

PRÁCTICA 3

A la hora de diseñar e implementar esta práctica nos hemos encontrado con diversos problemas y en este documento vamos a explicar nuestras decisiones.

Primeramente, decidimos que las sesiones, aparte de día, mes y año, tuvieran hora y minutos. Esto dificulta el diseño y la implementación ligeramente, ya que se hace necesario un nuevo atributo en película que indique la duración de esta para evitar que haya dos sesiones en la misma sala en horarios que no debería. Para su implementación, hemos utilizado la clase Calendar de Java, ya que la clase Date con hora y minutos esta deprecada.

Sin embargo, al utilizar esta clase nos encontramos con un nuevo problema. Los meses se expresan mediante una enumeración y, por tanto, al construir la sesión hacemos un switch para asociar cada uno de los meses con su número (1-12, enero-diciembre). Cuando se introducía un mes no válido no sabíamos cómo tratarlo puesto que, si se introduce un día no válido, por ejemplo 32-12-2018 la clase Calendar lo interpreta automáticamente como 01-01-2019, pero no ocurre lo mismo con los meses puesto que hemos utilizado las enumeraciones para asignar el valor del mes. Por tanto, decidimos utilizar el parámetro sala para comprobar si ha habido algún error. Si el mes introducido no es el correcto o falla el introducir la sesión en la sala (detallaremos esto ahora) sala se igualará a null. Cuando sala vale null, el toString de sesión imprime un mensaje de error.

El más grande de los problemas encontrados ha sido el comprobar que las sesiones no se solapen, ya que hay muchos casos posibles (que haya una sesión que haya terminado cuando empiece la nueva, que haya una sesión que cuando vaya a empezar la nueva no haya terminado, etc.). La solución final que hemos utilizado puede parecer un poco liosa y poco intuitiva. Lo que hacemos es comprobar caso por caso de error con cada sesión de la sala.

Dejando a un lado las fechas y los horarios, decidimos guardar en la sesión la sala en la que se va a dar lugar. Por ello, el atributo de butacas lo hemos omitido en sesión, puesto que se encuentra en sala y tenemos una relación directa para acceder a él. Además, el constructor de sesión, en adición a vincular la sesión con la sala, añade la nueva sesión a la lista de sesiones de la sala llamando al método anadirSesion en el constructor. Uno de los problemas, de esta decisión es que si se falla al añadir la sesión a la sala el objeto ya está creado, aunque sea inútil.

Por otra parte, hemos añadido un identificador a las entradas, puesto que cuando queremos borrar una entrada del cine sin este atributo solo podríamos borrar todas las entradas de una sesión determinada o una de ellas al azar. De esta forma, con el id cada entrada tiene uno propio y si queremos borrar una se busca por su id.

En relación con las entradas, hemos decidido que el precio es el mismo para todas las entradas (atributo estático), puesto a 9 euros (se puede cambiar en cualquier momento con el método setter). Cualquier descuento aplicable, en este caso el del día del espectador, es otro atributo fijo y que será aplicado al llamar al método getPrecio.

En cuanto a las pruebas realizadas, en el caso de película, como no contiene ningún método que no sea getter, setter o toString, comprobamos que al crear una película ésta se imprima correctamente.

Para sala y para sesión las pruebas realizadas son muy similares. En el caso de sala, creamos una sala y varias sesiones que se vayan a dar lugar en dicha sala. Ponemos fechas distintas y horarios que puedan dar conflicto para comprobar que cuando las sesiones son incorrectas se imprime el mensaje adecuado. Además, comprobamos que las sesiones se eliminan correctamente de la lista de sesiones. Por otro lado, en el tester de sesión se crean las salas y las sesiones, de nuevo comprobando que se hayan creado correctamente. Después, comprobamos la función de actualizarButacasVendidas cuando hay entradas disponibles y cuando no las hay.

Por último, para cine creamos el cine, además de una película y una sala que introducimos en las listas del cine. Después creamos una sesión con la película creada que metemos en la sala creada. Ahora creamos una película nueva que sustituye a la película de la sesión anterior. Creamos una sala nueva y añadimos la sala a la sesión anterior. Imprimimos la información de la cartelera, recaudación y sesiones. Ahora probamos a vender dos entradas, una del día del espectador y otra normal, imprimiendo la recaudación después de cada venta para comprobar que es correcto.

