

**Universidad Tecnológica Nacional Facultad
Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Apellido y nombre: Kevin Martin Plucci

Fecha: 12/07/2021

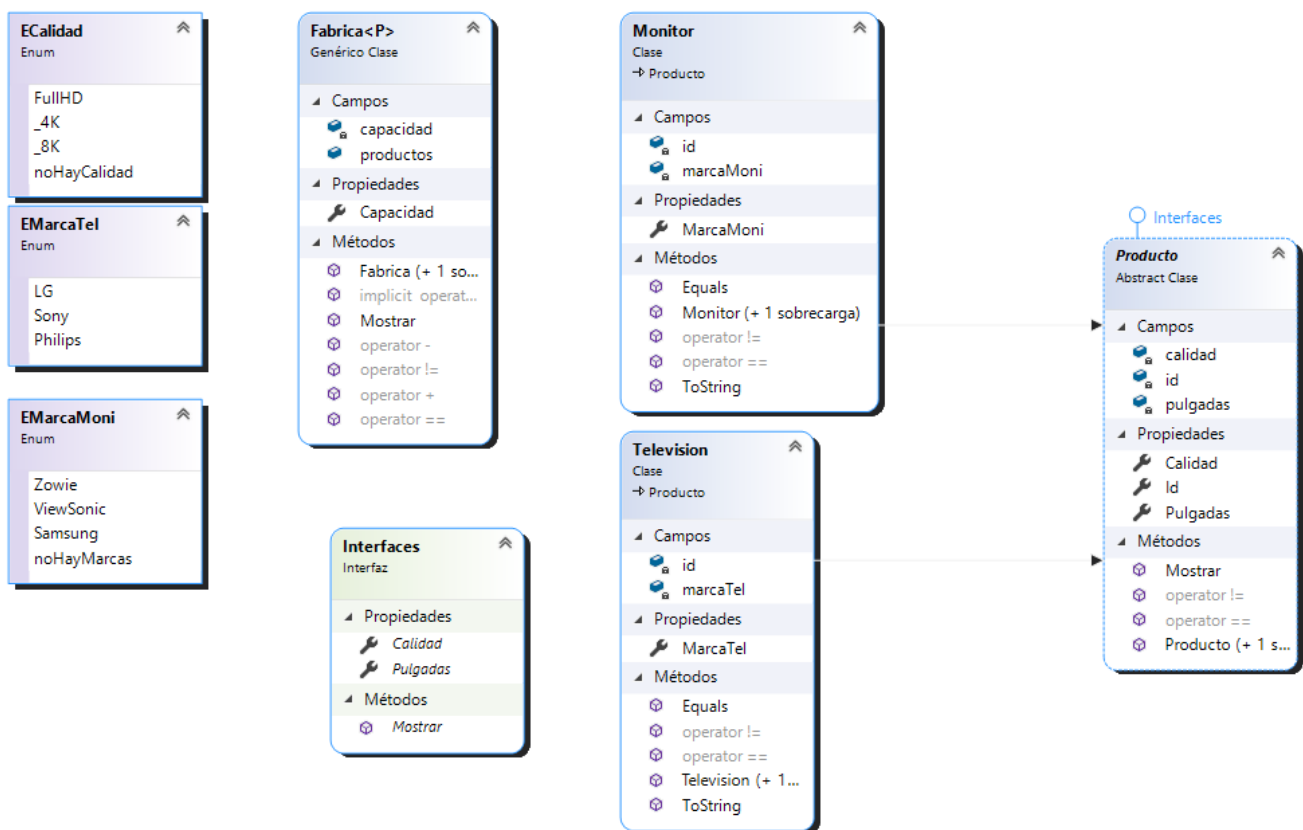
Trabajo Practico Número Cuatro

Nota:

División: 2A

Firma: Kmp.

• Diagrama de Clases:



Correcciones del TP Nº 3

Plucci, Kevin	Métodos de test sin nombre TestMethod1				
	Cumplir con las reglas de estilo: fabrica_				
	Comentar código, en el caso de los test poner que deberían hacer (sin nombre y sin documentar, se hace difícil)				
	Testear excepciones, archivos, cosas muy permeables a errores				
	Agregar funcionalidad. Solo se puede cargar info y guardar, no levanta dicho archivo, al cerrar el form se pierde todo, no se ejecuta ningún proceso. Flojo.				
	SI	SI	D	SI	Al cerrar el form no cierra la aplicación.
					Puedo agregar muchas veces el mismo producto, no hay excepción como nombra el documento adjunto.

Métodos de test sin nombre TestMethod1

Cada TestMethod fue cambiado por su nombre de proceso que hace cada uno (Igualdad y distinto).

Cumplir con las reglas de estilo: fabrica_

Agregamos el Where y ordene más el Summary. Con su regla de estilo

Comentar código, en el caso de los test poner que deberían hacer (sin nombre y sin documentar, se hace difícil)

Se comentó todo el código y acá haremos una explicación (e documentaremos) los test faltantes que tuvimos problemas con el primer punto a corregir.

Comprobamos la sobrecarga de == de los productos.

```
/// <summary>
/// Test Unitario de Sobrecarga del operador == productos
/// </summary>
[TestMethod]
0 referencias
public void VerificarIgualdad()
{
    int fabrica_ = 4;
    int pulgadasMonitor_ = 1;
    int pulgadasTelevisor_ = 1;
    int idMonitor = 1;
    int idTelevisor = 1;
    ECalidad unitario = ECalidad._4K;
    ECalidad secundario = ECalidad.FullHD;
    EMarcaTel marcaT = EMarcaTel.Philips;
    EMarcaMoni marcaM = EMarcaMoni.ViewSonic;

    Fabrica<Television> f1 = new Fabrica<Television>(fabrica_);
    Fabrica<Monitor> f2 = new Fabrica<Monitor>(fabrica_);
    Television t1 = new Television(idTelevisor, pulgadasTelevisor_, unitario, marcaT);
    Monitor m1 = new Monitor(idMonitor, pulgadasMonitor_, secundario, marcaM);

    //YA INGRESADO
    //f1 += c1;
    f1 += t1;

    //YA INGRESADO
    f2 += m1;

    Assert.AreEqual(t1.Id, idTelevisor);
    Assert.AreEqual(m1.Id, idMonitor);
    Assert.AreEqual(t1.Pulgadas, pulgadasTelevisor_);
    Assert.AreEqual(m1.Pulgadas, pulgadasMonitor_);
    Assert.AreEqual(t1.Calidad, unitario);
    Assert.AreEqual(m1.Calidad, secundario);
    Assert.AreEqual(t1.MarcaTel, marcaT);
    Assert.AreEqual(m1.MarcaMoni, marcaM);
}
```

Y el otro Test Unitario para probar los distintos productos

```
/// <summary>
/// Test Unitario de Sobrecarga del operador != productos
/// </summary>
[TestMethod]
0 referencias
public void VerificarDistinto()
{
    int fabrica_ = 4;
    int idMonitor = 1;
    int idTelevisor = 1;
    int pulgadasMonitor_ = 1;
    int pulgadasTelevisor_ = 1;
    ECalidad unitario = ECalidad._4K;
    ECalidad secundario = ECalidad.FullHD;
    EMarcaTel marcaT = EMarcaTel.Philips;
    EMarcaMoni marcaM = EMarcaMoni.ViewSonic;

    Fabrica<Television> f1 = new Fabrica<Television>(fabrica_);
    Fabrica<Monitor> f2 = new Fabrica<Monitor>(fabrica_);
    Television t1 = new Television(idTelevisor, pulgadasTelevisor_, unitario, marcaT);
    Monitor m1 = new Monitor(idMonitor, pulgadasMonitor_, secundario, marcaM);

    //YA INGRESADO
    //f1 += c1;
    f1 += t1;

    //YA INGRESADO
    f2 += m1;

    Assert.AreEqual(t1.Id, idTelevisor);
    Assert.AreEqual(m1.Id, idMonitor);
    Assert.AreEqual(t1.Pulgadas, pulgadasTelevisor_);
    Assert.AreEqual(m1.Pulgadas, pulgadasMonitor_);
    Assert.AreEqual(t1.Calidad, unitario);
    Assert.AreEqual(m1.Calidad, secundario);
    Assert.AreEqual(t1.MarcaTel, marcaT);
    Assert.AreEqual(m1.MarcaMoni, marcaM);

    Assert.IsTrue(f1 != m1 && f2 != t1);
}
```

Testear excepciones, archivos, cosas muy permeables a errores

Se le agrego
excepciones a
los temas
antiguos y a
los
incorporados.
Un ejemplo:

```
#region Sobrecargas de Operadores
/// <summary>
/// Sobrecarga del operador igual entre dos monitores
/// </summary>
/// <param name="m1">Primer monitor</param>
/// <param name="m2">Segundo monitor a ser comparado con el monitor</param>
/// <returns>retorna si los monitores de los productos son iguales</returns>
3 referencias
public static bool operator ==(Monitor m1, Monitor m2)
{
    bool rta = false;
    m1 = new Monitor();
    m2 = new Monitor();
    try
    {
        if (!((Producto)m1 == (Producto)m2 && m1.MarcaMoni == m2.MarcaMoni))
        {
            rta = true;
        }
    }
    catch (Exception ex) { }
    return rta;
}
```

A la sobrecarga de operador == de Monitor le agregamos una Excepciones para evitar errores (con su código comentado como los puntos anteriores). Otro ejemplo con un tema más actual en Archivos:

```
1 referencia
private void GuardarDataTable()
{
    try
    {
        this.dt.WriteXmlSchema(xml_Monitores_Schema);

        this.dt.WriteXml(xml_Monitores);

        System.Threading.Thread.Sleep(2000);
        MessageBox.Show("Se han guardado el esquema y los datos del DataTable!!!");
    }
    catch
    {
        MessageBox.Show("Error al guardar el DataTable. ",
            "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Y entre otros más...

Agregar funcionalidad. Solo se puede cargar info y guardar, no levanta dicho archivo, al cerrar el form se pierde todo, no se ejecuta ningún proceso.

```
1 referencia
private void btnVerLog_Click(object sender, EventArgs e)
{
    DialogResult rta = openFileDialog1.ShowDialog();//Reemplazar por la llamada al método correspondiente del OpenFileDialog

    if (rta == DialogResult.OK)
    {
        //leer el archivo seleccionado por el cliente y mostrarlo en txtVisorTickets

        System.Threading.Thread.Sleep(2000);
        string path = openFileDialog1.FileName;
        string texto;
        StreamReader f = new StreamReader(path);

        texto = f.ReadToEnd();

        this.txtVisorTickets.Text = texto;
    }
}
```

Agregue una visualización y carga de los tickets tanto monitor como televisor (tampoco se pierde la información, ya que uso el DataGreed para incorporar los datos que colocho, los cargo a la base de datos y de ahí los serializo en el mismo DataGreed.

Flojo.

Se documentó todo, se comentó todo, se agregó nuevas funcionalidades al forms, se arregló las excepciones, etc.

Al cerrar el form no cierra la aplicación.

Ya cierra y guarda los datos al cerrar el forms.

```
1 referencia
private void VolverM_Click(object sender, EventArgs e)
{
    //FormInicio frmInicio = new FormInicio();

    //this.Hide();

    //frmInicio.Show();

    this.Close();
}
```

Puedo agregar muchas veces el mismo producto, no hay excepción como nombra el documento adjunto.

Error de concepto a la hora de documentar, es ilimitado su stock.

Temas incorporados del TP N°4

Hilos

```
System.Threading.Thread.Sleep(2000);
```

Se le agrego este delay para evitar problemas a la hora de cargar un archivo o un Message Box

Ejemplo:

```
{
    this.dt.WriteXmlSchema(xml_Monitores_Schema);

    this.dt.WriteXml(xml_Monitores);

    System.Threading.Thread.Sleep(2000);
    MessageBox.Show("Se han guardado el esquema y los datos del DataTable!!!");
}
```

Método de extensión:

Uno solo hay y es solo para evitar la saturación de productos.

```
11 referencias
public static Fabrica<P> operator +(Fabrica<P> d, Producto p)
{
    try
    {
        if (d.Capacidad > d.productos.Count)
        {
            if (d != p)
                d.productos.Add(p);
            else
                Console.WriteLine("El producto ya esta en la Fabrica.\n");
        }
        else
            throw new Exception();
    }
    catch (Exception ex)
    {
        ex.InformarProductos();
    }
    return d;
}
```

Base de datos:

Se establece una carga de datos hacia la base de datos que el empleado fabrica a la hora de hacer tantos Monitores o Televisores, cada uno con su ID de producto, Marca, Calidad de Video y Pulgadas, todo en un table ordenado ascendentemente por la ID:

```
1 referencia
private void guardar_Click(object sender, EventArgs e)
{
    List<Monitor> auxMonitor = new List<Monitor>();

    try
    {
        string consultaP = "SELECT * FROM Monitor WHERE id = @id";

        cn.Open();

        SqlCommand comandP = new SqlCommand(consultaP, cn);

        GuardarDataTable();

        //comandP.Parameters.AddWithValue("@id", CrearM.Text);
        comandP.ExecuteNonQuery();
        SqlDataReader infoP = comandP.ExecuteReader();

        while (infoP.Read())
        {
            ECalidad Calidad = (ECalidad)Enum.Parse(typeof(ECalidad), infoP["calidad"].ToString());
            EMarcaMoni MarcaMoni = (EMarcaMoni)Enum.Parse(typeof(EMarcaMoni), infoP["marca"].ToString());
            int id = Convert.ToInt32(infoP["id"].ToString());
            int pulgadas = Convert.ToInt32(infoP["pulgadas"].ToString());

            Monitor monitores = new Monitor(id, pulgadas, Calidad, MarcaMoni);
        }
    }

    catch(Exception ex)
    {
    }
}
```