

TP 4 – SISTEMA DE GESTION DE SERVICIOS DE LIMPIEZA Y REPARACION DE DISPOSITIVOS

FERRERO VALENTIN – DIVISION 2A

Es un sistema utilizado para gestionar el ingreso de nuevos dispositivos ya sea para limpieza o reparación y darle seguimiento al servicio (fecha de entrega, servicio pendiente/completado, etc.). Permite ingresar clientes, ver datos y servicios asociados.

Utilización del programa:

En la pantalla principal se despliega una lista de los servicios próximos a entregar (si es que los hay). Para cargar un ingreso se pulsa el botón 'Atender', selecciona un cliente de la lista, o se da uno nuevo de alta, y se ingresan los datos del nuevo servicio.

Para ver todos los servicios cargados se pulsa el botón 'Gestionar Servicios', que abre un nuevo form donde se pueden ver todos los servicios, su fecha de entrega, que tipo de trabajo se requiere y un detalle. Si existen servicios con fecha de entrega anterior al día actual, se emite una alerta que notifica el atraso. Hay dos listas, una de pendientes y otra de completados, se puede seleccionar un servicio de una lista y pasarlos a pendiente/completado según corresponda. Desde el mismo menú también se puede cargar un nuevo servicio, eliminar uno ya cargado o guardar la lista de servicios.

Para ver todos los clientes cargados se pulsa el botón 'Gestionar Clientes', que abre un nuevo form donde se pueden ver todos los clientes en el sistema con sus datos. Desde el mismo menú se puede cargar, modificar o eliminar un cliente, así como también puede ver los servicios que el cliente tiene asociados, o guardar el listado de clientes del sistema. Toda esta gestión de datos se hace con SQL, con

un id auto incremental, ejecutar query antes de ejecutar el programa.

Para cargar o guardar los archivos del sistema, puede ir a los menús específicos de gestión de las diferentes listas, o puede pulsar el botón 'Gestionar Archivos', donde podrá realizar el guardado, la carga o la eliminación de datos (pulsando el botón 'Borrar Datos.').

Al cerrar el programa, se pregunta si se quieren guardar todos los cambios, en el caso de que responda si se guardaran las listas, caso contrario no se guardaran los datos, salvo que el usuario haya guardado manualmente.

10 – Excepciones:

Se genero un nuevo class library que contiene las excepciones propias. Excepciones tales como CampoVacioException, NoEsNumericoException y DniInvalidoException se utilizan para verificar el ingreso de datos desde distintos forms(ej. FormAtender o formDatosCliente). Por otro lado, ErrorArchivoException se arroja cuando falla la lectura o el guardado de cualquier archivo. Por último, ErrorBaseDeDatosException se arroja cuando ocurre un error al leer, escribir o borrar en la base de datos.

11 – Pruebas Unitarias:

Se realizan las pruebas unitarias de los métodos de la clase ListadoGenerico<T> para verificar el funcionamiento de métodos clave como EliminarElemento o ExisteElementoEnLaLista. Todos los test son completados con éxito.

12 – Tipos Genéricos:

Se creo una nueva clase llamada ListadoGenerico<T>, utilizada para manejar listas de distintos tipos, así como agregarle o eliminarle elementos, ver si un elemento existe en el listado o leer/guardar los

listados. Esta se utilizó para la creación de las listas de Clientes y Servicios, ya que, si bien no tenían los mismo atributos o usos, los requerimientos del sistema para con las listas eran los mismos, como el manejo de archivos o la carga o eliminación de elementos de las listas.

13 y 14 – Interfaces y archivos:

Se creo la interfaz IArchivos con los métodos Guardar y Leer que luego se utilizan en la serialización de archivos. Se creo la clase SerializadoraXml para el manejo de archivos .xml, tanto para leer como para guardar, la cual es utilizada dentro de ListadoGenerico<T>. También se utiliza un GestorTxt, para leer y guardar los ultimosIds tanto de los clientes como de los servicios.

15 – Base de datos:

Se utiliza para la gestion de una de las dos listas, el listado de clientes. Antes de comenzar hay que ejecutar la query que crea la base de datos con su tabla correspondiente y le carga datos. Utiliza SQL para cargar, modificar y eliminar clientes. El id del cliente ya no es cargado manualmente, se utiliza un id auto incremental, de 1 en 1.

16 – Delegados y expresiones Lambda:

Se utiliza delegados en dos oportunidades. En un primer momento se utiliza el delegado DelegadoActualizarListado en formPrincipal, al que se le cargan tres métodos, uno que ordena la lista de servicios, otro que la carga en el dataGridView y un último que verifica las fechas de entrega. Esto resulta muy útil porque estos tres métodos se utilizaban en varias oportunidades, por lo que resulta más sencillo agruparlos e invocarlos en una sola línea. Luego utilizo expresiones Lambda cuando ejecuto una task en el load del formPrincipal y cuando ordeno la lista de servicios antes mencionada.

17 – Hilos:

Hago uso del Task cuando cargo la lista de servicios al iniciar el programa, ya que ralentiza mucho la carga del programa si hay muchos servicios en el sistema, por lo que resulta necesario que dicha tarea se ejecute en otro hilo.

18 – Eventos:

Utilizo eventos para notificar al usuario que hay servicios vencidos, es decir que no se marcaron como completados antes de la fecha de entrega estipulada. Para que no resulte molesto, en el formPrincipal el evento hace visible un label que avisa del problema, pero en la sección Gestionar Servicios, en lugar de un label es un MessageBox.