

TP 3 / LABORATORIO II

Micaela Sol Vazzana



Enchulame la cámara Rental APP consiste en una aplicación de alquiler de equipos de cine y video destinado a productoras cinematográficas, realizadores audiovisuales independientes o estudiantes de carreras afines.

En esta aplicación podremos entonces realizar reservas accediendo a un listado de equipos que la empresa tiene en stock. La aplicación además permite realizar modificaciones sobre esa reserva, en caso de que el cliente quiera cancelar algún equipo que haya decidido no necesitar o en cambio, sumar más equipos a su reserva (siempre que estos estén en stock).

Por otra parte, podremos dar de baja la reserva si es que el cliente decide cancelarla, o bien si el cliente procede a la devolución una vez que haya terminado su jornada de alquiler.

La aplicación también nos permite guardarnos un listado de clientes y sus datos para poder tener una base de datos que servirá para realizar mailing con promociones.

EXCEPCIONES

Controlo las excepciones que pueden arrojarme aquellas instancias de objetos como el **StreamWriter**, **StreamReader**, arrojando una nueva excepción llamada **ArchivoException**, donde además guardo la excepción original con la propiedad `InnerException`.

También controlo las excepciones que me pueden arrojar los métodos como el **Parse** o cuando al hacer una búsqueda de índice esta esté fuera de rango. Cada posibilidad de uso de métodos que puedan arrojar excepciones controlo guardando la excepción con una excepción genérica que me permita saber qué sucedió.

PRUEBAS UNITARIAS

Realizo los test de las excepciones que puedan arrojarme los métodos que gestionaran los archivos. Verifico de esta manera, que la excepción **ArchivoException** creada por mí sea propiamente lanzada al manipular esos métodos.

Por otra parte, realizo test para controlar que los métodos que desarrollé en mi biblioteca de clases **Entidades** funcionen correctamente.

TIPOS GENERICOS

Realicé una clase gestora de archivos xml y json llamada **Serializadora**. Esta contiene los métodos que van a manejar estos distintos tipos de archivos.

Esta clase es genérica, ya que voy a utilizar los métodos que serializan con distintos tipos de objetos. Esto lo indico a la hora de utilizarlos y serán serializados o deserializados según esa indicación.

INTERFACES

La utilización de interfaces está implementada por un lado, en los formularios al mostrar un mensaje de error, que será el mensaje que contendrá el mensaje de las excepciones controladas, a su vez se muestran las **InnerException**. Estas se mostrarán a través de un **Message.Box**. Por otra parte, implementé un método que hace el cálculo del costo total de la reserva. Si este cliente es estudiante, el cálculo aplicará un 15% de descuento.

ARCHIVOS

La creación y lectura de archivos de texto se gestionan desde la clase **GestorArchivosTexto**. Esta es utilizada de distintas maneras. En primer lugar es utilizada para crear números auto-incrementales. Creo ids para el número de reserva y creo números de recibo. En segundo lugar, es utilizada para crear esos recibos que serán el comprobante del cliente. Estos contienen todos los datos de la reserva. Los guardo en un subdirectorio: **Recibos**.

SERIALIZACIÓN

La creación y lectura de archivos **xml** y **json** se gestionan desde la clase **Serializadora**.

Los archivos **xml** son gestionados para guardar o leer los datos de los clientes. En el caso de la lectura, solo podremos ver los mails de los clientes a través del botón **Mails Clientes**, ya que es lo único que nos importa del cliente en este sistema. Al verlos podemos copiarnos los mails que necesitamos de ellos para proceder a enviar las promociones de la empresa.

Por otra parte, los archivos **json** son gestionados para guardar o leer los listados que contienen la información de los equipos que la

empresa tiene en stock y las reservas de los clientes que se encuentran activas.

Algunas aclaraciones:

La modificación de reservas reutiliza el formulario que da de alta las reservas. se envían todos los datos que contiene esa reserva al formulario para poder realizar las modificaciones pertinentes. La baja de reservas elimina esa reserva, devolviendo el listado de equipos reservados al listado de equipos disponibles, ya que vuelven a poder ser elegibles. Estos vuelven acomodándose al final del listado.

Los mails de los clientes de momento se encuentran repetidos y sin filtrar. La idea es filtrarlos para que los muestre una única vez, a pesar de que el cliente haya venido varias veces a alquilarle a la empresa. Estos clientes pueden darse de alta muchas veces, ya que cada vez tendrán un id de reserva distinto, que me permitirá asociar a las reservas y saber qué es lo que el cliente alquiló en cada ocasión.