



ENTREGA 2DO TP INTEGRADOR

Alumno: Uriel Nicolás Marlés

División: 2C

Objetivo del programa:

Rent-a-tronic, es un programa utilizado por una empresa que alquila animatrónicos por un día a fiestas de niños, con el objetivo de ganar dinero y recolectar esencia de sus almas.

Funcionalidades del programa:

- Tiene un sistema de usuarios para separar lo que puede hacer un administrador y un empleado común, ocultando información siniestra a los empleados comunes.
- Cada animatrónico tiene una agenda personal, a la hora de intentar reservar una nueva fiesta, tras elegir la fecha, únicamente aparecerán los animatrónicos que no tengan una fiesta reservada en ese día
- Cada fiesta puede contener hasta tres animatrónicos (la relación entre animatrónicos y fiestas está en una tabla intermedia)
- Cada tabla de información (animatrónicos /fiestas/usuarios) cuenta con una interfaz para el editado de cada tipo de entrada y una barra de búsqueda para encontrar la entidad según el nombre.
- Desactiva las fiestas automáticamente una vez la fecha pasa, manteniendo un historial de fiestas pasadas y calculando los ingresos y almas recolectadas en las fiestas ya ocurridas
- Lleva un historial del empleado encargado de cada fiesta y de la cantidad de fiestas realizadas por cada empleado, en fin de poder repartir el dinero adecuadamente
- Permite ingreso de información desde CSV y guardado de Backups en CSV, ambas actividades se pueden hacer desde la interfaz de la aplicación, se pueden crear infinitos backups sin sobrescribir.

Aplicado de los temas requeridos:

- Excepciones
  - El archivo Scripts/Excepciones.cs contiene todas las excepciones personalizadas utilizadas en el programa

- Cada excepción es utilizada en un contexto específico para diferenciarla de excepciones no controladas, se utilizan tanto en el front como en el back
- Ante la aparición de una excepción no manejada hay un formulario creado para mostrar un desglose de lo ocurrido
- Pruebas unitarias
  - En el proyectos TESTS se encuentran pruebas unitarias que verifican la correcta comunicación con la base de datos, el envío de información, la modificación de información obtenida y la recepción
- Interfaces
  - La interfaz implementada es IEntidades, que conecta el comportamiento en común de todas las entidades, las cuales representan una fila existente o por existir en la base de datos.
- Genericos
  - Los genéricos son empleados principalmente en dos lugares: el form frmTablas<T> y la clase ColeccionEntidades<T>
  - El form FrmTablas recibe una clase que implemente IEntidades y crea un formulario mostrando una tabla y las formas de interactuar con ella con la información a la cual el usuario tiene acceso
  - La colección entidades tiene la principal función de filtrar las entidades dentro de su colección utilizando diferentes métodos.
- Archivos
  - La comunicación con archivos se realiza dentro de la clase estática FileManager, esta clase se encarga de la escritura, lectura y traducción con los archivos que el programa requiere.
- Delegados y eventos
  - Hay múltiples usos de eventos, los dos más notorios son eventoUsuarioObtenido y eventoCerrando
  - EventoUsuarioObtenido avisa a todos sus suscriptores que a través de un nuevo formulario de inicio de sesión (se puede cambiar la sesión tras abrir el programa) se recibió un nuevo usuario, esto requiere una serie de cambios tanto visuales como en la lógica y para evitar crear nuevamente los formularios ya instanciados, simplemente se utiliza el evento para actualizar la información.
  - eventoCerrando permite informar al form principal que se confirmó un cerrado de la aplicación desde un formulario hijo y que no tiene que activar su FrmClosing para confirmar.
- Lambda
  - Dos ejemplos de lambda son el asignado al eventoUsuarioObtenido de la función que guarda el usuario en el form y la función que muestra un error en la tabla durante 5 segundos cuando un usuario sin admin intenta realizar una acción para la cual no tiene permisos
- Multihilo/asincronismo

- El uso más evidente de asincronismo es en los botones del formulario de frmTabla, los mensajes que estos muestran están temporizados con un `TASK.sleep()` para luego desaparecer, esta demora no impide que se continúe apretando el mismo botón o que se interactúe con el resto del programa, debido a ser una TASK en un hilo diferente.
- Conexión a base de datos / SQL
  - El generado de comandos sql dinámicamente a esta realizado dentro de la clase `ConexionConTabla`, dicha clase se instancia con la `connectionString` y hay una única instancia a lo largo de todo el proyecto, lo que permite que si se llegase a cambiar el nombre de la base de datos únicamente habría que cambiar esa única mención de la `connectionString`
  - En caso de una migración la creación de las tablas los métodos dentro de `FileManager` permiten la creación rápida de las mismas y el cargado del backup realizado.