

Alumno: Fermin Gonzalez

División: 2A

Trabajo Práctico Nro 3 - Gimnasio

Metodos : amarillo.

Clases: verde.

El programa trata sobre un Gimnasio que permite el alta,baja y modificación de sus clientes.

Mostrar: Consiste de un formulario(frmClientes) donde en un DataGridView se muestra la lista de clientes, en este mismo formulario, están los botones que permiten dar de alta, modificar, eliminar o generar la credencial de un cliente.

Alta: Consiste de un formulario(frmAgregarCliente) donde se completan los campos con los atributos del cliente a agregar.

Modificación: Consiste de un formulario(frmModificarCliente) donde se pueden cambiar los valores de los atributos del cliente, a excepción del DNI que siempre permanece con el mismo valor.

Baja: Se da en el formulario(frmClientes) donde se elige un cliente de la lista y se lo elimina definitivamente.

Credencial: Consiste de un archivo.txt donde se guardará un texto con datos del cliente que se seleccionó en la **ruta o archivo que el usuario elija**.

Cargar Clientes: Permite cargar un archivo de extensión **.xml** o **.json** desde la **ruta que elija el usuario**, con el que se carga una lista de clientes para el gimnasio.

Guardar Clientes: Permite guardar la lista de clientes del gimnasio en un archivo de extensión **.xml** o **.json**, en la **ruta que elija el usuario**.

Temas vistos:

Excepciones (Clase 10) :

ArchivoIncorrectoException (IO): Excepción propia que se utiliza en los métodos, **ValidarSiExisteElArchivo** y **ValidarExtension** de la clase **Archivo**.

La mayoría de las excepciones se manejan en los formularios, todas utilizan el método **MostrarMensajeDeError** para mostrar el mensaje de error y los detalles del StackTrace.

Pruebas Unitarias (Clase 11) :

Se encuentran en la clase **Pruebas**, se testea:

Los métodos más relevantes de la clase **Cliente** y la clase **Gimnasio**.

y además, el método **ValidarExtensión** en las clases **PuntoXml**, **PuntoJson** y **PuntoTxt**.

Tipos Genericos (Clase 12) :

Se utilizan en las clases **PuntoJson**, **PuntoTxt**, **PuntoXml** y en la interfaz **IArchivo** de la biblioteca de clases IO, les permite serializar y deserializar contenido genérico.

PuntoJson y **PuntoXml** están implementadas en el **FrmPrincipal** para manejar la carga y el guardado de una lista de clientes(**List<Clientes>**).

PuntoTxt se implementa en el **FrmClientes** donde maneja el guardado de la credencial(string) del gimnasio.

Interfaces (Clase 13) :

IArchivo, está declarada en la biblioteca de clases IO y utilizada por las clases **PuntoJson**, **PuntoXml** y **PuntoTxt** para declarar los métodos **Guardar**, **GuardarComo** y **Leer** contenido genérico.

Archivos (Clase 14) :

Se utilizan en la clase **PuntoTxt** para escribir y guardar en un archivo.txt la credencial del gimnasio.

Serialización (Clase 15) :

Se utiliza en la carga y el guardado de archivos .xml y .json, las clases de la biblioteca de clases IO contienen todos los métodos necesarios para aplicarla. El frmPrincipal contiene los dos botones que permiten la implementación.

Aclaraciones:

1. La lista de clientes se puede cargar solo una vez, pero se puede guardar cuantas veces quiera en el formato que se elija.
2. En la baja, los clientes se eliminan definitivamente.
3. En el repositorio hay una carpeta con el nombre “**ARCHIVOS_CLIENTES**”, que contiene 2 listas de clientes, una en formato .json y otra en .xml.
4. Si se cargó un archivo con una lista de clientes, al cerrar el formulario frmClientes, se actualizará el archivo en el que se cargaron los clientes automáticamente.