

# **HITO 2 DEL 2DO TRIMESTRE LENGUAJE DE MARCAS**

Valentín Cortés Amadín

03/03/2024

# Contenidos

<b>1. Desarrollo del documento XML.....</b>	<b>2</b>
<b>2. Desarrollo de los documentos XSLT y transformación.....</b>	<b>3</b>
• A partir de un documento XML, mostrad la información de cada pedido en una tabla...3	
• A partir de un documento XML, mostrad la información de cada cliente en una tabla... 5	
• Crea una factura de un cliente que compra un producto a partir de la información de tu XML.....	7
<b>3. Validación del documento XML.....</b>	<b>9</b>

## 1. Desarrollo del documento XML

Siguiendo los criterios indicados en el enunciado se crea una plantilla que sirve como punto de partida. Esta contiene una marca principal que engloba todo el código, llamada <pedidos>. Dentro tenemos la marca <anio>, que se encuentra presente dos veces para englobar los años 2021 y 2022 tal y como pide el enunciado.

Dentro de cada marca <anio> encontramos tres marcas de <trimestre> y en cada una de estas la marca <pedido> (hay 10 por trimestre), que engloba todo lo referente a datos del cliente, su <direccion>, los <detalles\_pedido> y en <productos> se incluyen 3 marcas para cada <producto> (hay tres productos en cada pedido).

La estructura final tendrá la siguiente forma:

```
<pedidos>
  <anio year="">
    <trimestre num="">
      <pedido>
        <nombre></nombre>
        <apellidos></apellidos>
        <telefono></telefono>

        <direccion>
          <calle></calle>
          <ciudad></ciudad>
          <codigo_postal></codigo_postal>
          <provincia></provincia>
        </direccion>

        <correo></correo>
        <fecha_inclusion></fecha_inclusion>

        <detalles_pedido>
          <numero_pedido></numero_pedido>
          <fecha_compra></fecha_compra>
          <fecha_entrega></fecha_entrega>
          <total_factura></total_factura>

          <productos>
            <producto>
              <nombre></nombre>
              <referencia></referencia>
              <precio></precio>
              <unidades></unidades>
            </producto>
          </productos>
        </detalles_pedido>
      </pedido>
    </trimestre>
  </anio>
</pedidos>
```

## 2. Desarrollo de los documentos XSLT y transformación

Una vez rellenado el documento XML con todos los datos es momento de dar respuesta a las tres cuestiones que plantea el hito en la fase 2:

- A partir de un documento XML, mostrad la información de cada pedido en una tabla.

Organizaremos los datos de cada pedido en una tabla donde las columnas principales serán Cliente, Teléfono, Dirección, Correo, Fecha de inclusión, Número de pedido, Fecha de compra, Fecha de entrega y Total de la factura.

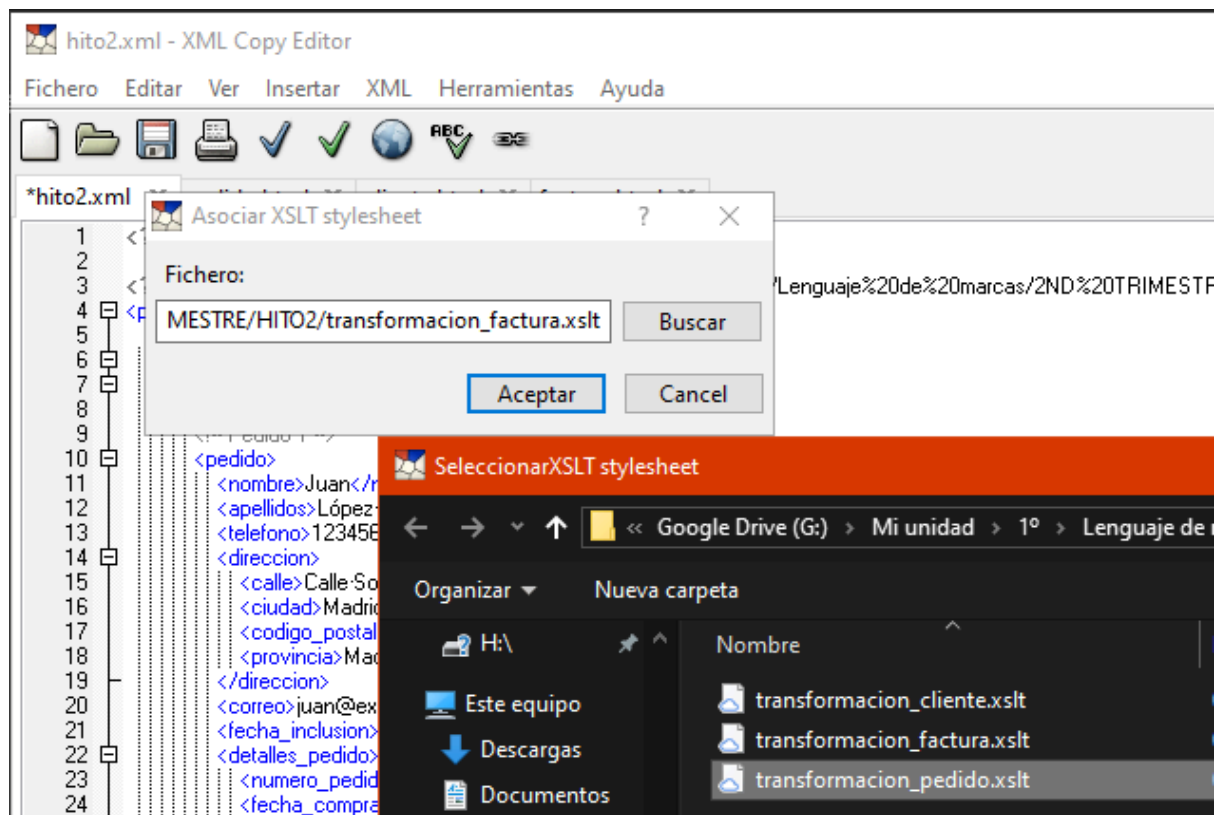
```
<body>
  <h2>Información de los Pedidos</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Cliente</th>
      <th>Teléfono</th>
      <th>Dirección</th>
      <th>Correo</th>
      <th>Fecha de Inclusión</th>
      <th>Número de Pedido</th>
      <th>Fecha de Compra</th>
      <th>Fecha de Entrega</th>
      <th>Total de la Factura</th>
    </tr>
```

Para seleccionar los elementos del documento XML usaremos la línea `<xsl:for-each>` acompañado de una sentencia SELECT:

```
<xsl:for-each select="pedidos/año/trimestre/pedido">
  <tr>
    <td><xsl:value-of select="nombre"/> <xsl:value-of select="apellido"/>
    <td><xsl:value-of select="telefono"/></td>
    <td><xsl:value-of select="concat(direccion/calle, ', ', direccion/calle2)/>
    <td><xsl:value-of select="correo"/></td>
    <td><xsl:value-of select="fecha_inclusion"/></td>
    <td><xsl:value-of select="detalles_pedido/numero_pedido"/></td>
    <td><xsl:value-of select="detalles_pedido/fecha_compra"/></td>
    <td><xsl:value-of select="detalles_pedido/fecha_entrega"/></td>
    <td><xsl:value-of select="detalles_pedido/total_factura"/></td>
  </tr>
</xsl:for-each>
```

Llamamos a este documento **transformacion\_pedido.xslt**.

Por último, con la herramienta XML Copy Editor, vinculamos el documento XML con este archivo XSLT para hacer una transformación y obtenemos como resultado un documento HTML con los datos de cada pedido organizados en una tabla.



Y este es el resultado:

The screenshot shows a web browser with three tabs: 'pedido.html', 'cliente.html', and 'factura.html'. The active tab is 'pedido.html', which displays a table titled 'Información de los Pedidos'. The table contains the following data:

Cliente	Teléfono	Dirección	Correo	Fecha de Inclusión	Número de Pedido	Fecha de Compra	Fecha de Entrega	Total de la Factura
JuanLópez García	123456789	Calle Sol 123, Madrid, 28001, Madrid	juan@example.com	2021-01-05	001	2021-01-05	2021-01-10	75.00
AnaGutiérrez Martínez	987654321	Avenida Libertad 456, Sevilla, 41001, Sevilla	ana@example.com	2021-02-12	002	2021-02-12	2021-02-17	150.00
CarlosRuiz Sánchez	654123987	Calle Mayor 789, Valencia, 46001, Valencia	carlos@example.com	2021-03-25	003	2021-03-25	2021-03-30	90.00
LauraMartínez Fernández	789456123	Calle Gran Vía 321, Barcelona, 08001, Barcelona	laura@example.com	2021-04-08	004	2021-04-08	2021-04-13	120.00
MartaGómez Rodríguez	741258963	Calle Primavera 456, Madrid, 28002, Madrid	marta@example.com	2021-05-20	005	2021-05-20	2021-05-25	200.00
DavidFernández López	852369741	Calle Mayor 789, Valencia, 46002, Valencia	david@example.com	2021-06-15	006	2021-06-15	2021-06-20	180.00

- A partir de un documento XML, mostrad la información de cada cliente en una tabla.

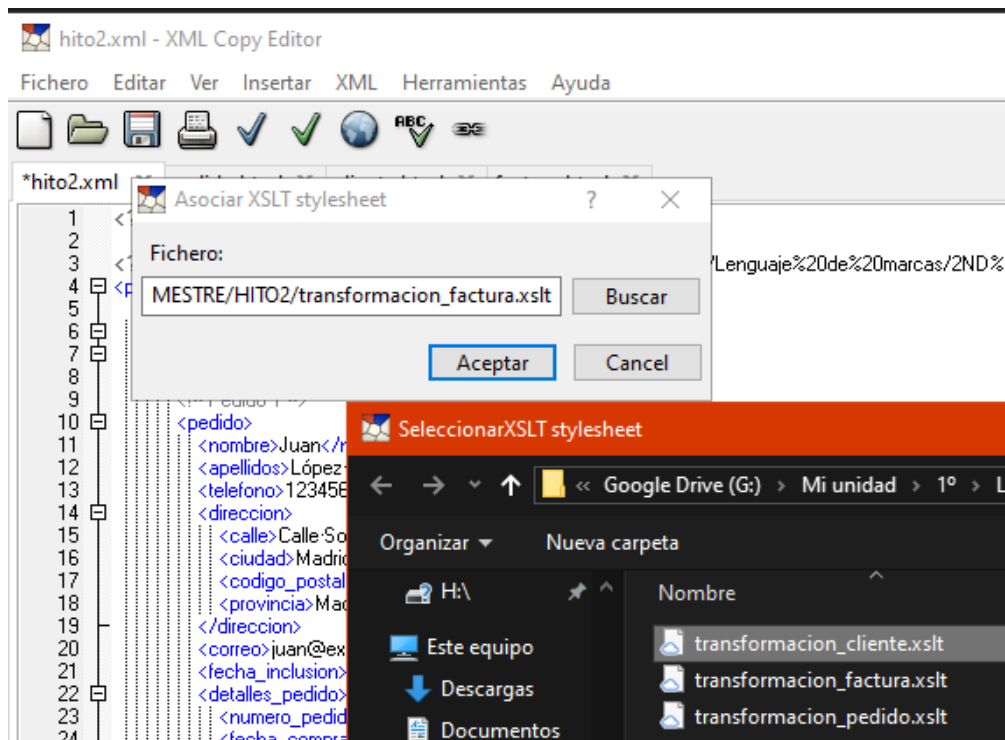
Repetimos el procedimiento documentado en el apartado anterior, pero en esta ocasión dentro del archivo XSLT la tabla que definimos contendrá información de cada cliente:

```
<xsl:template match="/">
  <html>
    <body>
      <h2>Información de los Cliente
      <table border="1">
        <tr bgcolor="#ff4500">
          <th>Cliente</th>
          <th>Teléfono</th>
          <th>Dirección</th>
          <th>Correo</th>
        </tr>
```

Hacemos uso de la marca `<xsl:for-each>` acompañado de un SELECT para seleccionar todos los datos del documento XML que queremos introducir en dicha tabla:

```
<xsl:for-each select="pedidos/anio/trimestre/pedido">
  <tr>
    <td><xsl:value-of select="nombre"/> <xsl:value-of select="apellidos"/></td>
    <td><xsl:value-of select="telefono"/></td>
    <td><xsl:value-of select="concat(direccion/calle, ', ', direccion/ciudad, ', ',
    <td><xsl:value-of select="correo"/></td>
  </tr>
</xsl:for-each>
```

Cerramos el XSLT con las marcas correspondientes y hacemos la transformación en XML Copy Editor:



El resultado es el siguiente:

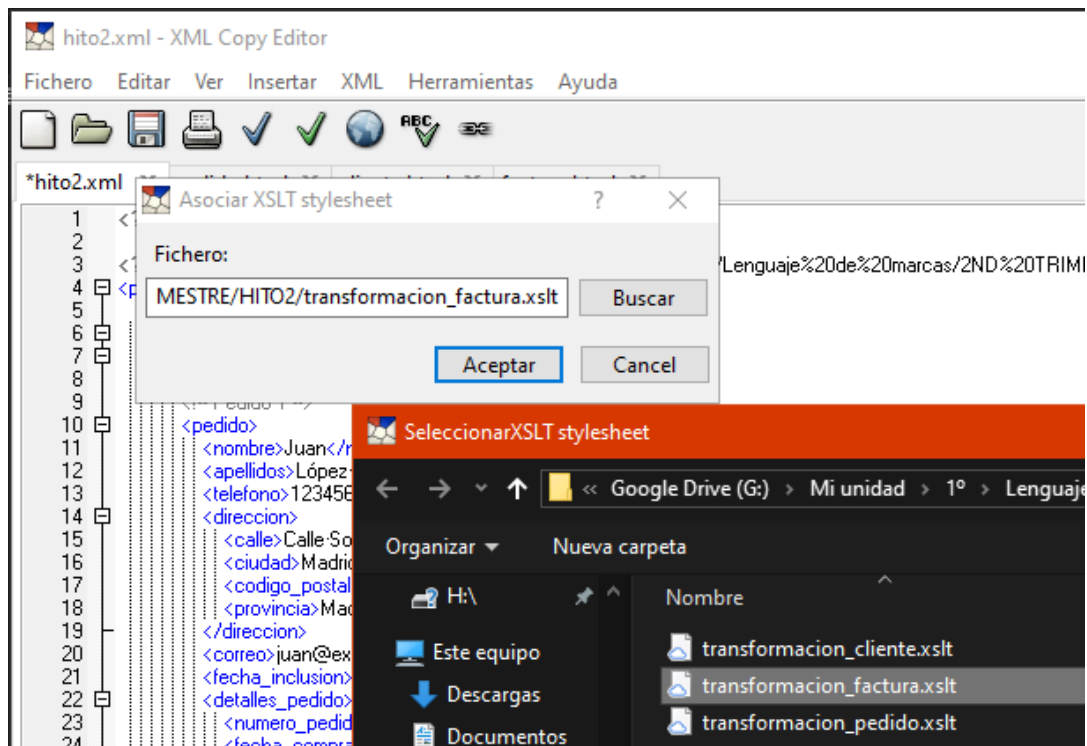
Información de los Clientes			
Cliente	Teléfono	Dirección	Correo
JuanLópez García	123456789	Calle Sol 123, Madrid, 28001, Madrid	juan@example.com
AnaGutiérrez Martínez	987654321	Avenida Libertad 456, Sevilla, 41001, Sevilla	ana@example.com
CarlosRuiz Sánchez	654123987	Calle Mayor 789, Valencia, 46001, Valencia	carlos@example.com
LauraMartínez Fernández	789456123	Calle Gran Vía 321, Barcelona, 08001, Barcelona	laura@example.com
MartaGómez Rodríguez	741258963	Calle Primavera 456, Madrid, 28002, Madrid	marta@example.com
DavidFernández López	852369741	Calle Mayor 789, Valencia, 46002, Valencia	david@example.com
SofíaJiménez Pérez	369852147	Avenida del Parque 789, Barcelona, 08002, Barcelona	sofia@example.com
ManuelRodríguez Pérez	987123456	Calle Real 123, Sevilla, 41002, Sevilla	manuel@example.com
LuisaGonzález Ruiz	632741985	Calle Mayor 456, Madrid, 28003, Madrid	luisa@example.com
JavierRuiz García	654987321	Avenida Libertad 789, Valencia, 46003, Valencia	javier@example.com
AnaGarcía Pérez	123456789	Calle Real 123, Madrid, 28001, Madrid	ana@example.com
PabloAlvarez Martínez	654123987	Calle Valencia 456, Barcelona, 08003, Barcelona	pablo@example.com
AnaRuiz Sánchez	741852963	Avenida del Sol 789, Sevilla, 41003, Sevilla	ana@example.com

- Crea una factura de un cliente que compra un producto a partir de la información de tu XML.

Repetimos por última vez el procedimiento de las dos cuestiones anteriores para crear una entrada para cada cliente que contenga sus datos y una pequeña tabla con el nombre del producto, el precio, las unidades y abajo una línea que indique el total de la factura.

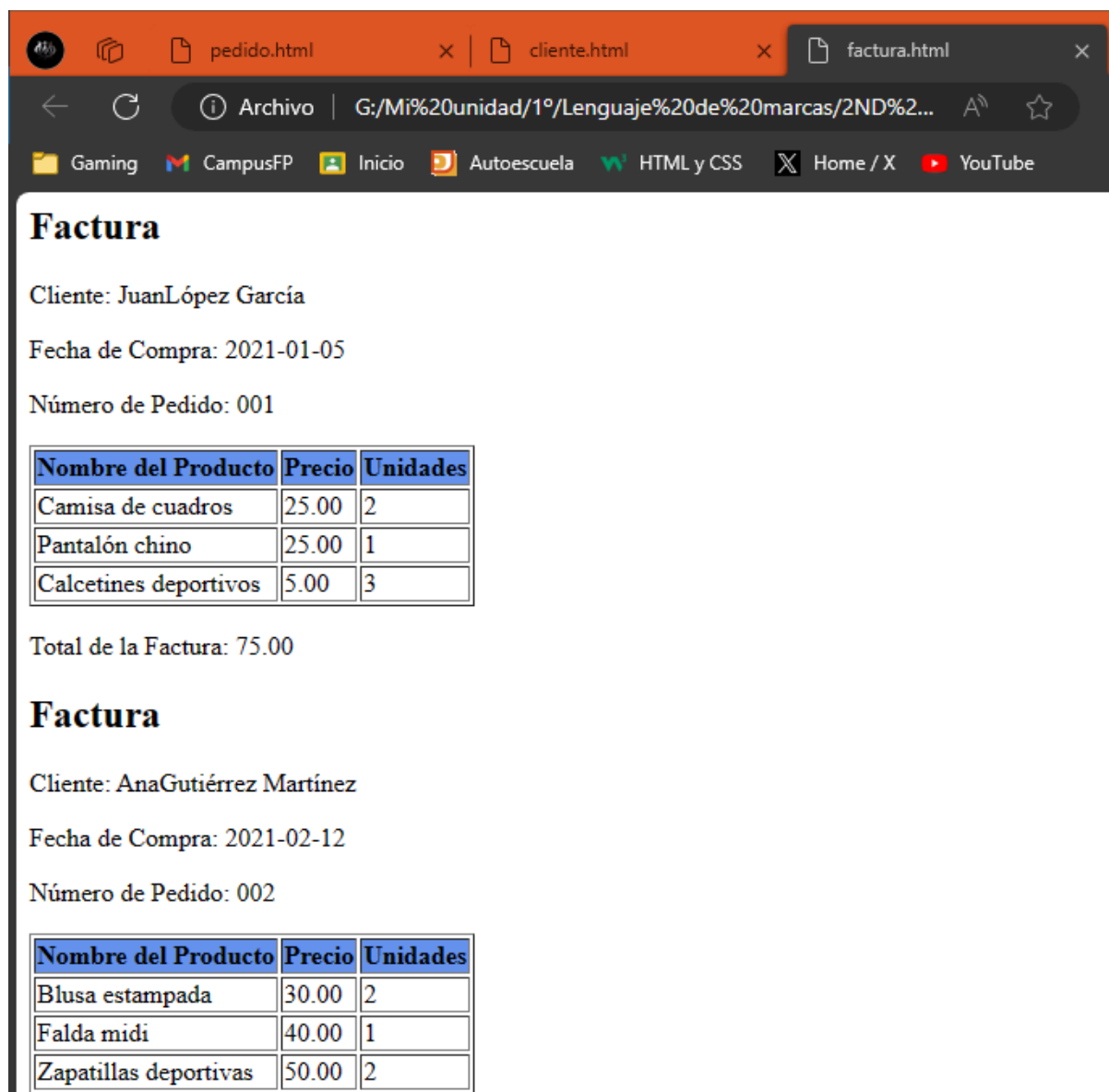
```
<xsl:template match="pedido">
  <html>
    <body>
      <h2>Factura</h2>
      <p>Cliente: <xsl:value-of select="nombre"/> <xsl:value-of select="apellidos"/></p>
      <p>Fecha de Compra: <xsl:value-of select="detalles_pedido/fecha_compra"/></p>
      <p>Número de Pedido: <xsl:value-of select="detalles_pedido/numero_pedido"/></p>
      <table border="1">
        <tr bgcolor="#6495ed">
          <th>Nombre del Producto</th>
          <th>Precio</th>
          <th>Unidades</th>
        </tr>
        <xsl:for-each select="detalles_pedido/productos/producto">
          <tr>
            <td><xsl:value-of select="nombre"/></td>
            <td><xsl:value-of select="precio"/></td>
            <td><xsl:value-of select="unidades"/></td>
          </tr>
        </xsl:for-each>
      </table>
      <p>Total de la Factura: <xsl:value-of select="detalles_pedido/total_factura"/></p>
    </body>
  </html>
</xsl:template>
```

Hacemos la transformación en XML Copy Editor:





Y el resultado es el siguiente:



**Factura**

Cliente: JuanLópez García

Fecha de Compra: 2021-01-05

Número de Pedido: 001

Nombre del Producto	Precio	Unidades
Camisa de cuadros	25.00	2
Pantalón chino	25.00	1
Calcetines deportivos	5.00	3

Total de la Factura: 75.00

**Factura**

Cliente: AnaGutiérrez Martínez

Fecha de Compra: 2021-02-12

Número de Pedido: 002

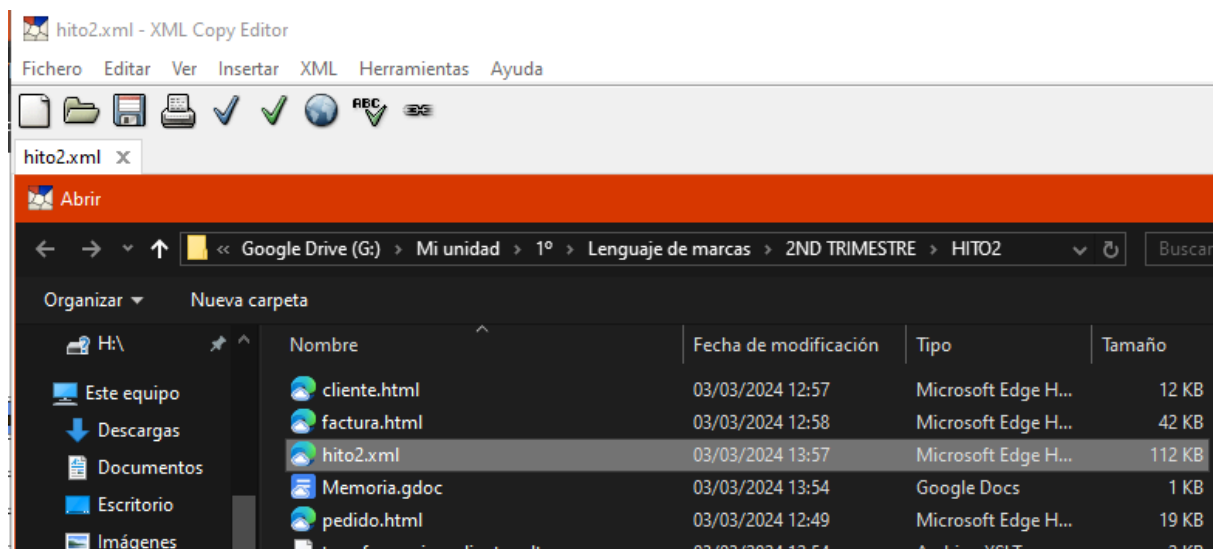
Nombre del Producto	Precio	Unidades
Blusa estampada	30.00	2
Falda midi	40.00	1
Zapatillas deportivas	50.00	2

### 3. Validación del documento XML

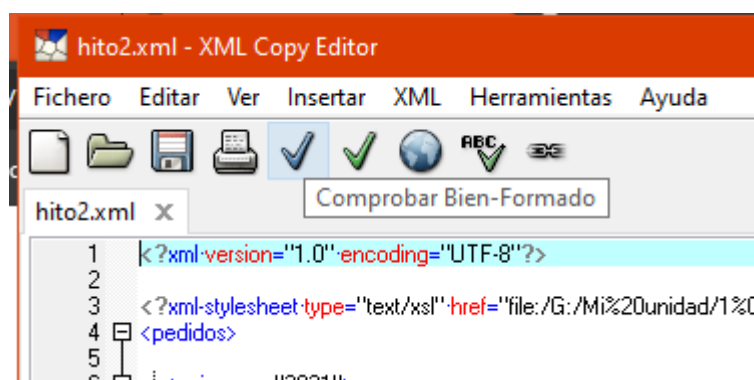
La herramienta XML Copy Editor incluye un botón con el cual podemos validar los documentos XML. Un documento XML validado es un documento que está “bien formado” (no contiene errores de sintaxis) y está hecho conforme a una definición de tipo de documento, ya sea XML schema o DTD.

Par validar un documento en XML Copy Editor hacemos lo siguiente:

1. Abrimos nuestro documento XML

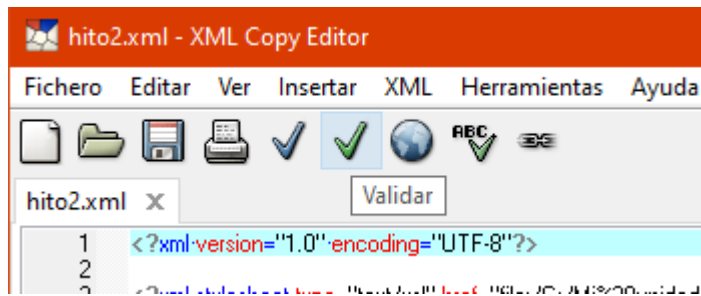


2. Hacemos una primera comprobación de que está bien formado haciendo click en el check azul:



Abajo aparecerá un panel de información indicando que el documento está bien formado.

3. Hacemos la validación final haciendo click en el check verde:



El documento está validado.

Como paso final unimos los tres documentos html mediante un cuarto documento que actuará como index y permitirá al usuario alternar cómodamente entre las tres páginas con su respectiva información: