MEMORIA PRACTICA ISD

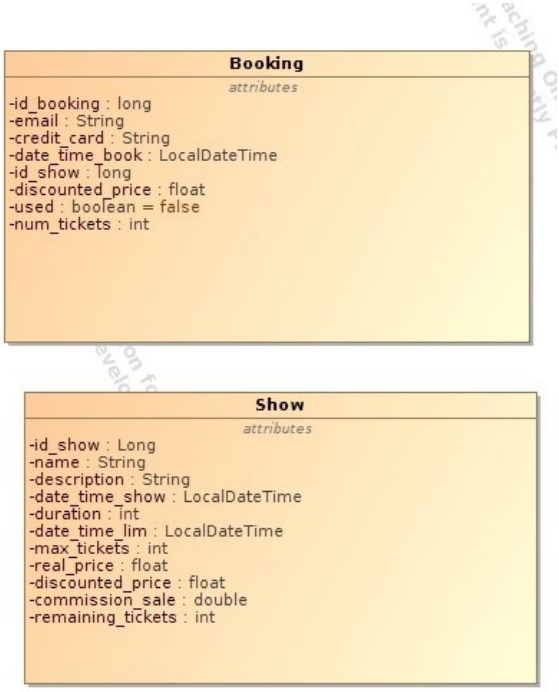
Autores:
Sergio García Rodríguez.
Alejandro Cabarcos Calvo.
Nicolás Calvo Gómez.

1. Introducción

En la siguiente memoria recogemos la implementación web de una empresa organizadora de espectáculos.
Esta aplicación sigue una arquitectura en capas. Se divide en capa modelo, capa servicio y capa cliente. Esta aplicación usa protocolo REST para su invocación remota.
La capa de acceso a datos usa una base de datos relacional que almacena los datos de los espectáculos y de las reservas.
En esta aplicación ha sido implementada la parte opcional de SOAP.

2. Capa modelo

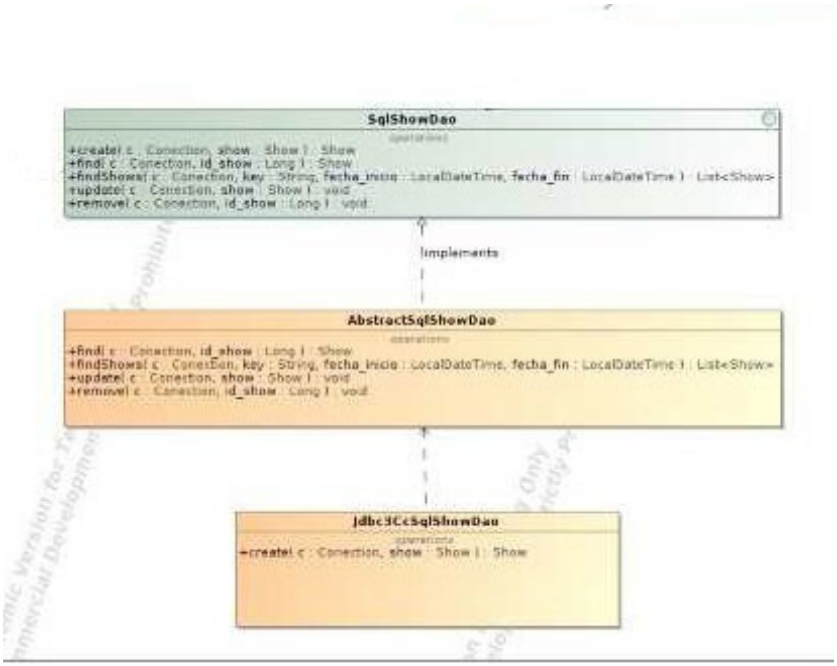
2.1. Entidades



La entidad Show almacena los espectáculos y la entidad Booking las reservas de estos espectáculos.
La relación entre Show y Booking es 1:N, es decir, un espectáculo (Show) puede tener asociados varias (N) reservas (Booking). Esta relación se mantiene gracias al atributo id_show que es clave foránea en Booking.

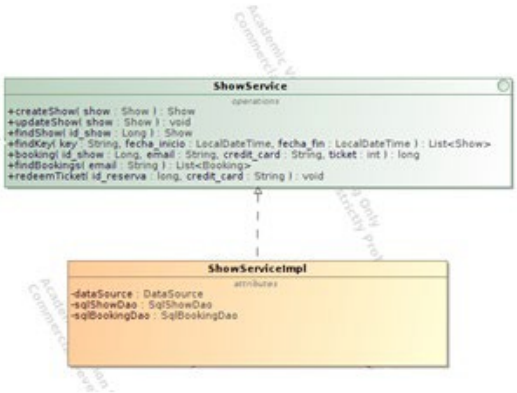
2.2. DAOs

-Show



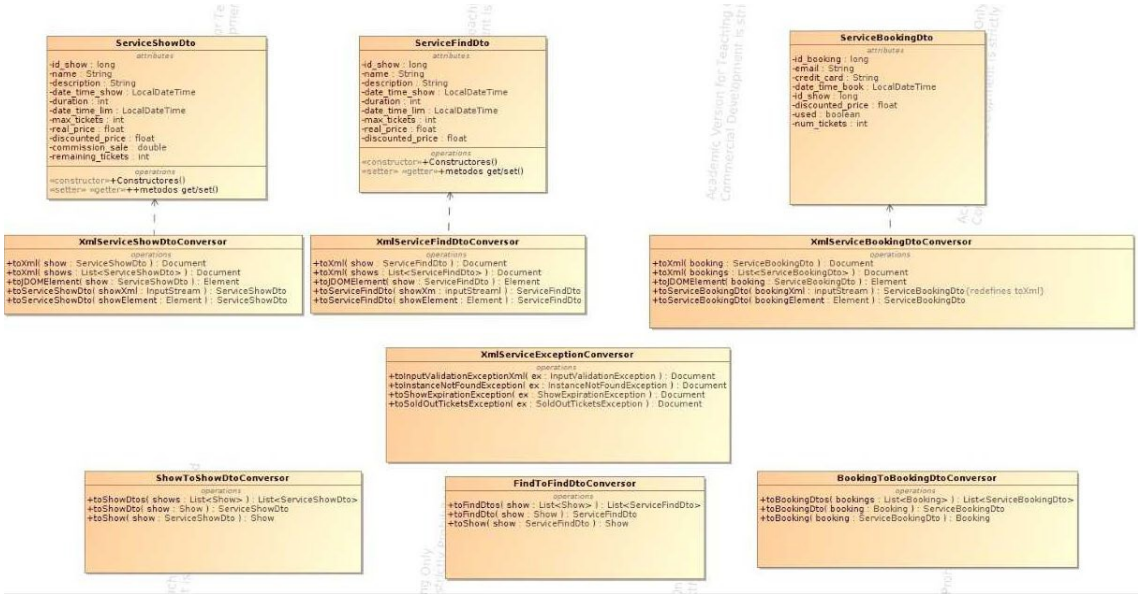
-Booking

2.3. Fachadas



3. Capa de servicios

3.1. DTOs



En este apartado, además de los diagramas DTOS usados en la capa servicio (ServiceShowDto, ServiceFindDto, ServiceBookingDto) añadimos los diagramas de las clase XmlServiceShowDtoConversor, XmlServiceFindDtoConversor, XmlServiceBookingDtoConversor y XmlServiceExceptionConversor.
XmlServiceShowDtoConversor es la clase encargada de realizar las conversiones de ServiceShowDto a XML y viceversa.
XmlServiceFindDtoConversor y XmlServiceBookingDtoConversor hacen lo mismo pero entre ServiceFindDto, ServiceBookingDto y XML.
XmlServiceExceptionConversor transforma las excepciones de la capa modelo a su representación a XML.

También se incluyen las clase ShowToShowDtoConversor, FindToFindDtoConversor, BookingToBookingDtoConversor que son las encargadas de transformar los Dtos usados por el servicio a objetos de tipo Show y Booking, pertenecientes a la capa modelo.

3.2. REST

-createShow :
URL: /show
Método de invocación : POST
Entrada: ClientShowDto show
Salida: Long

-updateShow :
URL: /show/id
Método de invocación : PUT
Entrada: ClientShowDto show
Salida: void

-findShow :
URL: /show/id
Método de invocación : GET
Entrada: Long id_show
Salida: ClientShowDto

-findShows :
URL: /show?keywords=””
Método de invocación : GET
Entrada: String keywords
Salida: List<ClientUserFindDto>

-booking:
URL: /booking
Método de invocación : POST
Entrada: Long id_show, String email, String credit_card, int tickets
Salida: Long id_booking

-findBookings :
URL: /booking?email=””
Método de invocación : GET
Entrada: String email
Salida: List<ClientBookingDto>

-redeemTickets :

URL: /booking/id

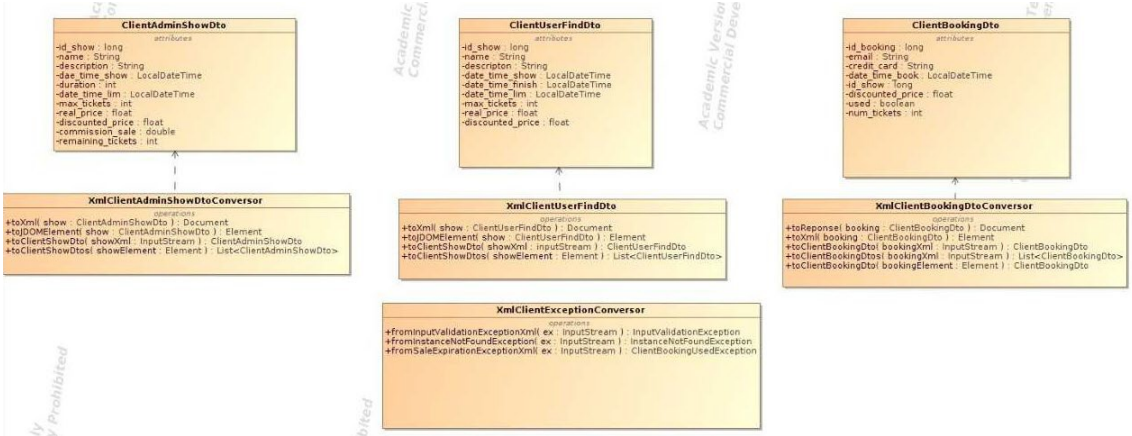
Método de invocación : POST

Entrada: Long id_booking, String credit_card

Salida: void

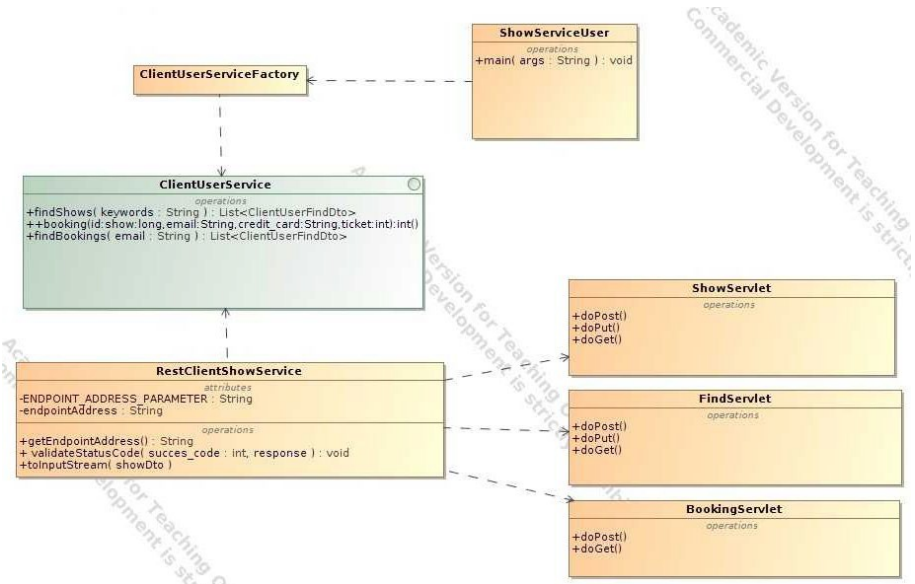
4. Aplicaciones cliente

4.1. DTOs

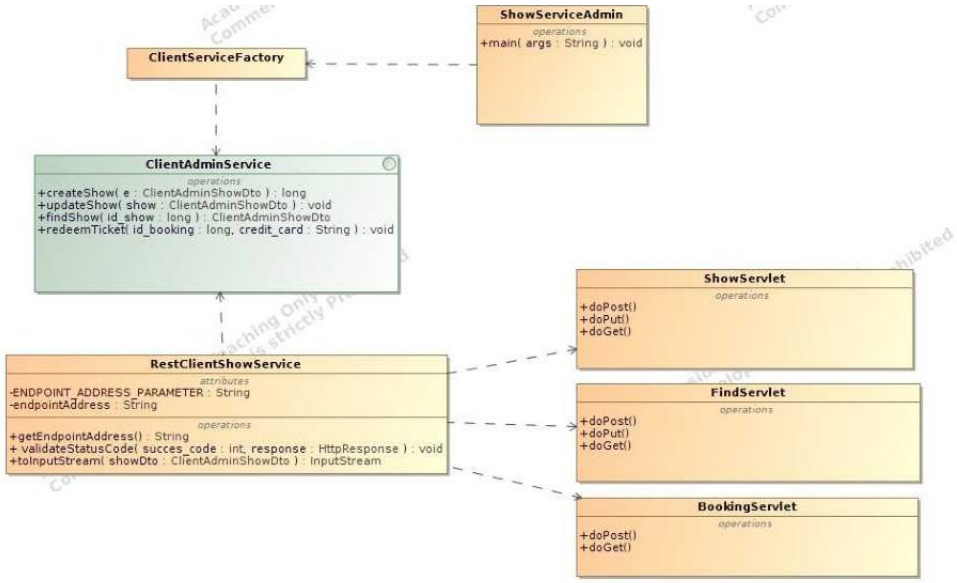


En este apartado se incluyen, a mayores de los DTOs usados por la aplicación cliente, las clases XmlClientShowDtoConversor, XmlClientUserFindDto, XmlClientBookingDtoConversor que son las encargadas de convertir las clases especificadas a formato xml.

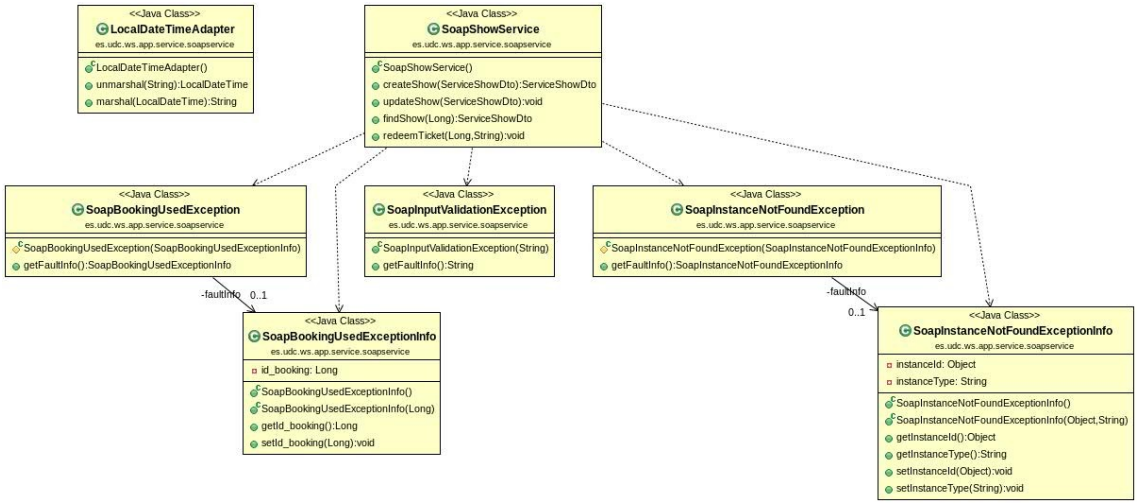
4.2. Capa de acceso al servicio: Usuario



4.3. Capa de acceso al servicio: Administrador



5. Trabajos tutelados



Cabe destacar en la implementación de SOAP la necesidad de crear un adaptador para LocalDateTime (LocalDateTimeAdapter) debido a problemas con un plugin de java.

6. Errores conocidos

Falta implementar la parte SOAP del cliente