliente seleccionar la cantidad de sillas disponibles. Al seleccionar el número de sillas seleccionadas se deja dar paso a la siguiente ventana.

Código	Nombre		Fecha	Grado de necesidad
RQF-006	Recibo de Boletas		02/08/2022	Alto
Descripción	El sistema debe permitir al usuario ver su recibo con los datos de la función y las sillas seleccionada.			
Entradas	Fuente	Salida	Destino	Regla de negocio
Función, usuario y sillas	Vista "Recibo Boletas"	Mensaje "Pago realizado correctamente"	Base de datos y Vista "Menú de Cliente"	Con todas las entradas se genera la petición para inyectar la base de datos.

Código	Nombre		Fecha	Grado de necesidad
RQF-007	Ventana Snacks		03/08/2022	Alto
Descripción	El sistema debe mostrarle al usuario cliente los snacks disponibles dependiendo del multiplex seleccionado y la posibilidad de comprar.			
Entradas	Fuente	Salida	Destino	Proceso
Cantidad a comprar de cada snack	Vista "Snacks"		Vista "Recibo Snacks"	Se hace una consulta a la base de datos para adquirir la cantidad máxima disponible de cada snack.

Código	Nombre		Fecha	Grado de necesidad
RQF-008	Recibo Snacks		03/08/2022	Alto
Descripción	El sistema debe mostrarle al usuario cliente el recibo que dice la cantidad y el total de los snacks.			
Entradas	Fuente	Salida	Destino	Proceso
Cantidad de snacks	Vista "Recibo de Snacks"		Base de datos y Vista "Menú de Cliente"	Dependiendo de la cantidad de snacks se hace una consulta de base de datos

				para cambiar la cantidad disponible de cada snack.
--	--	--	--	--

Código	Nombre		Fecha	Grado de necesidad
RQF-009	Calificar Películas		3/08/2022	Alto
Descripción	El sistema debe permitir a los usuarios calificar las películas que hayan visto anteriormente.			
Entradas	Fuente	Salida	Destino	Regla de negocio
Calificación de película	Vista "Calificación de Películas"		Base de Datos y vista "Menú Cliente"	Se hace una consulta con la base de datos para saber las películas que el usuario ha visto, con esto el usuario puede calificar de 0 a 5 las películas vistas.

Código	Nombre		Fecha	Grado de necesidad
RQF-010	Menú Administrador		7/08/2022	Alto
Descripción	El sistema debe mostrarle al usuario administrador las funcionalidades que posea, como lo son buscar empleados, cambiar cargo de empleado y añadir empleados.			
Entradas	Fuente	Salida	Destino	Regla de negocio
Botón presionado	Vista "Menú de Administrador"		Vista "Ver Empleado" o "Registro Empleados" o "Cambiar cargo empleado"	Dependiendo del botón que se elija se direcciona a la ventana correspondiente.

Código	Nombre		Fecha	Grado de necesidad
RQF-011	Buscar Empleado		7/08/2022	Alto
Descripción	El sistema debe permitir al usuario administrador			
Entradas	Fuente	Salida	Destino	Regla de negocio
Código Empleado	Vista "Buscar Empleado"	Mensaje "Empleado no encontrado".		Con el código de empleado ingresado se hace la consulta para encontrar el empleado, de encontrar, muestra sus datos, en caso contrario muestra mensaje.

Código	Nombre		Fecha	Grado de necesidad
RQF-012	Cambiar Cargo Empleado		7/08/2022	Alto
Descripción	El sistema debe permitir al usuario administrador la posibilidad de cambiar el cargo del empleado.			
Entradas	Fuente	Salida	Destino	Regla de negocio
Cargo nuevo del empleado y Código de Empleado	Vista "Cambiar Cargo Empleado"	Mensaje "No deje campos vacíos" o "Usuario no encontrado"	Base de Datos	Con el código del empleado, se busca el empleado en la base de datos por petición,

Código	Nombre		Fecha	Grado de necesidad
RQF-013	Añadir Empleado		8/03/2022	Alto
Descripción	El sistema debe permitir al usuario administrador la posibilidad de añadir empleado.			
Entradas	Fuente	Salida	Destino	Regla de negocio
Cédula, código, teléfono, cargo salario, fecha de contratación y multiplex de trabajo	Vista "Registro Empleado".	Mensaje "No deje espacio en blanco".	Base de datos.	Con los datos de entrada se hace una inserción a la base de datos para crear un nuevo empleado.

Código	Nombre		Fecha	Grado de necesidad
RQF-014	Menú Empleado		10/03/2022	Alto
Descripción	El sistema debe permitir al usuario empleado las funcionalidades que posea, como lo son buscar venta de boletos y snacks y registrar cliente.			
Entradas	Fuente	Salida	Destino	Regla de negocio
Botón presionado	Vista "Menú Registrar Cliente"		Vista Selección de Función, Vista Snacks o Vista Registrar Cliente	Dependiendo del botón que se elija se direcciona a la ventana correspondiente.

REQUISITOS NO FUNCIONALES

SISTEMA CONTROL DE AMBIENTES			
SRS – Especificación de Requerimientos			
Código	Nombre	Fecha	Grado Necesidad
RQF-015	Software	24/07/2022	Alto
Descripción	Disponer de un computador con los software necesarios para compilar y arrancar el programa.		

SISTEMA CONTROL DE AMBIENTES			
SRS – Especificación de Requerimientos			
Código	Nombre	Fecha	Grado Necesidad
RQF-016	Interfaz de usuario	10/08/2022	Medio
Descripción	El programa debe contar con una interfaz intuitiva y fácil de entender que permita que el usuario entienda a simple vista cómo funciona el aplicativo y pueda hacer uso de todas las funcionalidades que se le otorga por su rol de usuario. Así mismo el aplicativo debe contar con diseño y colores agradables para el usuario.		

SISTEMA CONTROL DE AMBIENTES			
SRS – Especificación de Requerimientos			
Código	Nombre	Fecha	Grado Necesidad
RQF-017	Aplicación de patrones de diseño	24/07/2022	Alto
Descripción	El programa debe contar con la aplicación de los patrones de diseño que según nuestro criterio pueda ayudar a que el código del programa posea una mejor estructura, prácticas de programación y orden.		

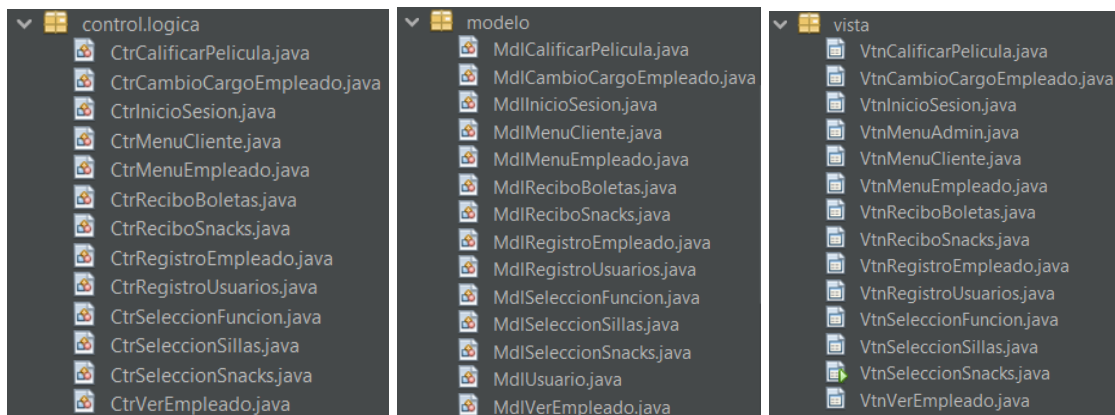
SISTEMA CONTROL DE AMBIENTES			
SRS – Especificación de Requerimientos			
Código	Nombre	Fecha	Grado Necesidad
RQF-018	Login	25/03/2022	Alto
Descripción	El programa cuenta con una validación de datos de usuario para así cumplir con los estándares de seguridad básicos para el manejo de la aplicación y su respectiva base de datos. Así mismo este login debe garantizar que cada usuario solo pueda hacer las funcionalidades que se le hayan dado según su rol.		

Justificación de los Patrones de Diseño Adoptados en el Desarrollo del Proyecto

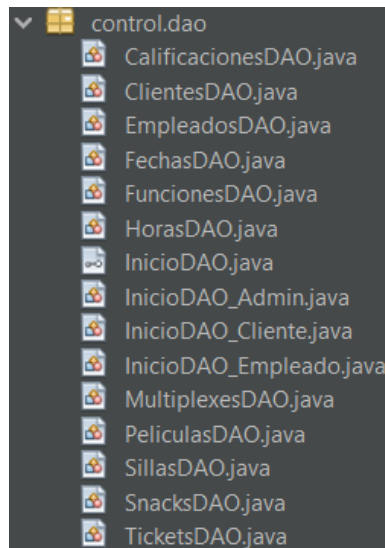
Patrón Singleton: Este patrón de diseño creación busca crear una única instancia de una clase para así evitar la ejecución múltiple de procedimientos del mismo, para esto se crea un punto de acceso global a la clase. Este patrón decidimos implementarlo en la clase Conexión, esto porque solo es necesario una única conexión a una única base de datos. Aplicando este patrón de diseño evitamos tener múltiples conexiones a la base de datos y generar posibles corrupciones de datos en la misma.

```
public class Conexion {  
    private static Connection cn = null;  
  
    // SINGLETON - retorna una unica instancia de la conexion  
    public static Connection getConexion() {  
        if (cn == null) {  
            new Conexion();  
        }  
  
        return cn;  
    }  
}
```

Patrón Mediador y Modelo - Vista - Controlador: El patrón Mediador busca reducir las dependencias entre objetos. Para esto restringe las comunicaciones directas entre ellos, obligándolos a comunicarse únicamente a través de un objeto mediador. El estilo de arquitectura de software Modelo-Vista-Controlador aplica este patrón donde el mediador viene siendo el controlador el cual restringe que objetos de la vista y los de los modelos se comuniquen entre sí únicamente por esta clase. Se aplicó esta arquitectura de software para que a su vez se tuviera una aplicación del patrón mediador y se redujeran en gran medida las dependencias entre objetos, ya que se vio que los alcances del proyecto llevaría a tener gran cantidad de clases y vistas.



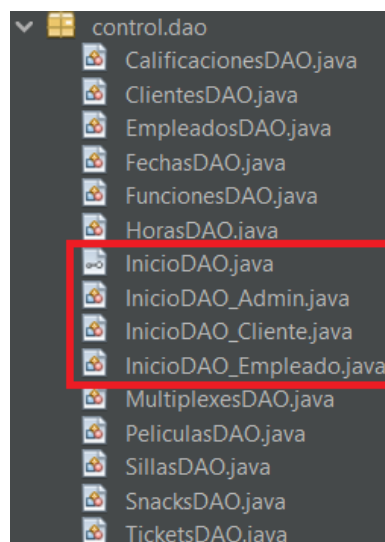
Patrón DAO: El patrón DAO es un patrón de diseño que suministra una clase para la comunicación común para varias clases para el almacenamiento de datos, tales como una base de datos o un archivo. Este patrón fue implementado para manejar las operaciones hacia la base de datos con sus respectivas sentencias SQL. Se creó una clase DAO independiente para cada tabla, para que de esta manera cada clase DAO posea las peticiones correspondientes a su tabla.



Inyección de Dependencias: este patrón de diseño hace que se suministre objetos a una clase en lugar de ser la propia clase la encargada instanciar dichos objetos. Estos objetos son objetos necesarios para el buen funcionamiento de la clase (de ahí el concepto de dependencia).

```
public class CtrInicioSesion implements ActionListener {  
  
    private MdlInicioSesion mdlInicioSesion;  
  
    public CtrInicioSesion(VtnInicioSesion aThis) {  
        mdlInicioSesion = aThis.getMdlInicioSesion();  
    }  
}
```

Patrón Strategy: El patrón de diseño estructural Strategy es un patrón que permite definir una interfaz y hacer los objetos de sus implementaciones intercambiables dependiendo de la tarea a resolver. Este patrón lo vimos muy útil a la hora del inicio de sesión en donde se requiere que según el tipo de usuario se despliegue una interfaz distinta con sus respectivas funciones.



Como podemos ver, en la clase MdInicioSesion tenemos una interface InicioDAO e implementaciones concretas como lo son las de Administrador, Cliente y Empleado, con esto ya podemos intercambiar el tipo de implementaciones.

```
// STRATEGY -----  
public int getCliente(String usuario, String contraseña) {  
    inicioDAO = new InicioDAO_Cliente();  
    return inicioDAO.comprobarUsuario(usuario, contraseña);  
}  
  
public int getAdmin(String usuario, String contraseña) {  
    inicioDAO = new InicioDAO_Admin();  
    return inicioDAO.comprobarUsuario(usuario, contraseña);  
}  
  
public int getEmpleado(String usuario, String contraseña) {  
    inicioDAO = new InicioDAO_Empleado();  
    return inicioDAO.comprobarUsuario(usuario, contraseña);  
}
```

En nuestra clase CtrInicioSesion tenemos un método actionPerformed en el que tenemos la lógica de cuando el botón de iniciar sesión es oprimido empieza a verificar por cada tipo de usuario para ver en cuál tipo recae el usuario que se está registrando, en cada cambio de verificación intercambia implementaciones de la clase InicioDAO.

Reflexión Final

- **Miguel Nicolás Díaz Vargas:** El desarrollo de este proyecto hizo que reforzaré y afianza lo aprendido sobre los patrones de diseño así como sus tipos, puesto que solo el saber sus significados y para qué servían no era suficiente, sino que me vi obligado a de verdad aplicarlos en código, evidenciando en tiempo real los beneficios que generan solo la implementación de los contados patrones que implementamos en el desarrollo del proyecto.
- **Jose Miguel Londoño Blandon:** La solución de este proyecto me deja como enseñanza, el uso práctico de los modelos de programación vistos en clase, ya que al momento, de tener problemas en vez de reinventar una solución, vi que identifique rápidamente que modelo o patrón de diseño se podía usar, y así lograr hacer un desarrollo más universal aceptado por la comunidad, que posee buenas prácticas y mantiene una alta cohesión y bajo acoplamiento en su desarrollo.
- **Daniel Felipe Páez Mosquera:** Trabajar en este proyecto me ayudó a afianzar los conocimientos adquiridos en clase, de forma que pude implementar junto con mis compañeros los patrones de diseño, esto nos sirvió para empezar a tener unas buenas prácticas de programación siguiendo el principio de alta cohesión y bajo acoplamiento, además de que fue una experiencia cercana de un ambiente laboral real, donde tendremos que trabajar junto con otros ingenieros y poder entendernos todos.