

App: GYM

La aplicación se trata de una atención al cliente de un gimnasio, donde se puede dar de alta y baja clientes, cobrarles el servicio mensual, y venderles productos.

TEMAS

EXCEPCIONES

Usado en:

- **“Archivo excepción”**: en la biblioteca **“WinFormsGym”**, siempre que lea o escriba un archivo
- **“SinStockExcepcion”**: en **“FrmVentaProductos.cs”**, cuando vendo un producto

Durante todo el código hago try catches constantemente, en cada función que pueda tirar una excepción.

También, declaré dos excepciones propias:

1. Una está en la biblioteca “Archivos”: “ArchivoExcepcion”, la cual, si ocurre un error en el archivo que quiero leer o escribir, se arroja.
2. La otra está en la carpeta “Entidades”: “SinStockExcepcion” que se arroja cuando el stock de un producto no alcanza para vender la cantidad de productos que quiere comprar el cliente.

UNIT TESTINGS

Usado en: “UnitTesting.cs”

Tienen su propia biblioteca: “UnitTesting”, en esta hago los test unitarios de algunas de las funciones del programa.

TIPOS GENÉRICOS

Usado en: “Gym.cs” y “SerializacionJson.cs”

Gym.cs permite cargar a una lista cualquier objeto. Sea “Persona” como los 2 tipos de productos que el Gym vende (“AccesoriosDeportivos” y “IndumentariaDeportiva”)

SerializacionJson.cs permite hacer escritura y lectura de archivos que contengan los 2 tipos de productos que el Gym vende (“AccesoriosDeportivos” y “IndumentariaDeportiva”)

INTERFACES

Usado en biblioteca de clases “FrmVentaProductos.cs”

Interfaz implementada en “Entidades.cs”: Implemento la interfaz “IVenta”, que maneja métodos y una propiedad que tenga que ver con la venta de productos

ARCHIVOS Y SERIALIZACIÓN

Usado en biblioteca de clases “Archivos”

Hago la escritura y lectura de los distintos archivos en la biblioteca de clases “WinFormGym”.

El Json lo uso para productos RW (En “VentaProductos.cs”). El Xml para personas RW (En “FrmAlta”, “FrmAtender”). Y el txt para detalles de venta W(En “FrmVentaProductos.cs”)