



---

# <Universidad Técnica del Norte>

---

FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS  
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

## **Sistema de Compra de Tickets de Cine**

## **Documento de la Arquitectura del Software**

**Versión 2.0**

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

## Historia de Revisión

Fecha	Versión	Descripción	Autor
03/09/2020	1.0	Versión inicial del documento, previa a la Fase de Construcción del proyecto	Ronny Diaz James Scarberry Diana Carlosama
10/09/2020	2.0	Versión inicial del documento, previa a la Fase de Construcción del proyecto	Ronny Diaz James Scarberry Diana Carlosama

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

## Tabla de Contenidos

1.	Introducción	4
1.1	Propósito	4
1.2	Alcance	4
1.3	Definiciones, Siglas, y Abreviaciones	4
1.4	Referencias	4
1.5	Vista Global	4
2.	Representación Arquitectónica	4
3.	Metas y Restricciones Arquitectónicas	5
4.	Vista de Casos de Uso	5
5.	Vista Lógica	6
5.1	Visión general	6
5.1.1	<b>Diagrama de Conceptual (Modelo de Dominio)</b>	6
5.1.2	<b>Diagrama de Clases</b>	7
5.2	Realizaciones de los casos de uso	8
5.2.1	<b>Crear Cuenta</b>	8
5.2.2	<b>Iniciar Sesión</b>	8
5.2.3	<b>Cerrar Sesión</b>	9
5.2.4	<b>Registrar pagos</b>	9
5.2.5	<b>Hacer Reservación</b>	9
6.	Vista de Implantación	10
6.1	Configuración Estándar	10
7.	Vista de Implementación	11
7.1	Vista General	11
7.2	Capas	11
7.2.1	<b>Vista</b>	11
7.2.2	<b>Controlador</b>	12
7.2.3	<b>Modelo</b>	13
8.	Vista de Datos	13
8.1	Diagrama Entidad-Relación (ER)	13
8.2	Diccionario de Datos	14
9.	Tamaño y Desempeño	16
10.	Calidad	17
11.	Figura Arquitectónica	18

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

# Documento de la Arquitectura del Software

## 1. Introducción

### 1.1 Propósito

La elaboración del documento presente tiene como propósito presentarles a los futuros usuarios del sistema CTC una visión clara de los distintos aspectos que conforman su arquitectura. El Documento de Arquitectura del Software busca el aumento de la productividad del equipo y sentar las bases de la arquitectura del software, con el fin de facilitar las actividades de análisis y toma de decisiones que puedan llevarse a cabo en un futuro.

### 1.2 Alcance

Definir y unificar los criterios de desarrollo del Sistema CTC con respecto a los aspectos arquitectónicos del proyecto, mostrando los detalles sobre los paquetes, capas y procesos que conformaran el sistema.

### 1.3 Definiciones, Siglas, y Abreviaciones

A continuación, presentamos las abreviaturas y definiciones de los términos de mayor importancia que se encuentran en el documento.

**JSF**: Java Server Faces.

**JEE**: Java Enterprise Edition.

**MVC**: Modelo – Vista – Controlador. Estilo de arquitectura de software, frecuentemente visto en aplicaciones Web, que divide los procesos en tres capas.

**PostgreSQL**: Base de datos relacional Postgres.

**JAVA**: lenguaje de programación semicompilado, diseñado para la construcción de sistemas web.

### 1.4 Referencias

- Documento de Visión del Sistema (v 1.0 – 02/05/2010).
- Documento de Especificación de Requerimientos del Sistema (v 1.0 – 19/05/2010).

### 1.5 Vista Global

En las siguientes diez (10) secciones del documento se detalla la arquitectura del Sistema CTC a desarrollarse. Se hará hincapié en los detalles de las distintas vistas (lógica, casos de uso, procesos, implementación e implantación) que conformaran la arquitectura del software (además de otros aspectos relevantes), empleando el lenguaje más sencillo posible y haciendo uso de diagramas donde se considere necesario con el fin de expresar las ideas de la forma más concisa.

## 2. Representación Arquitectónica

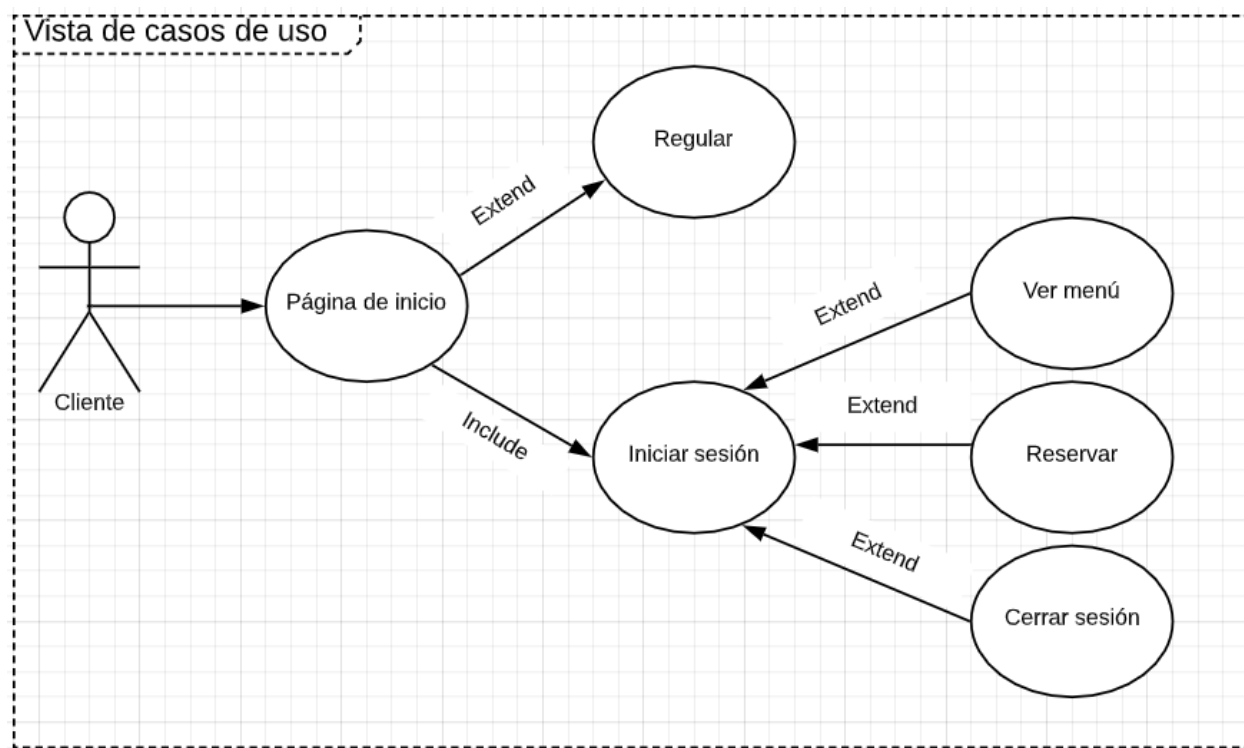
La representación arquitectónica del Sistema CTC se encuentra basada en el modelo de **4+1 Vistas** de Philippe Kruchten. En el transcurso del documento se tratará más a fondo los detalles de cada una.

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

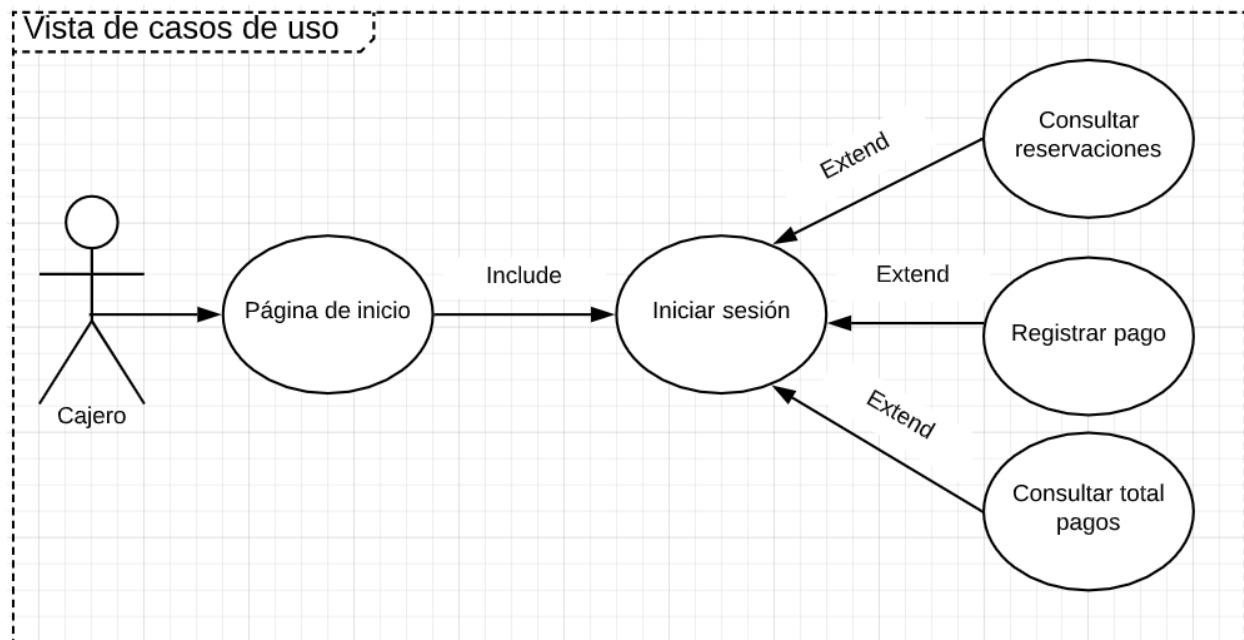
### 3. Metas y Restricciones Arquitectónicas

Con el desarrollo del sistema CTC se busca obtener una aplicación con un alto nivel de portabilidad, de fácil distribución y con un alto nivel de reusabilidad. En sus etapas iniciales, la aplicación será montada en el local host conectada a una base de datos remota cloudFICA, pero se busca en futuras versiones montarla en un servidor de aplicaciones usando los recursos de la UTN. Entre las restricciones principales del desarrollo del sistema CTC se encuentran: hacer uso de tecnologías de software libre (en nuestro caso, el lenguaje JAVA en su versión Enterprise y la base de datos PostgreSQL) y el uso del patrón MVC para el desarrollo de la herramienta (impuesto por el LISI). Por último, se tiene como meta lograr que el sistema pueda ejecutarse sin percances en los navegadores web Microsoft Internet Explorer, Opera, Google Chrome, Edge, Mozilla Firefox, con el fin de poder serle útil a la mayor cantidad de usuarios posible.

### 4. Vista de Casos de Uso



C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

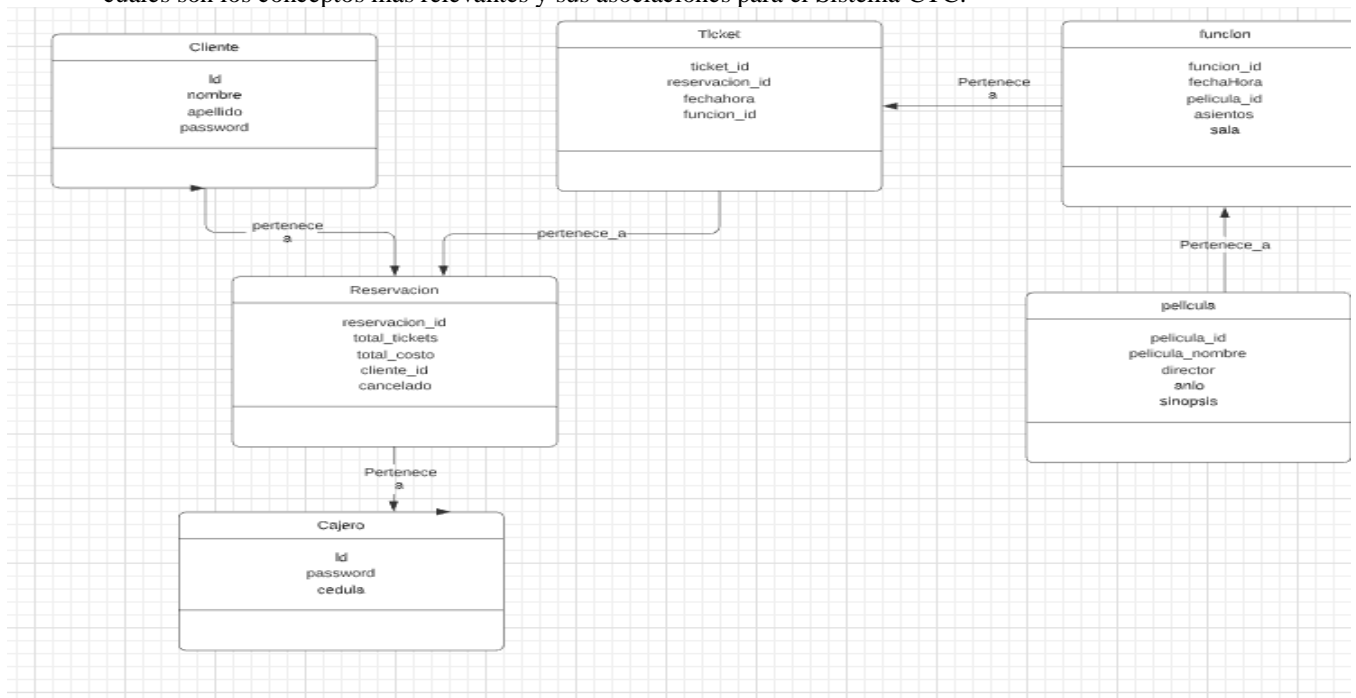


## 5. Vista Lógica

### 5.1 Visión general

#### 5.1.1 Diagrama de Conceptual (Modelo de Dominio)

A continuación, proponemos un Modelo Conceptual (diagrama de Modelo de Dominio) para demostrar cuales son los conceptos más relevantes y sus asociaciones para el Sistema CTC.



*Diagrama de Modelo del Dominio*

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

### 5.1.2 Diagrama de Clases

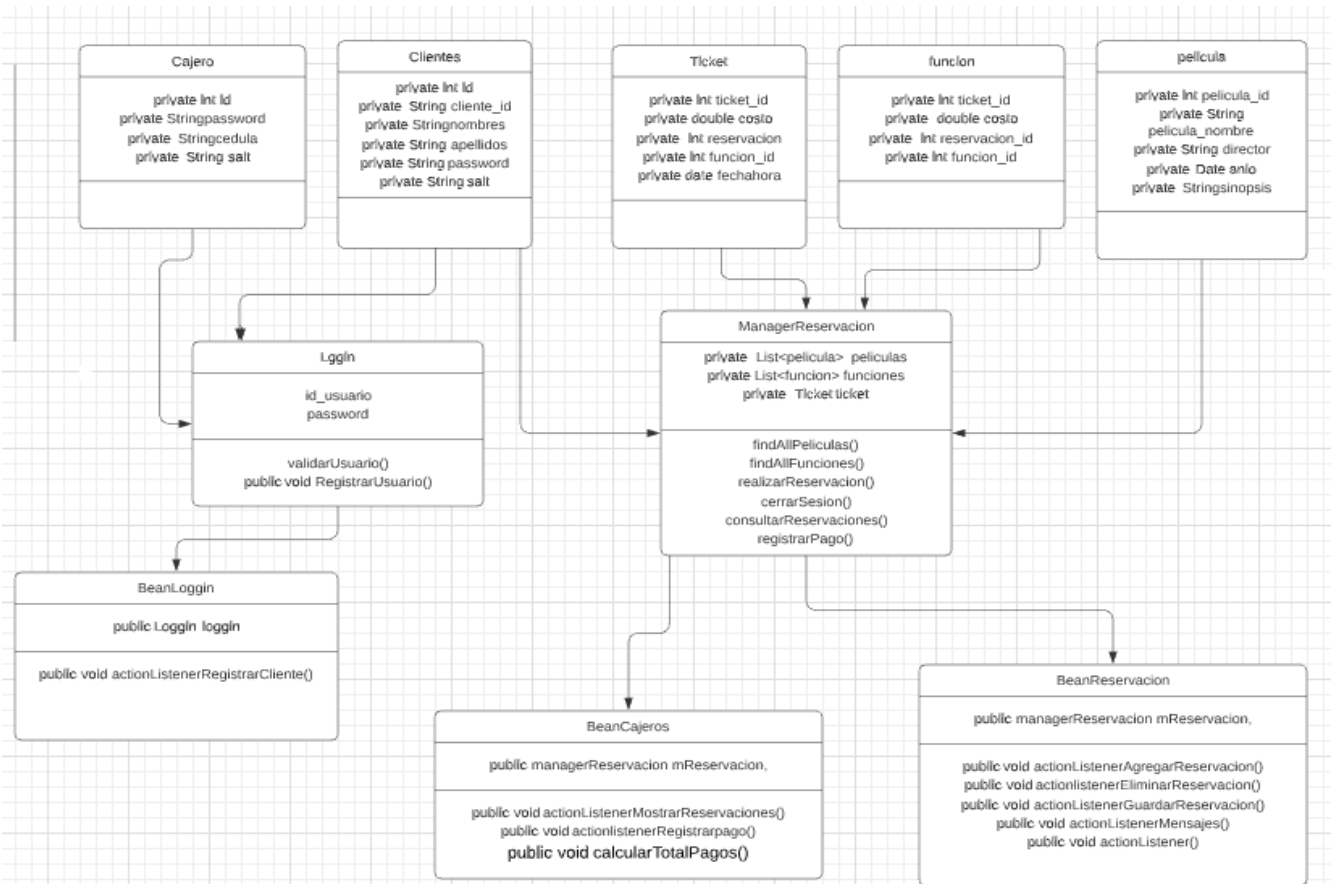
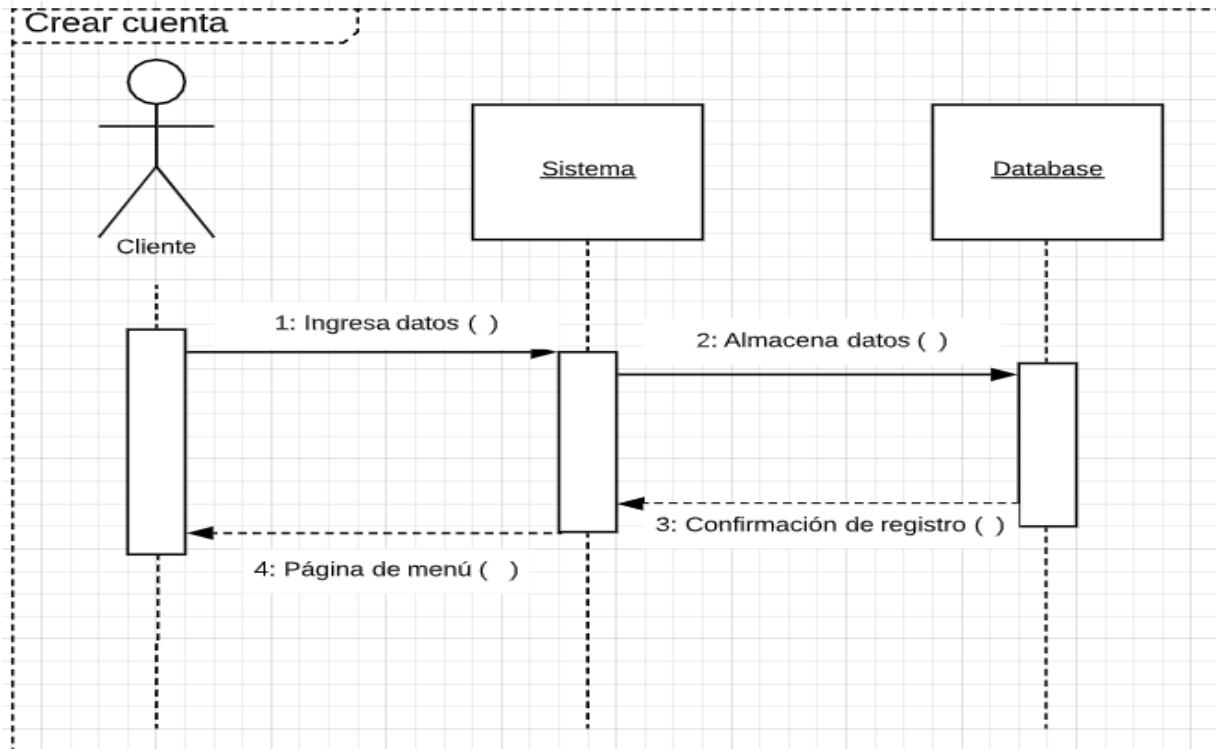


Diagrama de Clases

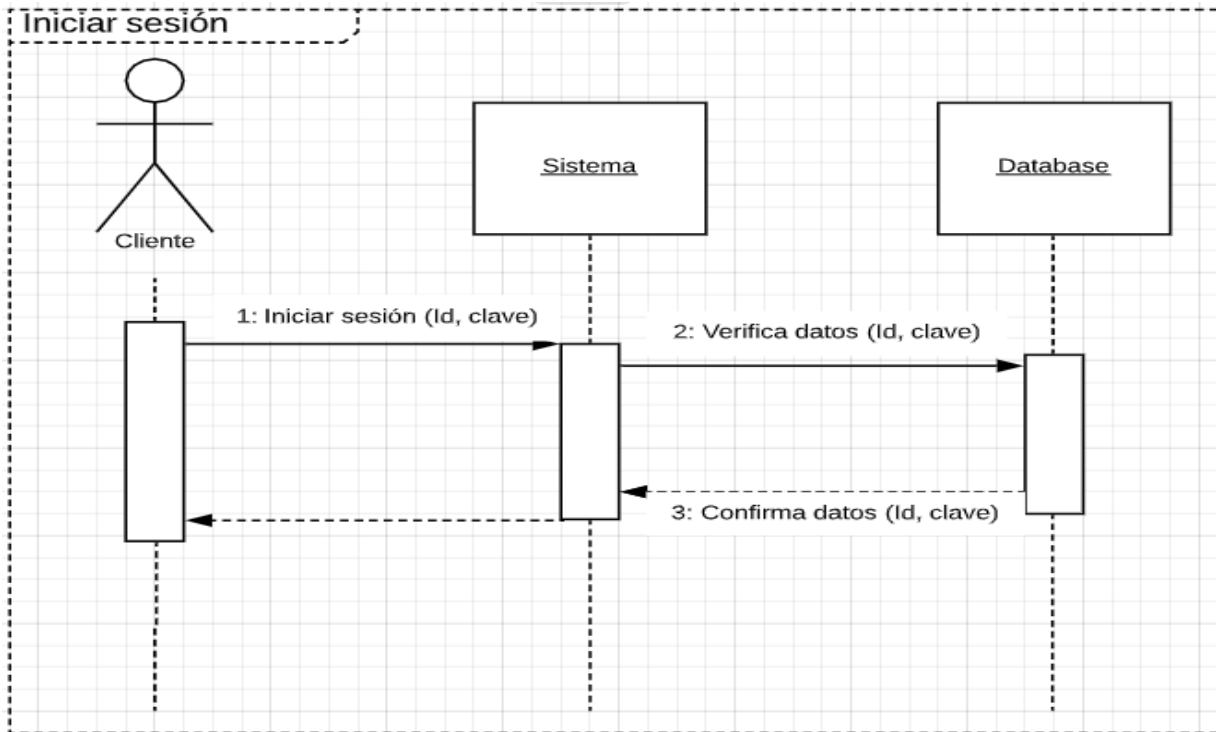
C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

## 5.2 Realizaciones de los casos de uso

### 5.2.1 Crear Cuenta



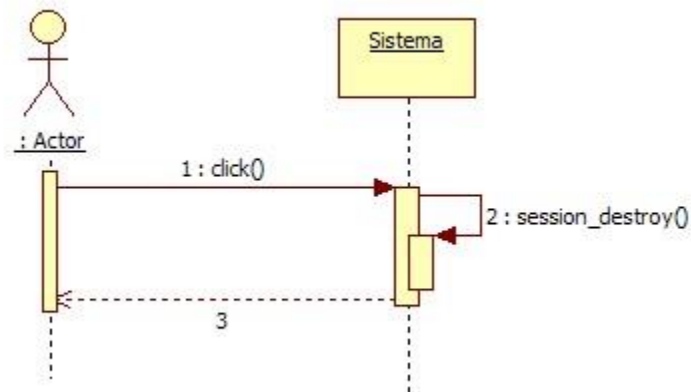
### 5.2.2 Iniciar Sesión



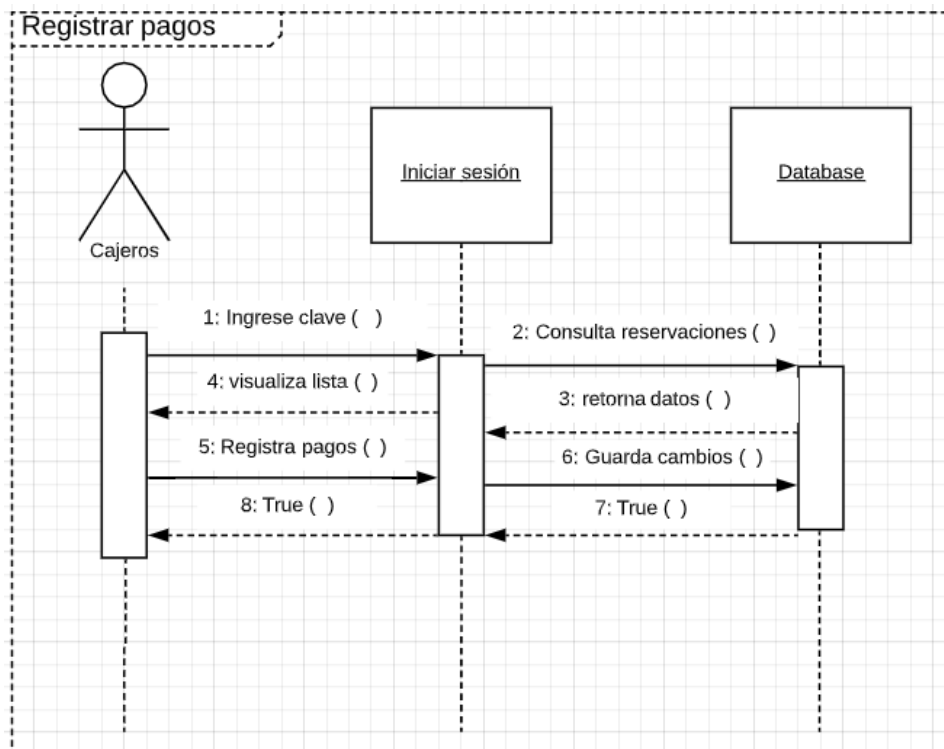


C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

### 5.2.3 Cerrar Sesión

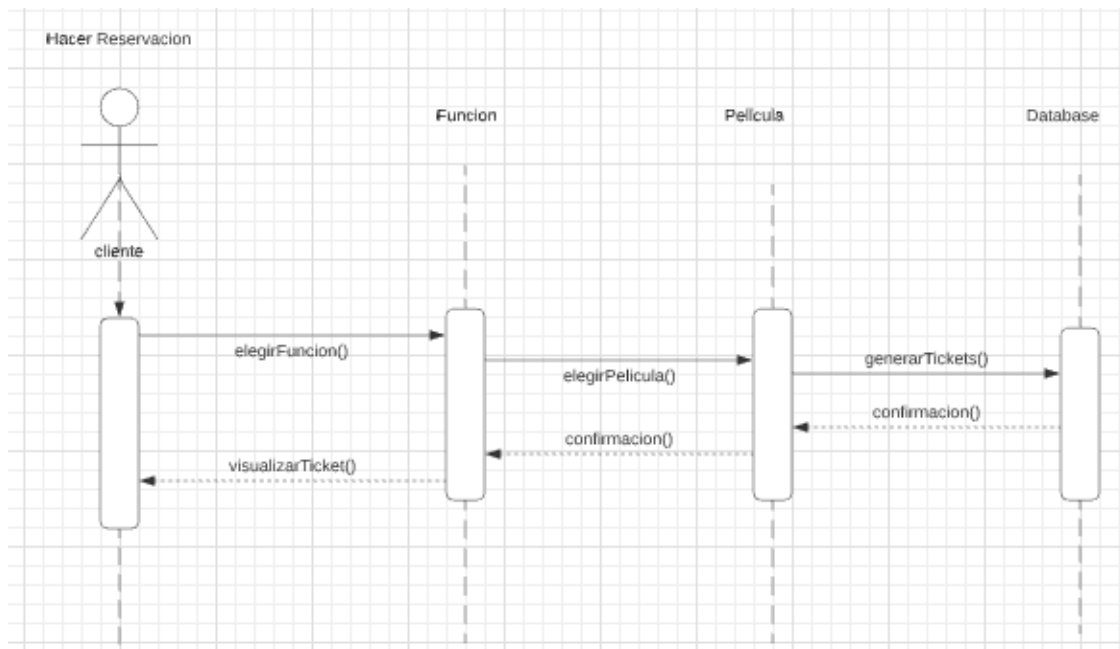


### 5.2.4 Registrar pagos



### 5.2.5 Hacer Reservación

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	



## 6. Vista de Implantación

### 6.1 Configuración Estándar

El Sistema CTC se está desarrollando con una configuración estándar sencilla. se hospedará y ejecutará desde un servidor local, junto con el manejador de base de datos cloudFICA asignados para cada grupo de trabajo y los datos de la aplicación. Los usuarios tendrán acceso al sistema mediante el uso de estaciones de trabajo conectadas directamente al servidor (mediante una red interna, por ejemplo). Dichos clientes deben ejecutar un navegador web para poder hacer uso del sistema. Se recomienda tener un servidor de respaldo, el cual monitoree el servidor principal y pueda detectar cuando haya fallas en el mismo.

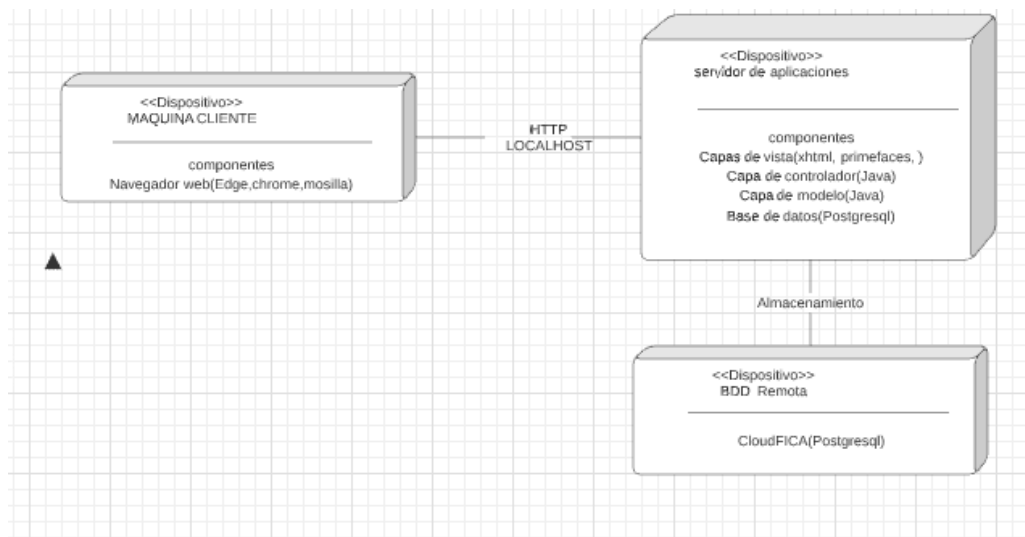


Diagrama de la Implantación Estándar del Sistema CTC

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

## 7. Vista de Implementación

### 7.1 Vista General

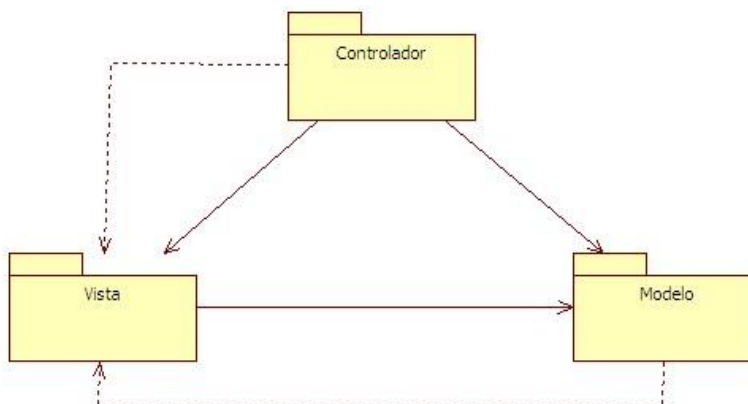
El Sistema CTC se desarrollará, a petición de los requerimientos de los docentes encargados, bajo el patrón conocido como **MVC**, el cual dividirá nuestro software en tres (3) capas principales. Con el uso de dicho patrón se contará con un mejor entendimiento del sistema y facilitará las labores de desarrollo y mantenimiento de la aplicación.

Como se mencionó anteriormente, el patrón **MVC** se encarga de separar la aplicación en 3 capas, conocidas como el Modelo, la Vista y el Controlador.

El Modelo (también conocida como la capa de datos) es la capa en donde reside la información que maneja el sistema CTC. Estará conformado por un gestor de bases de datos (en nuestro caso, PostgreSQL) para realizar el almacenamiento de datos. Esta capa deberá responder a las peticiones de información de estado de parte de la capa de Vista y a las instrucciones de modificación de estado, provenientes de la capa de Controlador.

La Vista (también conocida como la capa de aplicación) es la capa que se encarga de presentar la interfaz con el usuario. Contiene todo el código para generar la interfaz con el usuario, en lenguaje HTML, y se encarga tanto de mostrar la información del sistema como capturar los datos ingresados por los usuarios. Esta es la única capa que el usuario final llega a ver del sistema.

El Controlador (también conocida como la capa lógica) es la capa que se encarga de interactuar entre la capa de Vista y la capa de Modelo. Esta capa recibe peticiones del usuario a través de la capa de Vista y las transmite a la capa de Modelo. Luego, recibe los datos generados por la capa Modelo y se los hace llegar a la capa de Vista para que puedan ser mostradas al usuario. Este ciclo se repite tantas veces como el usuario genere peticiones (o acciones) en el sistema. Por último, esta capa es de suma importancia puesto que contiene toda la lógica del negocio y es quien se encarga de procesar los datos de entrada ingresados por los usuarios.



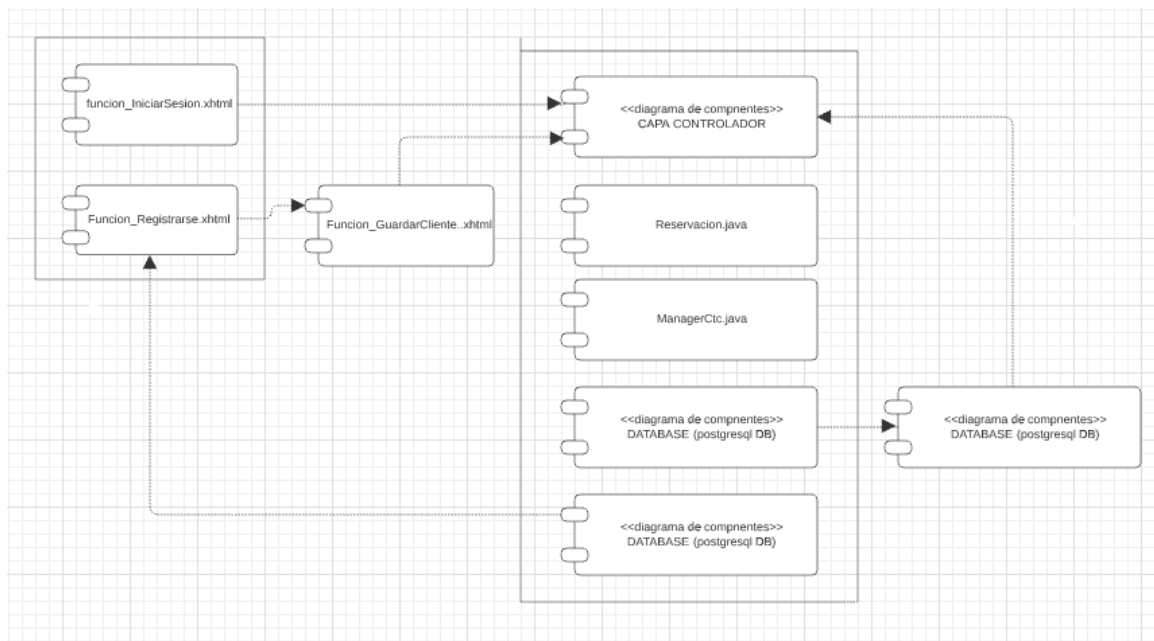
*Diagrama de Componentes del patrón MVC*

### 7.2 Capas

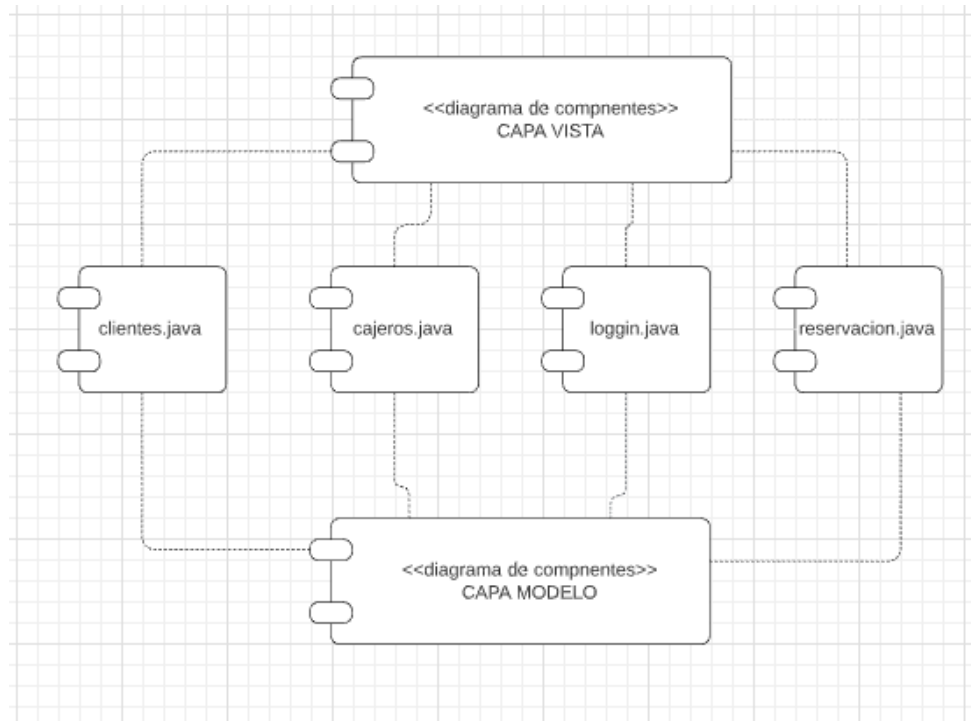
#### 7.2.1 Vista

*Diagrama de Componentes de la capa Vista*

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	



### 7.2.2 Controlador



*Diagrama de Componentes de la capa Controlador*

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

### 7.2.3 Modelo

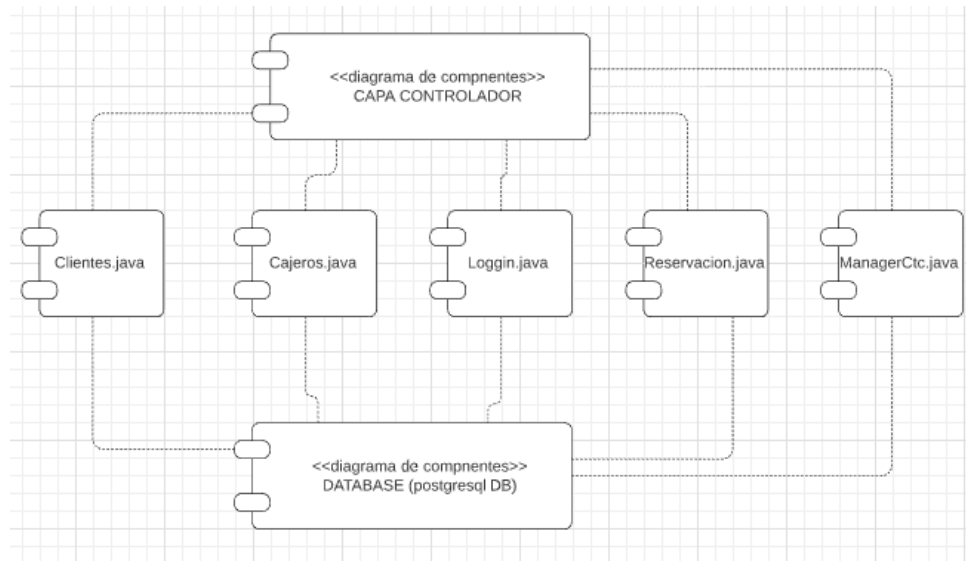
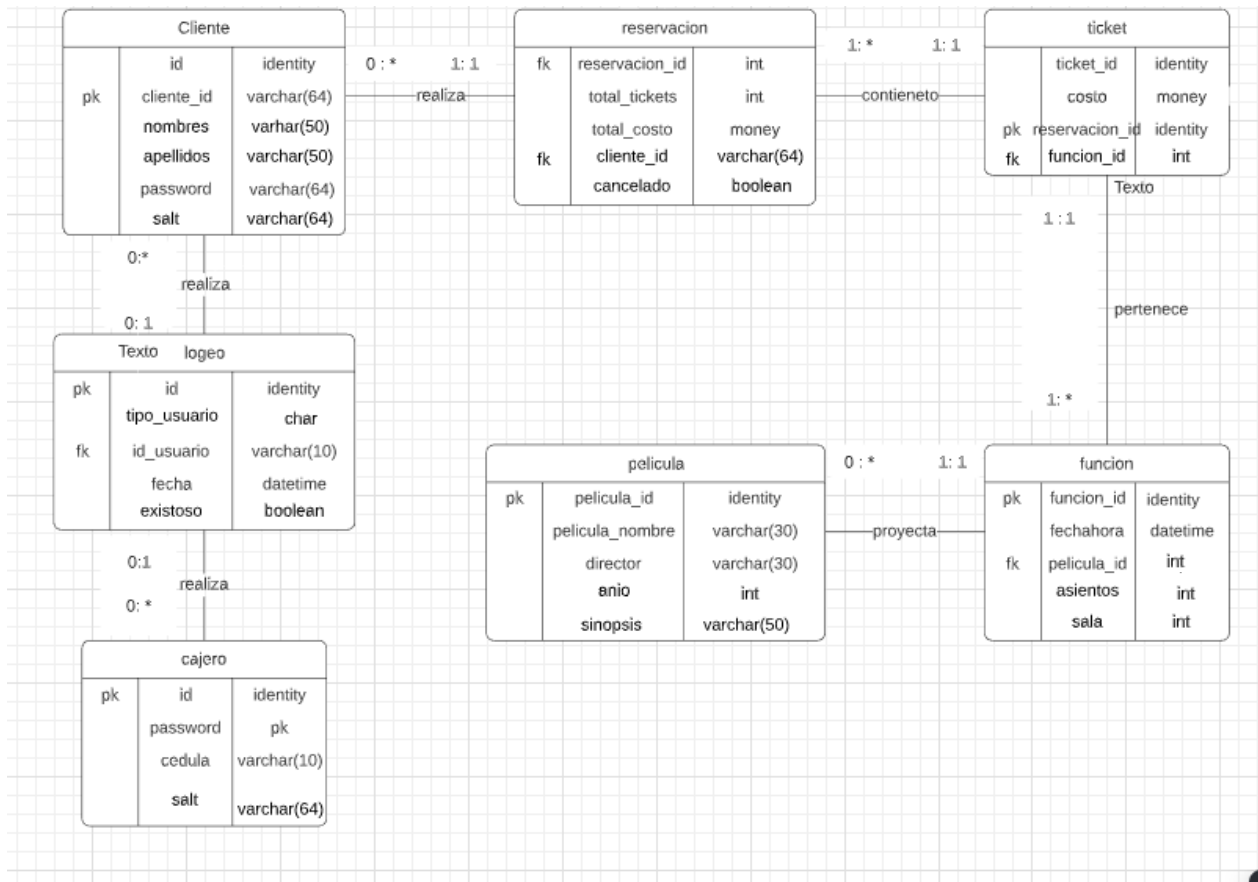


Diagrama de Componentes de la capa Modelo

## 8. Vista de Datos

### 8.1 Diagrama Entidad-Relación (ER)



C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

## 8.2 Diccionario de Datos

Cliente

Atributo	Tipo	Null	Valor por Defecto	Descripción
<u>id</u>	identity	no	-	identificador independiente de la clave primaria.
cliente_id	varchar(64)	no	-	indica el id del cliente como clave primaria.
nombres	varchar(50)	no	-	se registran los nombres del cliente.
apellidos	varchar(50)	no	-	se registran los apellidos del cliente.
password	varchar(64)	no	-	se solicita una clave del cliente.
<u>salt</u>	varchar(64)	no	-	valor que se agrega a la contraseña para hashearla.

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

#### reservación

Atributo	Tipo	Null	Valor por Defecto	Descripción
<u>reservación_id</u>	int(2)	No	-	Clave primaria de la tabla de reservaciones.
<u>total_tickets</u>	int(2)	No	-	Total de los tickets reservados por cliente.
<u>total_costo</u>	money	No	-	Total del costo de los boletos reservados
<u>cliente_id</u>	varchar(64)	No	-	Clave foránea que apunta a: Cliente → cliente_id.
cancelado	boolean	No	false	Estado que determina si el cliente ha pagado o no su reservación.

#### ticket

Atributo	Tipo	Null	Valor por Defecto	Descripción
<u>ticket_id</u>	identity	no	-	clave primaria de la tabla ticket.
<u>costo</u>	money	no	-	valor unitario de cada ticket
<u>reservación</u>	identity	no	-	código de reservación del boleto.
función_id	int	no	-	clave foránea que apunta a: función → funcion_id.

#### funcion

Atributo	Tipo	Null	Valor por Defecto	Descripción
<u>función_id</u>	identity	no	-	clave primaria de la tabla función
<u>fechahora</u>	datetime	no	-	fecha y hora que se realizara la función
<u>película_id</u>	int	no	-	clave foránea que apunta a: película→ película_id
<u>asientos</u>	int	no	20	número de asientos disponibles para esa función
<u>sala</u>	int	no	-	numero de sala en la que se proyectara la función

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

#### pelicula

Atributo	Tipo	Null	Valor por Defecto	Descripción
<u>pelicula_id</u>	identity	no	-	clave primaria de la tabla película
pelicula_nombre	varchar(30)	no	-	indica el nombre de la película
director	varchar(30)	no	-	indica el nombre del director de la película
anio	int	no	-	indica el año en que se estrenó la película
sinopsis	varchar(50)	no	-	indica el resumen de la película

#### logeo

Atributo	Tipo	Null	Valor por Defecto	Descripción
<u>id</u>	identity	no	-	clave primaria de la tabla logeo
tipo_usuario	char	no	-	carácter que determina que tipo de usuario inicio sesión o intento hacerlo
id_usuario	varchar(10)	no	-	clave foránea que apunta a: cliente → cliente_id o a su vez a cajero → cajero_id
fecha	datetime	no	-	fecha actual en la cual el usuario inicio sesión o intento hacerlo
exitoso	boolean	no	false	estado que indica si logro iniciar sesión o no

#### cajero

Atributo	Tipo	Null	Valor por Defecto	Descripción
<u>id</u>	identity	no	-	clave primaria de la tabla cajero
password	Varchar(64)	no	-	contraseña del cajero
cedula	varchar(10)	no	-	cedula del cajero
salt	varchar(64)	no	-	valor que se agrega a la contraseña para hashearla.

## 9. Tamaño y Desempeño

La aplicación web CTC se puede catalogar como un sistema de pequeño alcance, considerando que va enfocado en primera instancia a hacer posible que un cliente solo reserve los boletos para una función que ofrece la empresa, y que necesariamente el pago se lo realice de forma presencial. Como también que permita que el empleado solo



C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

registre el pago de las reservaciones. No se puede medir a priori el volumen de información que será cargada en el sistema debido a que esto dependerá de manera directamente proporcional a las actividades de la empresa cliente y el tamaño de información que ingrese al sistema. La cantidad máxima de información que pueda alojarse en el sistema dependerá de la capacidad de alojamiento del servidor local en el cual se implante el Sistema CTC.

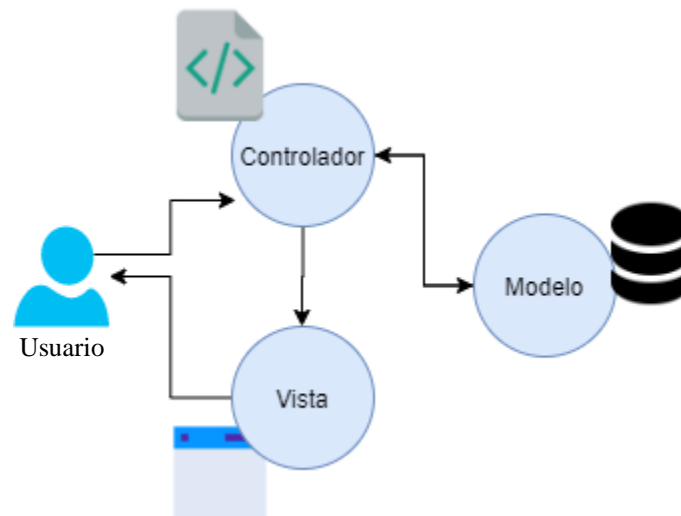
La aplicación CTC también puede considerarse como un sistema de alta portabilidad, debido a sus bajos requerimientos de sistema: un servidor con soporte de Java EE y PostgreSQL, un explorador Web (Microsoft Internet Explorer o Mozilla Firefox) y una conexión de Internet (o Intranet).

## 10. Calidad

Requerimiento	Diagrama	Elemento	Observaciones
Tolerancia a fallos	Diagrama de Implantación Estándar del Sistema CTC	Servidor de Respaldo	Se recomienda el uso de un servidor de respaldo que monitoree el estado del servidor de aplicación principal, con el fin de manejar aquellas fallas que puedan ocurrir en el mismo.
Lenguajes, Componentes Comprados y Licenciamiento	Diagrama de Implantación Estándar del Sistema CTC	Servidor de Aplicación	Como se puede ver en el diagrama, los componentes del software fueron desarrollados con JAVA EE. Adicionalmente, la base de datos utilizada es PostgreSQL, la cual también es software libre.
Arquitectura MVC	Diagrama de Componentes del Patrón MVC y Diagrama de Componentes de las Capas Vista, Controlador y Modelo	-	A pesar de que no se está utilizando un framework para el desarrollo de la aplicación, se está haciendo uso del patrón MVC durante el desarrollo del mismo.

C.T.C. (Compra de tickets de cine)	Versión 2.0
Documento de la Arquitectura de Software	Fecha: 10/09/2020
DOCUMENTO_ARQUITECTURA_SOFTWARE_CTC	

## 11. Figura Arquitectónica



**Ilustración 1: Modelo MVC**  
Fuente: Elaboración propia.