

# Producto 5. Transformación de un archivo XML mediante una hoja de estilo XSLT.

Rubén Vicente Gilabert

Uoc

## Índice

1. Objetivo del producto.....	2
2. Descripción de funcionalidades implementadas.....	2
3. Consideraciones técnicas.....	3
4. Resultados obtenidos.....	4
5. Archivos generados.....	4
6. Recursos utilizados.....	4

## 1. Objetivo del producto

El objetivo principal de este producto es transformar el archivo XML generado en el Producto 4 en un documento HTML usando una hoja de estilo XSLT.

## 2. Descripción de funcionalidades implementadas

Se han implementado las siguientes funcionalidades:

- Diseño de una hoja de estilo transformacion.xslt que convierte la estructura XML a un HTML estructurado y legible.

```
C:\Users\Rubo > Documents > UOC entregar > Programacion de comandos personalizados > Producto_4 > Producto_4 > transformacion.xslt
1  <?xml version="1.0" encoding="utf-8"?>
2  <xsl:stylesheet version="1.0"
3    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4
5    <xsl:output method="html" indent="yes" encoding="UTF-8"/>
6
7    <xsl:template match="/">
8      <html>
9        <head>
10         <title>Configuración de Red</title>
11         <style>
12           body {
13             font-family: Arial, sans-serif;
14             background-color: #f4f4f4;
15             padding: 20px;
16           }
17           h1, h2 {
18             color: #333;
19           }
20           table {
21             border-collapse: collapse;
22             width: 100%;
23             margin-bottom: 40px;
24           }
25           th, td {
26             border: 1px solid #ccc;
27             padding: 8px;
28             text-align: left;
29           }
30           th {
31             background-color: #007bff;
32             color: white;
33           }
34         </style>
35       </head>
36       <body>
37         <h1>Resumen de Configuración de Red</h1>
```

- Representación tabular de la información de red, incluyendo:
  - Nombre del adaptador.
  - IP, máscara de subred, gateway, DNS.
  - Latencia media del servidor DNS.

Resumen de Configuración de Red	
Campo	Valor
Adaptador	Wi-Fi
Dirección IP	192.168.0.22
Máscara de subred	192.168.0.0/24 (mascara 255.255.255.0)
Puerta de enlace	192.168.0.1
Servidor DNS	8.8.8.8
Latencia media	14 ms

- Sección dinámica de saltos hacia el DNS, mostrando:
  - Número de salto.
  - Tres tiempos de respuesta.
  - Dirección IP del salto o mensaje de error.

#### Saltos hacia el DNS

#	Tiempo 1	Tiempo 2	Tiempo 3	IP o mensaje
1	<1 ms	<1 ms	<1 ms	192.168.0.1
2	23 ms	16 ms	73 ms	10.24.16.1
3	*	*	*	3 * * * Tiempo de espera agotado para esta solicitud.
4	*	*	*	4 * * * Tiempo de espera agotado para esta solicitud.
5	45 ms	15 ms	27 ms	212.166.147.22
6	15 ms	61 ms	16 ms	108.170.225.251
7	15 ms	16 ms	15 ms	142.250.239.25
8	14 ms	15 ms	15 ms	8.8.8.8

- Detección y manejo de errores en los saltos, como "Tiempo de espera agotado".

### 3. Consideraciones técnicas.

El código está modularizado en archivos:

- main.c
- adaptador.c / adaptador.h
- info.c / info.h
- util.c / util.h
- xml.c / xml.h
- xslt.c / xslt.h

El XML de entrada (salida.xml) fue validado previamente para asegurar que estaba bien formado y cumplía una estructura consistente. Se detecta un error a la hora de transformar el XML debido a la siguiente línea:

```
<Salto> 1  <1 ms  <1 ms  <1 ms  192.168.0.1 </Salto>
```

Se modifica por:

```
<Salto> 1  &lt;1 ms  &lt;1 ms  &lt;1 ms  192.168.0.1 </Salto>
```

Y se resuelve el error. Se procede a modificar xml.c del producto 4 para evitar los símbolos < y > antes y después de un 1 haciendo que el programa los sustituya por &lt;

La transformación puede ejecutarse:

- Automáticamente desde el programa C.
- Manualmente desde PowerShell, con comandos basados en System.Xml.Xsl.XslCompiledTransform.

```
PS C:\Users\Rubo\Documents\UOC entregar\Programacion de comandos personalizados\Producto_4\Producto_4> $xml = New-Object System.Xml.XmlDocument
>> $xml.Load("C:\Users\Rubo\Documents\UOC entregar\Programacion de comandos personalizados\Producto_4\Producto_4\salida.xml")
>> $xslt = New-Object System.Xml.Xsl.XslCompiledTransform
>> $xslt.Load("C:\Users\Rubo\Documents\UOC entregar\Programacion de comandos personalizados\Producto_4\Producto_4\transformacion.xslt")
>> $writer = New-Object System.IO.StreamWriter("C:\Users\Rubo\Documents\UOC entregar\Programacion de comandos personalizados\Producto_4\Producto_4\salida.html")
>> $xslt.Transform($xml, $null, $writer)
>> $writer.Close()
>>
PS C:\Users\Rubo\Documents\UOC entregar\Programacion de comandos personalizados\Producto_4\Producto_4>
```

## 4. Resultados obtenidos

Tras ejecutar el proceso de transformación, se obtiene un archivo HTML con el siguiente contenido:

- Encabezado con el título y estilos.
- Tabla 1: Resumen de configuración de red.
- Tabla 2: Lista detallada de los saltos hacia el servidor DNS.
- Diseño limpio y responsivo, válido para visualización inmediata en cualquier navegador moderno.

Se comprobó que el archivo generado:

- Es accesible desde el navegador.
- Presenta información clara y alineada.
- No contiene errores sintácticos ni visuales.

## 5. Archivos generados

- **salida.xml**: Archivo XML con la información de red (desde producto 4, con una pequeña modificación).
- **transformacion.xslt**: Hoja de estilo XSLT para generar el HTML.
- **salida.html**: Resultado de la transformación, listo para visualizar.

## 6. Recursos utilizados

- W3Schools. (n.d.). XSLT Tutorial. Recuperado de:  
[https://www.w3schools.com/xml/xsl\\_intro.asp](https://www.w3schools.com/xml/xsl_intro.asp)
- W3Schools. (n.d.). XSLT Tryit Editor v1.0. W3Schools.  
<https://www.w3schools.com/xml/tryxslt.asp?xmlfile=catalog&xsltfile=catalog>
- Microsoft Docs. (n.d.). *System.Xml.Xsl.XslCompiledTransform Class*. Recuperado de:  
<https://learn.microsoft.com/en-us/dotnet/api/system.xml.xsl.xslcompiledtransform>