**M07\_EAC1 (Interfaces gráficas, Visual Studio (Windows Forms/WPF), XML, XSLT)**

**Ejercicio1 (Duplicar con una aplicación Windows Forms una interface):**

****

1. El primer paso es crear un nuevo proyecto <<Aplicación "con interfaz de usuario" para Windows Forms”.
2. La interface de la imagen se compone (de arriba abajo):
   1. Un menú superior (Archivo, Edición, etc…) 🡪 Componente “MenuStrip”.
   2. Un panel y una serie de botones que se convertirán en iconos.
   3. Una serie de pestañas (Editor, Mostrar, Configurar, Extensión) que se realiza con el componente 🡪 TabControl.
   4. Por último Se compone de un botón (Mostrar).

**Ejercicio 2 (Crear una interfaz gráfica sobre una encuesta en la que se pregunta si la interfaz de una imagen que seleccionemos, cumple con los requisitos de <<Accesibilidad e intuición, Uso de metáforas, Aprendizaje y uso fácil, Consistencia, Legibilidad y Reducción de la carga de memoria>>)**:

Esta aplicación la tenemos que hacer con Visual Studio y WPF. WPF se base en componentes gráficos y el lenguaje <<XAMEL>>, que es una lenguaje de etiquetas parecido a HTML.

Hay varios elementos del lenguaje XAMEL que debemos tener en cuenta:

1. StackPanel: Sirve para “apilar” componentes uno debajo de otro. Tiene sus atributos como Alineamiento Horizontal, margen, alineamiento vertical, alto, ancho, etc.. Un ejemplo podría ser:
   1. <StackPanel HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top" Height="1818" Width="772">
2. Dentro de un StackPanel, puede y debe haber otros componentes. Por ejemplo una imagen, un texto, etc.. Veamos un ejemplo:
   1. Imagen: <Image Source="background.jpg" Margin="6,-5,9,9" Width="721" VerticalAlignment="Center" HorizontalAlignment="Center" Stretch="Fill" Height="232"/>
      1. Vemos como la imagen tiene una serie de atributos:
         1. Source: ruta y nombre donde se ubica la imagen.
         2. Margin: Margen con el resto de componentes que lo envuelven.
         3. Width/Height: Ancho y alto de la imagen.
         4. Vertical/Horizontal Alignment: Alineamiento horizontal y vertial del componente.
         5. Stretch: Atributo para indicar como se posicionará la imagen respecto a su componente, en este caso se ajusta en ancho y alto al borde que lo envuelve.
   2. Texto: En este caso para incluir un texto se utiliza el componente “Label”:
      1. Label: <Label FontSize="24" HorizontalAlignment="Center" VerticalAlignment="Center" Content="Encuesta sobre Interfaces gráficas"/>
         1. FontSize: tamaño de letra.
         2. Vertical/Horizontal Alignment: horizontal y vertial del componente.
         3. Content: texto que se verá en el componente.
3. Componente **Border**: El componente border, como bien indica su nombre es un borde que puede encapsular otro componente. Por ejemplo, una imagen, un texto o ambos. Tienes una serie de atributos. Pondremos un ejemplo de un borde y explicaremos sus atributos:
   1. <Border CornerRadius="10" Height="271" Margin="10,10,21,10" BorderThickness="1" Padding="5">

<StackPanel>

<Label HorizontalAlignment="Center" FontSize="18" Content="MODELO INTERFAZ GRÁFICA" Margin="248,0,230,-40" Height="33" Width="251"/>

<Image Source="interfaz.png" Cursor="Arrow" Margin="39,41,39,9" Height="201" VerticalAlignment="Center" Stretch="UniformToFill"/>

</StackPanel>

</Border>

* + 1. Podemos ver que tenemos un borde que encapsula un componente Label y una imagen. Los atributos de los que dispone son:
       1. CornerRadius: Redondeo del componente.
       2. Height: altura del componente.
       3. Margin: margen del componente.
       4. BorderThickness: ancho de la línea del componente.
       5. Padding: distancia entre los componentes que lo rodean.

**Ejercicio 3 (Hay que realizar un fichero XML sobre un librería, su hoja de estilos “xsl” y después incluir ambos archivos en un webForm (ASP.NET) y añadir varios componentes)**:

La creación del fichero XML la podemos obviar, por lo que pasamos a definir la hoja de estilos:

**NOTA**: tengo que comentar que la hoja de estilos para el fichero XML es diferente a l hoja de estilos del WebForm.

Hoja de estilos XML directa:

Las notaciones del fichero “xsl” en la cabecera, son las siguientes:

**<?xml version="1.0" encoding="utf-8"?>**

**<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">**

Abrimos un template:

**<xsl:template match="/">** 🡪 Entre comillas dobles, indicamos la raíz del documento.

Podemos abrir etiquetas HTML, por ejemplo: <html>, <body>, <h1>, <table>, etc…

Debemos hacer un recorrido de todas las etiquetas del documento. Para ello utilizaremos la expresión:

**<xsl:for-each select="LLIBRES/LLIBRE">”**

Lo que hay entre comillas dobles, es de donde queremos que empiece a recorrer, es la ruta, en este caso, la etiqueta raíz (RootElement) es “LLibres”, después le sigue cada “LLIBRE”.

Ahora por cada etiqueta diferente tenemos que utilizar la expresión:

<xsl:value-of select="NOM"/>

<xsl:value-of select="AUTOR"/>

<xsl:value-of select="EDITORIAL"/>

…

Por último, siguiente con el recorrido de “**xsl:for-each**”, si tenemos un campo de imagen que apunta a un fichero con su ruta, debemos utilizar la siguiente expresión:

**<img>**

**<xsl:attribute name="src">**

**<xsl:value-of select="IMAGEN"/>**

**</xsl:attribute>**

**<xsl:attribute name="height">**

**100**

**</xsl:attribute>**

**<xsl:attribute name="width">**

**100**

**</xsl:attribute>**

**</img>**

1. Se abre una etiqueta “<img>” de tipo HTML
2. Se indica que esa etiqueta “img” tiene un atributo “src” que viene siendo el “source” o el origen del fichero 🡪 **<xsl:attribute name="src">**
   1. Indicamos que el nombre del campo (IMAGEN) 🡪 **<xsl:value-of select="IMAGEN"/>**
3. Indicamos que la imagen tendrá otros atributos que serán el alto y el ancho:
   1. **<xsl:attribute name="height">100</xsl:attribute>**
   2. **<xsl:attribute name="width">100 </xsl:attribute>**

Hoja de estilos en Visual Studio:

Hay varias diferencias respecto a la hoja de estilos directa. Para empezar hay que abrir un “template” para cada etiqueta “raíz” o etiqueta que contiene elementos:

Template LLIBRES (Principal):

<xsl:template match="LLIBRES">

<LLIBRES>

<xsl:apply-templates/>

</LLIBRES>

</xsl:template>

Template Hijo (contiene todos los libros):

<xsl:template match="LLIBRE">

<LLIBRE>

<xsl:for-each select="\*">

<xsl:attribute name="{name()}">

<xsl:value-of select="text()"/>

</xsl:attribute>

<xsl:attribute name="src">

<xsl:value-of select="IMAGEN" />

</xsl:attribute>

</xsl:for-each>

</LLIBRE>

</xsl:template>

El segundo template conviene explicarlo:

1. Abrimos una etiqueta a la que hacemos referencia en el template, en el atributo “match=”LLIBRE”--:
   1. <LLIBRE>
2. Recorremos sus nodos hijos:
   1. <xsl:for-each select="\*">
3. Indicamos que las etiquetas o nodos tendrán un atributo “name” que puede ser de tipo texto (todas las etiquetas menos el nodo de la imagen), o de tipo “src”:
   1. Tipo texto:
      1. **<xsl:attribute name="{name()}">**

**<xsl:value-of select="text()"/>**

**</xsl:attribute>**

* 1. Tipo “src”:
     1. **<xsl:attribute name="src">**

**<xsl:value-of select="IMAGEN" />**

**</xsl:attribute>**

**Pasamos a trasladar esta información a un WebForm**:

1. Añadimos un componente “XmlDataSource”:
   1. Le indicamos el fichero origen y destino. Esto lo hacemos desde el menú de propiedades el propio componente:
      1. DataFile: buscamos el fichero XML
      2. TransformFile: buscamos el fichero xslt (hoja de estilos
2. Añadimos un objeto “GridView” y lo configuramos de la siguiente manera:
   1. En el menú derecho de propiedades del componente, nos vamos a la opción “DataSourdceID” e incluimos componente “XmlDataSource” anteriormente creado.
   2. Sólo nos queda configurar el formato de la tabla.

Mostramos el fichero XML:

Original:

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type='text/xsl' href='llibres.xsl'?>

<LLIBRES>

<LLIBRE>

<NOM>PYTHON A FONDO</NOM>

<AUTOR>OSCAR RAMIREZ JIMENEZ</AUTOR>

<EDITORIAL>S.A. MARCOMBO</EDITORIAL>

<PREU>37.86</PREU>

<ANY\_PUB>2021</ANY\_PUB>

<IMAGEN>imagenes/PYTHON A FONDO.jpg</IMAGEN>

</LLIBRE>

<LLIBRE>

<NOM>C# 8 Y VISUAL STUDIO 2019: LOS FUNDAMENTOS DEL LENGUA</NOM>

<AUTOR>SEBASTIEN PUTIER</AUTOR>

<EDITORIAL>ENI</EDITORIAL>

<PREU>28.40</PREU>

<ANY\_PUB>2021</ANY\_PUB>

<IMAGEN>imagenes/VISUAL STUDIO.jpg</IMAGEN>

</LLIBRE>

<LLIBRE>

<NOM>PROGRAMACION EN GO</NOM>

<AUTOR>MARIO MACIAS LLORET</AUTOR>

<EDITORIAL>S.A. MARCOMBO</EDITORIAL>

<PREU>18.99</PREU>

<ANY\_PUB>2020</ANY\_PUB>

<IMAGEN>imagenes/GO.jpg</IMAGEN>

</LLIBRE>

<LLIBRE>

<NOM>TALLER DE ARDUINO. EXPERIMENTANDO CON ARDUINO MKR 1010 (2ª)</NOM>

<AUTOR>GERMAN TOJEIRO CALAZA</AUTOR>

<EDITORIAL>S.A. MARCOMBO</EDITORIAL>

<PREU>18.99</PREU>

<ANY\_PUB>2020</ANY\_PUB>

<IMAGEN>imagenes/ARDUINO.jpg</IMAGEN>

</LLIBRE>

<LLIBRE>

<NOM>PROBLEMAS DE INGENIERÍA DE COMPUTADORES I</NOM>

<AUTOR>VV.AA.</AUTOR>

<EDITORIAL>SANZ Y TORRES</EDITORIAL>

<PREU>23</PREU>

<ANY\_PUB>2010</ANY\_PUB>

<IMAGEN>imagenes/PROBLEMA ING.jpg</IMAGEN>

</LLIBRE>

</LLIBRES>

Fichero directo xsl:

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h1 align="center">Catàleg de llibres:</h1>

<xsl:for-each select="LLIBRES/LLIBRE">

<div align="center">

<strong> Título </strong>

<xsl:value-of select="NOM"/>

<br/>

<strong> Autor </strong>

<xsl:value-of select="AUTOR"/>

<br/>

<strong> Editorial </strong>

<xsl:value-of select="EDITORIAL"/>

<br/>

<strong> Precio </strong>

<xsl:value-of select="PREU"/>

<br/>

<strong> Año Publicación</strong>

<xsl:value-of select="ANY\_PUB"/>

<br/>

<strong> PORTADA</strong>

<br/>

<img>

<xsl:attribute name="src">

<xsl:value-of select="IMAGEN"/>

</xsl:attribute>

<xsl:attribute name="height">

100

</xsl:attribute>

<xsl:attribute name="width">

100

</xsl:attribute>

</img>

<br/>

</div>

</xsl:for-each>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

XML Visual Studio:

<?xml version="1.0" encoding="UTF-8"?>

<LLIBRES>

<LLIBRE>

<NOM>PYTHON A FONDO</NOM>

<AUTOR>OSCAR RAMIREZ JIMENEZ</AUTOR>

<EDITORIAL>S.A. MARCOMBO</EDITORIAL>

<PREU>37.86</PREU>

<ANY\_PUB>2021</ANY\_PUB>

<IMAGEN>imagenes/PYTHON A FONDO.jpg</IMAGEN>

</LLIBRE>

<LLIBRE>

<NOM>C# 8 Y VISUAL STUDIO 2019: LOS FUNDAMENTOS DEL LENGUA</NOM>

<AUTOR>SEBASTIEN PUTIER</AUTOR>

<EDITORIAL>ENI</EDITORIAL>

<PREU>28.40</PREU>

<ANY\_PUB>2021</ANY\_PUB>

<IMAGEN>imagenes/VISUAL STUDIO.jpg</IMAGEN>

</LLIBRE>

<LLIBRE>

<NOM>PROGRAMACION EN GO</NOM>

<AUTOR>MARIO MACIAS LLORET</AUTOR>

<EDITORIAL>S.A. MARCOMBO</EDITORIAL>

<PREU>18.99</PREU>

<ANY\_PUB>2020</ANY\_PUB>

<IMAGEN>imagenes/GO.jpg</IMAGEN>

</LLIBRE>

<LLIBRE>

<NOM>TALLER DE ARDUINO. EXPERIMENTANDO CON ARDUINO MKR 1010 (2ª)</NOM>

<AUTOR>GERMAN TOJEIRO CALAZA</AUTOR>

<EDITORIAL>S.A. MARCOMBO</EDITORIAL>

<PREU>18.99</PREU>

<ANY\_PUB>2020</ANY\_PUB>

<IMAGEN>imagenes/ARDUINO.jpg</IMAGEN>

</LLIBRE>

<LLIBRE>

<NOM>PROBLEMAS DE INGENIERÍA DE COMPUTADORES I</NOM>

<AUTOR>VV.AA.</AUTOR>

<EDITORIAL>SANZ Y TORRES</EDITORIAL>

<PREU>23</PREU>

<ANY\_PUB>2010</ANY\_PUB>

<IMAGEN>imagenes/PROBLEMA ING.jpg</IMAGEN>

</LLIBRE>

</LLIBRES>

Fichero XSLT Visual Studio:

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="LLIBRES">

<LLIBRES>

<xsl:apply-templates/>

</LLIBRES>

</xsl:template>

<xsl:template match="LLIBRE">

<LLIBRE>

<xsl:for-each select="\*">

<xsl:attribute name="{name()}">

<xsl:value-of select="text()"/>

</xsl:attribute>

<xsl:attribute name="src">

<xsl:value-of select="IMAGEN" />

</xsl:attribute>

</xsl:for-each>

</LLIBRE>

</xsl:template>

</xsl:stylesheet>

WEB FORM (Campo imagen):

<asp:ImageField DataImageUrlField="IMAGEN" HeaderText="Portada">

<ControlStyle Height="75px" Width="50px" />

<HeaderStyle HorizontalAlign="Center" VerticalAlign="Middle"></HeaderStyle>

<ItemStyle HorizontalAlign="Center" VerticalAlign="Middle"></ItemStyle>

</asp:ImageField>