SITIO WEB DE VENTAS DE ENTRADAS DE CINE

PRÁCTICA DE PROGRAMACIÓN AVANZADA

Ingeniería del Software
Curso académico 2019-2020

Julia Álvarez Gurdiel – julia.gurdiel
Jaime Cabero Creus – Jaime.cabero
Ángel Paderne Cózar – a.pcozar

UNIVERSIDADE DA CORUÑA

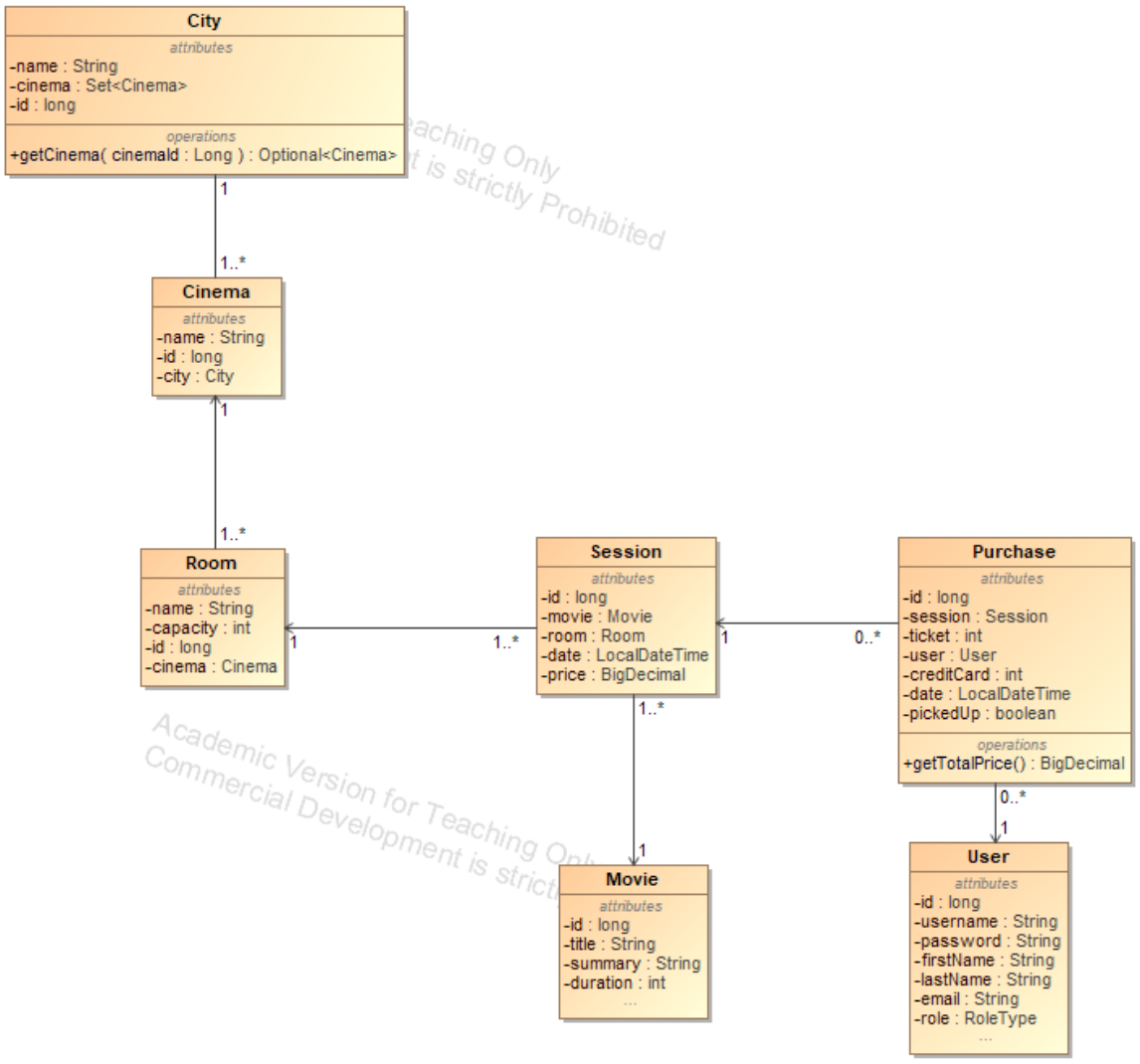
fic facultade de informática da coruña

ÍNDICE

1. BACKEND

1.1 Capa acceso a datos

Para cumplir con el acometido de la práctica, se han creado distintas entidades con sus correspondientes atributos y las siguientes relaciones entre ellas:



Podemos diferenciar siete entidades diferentes:

- **Room:** contiene todos los atributos que describen las salas de un determinado cine. Entre ellos podemos diferenciar un identificador para cada sala, un nombre, la capacidad máxima que tiene esa sala, y el cine al que pertenece. Como se puede observar en el diagrama, una sala puede albergar varias sesiones.

- **Session:** esta entidad representa un pase de una determinada película. Los atributos son: un identificador de la propia sesión, la película que se va a emitir, la sala en la que se va a emitir, la fecha completa correspondiente y el precio, que puede ser distinto para cada sesión. En una sala se pueden realizar múltiples sesiones, como se ha explicado anteriormente. Además, en una sesión se emite únicamente una película, y una compra es de una única sesión.

- **Movie:** define todos los atributos de la película, en este caso, un identificador único para cada película, el título de la misma, una pequeña sinopsis y la duración, en minutos. Una película se puede emitir en varias sesiones.

- **Purchase:** almacena los datos relacionados con la compra de una determinada sesión de una película. Cada compra tiene un identificador, la sesión que se va a comprar, el número de entradas que se desea comprar (con un límite máximo de 10 entradas por compra), el usuario que realiza la compra junto con la tarjeta de crédito con la que va a realizar la compra, la fecha completa en la que se hizo la compra y por último, el atributo booleano *pickedUp* que da información acerca de si las entradas han sido recogidas (true) o no (false) por el usuario. Además esta entidad tiene un método *transient* para poder calcular el precio total de la compra, que se calcula multiplicando el número total de entradas a comprar por el precio de cada entrada.

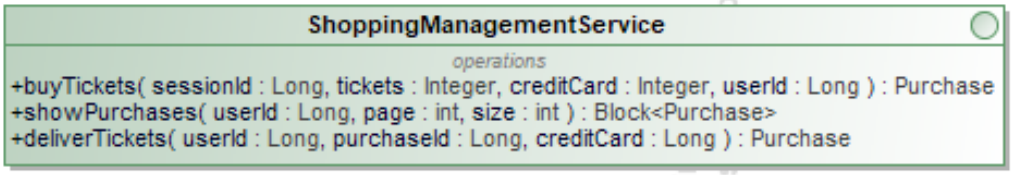
  Esta entidad tiene relación 1-N tanto con la entidad *Session*, como con la entidad *User*, ya que una compra siempre la realiza un usuario y es de una determinada sesión.

- **User:** comprende los datos relacionados con el perfil de usuario que interactúa con la aplicación web. Contiene un identificador propio para cada usuario, un *username* con su correspondiente contraseña, así como el nombre completo y un correo electrónico. A su vez un usuario puede tener diferentes roles, se tiene el rol "espectador" que hace referencia a todos aquellos usuario que utilizan la aplicación para comprar entradas de una película, y el rol "taquillero" que es la persona encargada de la parte administrativa, como por ejemplo entregar las entradas al cliente. Respecto a las relaciones de entidades, un mismo usuario puede realizar varias compras.

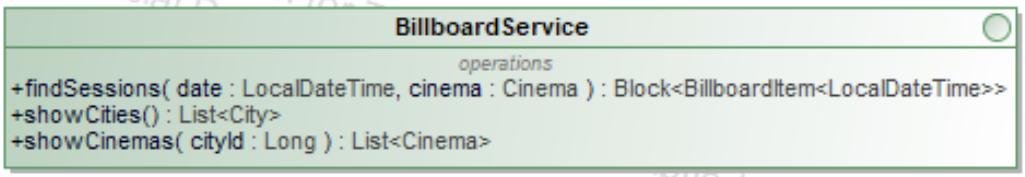
- **City:** hace referencia a las diferentes ciudades donde se pueden encontrar los cines. Tiene como atributos un identificador propio de cada ciudad, el nombre de la misma y los cines que contiene. Cada ciudad puede tener N cines. Además esta entidad tiene un método *transient* para poder obtener los cines existentes en cada ciudad.

- **Cinema:** entidad que guarda los datos de cada cine, el identificador de cada uno de ellos, la ciudad a la que pertenecen y el nombre del cine. La entidad cine está relacionada con las ciudades, es decir, un cine pertenece únicamente a una ciudad, y con las salas, ya que cada cine puede tener de 1 a N salas.

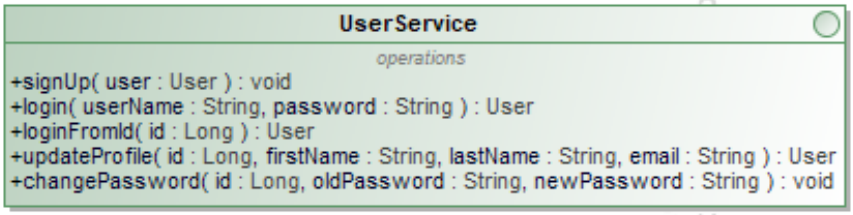
1.2 Capa lógica de negocio



- **ShoppingManagementService:** interfaz que contiene los casos de uso relacionado con la venta de entradas. Por un lado, un usuario de tipo "espectador" puede adquirir las entradas para una determinada sesión, indicando el número de tarjeta bancaria con la que va a pagar. Por otro lado, se puede obtener una lista con todas las compras realizadas por un usuario. Y por último, un usuario de tipo "taquillero" puede entregar las entradas de una determinada compra de un usuario, siempre y cuando éste le proporcione la tarjeta de crédito con la que pagó.



- **BillboardService:** es la interfaz encargada de las operaciones relacionadas con la cartelera del cine. Cualquier usuario puede ver las películas que existen en ese momento en la cartelera. Se buscan las sesiones del día de hoy, ordenadas por el título de la película, de tal manera que así sea más sencillo poder mostrarle al usuario las distintas películas con sus respectivas sesiones. Además se incluyen dos métodos de la parte opcional de la práctica para poder mostrar un listado de las ciudades existentes y de los cines existentes en cada ciudad.

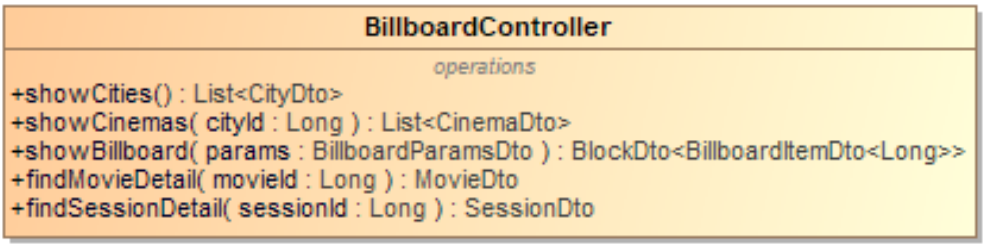


- **UserService:** interfaz que contiene las distintas operaciones que un usuario puede realizar en la aplicación relacionadas con la identificación y realización de distintas operaciones según los privilegios de cada usuario. Mediante este servicio un usuario no autenticado se puede registrar como espectador. Si ya está registrado, tanto si es espectador como si es taquillero, puede cambiar su contraseña, para una mayor seguridad en la aplicación, y su información, aportando el nombre completo del usuario y el email con el que se creó la cuenta.

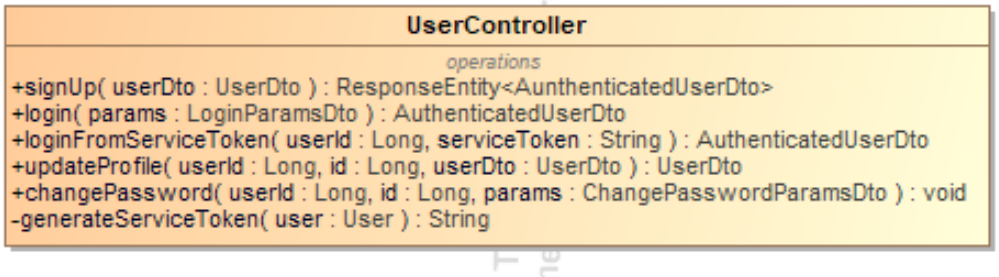
1.3 Capa servicios REST

- **ShoppingManagementController:** a través de este controlado se tratan todas las operaciones relacionadas con la compra y entrega de entradas. Por ello, un usuario puede comprar las entradas de una determinada sesión, restringiendo la compra a como máximo 10 entradas por usuario. Una vez realizada la compra, la aplicación le devolverá al cliente un identificador de compra.

  Por otro lado, también es posible recuperar todas las compras de un usuario. Para poder mostrarle los atributos correctos al usuario, se ha creado un *PurchaseSummaryDto* que devuelve la fecha de compra, la fecha completa de la sesión, el título de la película, el número de entradas compradas y el precio total, así el cliente puede tener un resumen de su compra.
  Y por último, a través de este controlador un usuario de tipo "taquillero" entrega las entradas de una compra. En esta operación se devuelve un *PurchaseDto* con los atributos más significativos para el usuario.



- **BillboardController:** es el controlador encargado de todas las operaciones relacionadas con la cartelera que puede ver cualquier usuario independientemente de su rol. Tenemos dos métodos para poder mostrar las ciudades con sus correspondientes cines, facilitando al usuario la búsqueda de su cine.
  Cualquier usuario de la aplicación puede ver la cartelera que existe en ese momento y las sesiones a las que aún puede asistir. Por último, los dos últimos métodos nos permiten visualizar los detalles de una determinada película y una determinada sesión, devolviendo en ambos casos un DTO específico para dar la mejor información al usuario.



- **UserController:** el controlador relacionado con las operaciones del usuario será el encargado de realizar las conversiones necesarias de código *JSON* a *Dto* y viceversa que permitan completar las operaciones de creación y modificación de los datos asociados a una cuenta de usuario.
  Este módulo también se encarga de la generación del *token JWT*, que permite identificar de manera unívoca a un usuario aportando la seguridad que se necesita a la aplicación, pues el *token* es un componente que no puede ser simulado por una persona que intente acceder a la aplicación de manera malintencionada para suplantar una identidad.