

Instancia TP3 (8/11)

La aplicación creada tiene por función la compra de entradas del cine, a través del uso de formularios se puede:

- 0- Se ingresan datos de Usuario (actualmente lo único válido es Usuario:admin Contraseña:admin, una vez que se creen las bases de datos se podrá registrar un usuario en su cuenta y que por ende se rellene el nombre al cargar el formulario de compra de entradas, mientras que admin tiene la capacidad de ver 2 botones invisibles para el usuario común, botones que no realizan nada de momento por ser relacionados directamente a alterar las bases de datos y realizar análisis de las mismas) en caso contrario se puede no crear una cuenta en el momento,
- 1- Visualizar una lista de películas (actualmente cargadas a mano y con picturebox vacíos, que para la instancia de TP4 serán obtenidas de una base de datos las películas junto a sus carátulas),
- 2- Al seleccionar una película y darle a comprar se abre otro formulario que permite la carga de datos referentes al usuario, como su nombre completo, la cantidad de entradas que desea, el horario y qué tipo de película quiere ver (2D,3D,4D),
- 3- Al confirmarse la compra se crea un Ticket.xml en la carpeta MisDocumentos o en caso contrario se guarda en la posición del proyecto si fuera a no poderse guardar en la carpeta MisDocumentos, el ticket contiene toda la información relevante para el comprador, como los datos ya dichos y el precio, y se le avisa al usuario que su carga fue exitosa permitiéndole ver el recién creado ticket, luego de eso puede cerrarlo y volver a abrirlo desde el menú principal (el menú con la lista de películas) utilizando el menustrip para abrir archivo y abrir su ticket.

NUEVA INFORMACIÓN: El análisis de datos de usuarios registrados está completamente aislado para el uso del usuario admin, para ingresar a él use (USUARIO: admin CONTRASEÑA: admin), ya se puede ingresar correctamente a otros usuarios sin necesidad de que esté hardcodeado en el programa, si necesita información de otros usuarios, como sus usuarios y contraseñas, se encuentran en la tabla UsuariosRegistrados, pero si no desea usar un usuario ya creado puede crearlo a través del botón Registrarse que tiene el formulario de Inicio, eso le va a dar las mismas funcionalidades que cualquier otro usuario registrado (excepto admin). Ya no tienen uso los métodos ordenar por Id de las clases que heredan de la interfaz ya que en su momento entendía que tenía que ver “facilitar” el análisis de datos, una vez que se explicó qué se refería por análisis de datos entendí que no tenía uso. Pero debido a su posible expansión como proyecto, el ordenamiento tiene sentido si se quisiera agregar una manera de visualizar las tablas ordenadas según el id que interese.

Clase 10: Excepciones

Las excepciones están utilizadas en distintos espacios de los proyectos, desde algunas propiedades de las diferentes clases, los formularios y el Test Unitario que comprueba que las propiedades designadas lanzan una excepción creada personal llamada PeliculaMalCargadaException de manera correcta. Las propiedades de la clase Película tienen varias implementaciones de esta excepción.

Clase 11: Prueba Unitaria

La prueba unitaria está llamada TestCine y tiene actualmente dos métodos de prueba (en la instancia de TP4 planeo crear tests para mi base de datos y sus distintas tablas, entre estas tablas estarían la tabla de películas, las de usuarios, la de tickets creados).

Clase 12: Tipos Genéricos

Los tipos genéricos fueron implementados en la clase estática FormGenerico que permite realizar formularios siguiendo el patrón de diseño singleton (si no existe lo crea, si existe lo reenvia), donde se realizó una sobrecarga de método para que se puedan crear constructores con un parámetro de tipo genérico, algo que se utilizó mucho para que varios formularios compartan información

Clase 13: Interfaces

Creada la interfaz IOrdenamiento cumple dos funciones, le obliga a crear una propiedad Id de solo lectura a todas las clases que la implementen, permitiendo relacionarse con una base de datos, sin que pueda alterarse el id (único) de la base de datos, además de que al ser de solo lectura fuerza a que este id no sea visible por el serializador de xml.

Clase 14: Archivos y Serialización

La clase estática Ticket es la que tiene el manejo de archivos y de serialización Xml, esta clase es llamada a través de la consola Test y de los formularios de creación y lectura de Tickets (formulario FormCompraEntrada crea los Tickets y los formularios FormExito y FormListaPelicula (este se debe al uso del ToolStripMenu Abrir Archivo)).

Clase 15 y 16: Base de Datos

La clase estática SqlHandler es la que se encarga de (a través de sus métodos estáticos) conectar con la base de datos y retornar la información necesitada de las tablas. Esta clase se utiliza en muchos formularios, desde el de listarPeliculas, hasta el de análisis de datos.

Clase 17: Delegados y expresiones lambda

Se utilizó un delegado en la clase Pelicula, con objetivo a usarse en un evento de suma de visitas. Mientras que la expresión Lambda se utilizó en algunas propiedades de Usuario Registrado en su get, pero más principalmente se usó la expresión Lambda en el test unitario TestCine que permite, en el método TestSumarVisitas, crear un método anónimo simulando un botón que haya sumado una visita como lo hace el formulario de películas.

Clase 18:Hilos

Los hilos se utilizaron para poder realizar una descarga de tickets en la nube, mientras que todavía sea utilizable la aplicación, al hacerlo en simultáneo la descarga se detiene correctamente al cerrar el formulario usando un token de cancelación. Esta descarga se encuentra en el formulario ListarPeliculas

Clase 19:Eventos

Los eventos se utilizan constantemente en formularios, pero el evento personalizado para la clase Peliculas se invoca cuando alguien intenta sumar un número de visitas a través de la propiedad Visitas.

Clase 20:Métodos de extensión

Estos no fueron creados para la instancia de TP4, ya que su uso tenía importancia desde un principio para alterar el comportamiento de la clase Lista y de la clase StringBuilder, se pueden encontrar en las clases StringBuidlerExtension y ListExtension.

Esto también se encuentra en el summary de la consola Test y de la clase Película (que fue la que empezó todo) para facilitar la corrección.