Especificación de Requisitos del Sistema de Registro.

Trabajo Práctico Final : MoviePass

Integrantes:

Jorge Villordo.

Ivan Ignacio Graciarena.

Ivan Lopez.

Metodología en Sistemas - Laboratorio IV.

Año : 2019 Comisión : 2.

Índice de contenido

1.Introducción

Este documento contiene la descripción detallada de los diferentes requisitos de software que debe cumplir el sistema de información utilizado para el registro, compra y venta de tickets de cines para los distintos cines en Mar del Plata y la zona.

1.1.Propósito

El propósito de este documento es presentar de manera formal la especificación de requisitos de este sistema, para su discusión y aceptación, por parte de los usuarios que utilizan dicho sistema. En esta especificación se detallan los requerimientos funcionales, las restricciones y los atributos de calidad que deberá satisfacer el sistema.

1.2.Alcance

Este sistema formará parte de todas las cadenas de cines de Mar del Plata y la zona con la finalidad de brindarle al usuario un servicio de compra y venta online de entradas.
En particular este sistema permitirá:

- Realizar el control físico de venta de tickets online.
- Realizar el recuento de tickets y los balances de ventas.
- Compra o venta de tickets con tarjeta de crédito.

1.3.Definiciones, acrónimos y abreviaturas

PHP: Hypertext Preprocessor, es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el preprocesado de texto plano en UTF-8

CSS: En español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.
HTML: Siglas en inglés de HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas
JS: Lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.
JSON: Es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.
RF:Requerimiento Funcional.
RNF:Requerimiento No Funcional.

1.4.Referencias

Para la elaboración de este escrito se han tomado como referencia todo lo visto durante el cuatrimestre de la materia de Metodologías en Sistemas dictada por la profesional Veronica Tomich de la Universidad Tecnológica Nacional, además del documento de "especificación de requisitos" elaborado por la Dirección de servicios de información administrativa de la Universidad de los Andes.

2.Descripción General del Sistema

2.1.Perspectiva del producto

Este producto (sistema de información) deberá funcionar en cualquier computador PC que soporte los sistemas operativos Windows (98 o superior) o Linux, así como también, que disponga de conexión a Internet y tenga instalado un navegador web.

2.2.Objetivos del Sistema

El sistema deberá cumplir con los siguientes objetivos:

- Consultar películas por fecha y/o categoría.
- Seleccionar una película para su compra. A continuación se visualizarán los cines donde se proyecta con sus horarios (solo aquellos que tengan aún entradas disponibles). Una vez seleccionado horario y cine se deben detallar la cantidad de entradas a comprar, visualizando el total de la compra.
- Consultar las entradas adquiridas, ordenadas por película ó por fecha.
- Ingresar películas en cartelera del cine con sus días y horarios de proyección.
- Consultar cantidades vendidas y remanentes de las proyecciones (Película, Cine, Turno).
- Administrar cines. Cada registro debe tener el nombre del cine, su capacidad total, dirección y valor único de entrada.
- Consultar totales vendidos en pesos (por película ó por cine, entre fechas).

3. Definición de Requisitos del Sistema

3.1. Definición de Requisitos Funcionales

Id	Nombre	Descripción	Usuario
RF-1	Registrar Cuenta	El usuario podrá registrar una cuenta con los campos nombre apellido correo electrónico fecha de nacimiento y contraseña, solo si el email no fue usado	Todos los usuarios

RF-2	Iniciar Sesión	Los dos roles (administrador y cliente) podrán iniciar sesión validando sus datos en los campos de “ingrese correo electrónico” e “ingrese contraseña”	Cliente - administrador
RF-3	Administrar Cines	El administrador podrá agregar modificar un cine especificando nombre dirección precio de entrada y ciudad o dar de baja un cine	Administrador

RF-4	Administrar Usuarios	El administrador podrá agregar o modificar un usuario especificando los campos de rol nombre fecha nacimiento email contraseña o dar de baja un usuario.	Administrador
RF-5	Administrar Tickets	El administrador podrá saber la cantidad de tickets que un cine ha vendido, además de saber la cantidad de tickets emitidos dependiendo de una función	Administrador
RF-6	Administrar Salas	El administrador podrá agregar o modificar una sala de un cine dado especificando la cantidad de asientos y el tipo de sala que es y el nombre o dar de baja	Administrador
RF-7	Administrar Funciones	El administrador podrá agregar o modificar una función de una sala dada	Administrador

		especificando la película la fecha y hora de la misma o dar de baja una función	
RF-8	Administrar Asientos	El administrador podrá visualizar la cantidad de entradas vendidas por función y los remanentes además de los estados de cada uno de los asientos	Administrador
RF-9	Búsqueda	Se podrá realizar una búsqueda tanto de la fecha de una función como de un género de una película	Administrado - Cliente
RF-10	Ver Función	Se podrá visualizar las funciones de todas las películas que tengan funciones	Administrador -Cliente

RF-11	Elegir Asiento	El Cliente podrá elegir sus asiento para la película que previamente eligió	Cliente
RF-12	Comprar Entrada	El Cliente podrá realizar una compra de la cantidad de tickets que previamente elegido habiendo o no un descuento por los días jueves o martes	Cliente
RF-13	Cerrar Sesión	Podrán cerrar sesión si previamente iniciaron sesión	Cliente - Administrador

3.2. Definición de Requisitos No Funcionales

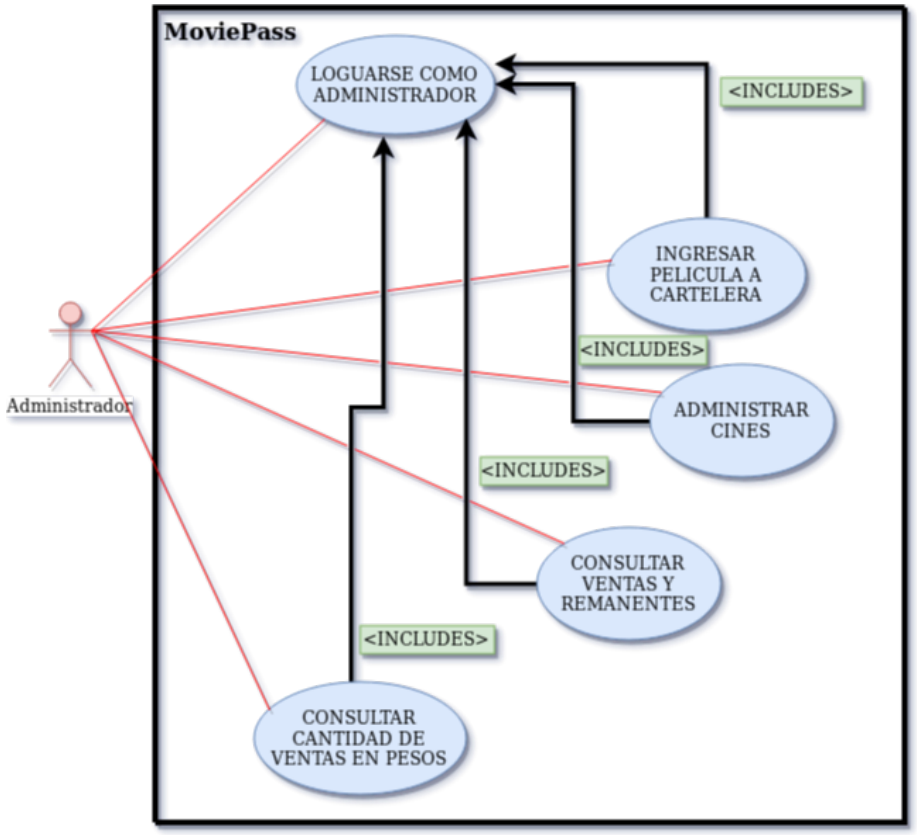
ID	Descripción
RNF-1	La interfaz del sistema deberá ser implementada como una aplicación web
RNF-2	El sistema debe ser diseñado según la arquitectura cliente/servidor de programación en capas de la aplicación respetando la arquitectura de 3 capas lógicas.
RNF-3	El sistema deberá ser desarrollado utilizando el lenguaje de programación PHP y utilizará el estándar HTML para el diseño de las páginas web del sistema.
RNF-4	La organización y administración de los datos deberá ser bajo un gestor de base de datos en este caso Mysql
RNF-5	El acceso a las películas y categorías (temas) de las mismas será efectuado a través del uso de una API pública del sitio TheMovieDb

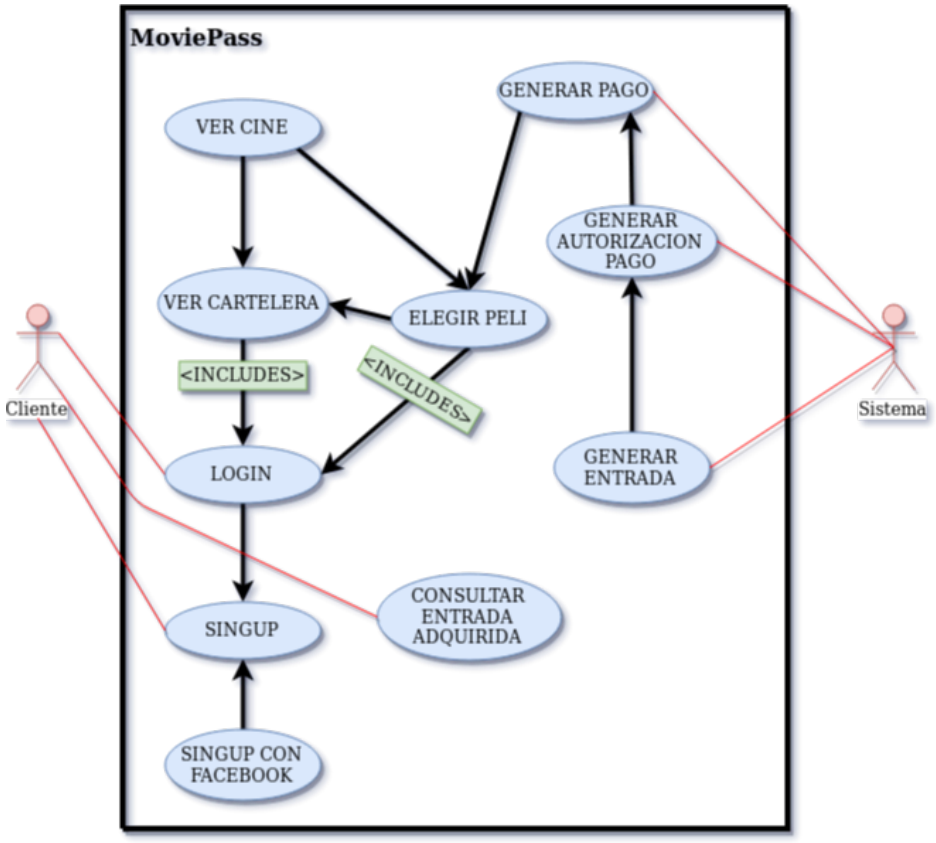
4. Especificación de Requisitos del Sistema

4.1. Definición de actores

NOMBRE	DESCRIPCIÓN
ACTOR 1 USUARIO CLIENTE	Es aquel que una vez registrado goza de los accesos a las funciones, a su compra y a su misma visualización
ACTOR 2 USUARIO ADMINISTRADOR	Es aquel que posee todos los privilegios y permisos para poder editar crear y borrar funciones cines usuarios y salas

4.2. Diagrama de casos de uso





4.3. Especificación de caso de uso “Alta de función”

NOMBRE	ALTA DE FUNCION
ACTOR	ADMINISTRADOR
Precondicion	El usuario “administrador” debera haberse logueado satisfactoriamente y haber ido a la seccion de “administrar cines” para poder operar con el abm de funciones.
Escenario de Exito	Paso 1 : el administrador clickea en el campo del cine elegido la opcion de administrar funciones
	Paso 2 : el sistema generara una insterfaz para ingresar datos de una funcion
	Paso 3 : el administrador rellenara esos datos especificando, la pelicula que desea, la fecha y hora de la misma y el lenguaje en que se emitira
	Paso 4 : el administrador debera clickear en el boton de agregar funcion.
	Paso 5 : el sistema comprobara que haya ingresado en todos lso campos los datos validos para esos campos y ademas que haya ingresado datos en cada uno de los campos
	Paso 6 : el sistema mostrara una advertencia de si quiere o no agregar una funcion
Postcondicion	La funcion fue creada exitosamente

Escenario de Fracaso	Paso 3.1: No se puede seleccionar un cine para la funcion
	Paso 3.1.1 : El sistema mostrara un listado en blanco si no hay cines existentes.
	Paso 3.2: No se puede seleccionar una sala para la funcion
	Paso 3.2.1 : El sistema mostrara un listado en blanco si no hay salas existentes.

4.4. Especificación de caso de uso “Compra de entrada”

NOMBRE	COMPRA DE ENTRADA
ACTOR	CLIENTE
Precondicion	el usuario debera haber seleccionado una pelicula con su respectiva funcion
Escenario de exito	Paso 1 : el usuario elige dentro de la interfaz los asientos disponibles y la cantidad de asientos que desea
	Paso 2 : el usuario debera clickear en el boton proceder al pago una vez seleccionado los asientos que quisiese
	Paso 3 : el usuario debera corroborar toda la informacion proveida por el sistema el cual especifica cantidad de asientos numeros de asientos fecha y hora de funcion y monto total habiendo o no tenido descuento

	por ser dia martes o jueves
	Paso 4 : el usuario una vez corroborada dicha informacion debera presionar en el boton pagar donde la interfaz de mercado pago pedira al usuario datos sensibles como numero de tarjeta vencimiento CBU y nombre del titular
	Paso 5 : una vez completado los campos el usuario debera aceptar el pago y luego el sistema informara al usuario que la compra ah sido efectuada satifactoriamente
Postcondicion	Se generara un registro de compra especificando el usuario el monto la cantidad de tickets para una funcion determinada en un dia determinado
Escenario de Fracaso	Paso 2.1 : el usuario no selecciono ningun asiento
	Paso 2.1.1 : el sistema tomara como consecuencia el primer espacio que encuentre y le generara una cantidad de entradas de un solo item
	Paso 4.1 : no se completa el formulario para ingresar datos en mercado pago
	Paso 4.1.1 : el sistema se encargara de cancelar la operacion y redirigir al usuario al inicio

5. Diagrama de clases con la arquitectura implementada

Diagrama de clases de la capa “Controllers”:

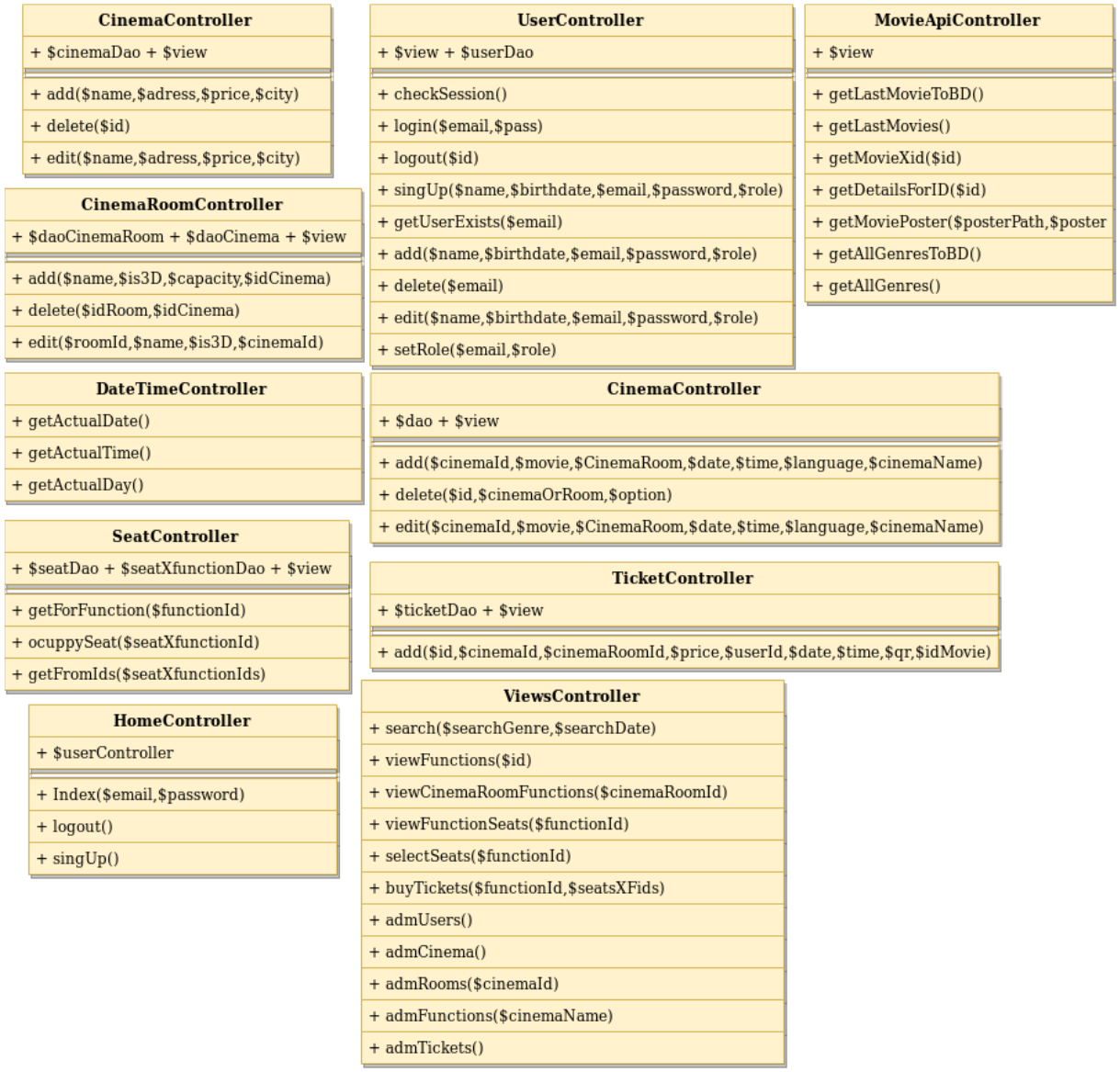

CinemaController
+ $cinemaDao + $view
+ add($name,$adress,$price,$city)
+ delete($id)
+ edit($name,$adress,$price,$city)
UserController
+ $view + $userDao
+ checkSession()
+ login($email,$pass)
+ logout($id)
+ singUp($name,$birthdate,$email,$password,$role)
+ getUserExists($email)
+ add($name,$birthdate,$email,$password,$role)
+ delete($email)
+ edit($name,$birthdate,$email,$password,$role)
+ setRole($email,$role)
MovieApiController
+ $view
+ getLastMovieToBD()
+ getLastMovies()
+ getMovieXid($id)
+ getDetailsForID($id)
+ getMoviePoster($posterPath,$poster
+ getAllGenresToBD()
+ getAllGenres()
CinemaRoomController
+ $daoCinemaRoom + $daoCinema + $view
+ add($name,$is3D,$capacity,$idCinema)
+ delete($idRoom,$idCinema)
+ edit($roomId,$name,$is3D,$cinemaId)
DateTimeController
+ getActualDate()
+ getActualTime()
+ getActualDay()
CinemaController
+ $dao + $view
+ add($cinemaId,$movie,$CinemaRoom,$date,$time,$language,$cinemaName)
+ delete($id,$cinemaOrRoom,$option)
+ edit($cinemaId,$movie,$CinemaRoom,$date,$time,$language,$cinemaName)
SeatController
+ $seatDao + $seatXfunctionDao + $view
+ getForFunction($functionId)
+ ocuppySeat($seatXfunctionId)
+ getFromIds($seatXfunctionIds)
TicketController
+ $ticketDao + $view
+ add($id,$cinemaId,$cinemaRoomId,$price,$userId,$date,$time,$qr,$idMovie)
HomeController
+ $userController
+ Index($email,$password)
+ logout()
+ singUp()
ViewsController
+ search($searchGenre,$searchDate)
+ viewFunctions($id)
+ viewCinemaRoomFunctions($cinemaRoomId)
+ viewFunctionSeats($functionId)
+ selectSeats($functionId)
+ buyTickets($functionId,$seatsXFids)
+ admUsers()
+ admCinema()
+ admRooms($cinemaId)
+ admFunctions($cinemaName)
+ admTickets()

Diagrama de clases de la capa “Models”:

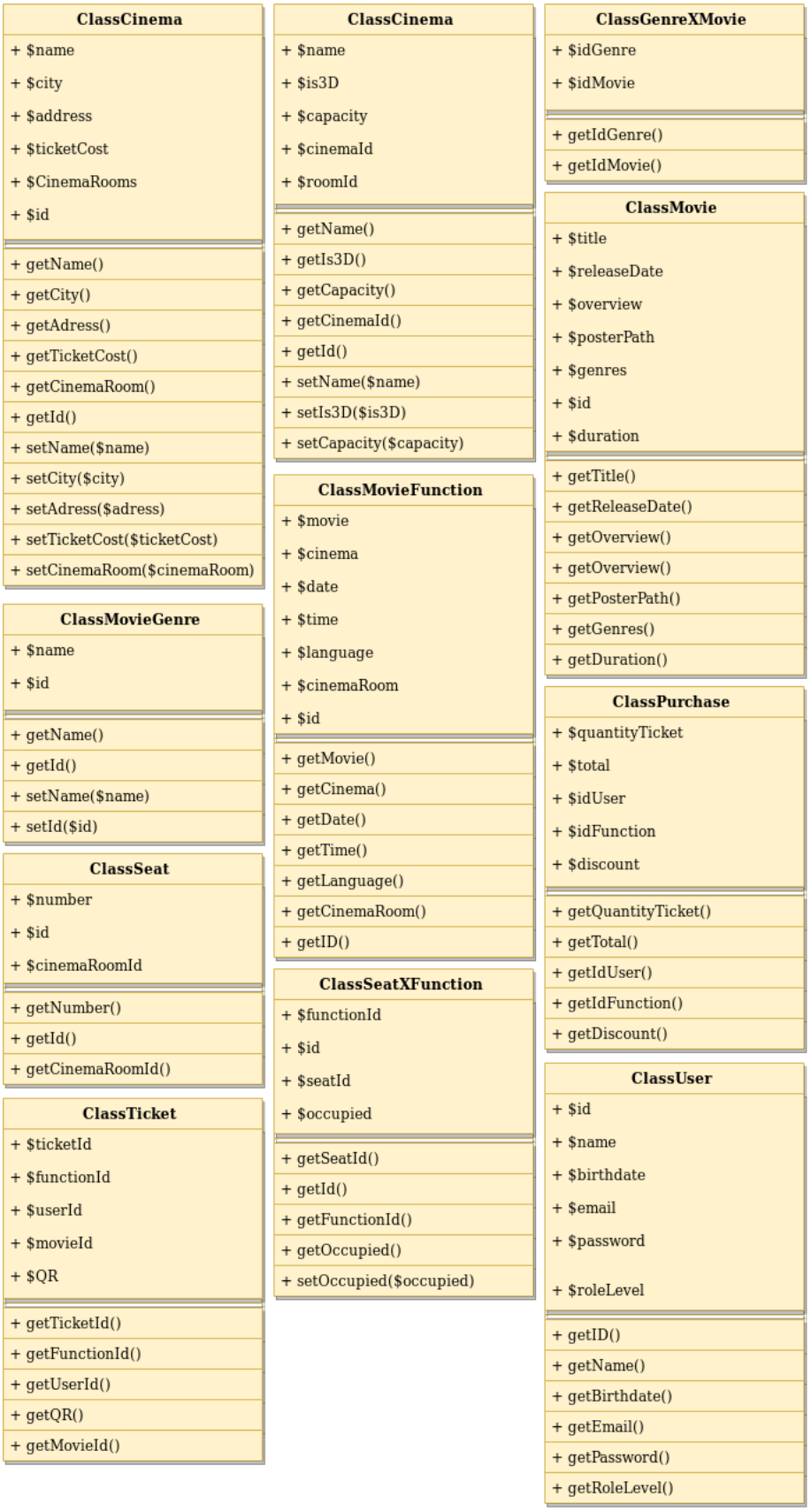
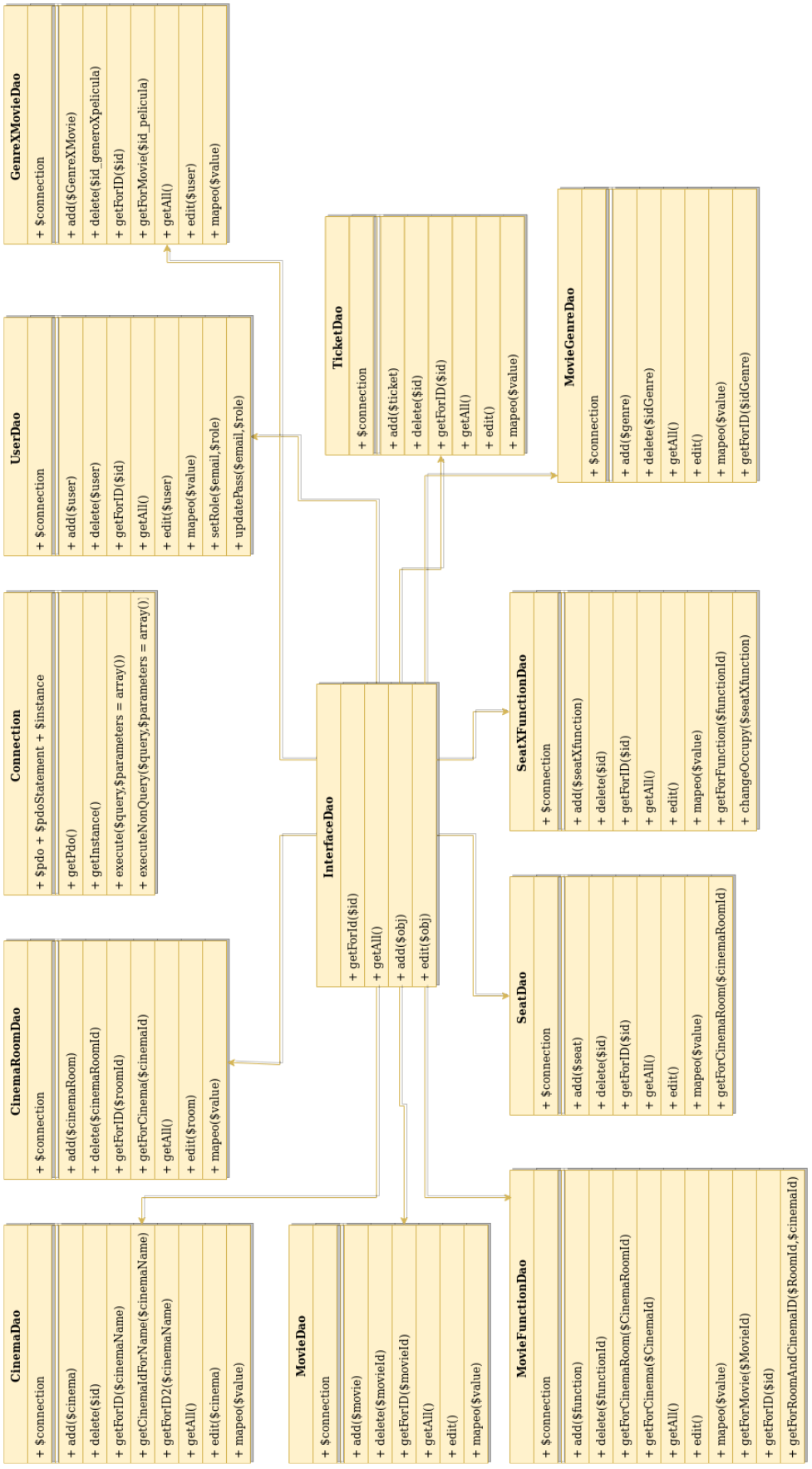


Diagrama de clases de la capa “Daos”:



6. Diagrama de entidades y relaciones