TP4 – Federico Gutierrez – 2D – 20/11/20

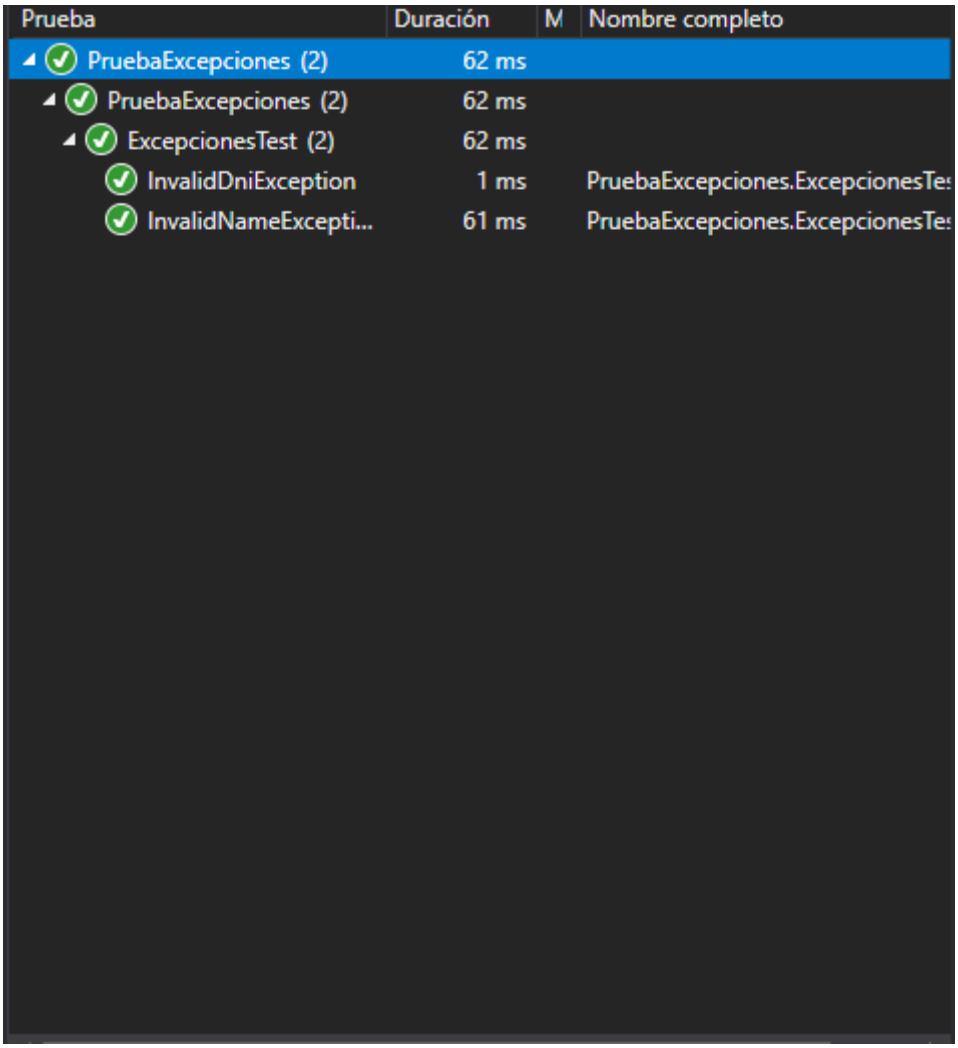
El siguiente programa pretende dar el servicio de Registro, Ingreso, Compra e Ingreso de Productos.

Excepciones

Las excepciones, en el programa son utilizadas para mostrar errores en tiempo de ejecución tales como el Ingreso invalido de el Nombre o el DNI a la hora del registro o cuando haya algún problema a la hora de cargar las compras.

Test Unitarios

En este caso, para comprobar que se lancen correctamente las excepciones en tiempo de ejecución



Tipos Genéricos e Interfaces

Los tipos genéricos son útiles en este programa, por ejemplo, para el manejador de la base de datos, en el cual los métodos que utiliza, tiene parámetros tipo T que a su vez el tipo T tiene que implementar una Interfaz.

```
/// <summary> Inserta el Objeto a la DB
2 referencias
public static Boolean Insert<T>(T objeto) where T: IDb
{
    Boolean rtrn = false;
    objeto.Object_To_DB(ref DB.command);
    try
    {
        if (sqlConn.State == System.Data.ConnectionState.Closed)
        {
            sqlConn.Open();
        }
```

```
public interface IDb
{
    /// <summary> Paso por referencia la sentencia SQL para hacer el INSERT
    3 referencias
    void Object_To_DB(ref SqlCommand command);

    /// <summary> Paso por referencia la sentencia SQL para hacer el SELECT
    3 referencias
    void DB_Select_Query(ref SqlCommand command);

    /// <summary> Paso por referencia la sentencia SQL para hacer el UPDATE
    3 referencias
    void DB_UpdateObject(ref SqlCommand command);

    /// <summary> Paso el SqlDataReader de la clase DB que contiene los datos leidos ...
    3 referencias
    object DB_GetObjectFromRegistro(SqlDataReader reader);

    /// <summary> Paso por referencia la sentencia SQL para hacer el DELETE
```

Archivos y Serialización

En este caso en archivos se guarda todas las compras realizadas, serializadas para que al iniciar el programa que carguen en memoria.

```
1 referencia
public static void LeerCompras()
{
    List<Compra> listaCompras;
    Xml<Compra> archivoXml = new Xml<Compra>();

    try
    {
        StringBuilder rutaStr = new StringBuilder(AppDomain.CurrentDomain.BaseDirectory);
        string nombreArchivo = "Compras.xml";
        rutaStr.Append(nombreArchivo);
        archivoXml.Leer(rutaStr.ToString(), out listaCompras);
        Negocio.ListaCompras = listaCompras;
    }
    catch (Exception e)
    {
        throw new LeerComprasException(e.Message, e.InnerException);
    }
}
```

Base de datos y SQL

Se utilizo una base de datos SQL para almacenar los datos de los Productos y Clientes.

```
11 referencias
public static class DB
{
    const string STRINGCONNEC = "Server=.\\sqlexpress;Database=Negocio;Trusted_Connection=True;";

    static SqlConnection sqlConn;
    static SqlCommand command;

    0 referencias
    static DB()
    {
        sqlConn = new SqlConnection();
        sqlConn.ConnectionString = STRINGCONNEC;
        command = new SqlCommand();
        command.Connection = sqlConn;
    }
```

Threads

Se uso Hilos para levantar los datos de la DB y las Compras, casi en simultaneo.

```
3 referencias
public static void IniciarNegocio()
{
    List<Thread> listaHilos = new List<Thread>();
    listaHilos.Add(new Thread(CargarUsuarios));
    listaHilos.Add(new Thread(CargarProductos));
    listaHilos.Add(new Thread(CargarCompras));

    try
    {
        foreach (Thread item in listaHilos)
        {
            Thread.Sleep(500);
            item.Start();
        }
    }
```

Eventos y Delegados.

Se uso eventos y delegados para que al cerrar el Formulario de Agregar Producto se actualice el número de Stock total.

Métodos de Extensión

Se utilizo un método de extensión para las Listas de Clientes ya que se requería buscar por DNI en la lista y que retornase el Cliente

```
0 referencias
static class ListaExtends
{
    1 referencia
    public static Cliente BuscarPorDNI(this List<Cliente> listaClientes, int dni)
    {
        if (listaClientes.Count == 0 || listaClientes == null)
            throw new NoHayUsuariosException();

        foreach (Cliente cli in listaClientes)
        {
            if(cli.DniUsuario == dni)
            {
                return cli;
            }
        }
        return null;
    }
}
```