

Big Data Aplicado

Profesor/a: José Manuel González Rodríguez

ACTIVIDAD EVALUABLE 1.3 (TECNOLOGÍA XML)
28/11/2024

Índice

1.- Introducción	4
1.1.- Objetivo de la Actividad	4
1.2.- Descripción general del proyecto	4
1.3.- Menú de la Pizzería Divan	5
1.4.- Tabla de Ingredientes	6
1.5.- Tabla de Ingredientes Extra	7
1.6.- Tabla de Alérgenos (Reglamento UE 1169/2011)	8
2.- Ejercicio 1: Creación del XML	8
2.1.- Descripción del ejercicio	8
2.2.- Estructura del archivo XML	8
2.3.- Detalles de los elementos y atributos utilizados	9
2.4.- Explicación del Archivo XML	10
3.- Ejercicio 2: Creación del DTD	22
3.1.- Descripción del ejercicio	22
3.2.- Explicación del DTD	22
3.3.- Definición de elementos y atributos en el DTD	23
3.4.- Explicación del Archivo DTD	24
4.- Ejercicio 3: Creación del XML Schema (XSD)	26
4.1.- Descripción del ejercicio	26
4.2.- Explicación del XSD	27
4.3.- Validación de datos en el XSD	27
4.4.- Explicación del archivo XSD	28
5.- Ejercicio 4: Transformación con XSLT	34
5.1. Descripción del ejercicio	34
5.2.- Primer archivo XSLT: Conteo de productos por tipo	34
5.3.- Resultados del primer archivo	35
5.4.- Segundo archivo XSLT: Generación de una tabla HTML con el menú de pizzas	36
5.5.- Resultado del segundo	38
6. Ejercicio 5: Consultas con XQuery	39
6.1.- Descripción del ejercicio	39
6.2.- Primer archivo XQuery: Productos aptos para celíacos	40
6.3.- Instalación de las aplicaciones	41

6.4.- Resultado del primer archivo	42
6.5.- Segundo archivo XQuery: Ordenar pizzas por número de ingredientes	43
7. Conclusiones	44
7.1. Reflexiones sobre el proceso	44
7.2. Desafíos enfrentados y soluciones implementadas	44
7.3. Importancia del trabajo	45
8. Bibliografía	45
8.1. Fuentes utilizadas para la creación de los archivos XML, DTD, XSD, XSLT y XQuery	45
8.2. Fuentes adicionales consultadas	46
9.- Mapa Mental	47
10.- Anexos	48

1.- Introducción

1.1.- Objetivo de la Actividad

El objetivo principal de esta actividad es aplicar y desarrollar habilidades en el manejo de tecnologías relacionadas con la manipulación, validación, transformación y consulta de datos en formatos estructurados, específicamente utilizando XML, DTD, XSD, XSLT y XQuery. A través de esta actividad, se busca aprender a crear y gestionar un sistema de información basado en XML que contenga la estructura de un menú de pizzería. Además, se debe validar este XML mediante DTD y XSD, realizar transformaciones mediante XSLT y consultar datos utilizando XQuery. De esta manera, el ejercicio permite comprender cómo estas tecnologías se utilizan en el desarrollo de aplicaciones web y en la gestión de grandes volúmenes de datos (Big Data), así como su aplicación práctica en un contexto del mundo real.

1.2.- Descripción general del proyecto

En este proyecto, se ha diseñado un sistema de almacenamiento de información para una pizzería que pretende crear una página web para mostrar su menú y otros datos relevantes. El sistema tiene como base un archivo XML que contiene todos los productos ofrecidos por la pizzería, incluyendo pizzas, entrantes y postres, así como información adicional como ingredientes, alérgenos y precios.

La pizzería desea almacenar datos clave, como:

- Información sobre las sucursales, incluyendo dirección, localidad, horarios de apertura y teléfono.
- Detalles sobre los ingredientes de cada producto, con sus precios en caso de ser añadidos como extras.
- Alérgenos de acuerdo con la normativa europea UE 1169/2011.
- Tres categorías de productos: entrantes, pizzas y postres.
- Tamaños de pizzas (30cm, 40cm, 50cm) y sus precios correspondientes.

Este proyecto ha sido dividido en varios ejercicios que incluyen la creación y validación de documentos XML, la transformación de datos usando XSLT para generar una vista web, y la ejecución de consultas con XQuery para filtrar y ordenar la información de manera eficiente. Los ejercicios se han desarrollado siguiendo un enfoque de Big Data, donde se aprende a manejar grandes volúmenes de datos en formato

estructurado, y también se prepara el terreno para la futura expansión de la pizzería a nuevas ubicaciones.

Además de la creación de los archivos XML, DTD, XSD, XSLT y XQuery, el proyecto también incluye una validación rigurosa de los datos, asegurando que el formato, los valores y las relaciones entre los distintos elementos sean correctos y consistentes. Esta actividad demuestra cómo las herramientas de procesamiento de datos estructurados pueden facilitar la creación de sistemas complejos de información y cómo las tecnologías de Big Data pueden aplicarse en situaciones cotidianas del negocio.

1.3.- Menú de la Pizzería Divan

Categoría	Producto	Descripción	Alérgenos	Precio
Entrantes	Nachos con queso	Deliciosos nachos gratinados con queso mozzarella de primera calidad y salsa guacamole.	Huevo, lácteos, soja	3,50 €
Pizzas	Queso	Base de tomate y queso mozzarella.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 12 €, 50 cm: 16 €
Pizzas	Jamón y queso	Base de tomate y queso mozzarella con jamón york.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 12 €, 50 cm: 16 €
Pizzas	Vegetal	Base de tomate y queso mozzarella, calabacín, cebolla, tomate y pimientos.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Pizzas	Cuatro quesos	Base de tomate y queso mozzarella, cheddar, gouda y parmesano.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Pizzas	Barbacoa	Base de tomate y queso mozzarella, carne de ternera picada, bacon, cebolla y salsa barbacoa.	Lácteos, gluten, soja, mostaza	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Pizzas	Romana	Base de tomate y queso mozzarella, jamón york, aceitunas negras, champiñones y orégano.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Pizzas	Hawaiana	Base de tomate y queso mozzarella, jamón york, piña y champiñones.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Pizzas	Mexicana	Base de tomate y queso mozzarella, pepperoni, carne de ternera picada, cebolla, jalapeños y salsa picante.	Lácteos, gluten, sulfitos, soja, mostaza	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Pizzas	Atún	Base de tomate y queso mozzarella, atún, tomate natural, aceitunas negras, pimientos, cebolla y orégano.	Lácteos, gluten, pescado	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Pizzas	Carbonara	Base de nata, queso mozzarella, bacon, cebolla y champiñones.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Pizzas	Pepperoni	Base de tomate, queso mozzarella y pepperoni.	Lácteos, gluten, soja, mostaza	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Novedades	Pollo y bacon	Base de tomate, queso mozzarella, pollo, bacon y queso cheddar.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Novedades	Díjola	Base de tomate, queso mozzarella, chorizo picante y queso (Gorgonzola o queso de cabra).	Lácteos, gluten, mostaza, soja	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Novedades	Pizza Shawarma	Base de tomate, queso mozzarella, cebolla, tomate natural, carne de pollo kebab con salsa de yogur y/o picante.	Lácteos, contiene gluten	30 cm: 9 €, 40 cm: 13 €, 50 cm: 17 €
Postres	Mousse de limón	Deliciosa mousse adornada con ralladura de limón natural.	Ninguno	3 €
Postres	Mousse de chocolate	Suave mousse de chocolate fondant (50% cacao) con virutas de chocolate negro.	Ninguno	3 €
Postres	Capricho blanco	Crema de chocolate blanco con coulis de frutos rojos.	Ninguno	3,50 €
Postres	Mini cheesecake con mango	Exquisita tarta de queso con base de galleta molida y coulis de mango natural.	Ninguno	3,50 €
Postres	Cheesecake con dulce	Riquísimo pastel de queso con base de galleta molida y dulce de leche.	Ninguno	3,50 €
Postres	Conos helados	Fresa, vainilla, chocolate y nata.	Ninguno	1 €

El contenido presenta un menú detallado de una pizzería, incluyendo entrantes, una variedad de pizzas en diferentes tamaños, novedades y postres, con descripciones, alérgenos y precios para cada producto.

1.4.- Tabla de Ingredientes

ID	Nombre del Ingrediente
I1	Tomate
I2	Queso Mozarela
I3	Jamón York
I4	Calabacín
I5	Cebolla
I6	Pimientos
I7	Cheddar
I8	Gouda
I9	Parmesano
I10	Carne de ternera picada
I11	Bacón
I12	Aceitunas negras
I13	Champiñones
I14	Piña
I15	Pepperoni
I16	Atún
I17	Nata
I18	Yogurt
I19	Jalapeños
I20	Salsa barbacoa
I21	Chorizo picante
I22	Queso Gorgonzola
I23	Queso de rulo de cabra
I24	Carne de pollo kebab
I25	Salsa de yogurt

Este contenido es una tabla de ingredientes con identificadores únicos (ID) que lista 25 ingredientes diferentes utilizados en las recetas de la pizzería, incluyendo verduras, carnes, quesos, salsas y otros componentes culinarios.

1.5.- Tabla de Ingredientes Extra

Nombre del Ingrediente Extra	Precio Extra (€)
Aceitunas	0,5
Champiñones	0,5
Piña	0,5
Pepperoni	0,5
Queso Mozarela	0,5
Cebolla	0,5
Jalapeños	0,5
Bacón	0,5
Tomate	0,5
Calabacín	0,5
Queso Cheddar	0,5
Queso Gouda	0,5
Queso Parmesano	0,5
Carne de ternera picada	0,5
Atún	0,5
Pimientos	0,5
Aceitunas negras	0,5
Yogurt	0,5
Salsa barbacoa	0,5
Carne de pollo kebab	0,5
Salsa de yogurt	0,5
Chorizo picante	0,5
Queso Gorgonzola	0,5
Queso de rulo de cabra	0,5

Esta tabla muestra una lista de ingredientes extra disponibles para personalizar las pizzas, con cada ingrediente adicional costando 0,50€, permitiendo a los clientes modificar sus pedidos según sus preferencias.

1.6.- Tabla de Alérgenos (Reglamento UE 1169/2011)

ID del Alérgeno	Nombre del Alérgeno
A01	Gluten
A02	Crustáceos
A03	Huevo
A04	Pescado
A05	Cacahuetes
A06	Soja
A07	Lácteos
A08	Frutos de Cáscara
A09	Apio
A10	Mostaza
A11	Granos de Sésamo
A12	Dióxido de Azufre y Sulfitos
A13	Altramuces
A14	Moluscos

Esta tabla presenta una lista de 14 alérgenos comunes en alimentos, cada uno con un identificador único (ID) y su nombre correspondiente. Los alérgenos incluyen desde gluten y lácteos hasta moluscos y altramuces, abarcando los principales alérgenos que deben ser declarados en productos alimenticios según las regulaciones de seguridad alimentaria.

2.- Ejercicio 1: Creación del XML

2.1.- Descripción del ejercicio

En este ejercicio se nos ha encargado la creación de un archivo XML que contenga toda la información necesaria para representar el menú de una pizzería. Este archivo XML debe almacenar los productos, las sucursales de la pizzería, los ingredientes, los alérgenos y otros datos relevantes de manera estructurada. El objetivo es crear un archivo bien organizado que se pueda utilizar más tarde para alimentar una futura página web o sistema de gestión para la pizzería.

2.2.- Estructura del archivo XML

El archivo XML se organiza de forma jerárquica. La estructura debe permitir almacenar los siguientes elementos clave:

- **Pizzería:** Elemento principal que contendrá la información general de la pizzería.
- **Sucursales:** Información de las diferentes ubicaciones de la pizzería.
- **Categorías de productos:** Productos agrupados por categorías, como entrantes, pizzas y postres.
- **Productos:** Detalles sobre cada producto: nombre, precio, ingredientes, alérgenos, tamaños (en el caso de las pizzas).
- **Horarios de apertura:** Detalle de los días de la semana con su horario de apertura y cierre.

A continuación, se muestra la jerarquía básica de elementos:

- <pizzeria>
 - <sucursales>
 - <sucursal> (con detalles como dirección, teléfono, horarios)
 - <categorias>
 - <categoria> (con productos como entrantes, pizzas y postres)
 - <producto> (con atributos como nombre, precio, ingredientes, alérgenos, tamaños, etc.)
 -

2.3.- Detalles de los elementos y atributos utilizados

A continuación, se describen los elementos y atributos clave utilizados en el archivo XML:

- <pizzeria>: Elemento raíz que agrupa toda la información de la pizzería.
 - <sucursales>: Contenedor de las diferentes sucursales de la pizzería.
 - <sucursal>: Elemento que describe una sucursal específica. Tiene los siguientes atributos:
 - **id**: Identificador único de la sucursal.
 - <direccion>: Dirección de la sucursal.
 - <localidad>: Localidad de la sucursal.
 - <telefono>: Número de teléfono de la sucursal.
 - <horarios>: Contenedor que agrupa los horarios de apertura de cada día.
 - <dia>: Elemento que especifica el día de la semana con su horario de apertura y cierre.
 - Atributos: **nombre, apertura, cierre**.

- <categorías>: Contenedor que agrupa las categorías de productos.
 - < categoría >: Cada tipo de producto (entrantes, pizzas, postres). Tiene el atributo `nombre` para especificar el tipo de categoría.
 - < producto >: Elemento que describe un producto específico de la pizzería. Tiene los siguientes atributos:
 - `id`: Identificador único del producto.
 - < nombre >: Nombre del producto.
 - < precio >: Precio del producto.
 - < descripción >: Descripción del producto.
 - < ingredientes >: Listado de ingredientes del producto (especialmente importante para las pizzas).
 - < ingrediente >: Cada ingrediente listado por nombre.
 - < alérgenos >: Listado de alérgenos presentes en el producto según el Reglamento UE 1169/2011.
 - < alérgeno >: Cada alérgeno listado por nombre.
 - < tamaños >: Sólo para las pizzas, contiene los diferentes tamaños y sus precios.
 - < tamaño >: Elemento que describe un tamaño de pizza con su diámetro y precio.

2.4.- Explicación del Archivo XML

A continuación, se muestra la estructura y contenido del archivo XML con datos representativos.

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

Esta línea declara un documento XML, especificando la versión 1.0 y la codificación de caracteres UTF-8.

```
<!-- Cambiar esto para usar el fichero pizzeria_divan -->
<!-- <!DOCTYPE pizzeria SYSTEM "pizzeria_divan.dtd">
<pizzeria> -->
```

Este código comenta una declaración DOCTYPE y el inicio de un elemento raíz 'pizzeria', sugiriendo un cambio para usar un archivo DTD específico llamado "pizzeria_divan.dtd".

```
<pizzeria xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:noNamespaceSchemaLocation="pizzeria_divan.xsd">
<?xmlstylesheet type="text/xsl" href="pizzeria_divan1.xsl"?>
```

Este código define el elemento raíz "pizzeria" de un documento XML, especificando un esquema XSD para validación y una hoja de estilo XSL para su presentación

```
<!-- Sección de sucursales -->
<sucursales>
    <sucursal id="S1">
        <direccion>Calle Pedro Barbabosa Parreño, 1</direccion>
        <localidad>Cortegana</localidad>
        <telefono>959879545</telefono>
        <horarios>
            <horario dia="lunes">Cerrado</horario>
            <horario dia="martes">Cerrado</horario>
            <horario dia="miércoles">Cerrado</horario>
            <horario dia="jueves">20:30-00:00</horario>
            <horario dia="viernes">20:30-00:00</horario>
            <horario dia="sábado">20:30-00:00</horario>
            <horario dia="domingo">20:30-00:00</horario>
        </horarios>
    </sucursal>
    <!-- Otras sucursales -->
</sucursales>
```

Este código XML define la información detallada de una sucursal de pizzería, incluyendo su dirección, localidad, teléfono y horarios de apertura para cada día de la semana.

```
<!-- Sección de ingredientes -->
<ingredientes>
    <ingrediente id="I1" nombre="Tomate" />
    <ingrediente id="I2" nombre="Queso Mozarella" />
    <ingrediente id="I3" nombre="Jamón York" />
    <ingrediente id="I4" nombre="Calabacín" />
    <ingrediente id="I5" nombre="Cebolla" />
    <ingrediente id="I6" nombre="Pimientos" />
    <ingrediente id="I7" nombre="Cheddar" />
    <ingrediente id="I8" nombre="Gouda" />
    <ingrediente id="I9" nombre="Parmesano" />
    <ingrediente id="I10" nombre="Carne de ternera picada" />
    <ingrediente id="I11" nombre="Bacón" />
    <ingrediente id="I12" nombre="Aceitunas negras" />
    <ingrediente id="I13" nombre="Champiñones" />
    <ingrediente id="I14" nombre="Piña" />
    <ingrediente id="I15" nombre="Pepperoni" />
    <ingrediente id="I16" nombre="Atún" />
    <ingrediente id="I17" nombre="Nata" />
    <ingrediente id="I18" nombre="Yogurt" />
    <ingrediente id="I19" nombre="Jalapeños" />
    <ingrediente id="I20" nombre="Salsa barbacoa" />
    <ingrediente id="I21" nombre="Chorizo picante" />
    <ingrediente id="I22" nombre="Queso Gorgonzola" />
    <ingrediente id="I23" nombre="Queso de rulo de cabra" />
    <ingrediente id="I24" nombre="Carne de pollo kebab" />
    <ingrediente id="I25" nombre="Salsa de yogurt" />
</ingredientes>
```

Este código XML define una lista completa de ingredientes para una pizzería, asignando a cada uno un identificador único (ID) y un nombre, abarcando una amplia variedad de ingredientes como quesos, carnes, verduras y salsas.

```
<!-- Sección de ingredientes extra -->
<ingredientes_extra>
    <ingrediente nombre="Aceitunas" precio_extra="0.50" />
    <ingrediente nombre="Champiñones" precio_extra="0.50" />
    <ingrediente nombre="Piña" precio_extra="0.50" />
    <ingrediente nombre="Pepperoni" precio_extra="0.50" />
    <ingrediente nombre="Queso Mozarela" precio_extra="0.50" />
    <ingrediente nombre="Cebolla" precio_extra="0.50" />
    <ingrediente nombre="Jalapeños" precio_extra="0.50" />
    <ingrediente nombre="Bacón" precio_extra="0.50" />
    <ingrediente nombre="Tomate" precio_extra="0.50" />
    <ingrediente nombre="Calabacín" precio_extra="0.50" />
    <ingrediente nombre="Queso Cheddar" precio_extra="0.50" />
    <ingrediente nombre="Queso Gouda" precio_extra="0.50" />
    <ingrediente nombre="Queso Parmesano" precio_extra="0.50" />
    <ingrediente nombre="Carne de ternera picada" precio_extra="0.50" />
    <ingrediente nombre="Atún" precio_extra="0.50" />
    <ingrediente nombre="Pimientos" precio_extra="0.50" />
    <ingrediente nombre="Aceitunas negras" precio_extra="0.50" />
    <ingrediente nombre="Yogurt" precio_extra="0.50" />
    <ingrediente nombre="Salsa barbacoa" precio_extra="0.50" />
    <ingrediente nombre="Carne de pollo kebab" precio_extra="0.50" />
    <ingrediente nombre="Salsa de yogurt" precio_extra="0.50" />
    <ingrediente nombre="Chorizo picante" precio_extra="0.50" />
    <ingrediente nombre="Queso Gorgonzola" precio_extra="0.50" />
    <ingrediente nombre="Queso de rulo de cabra" precio_extra="0.50" />
</ingredientes_extra>
```

Este código XML define una lista de ingredientes extra disponibles en una pizzería, especificando el nombre de cada ingrediente y un precio adicional fijo de 0.50 para cada uno.

```
<!-- Sección de alérgenos según el Reglamento UE 1169/2011 -->
<alergenos>
    <!-- Cada alérgeno tiene un id único y un nombre -->
    <alergeno id="A01" nombre="Gluten"/>
    <alergeno id="A02" nombre="Crustáceos"/>
    <alergeno id="A03" nombre="Huevo"/>
    <alergeno id="A04" nombre="Pescado"/>
    <alergeno id="A05" nombre="Cacahuetes"/>
    <alergeno id="A06" nombre="Soja"/>
    <alergeno id="A07" nombre="Lácteos"/>
    <alergeno id="A08" nombre="Frutos de Cáscara"/>
    <alergeno id="A09" nombre="APIO"/>
    <alergeno id="A10" nombre="Mostaza"/>
    <alergeno id="A11" nombre="Granos de Sésamo"/>
    <alergeno id="A12" nombre="Dióxido de Azufre y Sulfitos"/>
    <alergeno id="A13" nombre="Altramuces"/>
    <alergeno id="A14" nombre="Moluscos"/>
    <!-- Se pueden añadir más alérgenos aquí -->
</alergenos>
```

Este código XML define una lista de alérgenos según el Reglamento UE 1169/2011, asignando a cada uno un identificador único y un nombre.

```
<!-- Sección de categorías de productos -->
<categorias>
    <!-- Categoría de entrantes -->
    <categoria nombre="entrantes">
        <producto>
            <nombre>Nachos con queso</nombre>
            <alergenos>
                <alergeno nombre="Huevo"/>
                <alergeno nombre="Lácteos"/>
                <alergeno nombre="Soja"/>
            </alergenos>
            <descripcion>Deliciosos nachos gratinados con queso mozzarella de primera calidad y salsa guacamole</descripcion>
            <tamanos>
                <tamano tamao="Pequeño" precio="3.50"/>
            </tamanos>
        </producto>
        <!-- Otros productos de entrantes -->
    </categoria>
```

Este código XML define la categoría de entrantes en el menú de una pizzería, detallando un producto específico (Nachos con queso) con su nombre, alérgenos, descripción y precio.

```
<!-- Categoría de pizzas -->
<categoria nombre="pizzas">
    <producto>
        <nombre>Queso</nombre>
        <alergenos>
            <alergeno nombre="Lácteos"/>
            <alergeno nombre="Gluten"/>
        </alergenos>
        <descripcion>Base de tomate y queso mozarella</descripcion>
        <tamanos>
            <tamano tamao="30cm" precio="9.00"/>
            <tamano tamao="40cm" precio="12.00"/>
            <tamano tamao="50cm" precio="16.00"/>
        </tamanos>
    </producto>
```

Este código XML define un producto específico (Pizza de Queso) dentro de la categoría de pizzas, incluyendo sus alérgenos, descripción y precios para diferentes tamaños.

```
<producto>
    <nombre>Jamón y queso</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripcion>Base de tomate y queso mozarella y jamón york</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="12.00"/>
        <tamano tamao="50cm" precio="16.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza de Jamón y Queso) dentro de una categoría de pizzas, detallando sus alérgenos, descripción y precios para diferentes tamaños.

```
<producto>
    <nombre>Vegetal</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripcion>Base de tomate y queso mozarela, calabacín, cebolla, tomate y pimientos</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Vegetal) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nombre>Cuatro quesos</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripcion>Base de tomate y queso mozarela, cheddar, gouda y parmesano</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Cuatro Quesos) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nombre>Barbacoa</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Soja"/>
        <alergeno nombre="Mostaza"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripcion>Base de tomate y queso mozarela, carne de ternera picada, bacón, cebolla y salsa barbacoa</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Barbacoa) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nOMBRE>Romana</nOMBRE>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripción>Base de tomate y queso mozarella, jamón york, aceitunas negras, champiñones y orégano</descripción>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Romana) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nOMBRE>Hawaiana</nOMBRE>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripción>Base de tomate y queso mozarella, jamón york, piña y champiñones</descripción>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Hawaiana) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nOMBRE>Mexicana</nOMBRE>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
        <alergeno nombre="Dióxido de Azufre y Sulfitos"/>
        <alergeno nombre="Soja"/>
        <alergeno nombre="Mostaza"/>
    </alergenos>
    <descripción>Base de tomate y queso mozarella, pepperoni, carne de ternera picada, cebolla, jalapeños y salsa picante del chef</descripción>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Mexicana) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nombre>Atún</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
        <alergeno nombre="Pescado"/>
    </alergenos>
    <descripcion>Base de tomate y queso mozarella, atún, tomate natural, aceitunas negras, pimientos, cebolla y orégano</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza de Atún) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nombre>Carbonara</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripcion>Base de nata con un toque de pimienta negra, queso mozarella, bacon, cebolla y champiñones</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Carbonara) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nombre>Pepperoni</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
        <alergeno nombre="Soja"/>
        <alergeno nombre="Mostaza"/>
    </alergenos>
    <descripcion>Base de tomate, queso mozarella y pepperoni</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="12.00"/>
        <tamano tamao="50cm" precio="16.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Pepperoni) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nombre>Pollo y bacon</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
        <alergeno nombre="Mostaza"/>
        <alergeno nombre="Soja"/>
    </alergenos>
    <descripcion>Base de tomate, queso mozarella, pollo, bacon y queso cheddar</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza de Pollo y Bacon) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nombre>Diávola</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripcion>Base de tomate, queso mozarella, chorizo picante y queso a elegir (gorgonzola o queso de rulo de cabra)</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Pizza Diávola) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<producto>
    <nombre>Pizza Shawarma</nombre>
    <alergenos>
        <alergeno nombre="Lácteos"/>
        <alergeno nombre="Gluten"/>
    </alergenos>
    <descripcion>Base de tomate, queso mozarella, cebolla, tomate natural, carne de pollo kebab con salsa de yogurt y/o picante</descripcion>
    <tamanos>
        <tamano tamao="30cm" precio="9.00"/>
        <tamano tamao="40cm" precio="13.00"/>
        <tamano tamao="50cm" precio="17.00"/>
    </tamanos>
</producto>
</categoria>
```

Este código XML define un producto específico (Pizza Shawarma) en un menú de pizzería, detallando sus alérgenos, ingredientes y precios para diferentes tamaños.

```
<!-- Categoría de postres -->
<categoria nombre="postres">
    <producto>
        <nombre>Mouse de limón</nombre>
        <descripcion>Deliciosa Mousse adornada con rayadura de limón natural</descripcion>
        <tamanos>
            <tamano tamao="Pequeño" precio="3.00"/>
        </tamanos>
    </producto>
```

Este código XML define una categoría de postres en un menú de pizzería, incluyendo un producto específico (Mousse de Limón) con su descripción y precio para un tamaño pequeño.

```
<producto>
    <nombre>Mousse de chocolate</nombre>
    <descripcion>Suave Mousse de chocolate fondant (50% cacao) con virutas de chocolate negro</descripcion>
    <tamanos>
        <tamano tamao="Pequeño" precio="3.00"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Mousse de Chocolate) en un menú de postres, incluyendo su descripción y precio para un tamaño pequeño.

```
<producto>
    <nombre>Capricho blanco</nombre>
    <descripcion>Crema de chocolate blanco con coulis de frutos rojos</descripcion>
    <tamanos>
        <tamano tamao="Pequeño" precio="3.50"/>
    </tamanos>
</producto>
```

Este código XML define un producto específico (Capricho Blanco) en un menú de postres, describiendo su contenido (crema de chocolate blanco con coulis de frutos rojos) y su precio para un tamaño pequeño.

```
<producto>
    <nOMBRE>Mini cheesecake con mango</nOMBRE>
    <descripción>Exquisita tarta de queso con base de galleta molida y coulis de mango natural</descripción>
    <tAMANOS>
        <tAMANO tamao="Pequeño" precio="3.50"/>
    </tAMANOS>
</producto>
```

Este código XML define un producto específico (Mini Cheesecake con Mango) en un menú de postres, detallando su descripción (tarta de queso con base de galleta molida y coulis de mango natural) y su precio para un tamaño pequeño.

```
<producto>
    <nOMBRE>Cheesecake con dulce de leche</nOMBRE>
    <descripción>Riquísimo pastel de queso con base de galleta molida y cremosa de dulce de leche</descripción>
    <tAMANOS>
        <tAMANO tamao="Pequeño" precio="3.50"/>
    </tAMANOS>
</producto>
```

Este código XML define un producto específico (Cheesecake con Dulce de Leche) en un menú de postres, describiendo su contenido (pastel de queso con base de galleta molida y crema de dulce de leche) y su precio para un tamaño pequeño.

```
<producto>
    <nOMBRE>Conos helados</nOMBRE>
    <descripción>Fresa, vainilla, chocolate y nata</descripción>
    <tAMANOS>
        <tAMANO tamao="Pequeño" precio="1.00"/>
    </tAMANOS>
</producto>
</cATEGORIAS>
```

Este código XML define un producto específico (Conos Helados) en un menú de postres, describiendo los sabores disponibles (fresa, vainilla, chocolate y nata) y su precio para un tamaño pequeño. Además, indica que este producto pertenece a una categoría de postres.

```
<!-- Sección de redes sociales -->
<redes_sociales>
    <web>http://www.pizzeriadivan.metro.rest</web>
    <facebook>https://www.facebook.com/pizzeriadivan</facebook>
    <instagram>https://www.instagram.com/pizzeriadivan</instagram>
    <twitter>https://twitter.com/pizzeriadivan</twitter>
    <email>pizza.divan2019@gmail.com</email>
</redes_sociales>

</pizzeria>
```

Este código XML define una sección de redes sociales para una pizzería, incluyendo enlaces a su sitio web, cuentas de Facebook, Instagram, Twitter y una dirección de correo electrónico.

3.- Ejercicio 2: Creación del DTD

3.1.- Descripción del ejercicio

En este ejercicio se nos solicita la creación de un Documento Tipo de Definición (DTD) para el archivo XML previamente creado en el ejercicio anterior. El objetivo es definir las reglas que estructuran los datos del archivo XML y garantizar que los elementos y atributos sean consistentes con los requerimientos del modelo. El DTD actúa como una especie de "contrato" que especifica las estructuras válidas para el XML, validando así su formato y contenido.

3.2.- Explicación del DTD

Un DTD (Documento de Tipo de Definición) es una colección de reglas que describe la estructura y los tipos de datos permitidos en un archivo XML. Los DTD se pueden definir internamente dentro de un archivo XML o externamente como un archivo separado.

Un DTD define:

- **Elementos:** Las etiquetas o estructuras de datos permitidas en el XML.
- **Atributos:** Las propiedades que pueden tener los elementos.

- **Relaciones entre elementos:** Cómo se deben organizar y anidar los elementos.
- **Restricciones de los datos:** Qué tipo de contenido (texto, número, etc.) es válido para cada elemento.

3.3.- Definición de elementos y atributos en el DTD

En este DTD, se define la estructura de los elementos y atributos utilizados en el archivo XML de la pizzería. Los elementos principales son:

- **pizzeria:** El elemento raíz que contiene toda la información.
- **sucursales:** Elemento que agrupa las diferentes sucursales.
- **sucursal:** Elemento que describe una sucursal específica.
- **categorias:** Contenedor de las categorías de productos.
- **categoria:** Elemento que agrupa los productos por tipo (entrantes, pizzas, postres).
- **producto:** Define un producto específico de la pizzería.
- **ingredientes:** Lista de los ingredientes de un producto.
- **alergenos:** Lista de alérgenos asociados a un producto.
- **tamanos:** Elemento que define los tamaños de un producto (en el caso de las pizzas).

A continuación, se definen los elementos y atributos:

- **<pizzeria>:** Elemento raíz, sin atributos, que contiene toda la estructura.
- **<sucursales>:** Contenedor de las sucursales, sin atributos.
- **<sucursal>:** Elemento que contiene los datos de una sucursal, con un atributo `id` que es único para cada sucursal.
- **<direccion>, <localidad>, <telefono>:** Elementos dentro de sucursal que contienen texto.
- **<horarios>:** Elemento contenedor para los horarios de apertura de cada día.
- **<dia>:** Elemento que representa un día de la semana, con atributos `nombre`, `apertura` y `cierre`.
- **<categorias>:** Contenedor de las categorías de productos, sin atributos.
- **<categoria>:** Elemento que representa una categoría de producto, con el atributo `nombre`.
- **<producto>:** Elemento que describe un producto, con atributos `id`.
- **<nombre>, <precio>, <descripcion>:** Elementos que contienen datos textuales del producto.
- **<ingredientes>:** Contenedor de ingredientes de un producto.

- <ingrediente>: Elemento que lista un ingrediente, con el atributo nombre.
- <alergenos>: Contenedor de alérgenos del producto.
- <alergeno>: Elemento que lista un alérgeno, con el atributo nombre.
- <tamanos>: Contenedor de los diferentes tamaños del producto (solo para pizzas).
- <tamano>: Elemento que describe un tamaño de pizza, con los atributos diametro y precio.

3.4.- Explicación del Archivo DTD

A continuación, se ofrece una explicación sobre la definición del DTD para el archivo XML de la pizzería:

```
<!ELEMENT pizzeria (sucursales, ingredientes, ingredientes_extra, alergenos, categorias, redes_sociales)>
```

Este código define la estructura de un documento XML para una pizzería, especificando que el elemento raíz "pizzeria" debe contener elementos hijos para sucursales, ingredientes, ingredientes extra, alérgenos, categorías de productos y redes sociales.

```
<!-- Definición de sucursales -->
<!ELEMENT sucursales (sucursal+)>
<!ELEMENT sucursal (direccion, localidad, telefono, horarios)>
<!ATTLIST sucursal id ID #REQUIRED>
<!ELEMENT direccion (#PCDATA)>
<!ELEMENT localidad (#PCDATA)>
<!ELEMENT telefono (#PCDATA)>
<!ELEMENT horarios (horario+)>
<!ELEMENT horario (#PCDATA)>
<!ATTLIST horario dia CDATA #IMPLIED>
```

Este código DTD define la estructura y atributos de los elementos relacionados con las sucursales de una pizzería, incluyendo detalles como dirección, localidad, teléfono y horarios.

```
<!-- Definición de ingredientes -->
<!ELEMENT ingredientes (ingrediente+)>
<!ELEMENT ingrediente EMPTY>
<!ATTLIST ingrediente id ID #IMPLIED
          nombre CDATA #REQUIRED
          precio_extra CDATA #IMPLIED>
```

Este código DTD define la estructura de los ingredientes en una pizzería, especificando que cada ingrediente es un elemento vacío con atributos para id, nombre y precio extra.

```
<!-- Definición de ingredientes extra -->
<!ELEMENT ingredientes_extra (ingrediente+)>
```

Este código DTD define la estructura para los ingredientes extra en una pizzería, indicando que el elemento "ingredientes_extra" debe contener uno o más elementos "ingrediente".

```
<!-- Definición de alérgenos -->
<!ELEMENT alergenos (alergeno+)>
<!ELEMENT alergeno EMPTY>
<!ATTLIST alergeno id ID #IMPLIED
          nombre CDATA #REQUIRED>
```

Este código DTD define la estructura de los alérgenos en un menú de restaurante, especificando que cada alérgeno es un elemento vacío con atributos para id y nombre.

```
<!-- Definición de categorías -->
<!ELEMENT categorias (categoria+)>
<!ELEMENT categoria (producto+)>
<!ATTLIST categoria id ID #IMPLIED
          nombre CDATA #REQUIRED>
<!ELEMENT producto (nombre, alergenos?, descripcion, tamanos, precio?)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT descripcion (#PCDATA)>
<!ELEMENT tamanos (tamano+)>
<!ELEMENT tamano EMPTY>
<!ATTLIST tamano tamao CDATA #REQUIRED
          precio CDATA #REQUIRED>
<!ELEMENT precio (#PCDATA)>
```

Este código DTD define la estructura jerárquica de categorías y productos en un menú, incluyendo detalles como nombres, descripciones, alérgenos, tamaños y precios para cada producto.

```
<!-- Definición de redes sociales -->
<!ELEMENT redes_sociales (web, facebook, instagram, twitter, email)>
<!ELEMENT web (#PCDATA)>
<!ELEMENT facebook (#PCDATA)>
<!ELEMENT instagram (#PCDATA)>
<!ELEMENT twitter (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

Este código DTD define la estructura de las redes sociales de una pizzería, especificando elementos para web, Facebook, Instagram, Twitter y email.

4.- Ejercicio 3: Creación del XML Schema (XSD)

4.1.- Descripción del ejercicio

El objetivo de este ejercicio es la creación de un archivo XML Schema (XSD) que defina la estructura, los elementos y las restricciones de los datos que serán utilizados en el archivo XML de la pizzería. El archivo XSD tiene como finalidad proporcionar un marco de validación para los documentos XML, garantizando que los datos contenidos en ellos sean consistentes y sigan la estructura

especificada. Mediante la creación del XSD, se establece un modelo de los elementos XML, sus tipos de datos y las relaciones entre ellos.

4.2.- Explicación del XSD

El archivo XML Schema (XSD) es un estándar para describir la estructura y los tipos de datos en un documento XML. Es similar al DTD, pero ofrece mayores capacidades, como la definición de tipos de datos complejos, restricciones sobre valores y elementos, y la capacidad de especificar relaciones más detalladas entre elementos.

En este ejercicio, el archivo XSD define los elementos de la pizzería, sucursales, categorías de productos, productos y sus atributos de la siguiente manera:

- **Elementos:** Los elementos definidos en el XSD corresponden a los elementos del archivo XML, como `pizzeria`, `sucursales`, `categoria`, `producto`, etc.
- **Tipos de datos:** Se definen tipos de datos específicos para cada elemento, como `xs:string`, `xs:int`, `xs:decimal`, entre otros, para garantizar que los valores de los elementos XML sean del tipo correcto.
- **Restricciones:** Se especifican restricciones sobre los elementos, como si son obligatorios u opcionales, y se definen valores predeterminados para algunos de los atributos.
- **Estructura jerárquica:** El XSD describe cómo los elementos se relacionan entre sí, estableciendo si un elemento puede contener otros elementos, cuántas veces pueden repetirse y si pueden tener atributos.

4.3.- Validación de datos en el XSD

El archivo XSD permite la validación de los datos contenidos en un archivo XML. Al definir reglas claras sobre la estructura y los tipos de datos permitidos, el XSD ayuda a asegurarse de que los documentos XML sean válidos y consistentes. Las validaciones posibles incluyen:

- **Tipificación de datos:** Asegura que los datos dentro de cada elemento coincidan con el tipo de dato esperado (por ejemplo, un número entero para la cantidad de un producto, o una cadena de texto para la descripción de un producto).
- **Restricciones sobre valores:** El XSD puede especificar valores concretos o rangos para ciertos atributos o elementos. Por ejemplo, se puede establecer que el precio de un producto debe

ser un número positivo o que el atributo `id` de un producto debe ser único.

- **Cardinalidad de los elementos:** Permite definir cuántas veces pueden repetirse los elementos dentro de otros, como en el caso de la cantidad de productos dentro de una categoría, o las sucursales dentro de la pizzería.

4.4.- Explicación del archivo XSD

A continuación, se presenta una explicación del archivo XSD que valida el documento XML de la pizzería. Este archivo XSD establece la estructura y las normas de los datos:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

Este código XML define el encabezado de un esquema XSD, especificando que se trata de un documento XML que utiliza el estándar de XML Schema para definir la estructura y las reglas de validación de otros documentos XML, estableciendo así un marco normativo para la correcta organización y tipificación de los datos.

```
<!-- Elemento raíz de la estructura XML -->
<xs:element name="pizzeria">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="sucursales" type="SucursalesType" />
            <xs:element name="ingredientes" type="IngredientesType" />
            <xs:element name="ingredientes_extra" type="IngredientesExtraType" />
            <xs:element name="alergenos" type="AlergenosType" />
            <xs:element name="categorias" type="CategoriasType" />
            <xs:element name="redes_sociales" type="RedesSocialesType" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Este código define el elemento raíz del esquema XML, denominado "pizzeria", que es un tipo complejo que contiene una secuencia de elementos hijos obligatorios, incluyendo "sucursales", "ingredientes", "ingredientes_extra", "alergenos", "categorias" y "redes_sociales", cada uno de los cuales tiene un tipo específico que determina su estructura y contenido.

```
<!-- Definición de la sección <sucursales> -->
<xs:complexType name="SucursalesType">
    <xs:sequence>
        <xs:element name="sucursal" type="SucursalType" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
```

Este código define un tipo complejo llamado "SucursalesType" para la sección `<sucursales>`, que consiste en una secuencia de elementos `<sucursal>`, cada uno de los cuales es del tipo "SucursalType" y puede aparecer un número ilimitado de veces (`maxOccurs="unbounded"`), permitiendo así la inclusión de múltiples sucursales en el documento XML.

```
<!-- Definición de cada <sucursal> -->
<xs:complexType name="SucursalType">
    <xs:sequence>
        <xs:element name="direccion" type="xs:string" />
        <xs:element name="localidad" type="xs:string" />
        <xs:element name="telefono" type="xs:string" />
        <xs:element name="horarios" type="HorariosType" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required" />
</xs:complexType>
```

Este código define un tipo complejo llamado "SucursalType" para el elemento `<sucursal>`, que consiste en una secuencia de elementos obligatorios que incluyen "direccion", "localidad", "telefono" y "horarios", todos de tipo cadena, y también especifica un atributo requerido llamado "id" de tipo ID, que identifica de manera única cada sucursal dentro del documento XML.

```
<!-- Definición de la sección <horarios> -->
<xs:complexType name="HorariosType">
    <xss:sequence>
        <xs:element name="horario" maxOccurs="unbounded">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:string">
                        <xs:attribute name="dia" type="DiaSemanaType" use="optional" />
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xss:sequence>
</xs:complexType>

<xs:simpleType name="DiaSemanaType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="lunes" />
        <xs:enumeration value="martes" />
        <xs:enumeration value="miércoles" />
        <xs:enumeration value="jueves" />
        <xs:enumeration value="viernes" />
        <xs:enumeration value="sábado" />
        <xs:enumeration value="domingo" />
    </xs:restriction>
</xs:simpleType>
```

Este código define un tipo complejo llamado "HorariosType" para la sección `<horarios>`, que consiste en una secuencia de elementos `<horario>`, cada uno de los cuales puede aparecer un número ilimitado de veces (`maxOccurs="unbounded"`) y tiene un contenido de texto que se extiende a partir de una cadena; además, cada `<horario>` puede incluir opcionalmente un atributo "dia" que utiliza el tipo simple "DiaSemanaType". Este último tipo restringe los valores posibles a los días de la semana: lunes, martes, miércoles, jueves, viernes, sábado y domingo.

```
<!-- Definición de la sección <ingredientes> -->
<xs:complexType name="IngredientesType">
    <xss:sequence>
        <xs:element name="ingrediente" maxOccurs="unbounded">
            <xs:complexType>
                <xs:attribute name="id" type="xs:ID" use="required" />
                <xs:attribute name="nombre" type="xs:string" use="required" />
            </xs:complexType>
        </xs:element>
    </xss:sequence>
</xs:complexType>
```

Este código define un tipo complejo llamado "IngredientesType" para la sección `<ingredientes>`, que consiste en una secuencia de elementos `<ingrediente>`, cada uno de los cuales puede aparecer un número ilimitado de veces (`maxOccurs="unbounded"`). Cada `<ingrediente>` tiene dos atributos obligatorios: "id", que es de tipo ID y debe ser único, y "nombre", que es de tipo cadena y representa el nombre del ingrediente.

```
<!-- Definición de la sección <ingredientes_extra> -->
<xs:complexType name="IngredientesExtraType">
    <xs:sequence>
        <xs:element name="ingrediente" maxOccurs="unbounded">
            <xs:complexType>
                <xs:attribute name="nombre" type="xs:string" use="required" />
                <xs:attribute name="precio_extra" type="xs:decimal" use="required" />
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Este código define un tipo complejo llamado "IngredientesExtraType" para la sección `<ingredientes_extra>`, que consiste en una secuencia de elementos `<ingrediente>`, cada uno de los cuales puede aparecer un número ilimitado de veces (`maxOccurs="unbounded"`). Cada `<ingrediente>` tiene dos atributos obligatorios: "nombre", que es de tipo cadena y representa el nombre del ingrediente adicional, y "precio_extra", que es de tipo decimal y especifica el costo adicional asociado a ese ingrediente.

```
<!-- Definición de la sección <alergenos> -->
<xs:complexType name="AlergenosType">
    <xs:sequence>
        <xs:element name="alergeno" maxOccurs="unbounded">
            <xs:complexType>
                <xs:attribute name="id" type="xs:ID" use="required" />
                <xs:attribute name="nombre" type="xs:string" use="required" />
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Este código define un tipo complejo llamado "AlergenosType" para la sección `<alergenos>`, que consiste en una secuencia de elementos `<alergeno>`, cada uno de los cuales puede aparecer un número ilimitado de veces (`maxOccurs="unbounded"`). Cada

<alergeno> tiene dos atributos obligatorios: "id", que es de tipo ID y debe ser único, y "nombre", que es de tipo cadena y representa el nombre del alérgeno, asegurando así que se identifiquen adecuadamente los posibles alérgenos presentes en los productos.

```
<!-- Definición de la sección <categorias> -->
<xss:complexType name="CategoriasType">
    <xss:sequence>
        <xss:element name="categoria" maxOccurs="unbounded">
            <xss:complexType>
                <xss:sequence>
                    <xss:element name="producto" maxOccurs="unbounded" type="ProductoType" />
                </xss:sequence>
                    <xss:attribute name="nombre" type="xs:string" use="required" />
                </xss:complexType>
            </xss:element>
        </xss:sequence>
    </xss:complexType>
```

Este código define un tipo complejo llamado "CategoriasType" para la sección <categorias>, que consiste en una secuencia de elementos <categoria>, cada uno de los cuales puede aparecer un número ilimitado de veces (maxOccurs="unbounded"). Cada <categoria> contiene una secuencia de elementos <producto>, que también pueden aparecer múltiples veces y están definidos por el tipo "ProductoType". Además, cada <categoria> tiene un atributo obligatorio llamado "nombre", que es de tipo cadena y representa el nombre de la categoría, permitiendo así organizar los productos en distintas categorías dentro del menú.

```
<!-- Definición del tipo <producto> -->
<xss:complexType name="ProductoType">
    <xss:sequence>
        <xss:element name="nombre" type="xss:string" />
        <xss:element name="alergenos" minOccurs="0">
            <xss:complexType>
                <xss:sequence>
                    <xss:element name="alergeno" maxOccurs="unbounded">
                        <xss:complexType>
                            <xss:attribute name="nombre" type="xss:string" use="required" />
                        </xss:complexType>
                    </xss:element>
                </xss:sequence>
            </xss:complexType>
        </xss:element>
        <xss:element name="descripcion" type="xss:string" />
        <xss:element name="tamanos">
            <xss:complexType>
                <xss:sequence>
                    <!-- El precio es ahora un atributo de <tamano>, no un elemento -->
                    <xss:element name="tamano" maxOccurs="unbounded">
                        <xss:complexType>
                            <xss:attribute name="tamao" type="xss:string" use="required" />
                            <xss:attribute name="precio" type="xss:decimal" use="required" />
                        </xss:complexType>
                    </xss:element>
                </xss:sequence>
            </xss:complexType>
        </xss:element>
    </xss:sequence>
</xss:complexType>
```

Este código define un tipo complejo llamado "ProductoType" para el elemento `<producto>`, que consiste en una secuencia de elementos obligatorios y opcionales. Incluye un elemento `<nombre>` de tipo cadena, un elemento `<alergenos>` que es opcional (`minOccurs="0"`) y puede contener múltiples elementos `<alergeno>`, cada uno con un atributo obligatorio "nombre". También incluye un elemento `<descripcion>` de tipo cadena y un elemento `<tamanos>`, que a su vez contiene una secuencia de elementos `<tamano>`, cada uno de los cuales puede aparecer múltiples veces y tiene dos atributos obligatorios: "tamao" (que indica el tamaño) y "precio" (que especifica el costo del tamaño), permitiendo así una representación detallada de los productos ofrecidos en la pizzería.

```
<!-- Definición de la sección <redes_sociales> -->
<xss:complexType name="RedesSocialesType">
    <xss:sequence>
        <xss:element name="web" type="xs:string" />
        <xss:element name="facebook" type="xs:string" />
        <xss:element name="instagram" type="xs:string" />
        <xss:element name="twitter" type="xs:string" />
        <xss:element name="email" type="xs:string" />
    </xss:sequence>
</xss:complexType>

</xss:schema>
```

Este código define un tipo complejo llamado "RedesSocialesType" para la sección `<redes_sociales>`, que consiste en una secuencia de elementos obligatorios que representan los enlaces a las diferentes plataformas de redes sociales y contacto de la pizzería. Incluye elementos para la "web", "facebook", "instagram", "twitter" y "email", todos de tipo cadena, permitiendo así que se especifiquen las direcciones correspondientes para cada una de estas redes sociales y métodos de contacto en el documento XML.

5.- Ejercicio 4: Transformación con XSLT

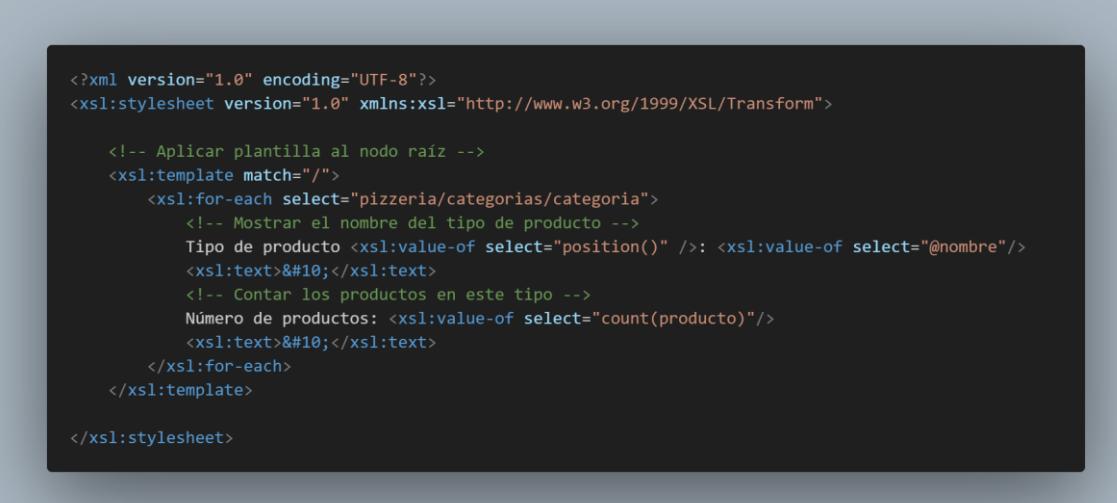
5.1. Descripción del ejercicio

Este ejercicio se centra en el uso de XSLT (eXtensible Stylesheet Language Transformations) para transformar un archivo XML que contiene información sobre una pizzería. Se busca aplicar transformaciones específicas que permitan extraer y presentar los datos de manera estructurada y visualmente atractiva, facilitando así su comprensión y análisis.

5.2.- Primer archivo XSLT: Conteo de productos por tipo

El primer archivo XSLT se encarga de contar la cantidad de productos disponibles en cada categoría dentro del menú de la pizzería. Utiliza estructuras como `<xsl:for-each>` para iterar sobre los elementos de productos y `<xsl:value-of>` para extraer los nombres de las categorías y sus respectivos conteos. Este proceso permite generar un

resumen que muestra cuántos productos hay en cada tipo, lo que es útil para el análisis del inventario.



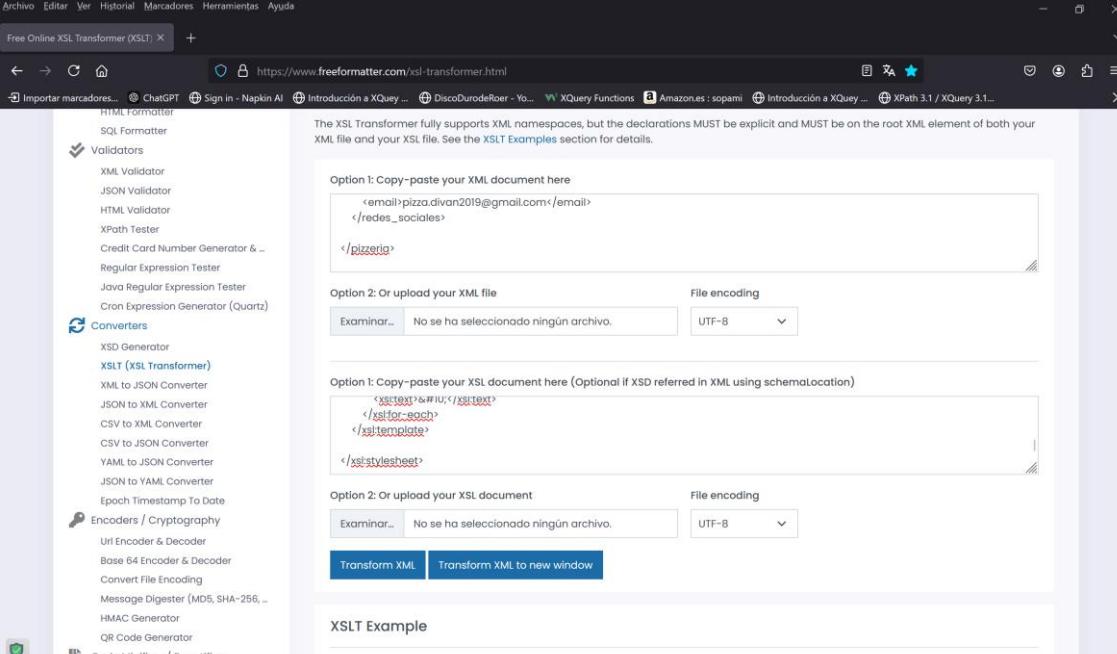
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <!-- Aplicar plantilla al nodo raíz -->
    <xsl:template match="/">
        <xsl:for-each select="pizzeria/categorias/categoría">
            <!-- Mostrar el nombre del tipo de producto -->
            Tipo de producto <xsl:value-of select="position()" />: <xsl:value-of select="@nombre"/>
            <xsl:text>&#10;</xsl:text>
            <!-- Contar los productos en este tipo -->
            Número de productos: <xsl:value-of select="count(producto)" />
            <xsl:text>&#10;</xsl:text>
        </xsl:for-each>
    </xsl:template>

</xsl:stylesheet>
```

Este código XSLT establece una transformación que itera sobre cada categoría de productos en un documento XML de pizzería, mostrando el tipo de producto y el número de productos disponibles en cada categoría, utilizando elementos como `<xsl:template>`, `<xsl:for-each>`, y `<xsl:value-of>` para extraer y presentar la información correspondiente.

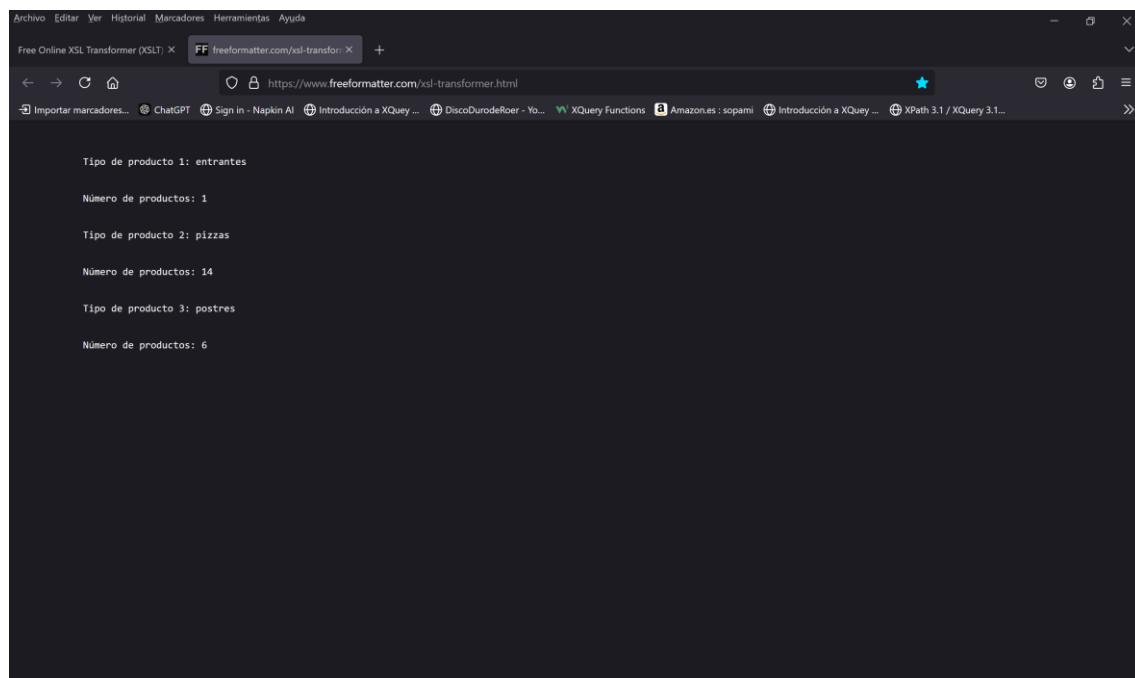
5.3.- Resultados del primer archivo



The screenshot shows the Free Online XSL Transformer interface. On the left, there's a sidebar with various tools like HTML Validator, JSON Validator, XPath Tester, Credit Card Number Generator, Regular Expression Tester, Java Regular Expression Tester, Cron Expression Generator, XSD Generator, XSLT (XSL Transformer), Epoch Timestamp To Date, Url Encoder & Decoder, Base 64 Encoder & Decoder, Convert File Encoding, Message Digester, HMAC Generator, QR Code Generator, and Code Minifiers / Beautifier. The main area has tabs for "Free Online XSL Transformer (XSLT)" and "XQuery Functions". The "XSLT (XSL Transformer)" tab is active. It has two main input fields: "Option 1: Copy-paste your XML document here" containing XML for a pizza menu, and "Option 2: Or upload your XML file" which is currently empty. Below these is a "File encoding" dropdown set to "UTF-8". Further down, there's another "Option 1: Copy-paste your XSL document here" field containing the XSLT code shown earlier, and a "File encoding" dropdown also set to "UTF-8". At the bottom, there are two buttons: "Transform XML" and "Transform XML to new window". A "XSLT Example" section follows, with a note about learning more at W3Schools.com.

Accedo al sitio web <https://www.freeformatter.com/xsl-transformer.html>, donde cargo los archivos pizzeria_divan.xml y

pizzeria_divan1.xls, y luego hago clic en el botón para transformar el XML en una nueva ventana.



El resultado en pantalla presenta un resumen de las categorías del menú de la pizzería, indicando que hay 1 producto en "entrantes", 14 productos en "pizzas" y 6 productos en "postres", lo que refleja la variedad y cantidad de opciones disponibles.

5.4.- Segundo archivo XSLT: Generación de una tabla HTML con el menú de pizzas

El segundo archivo XSLT transforma el XML del menú de pizzas en una tabla HTML. Este archivo utiliza etiquetas como `<table>`, `<tr>`, y `<td>` para estructurar la presentación visual del menú, mostrando cada pizza con su nombre, descripción y precios en diferentes tamaños. La transformación facilita la visualización del menú en un formato amigable para los usuarios, permitiendo una navegación intuitiva.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="UTF-8"/>

<xsl:template match="/">
<html>
<head>
<title>Menú de Pizzas</title>
<style>
    table {
        border-collapse: collapse;
        width: 100%;
    }
    th, td {
        border: 1px solid black;
        padding: 8px;
        text-align: left;
    }
    th {
        background-color: #f2f2f2;
    }
</style>
</head>
<body>
<h1>Menú de Pizzas</h1>
<table>
<tr>
<th>Nombre de la Pizza</th>
<th>Precio 30cm</th>
<th>Precio 40cm</th>
<th>Precio 50cm</th>
</tr>
<xsl:for-each select="pizzeria/categorias/categoría[@nombre='pizzas']/producto">
<tr>
<td><xsl:value-of select="nombre"/></td>
<xsl:for-each select="tamanos/tamano">
<td><xsl:value-of select="@precio"/>€</td>
</xsl:for-each>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

Este código XSLT transforma un documento XML de un menú de pizzería en una página HTML que presenta una tabla con los nombres de las pizzas y sus precios en diferentes tamaños, utilizando estilos CSS para mejorar la presentación visual.

5.5.- Resultado del segundo archivo

The screenshot shows the Free Online XSLT Transformer (XSLT) interface. On the left, there is a sidebar with various tools: XML Validator, JSON Validator, HTML Validator, XPath Tester, Credit Card Number Generator & ..., Regular Expression Tester, Java Regular Expression Tester, Cron Expression Generator (Quartz), Converters (XSD Generator, XSLT (XSL Transformer), XML to JSON Converter, JSON to XML Converter, CSV to XML Converter, CSV to JSON Converter, YAML to JSON Converter, JSON to YAML Converter, Epoch Timestamp To Date), Encoders / Cryptography (Url Encoder & Decoder, Base 64 Encoder & Decoder, Convert File Encoding, Message Digester (MD5, SHA-256, ...), HMAC Generator, QR Code Generator), Code Minifiers / Beautifier (JavaScript Beautifier, JavaScript Minifier, CSS Beautifier), and a shield icon.

The main area has two sections for XML and XSLT documents. Under "Option 1: Copy-paste your XML document here", the XML code is:

```
<email:pizza_divan2019@gmail.com>/email>
</redes_sociales>
</pizzeria>
```

Under "Option 2: Or upload your XML file", there is a "Examinar..." button and a dropdown for "File encoding" set to "UTF-8".

Below these, under "Option 1: Copy-paste your XSL document here (Optional if XSD referred in XML using schemaLocation)", the XSLT code is:

```
</html>
<xsl:template>
<xsl:stylesheet>
```

Under "Option 2: Or upload your XSL document", there is a "Examinar..." button and a dropdown for "File encoding" set to "UTF-8".

At the bottom, there are two buttons: "Transform XML" and "Transform XML to new window".

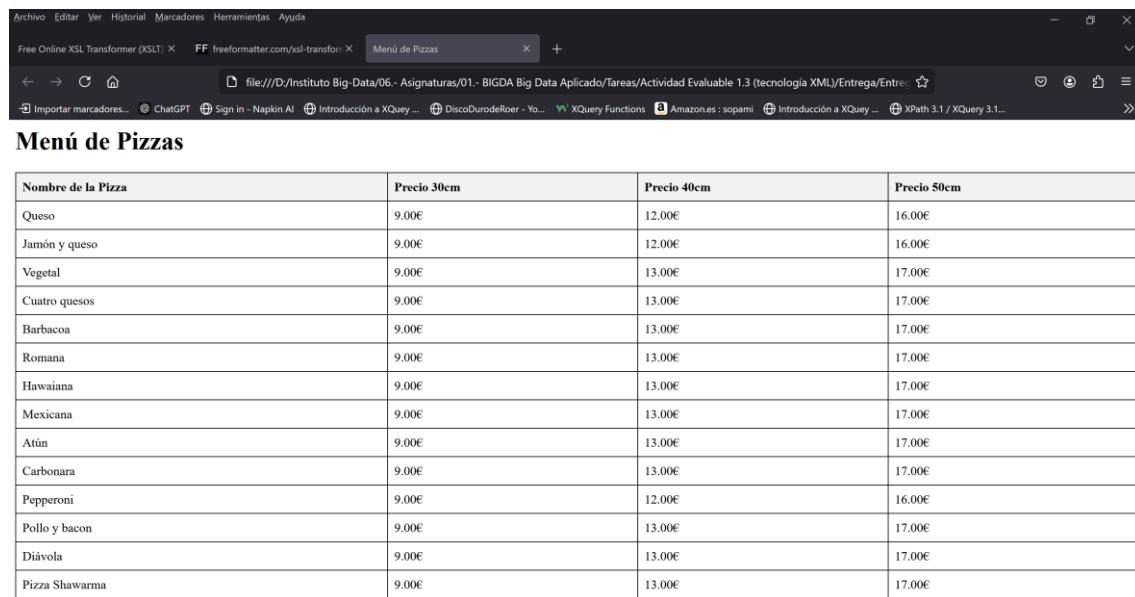
Accedo al sitio web <https://www.freeformatter.com/xsl-transformer.html>, donde cargo los archivos pizzeria_divan.xml y pizzeria_divan2.xsl, y luego hago clic en el botón para transformar el XML en una nueva ventana.

The screenshot shows the browser displaying the transformed HTML output. The page title is "Menú de Pizzas". The content is a table with three columns: Nombre de la Pizza, Precio 30cmc, Precio 40cmc, and Precio 50cmc. The rows represent different pizza types with their respective prices for three sizes.

Nombre de la Pizza	Precio 30cmc	Precio 40cmc	Precio 50cmc
Queso	9.00€	12.00€	16.00€
Jamón y queso	10.00€	13.00€	17.00€
Vegetal	9.00€	13.00€	17.00€

El segundo archivo genera una página HTML que presenta un menú de pizzas en formato de tabla, mostrando el nombre de cada pizza junto con sus precios correspondientes para tres tamaños

diferentes (30 cm, 40 cm y 50 cm), lo que permite a los usuarios visualizar fácilmente las opciones y precios disponibles.



The screenshot shows a browser window titled "Menú de Pizzas". The page contains a table with four columns: "Nombre de la Pizza", "Precio 30cm", "Precio 40cm", and "Precio 50cm". The table lists various pizza types and their prices for each size. The browser's address bar shows the URL: file:///D:/Instituto Big-Data/06.- Asignaturas/01.- BIGDA Big Data Aplicado/Tareas/Actividad Evaluable 1.3 (tecnología XML)/Entrega/Entrega 1.3.

Nombre de la Pizza	Precio 30cm	Precio 40cm	Precio 50cm
Queso	9.00€	12.00€	16.00€
Jamón y queso	9.00€	12.00€	16.00€
Vegetal	9.00€	13.00€	17.00€
Cuatro quesos	9.00€	13.00€	17.00€
Barbacoa	9.00€	13.00€	17.00€
Romana	9.00€	13.00€	17.00€
Hawaiana	9.00€	13.00€	17.00€
Mexicana	9.00€	13.00€	17.00€
Atún	9.00€	13.00€	17.00€
Carbonara	9.00€	13.00€	17.00€
Pepperoni	9.00€	12.00€	16.00€
Pollo y bacon	9.00€	13.00€	17.00€
Diávola	9.00€	13.00€	17.00€
Pizza Shawarma	9.00€	13.00€	17.00€

Al guardar el contenido como un archivo con extensión HTML, se genera una página web que presenta un menú de pizzas en un formato tabular. La tabla incluye encabezados para el nombre de la pizza y los precios correspondientes para tres tamaños diferentes (30 cm, 40 cm y 50 cm). Cada fila de la tabla muestra una pizza específica junto con sus precios, permitiendo a los usuarios visualizar de manera clara y organizada las opciones del menú, lo que facilita la elección de productos. La estructura HTML y el estilo CSS aplicados aseguran que la presentación sea atractiva y fácil de leer.

6. Ejercicio 5: Consultas con XQuery

6.1.- Descripción del ejercicio

El objetivo de este ejercicio es utilizar XQuery, un lenguaje de consultas diseñado para trabajar con documentos XML, para realizar consultas sobre un archivo XML que representa una pizzería. A través de XQuery, se podrán extraer, filtrar, y organizar los datos de forma dinámica y flexible. En este caso, se realizarán dos tipos de consultas:

1. **Productos aptos para celíacos:** Se creará una consulta que filtre los productos que no contengan ingredientes con gluten, identificando aquellos que son aptos para personas celíacas.
2. **Ordenar pizzas por número de ingredientes:** Se realizará una consulta que ordene las pizzas en función de la cantidad de ingredientes que contienen, permitiendo conocer qué pizzas tienen más o menos ingredientes.

El ejercicio tiene como propósito familiarizarse con las consultas XQuery y cómo manipular los datos dentro de un documento XML de manera eficiente.

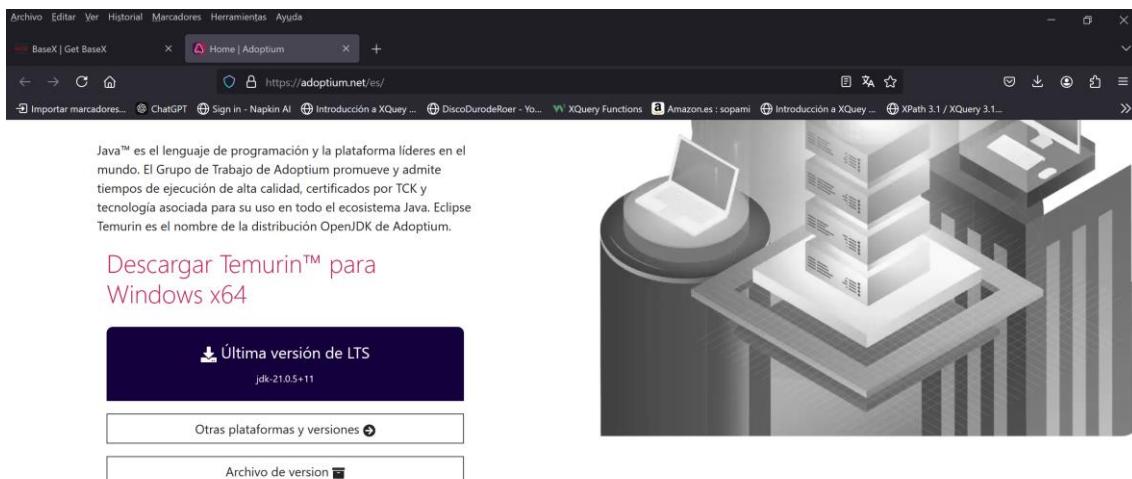
6.2.- Primer archivo XQuery: Productos aptos para celíacos

En este archivo XQuery, la consulta se enfoca en identificar productos que son aptos para celíacos. Esto se logra al filtrar los productos cuyo listado de ingredientes no contiene elementos relacionados con el gluten. Supongamos que en el archivo XML de la pizzería, los ingredientes de los productos incluyen alégenos que podrían indicar la presencia de gluten.

```
for $producto in doc("pizzeria_divan.xml")//producto
where not(contains(string-join($producto/alergenos/alergeno/@nombre, ' '), 'Gluten'))
let $tipo := $producto/ancestor::categoria/@nombre
return
<apto_celiacos tipo="${$tipo}">
  {data($producto/nombre)}
</apto_celiacos>
```

Este código XQuery selecciona productos del archivo XML "pizzeria_divan.xml" que no contienen alégenos relacionados con el gluten, y para cada uno de estos productos, genera un elemento XML <apto_celiacos> que incluye el nombre del producto y un atributo que indica su categoría, permitiendo así identificar claramente los productos aptos para celíacos.

6.3.- Instalación de las aplicaciones



Java™ es el lenguaje de programación y la plataforma líderes en el mundo. El Grupo de Trabajo de Adoptium promueve y admite tiempos de ejecución de alta calidad, certificados por TCK y tecnología asociada para su uso en todo el ecosistema Java. Eclipse Temurin es el nombre de la distribución OpenJDK de Adoptium.

Descargar Temurin™ para Windows x64

Última versión de LTS
jdk-21.0.5+11

Otras plataformas y versiones

Archivo de versión

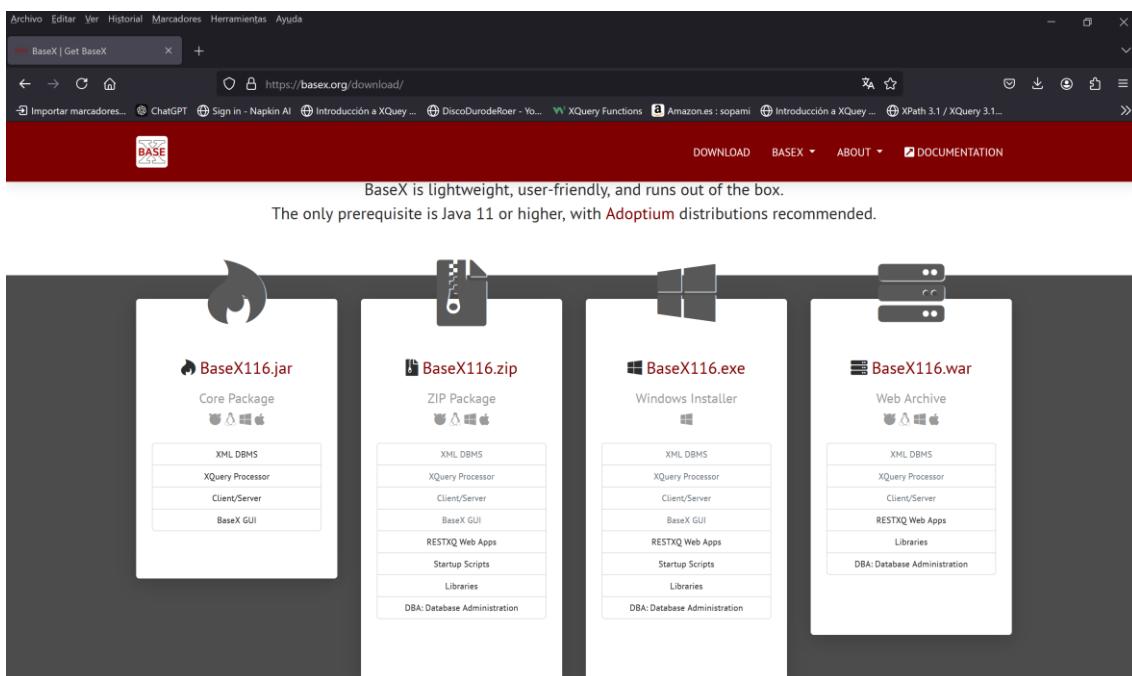


The Adoptium® Working Group

El grupo de trabajo Adoptium promueve y soporta entornos de ejecución de alta calidad y tecnologías asociadas para usar a través de todo el ecosistema Java. Nuestra visión es satisfacer las necesidades de Eclipse y de la amplia comunidad Java proveyendo entornos de ejecución para aplicaciones hechas en Java. Nosotros armonizamos estándares existentes y una gran variedad de

Cambiar idioma

Para poder visualizar los resultados, fue necesario instalar la aplicación Java 11 desde el sitio web <https://adoptium.net/es/>.



BaseX is lightweight, user-friendly, and runs out of the box.
The only prerequisite is Java 11 or higher, with Adoptium distributions recommended.

DOWNLOAD BASEX ABOUT DOCUMENTATION

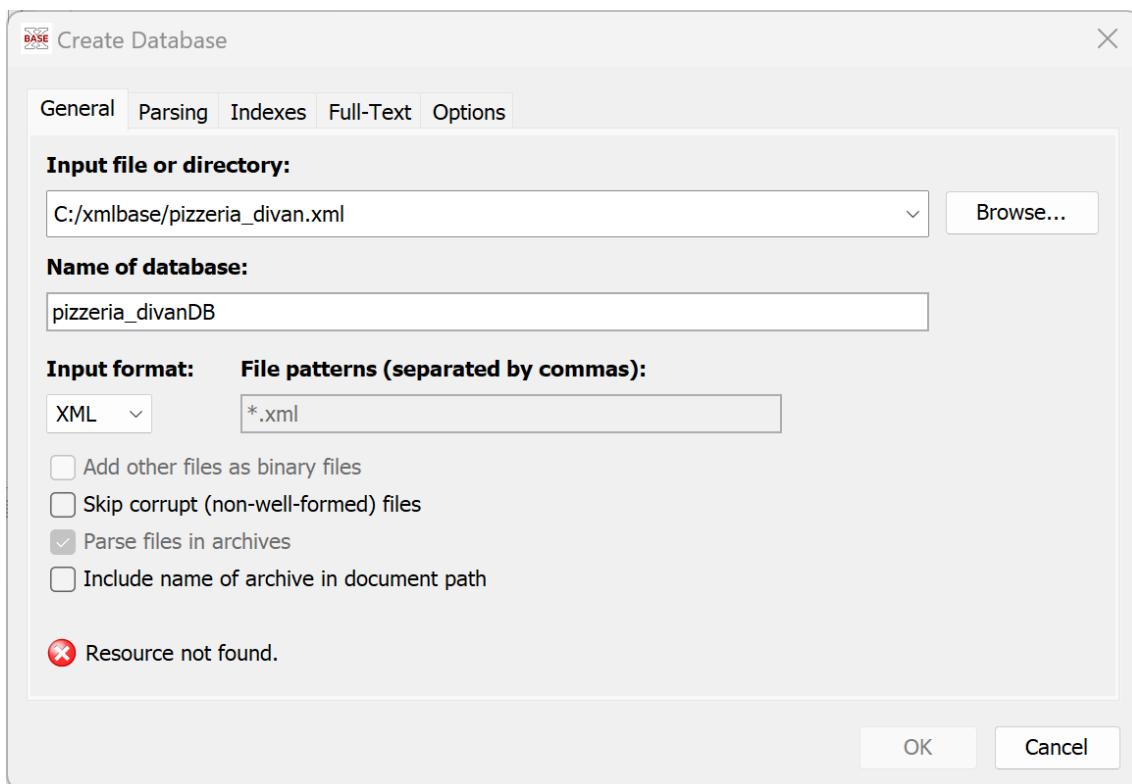
BaseX116.jar
Core Package
XML DBMS
XQuery Processor
Client/Server
BaseX GUI

BaseX116.zip
ZIP Package
XML DBMS
XQuery Processor
Client/Server
BaseX GUI
RESTXQ Web Apps
Startup Scripts
Libraries
DBA: Database Administration

BaseX116.exe
Windows Installer
XML DBMS
XQuery Processor
Client/Server
BaseX GUI
RESTXQ Web Apps
Startup Scripts
Libraries
DBA: Database Administration

BaseX116.war
Web Archive
XML DBMS
XQuery Processor
Client/Server
RESTXQ Web Apps
Libraries
DBA: Database Administration

Para poder visualizar los resultados, fue necesario instalar la aplicación BaseX para Windows desde <https://basex.org/download/>.



A continuación, opté por la opción para crear una nueva base de datos, seleccionando "Database - New". En la sección "Input File or Directory", elegí el archivo XML titulado "pizzeria_divan.xml" y asigné el nombre "pizzeria_divanDB" a la base de datos; posteriormente, hice clic en el botón "OK".

6.4.- Resultado del primer archivo

```

file* - BaseX 11.6
Database Editor View Visualization Options Help
File Edit View Visualization Options Help
Command Command...
Data Aplicado/Tareas/Actividad Evaluable 1.3 (tecnología XML)/Entrega/
*.xml *.xq
Find contents...
Entrega
+ Entrega
  - consulta.xp.txt (68 b)
  - consulta.xq (69 b)
  - ejecutar.py (952 b)
  - pizzeria.xml (15 kb)
  - pizzeria1.xml (261 b)
  - pizzeria1xp.py (1120 b)
  - pizzeria2.xml (264 b)
  - PIZZERIA DIVAN.xml (7045 b)
  - pizzeria_divan.ddf (1871 b)
  - pizzeria_divan.xq (69 b)
  - pizzeria_divan.xsd (659 b)
  - pizzeria_divan1.xsl (703 b)
  - pizzeria_divan2 - copia.xsl (1563 b)
  - pizzeria_divan2.xml (1215 b)

27 pizzeria_divan2.xq 27 file* 
Editor
1 for $producto in doc("pizzeria_divan.xml")//producto
2 where not(contains(string-join($producto/alergenos/alergeno/@nombre, ' '), ' Gluten'))
3 let $tipo := $producto/ancestor::categoria/@nombre
4 let $celiacos := $data($producto/nombre)
5 </apto_cecarios tipo="$tipo">
6   {data($producto/nombre)}
7 </apto_cecarios>
8

OK

```

Result

```

<apto_cecarios tipo="entrantes">Nachos con queso</apto_cecarios>
<apto_cecarios tipo="postres">Mouse de limón</apto_cecarios>
<apto_cecarios tipo="postres">Mousse de chocolate</apto_cecarios>
<apto_cecarios tipo="postres">Capricho blanco</apto_cecarios>
<apto_cecarios tipo="postres">Mini cheesecake con mango</apto_cecarios>
<apto_cecarios tipo="postres">Cheesecake con dulce de leche</apto_cecarios>
<apto_cecarios tipo="postres">Conos helados</apto_cecarios>

```

Result: - Hit(s): 7 Items
- Updated: 0 Items
- Printed: 468 b
- Read Locking: pizzeria_divan.xml
- Write Locking: (none)

Optimized Query:
document { "file:///C:/Instituto%20Big-Data/06 -%20Asignaturas/01 -%20IGDA%20Big%20Data%20Aplicado/Tareas/Actividad%20Evaluable%201.3%20(tecnología%20XML)/Entrega/Entregar/pizzeria_divan.xml" }/descendant::producto[not(contains(string-join(alergenos/alergeno/@nombre, " "), "Gluten"))] ! <apto_cecarios tipo="{ancestor::categoria/@nombre}">{ data(nombre) }</apto_cecarios>

Optimizing:
- rewrite fn:doc(source): doc("pizzeria_divan.xml") -> document { "file:///C:/Instituto%20Big-Data/06 -%20Asignaturas/01 -%20IGDA%20Big%20Data%20%20Evaluable%201.3%20(tecnología%20XML)/Entrega/Entregar/pizzeria_divan.xml" }

El resultado muestra una lista de productos aptos para celíacos en la pizzería Divan, incluyendo un entrante (Nachos con queso) y varios postres, todos ellos sin gluten y categorizados por tipo de plato.

6.5.- Segundo archivo XQuery: Ordenar pizzas por número de ingredientes

```
for $pizza in doc("pizzeria_divan.xml")//categoria[@nombre="pizzas"]/producto
let $num_ingredientes := count(tokenize($pizza/descripcion, ","))
order by $num_ingredientes descending
return concat($pizza/nombre/text(), ":", $num_ingredientes, " ingredientes")
```

Este código XQuery busca todas las pizzas en el documento XML, cuenta el número de ingredientes en cada una basándose en las comas en su descripción, y luego devuelve una lista ordenada de mayor a menor número de ingredientes, mostrando el nombre de cada pizza seguido de la cantidad de ingredientes que contiene.

The screenshot shows the BaseX 11.6 interface. The top bar includes 'file2*- BaseX 11.6', 'Database', 'Editor', 'View', 'Visualization', 'Options', and 'Help'. The 'Editor' tab is active, showing the XQuery code:

```
for $pizza in doc("pizzeria_divan.xml")//categoria[@nombre="pizzas"]/producto
let $num_ingredientes := count(tokenize($pizza/descripcion, ","))
order by $num_ingredientes descending
return concat($pizza/nombre/text(), ":", $num_ingredientes, " ingredientes")
```

The results pane shows '14 Results' and the output:

Nombre	Cantidad de Ingredientes
Atún	6
Mexicana	5
Pizza Shawarma	5
Vegetal	4
Barbacoa	4
Romana	4
Carbonara	4
Pollo y bacon	4
Cuatro quesos	3
Hawaiana	3
Diávola	3
Pepperoni	2
Queso	1
Jamón y queso	1

The bottom pane shows the 'Result' and 'Info' sections with detailed query statistics.

El resultado muestra una lista de pizzas ordenadas de mayor a menor según la cantidad de ingredientes que contienen, comenzando con la pizza de Atún que tiene 6 ingredientes y terminando con las

pizzas de Queso y Jamón y queso que tienen solo 1 ingrediente cada una.

7. Conclusiones

7.1. Reflexiones sobre el proceso

El desarrollo de este proyecto permitió explorar diferentes tecnologías relacionadas con la gestión y procesamiento de datos estructurados, como XML, DTD, XSD, XSLT y XQuery. Durante el proceso, se comprendió cómo estas herramientas se complementan para diseñar, validar, transformar y consultar datos. Cada etapa proporcionó un entendimiento más profundo sobre la flexibilidad de XML como estándar de intercambio de información y la importancia de las especificaciones que aseguran la consistencia y validez de los datos.

Además, trabajar con estas tecnologías resaltó la relevancia de estructurar correctamente los datos desde su creación, ya que una estructura bien diseñada facilita tareas posteriores como validaciones y transformaciones. Esto también subrayó la necesidad de documentar cada paso, garantizando la comprensión y la reutilización del trabajo.

7.2. Desafíos enfrentados y soluciones implementadas

El proceso presentó varios desafíos:

1. **Diseño del XML y el DTD:** Asegurar que el esquema del XML reflejara adecuadamente la estructura jerárquica de los datos requería un análisis cuidadoso. La solución fue iterar sobre el diseño inicial, haciendo ajustes tras validar el archivo con el DTD.
2. **Creación del XSD:** Convertir reglas previamente definidas en DTD a un esquema más detallado y robusto en XSD fue un desafío debido a las diferencias entre ambas tecnologías. Se resolvió utilizando ejemplos concretos para mapear las reglas de DTD a XSD, aprovechando las capacidades adicionales del XSD, como los tipos de datos.
3. **Transformación con XSLT:** Lograr que las transformaciones produjeran salidas claras y útiles implicó comprender cómo navegar por las jerarquías XML. La solución fue realizar pruebas incrementales, verificando cada transformación antes de avanzar al siguiente paso.

4. **Consultas con XQuery:** Diseñar consultas eficientes que extrajeran y ordenaran los datos deseados fue un reto, especialmente al trabajar con múltiples niveles de jerarquía. Esto se resolvió creando consultas sencillas primero y luego ampliándolas para cubrir casos más complejos.

7.3. Importancia del trabajo con XML, DTD, XSD, XSLT y XQuery en la gestión de datos

El trabajo con estas tecnologías es esencial en el contexto actual de la gestión de datos. XML proporciona un estándar ampliamente aceptado para estructurar y compartir información en diversos dominios, desde bases de datos hasta servicios web.

- **DTD y XSD:** Estos formatos garantizan que los datos estructurados cumplan con reglas predefinidas, asegurando la validez e integridad de la información. Esto es fundamental en sistemas donde la calidad y consistencia de los datos son críticas.
- **XSLT:** Esta herramienta facilita la transformación de datos XML en otros formatos, como HTML o CSV, lo que es crucial para hacer que la información sea accesible y utilizable en diferentes contextos.
- **XQuery:** Permite realizar consultas complejas sobre datos XML, haciéndolo una herramienta poderosa para analizar y manipular grandes volúmenes de datos estructurados.

En conjunto, estas herramientas ofrecen un ecosistema robusto para trabajar con datos estructurados, permitiendo integrarlos en flujos de trabajo modernos y optimizando la toma de decisiones basada en datos. Este proyecto ha demostrado su utilidad en escenarios prácticos, desde la creación de bases de datos XML bien definidas hasta la extracción y transformación eficiente de información.

8. Bibliografía

8.1. Fuentes utilizadas para la creación de los archivos XML, DTD, XSD, XSLT y XQuery

- *W3C XML Specification:* La especificación oficial de XML proporcionada por el W3C, que sirvió como base para estructurar y validar los archivos XML creados durante este proyecto.
Disponible en: <https://www.w3.org/XML/>

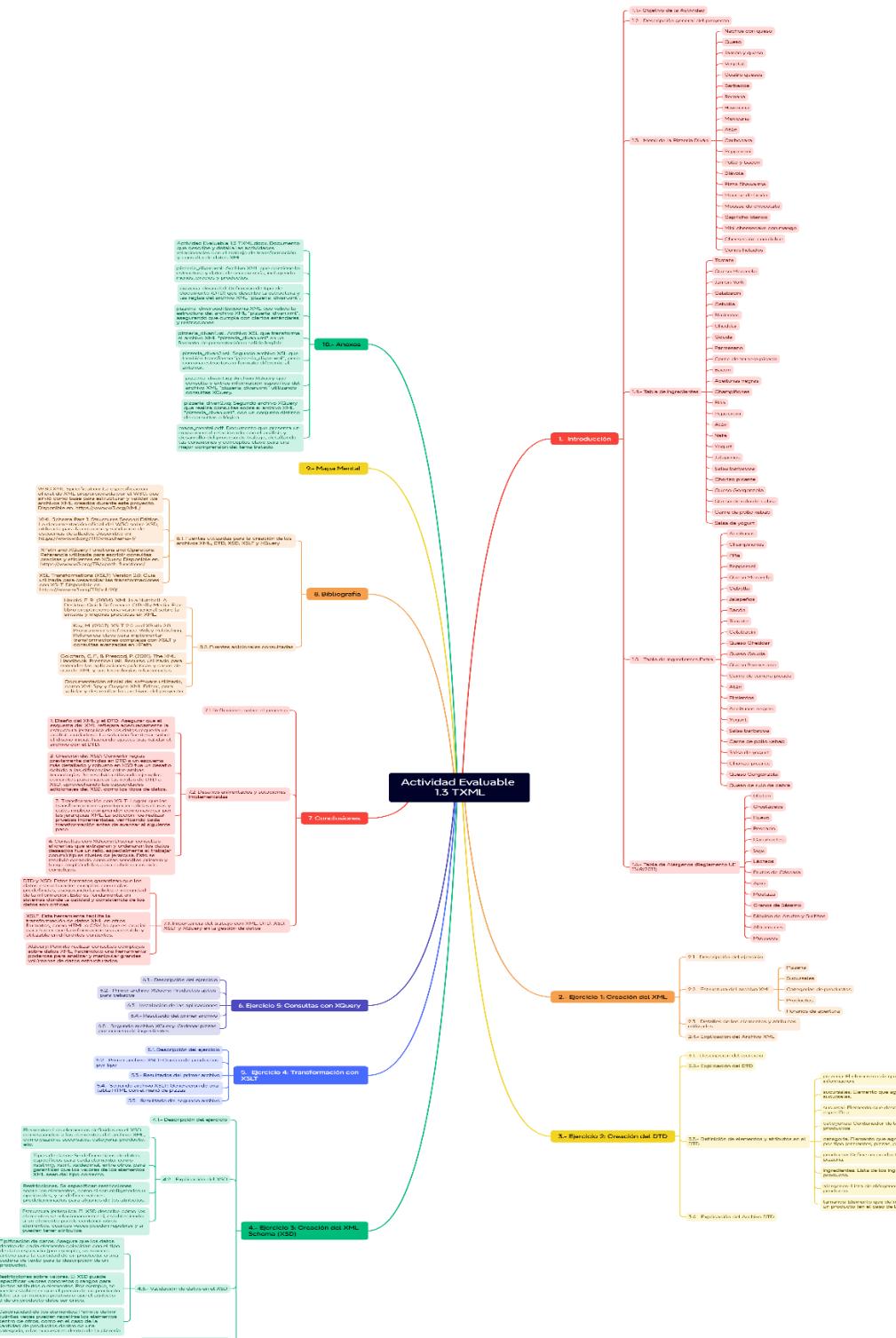
- *XML Schema Part 1: Structures Second Edition*: La documentación oficial del W3C sobre XSD, utilizada para la creación y validación de esquemas detallados. Disponible en: <https://www.w3.org/TR/xmlschema-1/>
- *XPath and XQuery Functions and Operators*: Referencia utilizada para escribir consultas precisas y eficientes en XQuery. Disponible en: <https://www.w3.org/TR/xpath-functions/>
- *XSL Transformations (XSLT) Version 2.0*: Guía utilizada para desarrollar las transformaciones con XSLT. Disponible en: <https://www.w3.org/TR/xslt20/>

8.2. Fuentes adicionales consultadas

- Harold, E. R. (2004). *XML in a Nutshell: A Desktop Quick Reference*. O'Reilly Media. Este libro proporcionó una visión general sobre la sintaxis y mejores prácticas en XML.
- Kay, M. (2007). *XSLT 2.0 and XPath 2.0 Programmer's Reference*. Wiley Publishing. Referencia clave para implementar transformaciones complejas con XSLT y consultas avanzadas en XPath.
- Goldfarb, C. F., & Prescod, P. (2001). *The XML Handbook*. Prentice Hall. Recurso utilizado para entender las aplicaciones prácticas y casos de uso de XML y sus tecnologías relacionadas.
- Documentación oficial del software utilizado, como XMLSpy y Oxygen XML Editor, para validar y desarrollar los archivos del proyecto.

Estas fuentes permitieron asegurar la calidad y validez técnica de cada componente creado, además de enriquecer el conocimiento sobre las capacidades y aplicaciones prácticas de XML y sus tecnologías complementarias.

9.- Mapa Mental



El mapa mental resume las conexiones y conceptos clave del proyecto, destacando las etapas de trabajo con XML, DTD, XSD, XSLT y XQuery, y cómo estas tecnologías se integran para gestionar y transformar datos estructurados de manera eficiente.

10.- Anexos

Actividad_Evaluable_1.3_TXML.docx: Documento que describe y detalla las actividades relacionadas con el trabajo de transformación y consulta de datos XML.

pizzeria_divan.xml: Archivo XML que contiene la estructura y datos de una pizzería, incluyendo menús, precios y productos.

pizzeria_divan.dtd: Definición de tipo de documento (DTD) que describe la estructura y las reglas del archivo XML "pizzeria_divan.xml".

pizzeria_divan.xsd: Esquema XML que valida la estructura del archivo XML "pizzeria_divan.xml", asegurando que cumpla con ciertos estándares y restricciones.

pizzeria_divan1.xsl: Archivo XSL que transforma el archivo XML "pizzeria_divan.xml" en un formato de presentación o salida legible.

pizzeria_divan2.xsl: Segundo archivo XSL que también transforma "pizzeria_divan.xml", pero con una estructura o formato diferente al anterior.

pizzeria_divan1.xq: Archivo XQuery que consulta y extrae información específica del archivo XML "pizzeria_divan.xml" utilizando consultas XQuery.

pizzeria_divan2.xq: Segundo archivo XQuery que realiza consultas sobre el archivo XML "pizzeria_divan.xml", con un conjunto distinto de consultas o lógica.

mapa_mental_Actividad_Evaluable_1.3_TXML.pdf :
Documento que presenta un mapa mental relacionado con el análisis y desarrollo del proceso de trabajo, detallando las conexiones y conceptos clave para una mejor comprensión del tema tratado.

Índice Alfabético

A

abarcando _____ 8, 12
actividad _____ 4, 5
adecuadamente _____ 32, 44
además _____ 30, 46
adicional _____ 4, 7, 13, 31
adicionales _____ 3, 44, 46
agrupa _____ 9, 10, 23
alergeno _____ 10, 24, 31, 33
alérgeno _____ 10, 24, 25, 32
alergenos _____ 10, 23, 24, 28, 31, 33, 40
alérgenos 4, 5, 8, 9, 10, 14, 15, 16, 17, 18, 19, 20, 23, 24, 25, 26, 32, 40
almacenar _____ 4, 8
análisis _____ 34, 35, 44, 48
anterior _____ 22, 48
aparecer _____ 29, 30, 31, 32, 33
apertura _____ 4, 9, 11, 23
aplicación _____ 4, 41
aplicaciones _____ 2, 4, 41, 46
aplicar _____ 4, 34
aptos _____ 2, 40, 43
archivo 2, 3, 4, 8, 9, 10, 11, 22, 23, 24, 26, 27, 28, 34, 35, 36, 38, 39, 40, 42, 43, 44, 48
archivos _____ 3, 5, 35, 38, 45, 46
aseguran _____ 39, 44
asegurando _____ 5, 32, 45, 48
asegurar _____ 46
así 4, 22, 28, 29, 32, 33, 34, 40
asignando _____ 12, 14
atractiva _____ 34, 39
atributo _____ 10, 23, 24, 28, 29, 30, 32, 33, 40
atributos _____ 2, 9, 10, 22, 23, 24, 25, 27, 31, 32, 33

B

base _____ 4, 21, 42, 45
bases _____ 45
baseX _____ 41
bien _____ 8, 44, 45
blanco _____ 20
botón _____ 36, 38, 42
busca _____ 4, 34, 43

C

cada _____ 4, 5, 7, 8, 9, 11, 12, 13, 14, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 38, 40, 43, 44, 46
cadena _____ 27, 29, 30, 31, 32, 33, 34
calidad _____ 45, 46
cantidad _____ 27, 28, 34, 36, 40, 43
capacidades _____ 27, 44, 46
cargo _____ 35, 38
carnes _____ 6, 12
caso _____ 4, 9, 23, 28, 39
casos _____ 45, 46
categoria _____ 9, 10, 23, 27, 32
categoría _____ 10, 14, 15, 20, 21, 23, 28, 32, 34, 35, 40
categorias _____ 9, 10, 23, 28, 32
categorías _____ 4, 9, 10, 23, 24, 26, 27, 32, 34, 36
celíacos _____ 2, 40, 43

Ch

chocolate _____ 20, 21

C

cierra _____ 9, 23
ciertos _____ 27, 48
claras _____ 27, 44
clave _____ 4, 8, 9, 46, 47, 48
clic 36, 38, 42

código _____ 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 37, 40, 43
cómo _____ 4, 5, 27, 40, 44, 47
complejas _____ 45, 46
complejo _____ 28, 29, 30, 31, 32, 33, 34
complejos _____ 5, 27, 45
comprender _____ 4, 44
comprensión _____ 34, 44, 48
conceptos _____ 47, 48
concretos _____ 27, 44
conexiones _____ 47, 48
conjunto _____ 45, 48
consiste _____ 29, 30, 31, 32, 33, 34
consistencia _____ 44, 45
consistentes _____ 5, 22, 26, 27
consulta _____ 4, 40, 48
consultar _____ 4, 44
consultas _____ 4, 39, 40, 45, 46, 48
contacto _____ 34
contenedor _____ 23
contener _____ 24, 25, 27, 33
contenga _____ 4, 8
contenido _____ 5, 6, 10, 20, 21, 22, 23, 28, 30, 39
contenidos _____ 26, 27
contexto _____ 4, 45
contiene _____ 4, 10, 23, 28, 32, 33, 34, 40, 43, 48
contienen _____ 23, 40, 43
continuación _____ 9, 10, 23, 24, 28, 42
correspondiente _____ 8, 35
correspondientes _____ 4, 34, 38, 39
costo _____ 31, 33
coulis _____ 20, 21
creación _____ 3, 4, 5, 8, 22, 26, 44, 45, 46
creado _____ 22, 46
crear _____ 4, 8, 42
crema _____ 20, 21
cuales _____ 28, 29, 30, 31, 32, 33
cuántas _____ 27, 28

D

datos 2, 3, 4, 5, 8, 10, 22, 23, 26, 27, 28, 34, 39, 40, 42, 44, 45, 47, 48
debe _____ 4, 8, 24, 25, 27, 31, 32
deben _____ 8, 23
decimal _____ 27, 31
define 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
definen _____ 23, 27
definición _____ 24, 27
definidas _____ 44, 45
definidos _____ 27, 32
definir _____ 22, 27, 28
dentro _____ 15, 22, 23, 27, 28, 29, 32, 34, 40
desafíos _____ 44
desarrollar _____ 4, 46
desarrollo _____ 4, 44, 48
describe _____ 9, 10, 22, 23, 24, 27, 48
describiendo _____ 20, 21
descripción _____ 10, 23, 33
descripción _____ 14, 15, 20, 21, 27, 36, 43
descripciones _____ 5, 26
detallada _____ 11, 33
detallado _____ 5, 44
detallando _____ 14, 15, 16, 17, 18, 19, 20, 21, 48
detalles _____ 9, 24, 26
día 9, 23, 30
día 9, 11, 23
días _____ 9, 30
diferentes 5, 6, 9, 10, 15, 16, 17, 18, 19, 20, 23, 24, 34, 36, 37, 39, 44, 45
dirección _____ 9, 23, 29
dirección _____ 4, 9, 11, 22, 24
diseñado _____ 4, 39
diseñar _____ 44
diseño _____ 44
disponibles _____ 7, 13, 21, 34, 35, 36, 39
divan _____ 11, 35, 38, 40, 42, 48
documentación _____ 46
documento 10, 11, 24, 27, 28, 29, 34, 35, 37, 40, 43, 48
documentos _____ 4, 26, 27, 28, 39
dos 31, 32, 33, 39

Pedro Manuel García Álvarez
Actividad Evaluable 1.3 (tecnología XML)

DTD 11, 48
dulce _____ 21

E

eficiente _____ 4, 40, 45, 47
eficientes _____ 45, 46
ejemplo _____ 27
ejercicio _____ 2, 4, 8, 22, 26, 27, 34, 39, 40
ejercicios _____ 4
elemento _____ 11, 23, 24, 25, 27, 28, 29, 33, 40
elementos 2, 5, 8, 9, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 40
email _____ 26, 34
enlaces _____ 22, 34
entrantes _____ 4, 5, 9, 10, 14, 23, 36
especialmente _____ 10, 45
específica _____ 9, 22, 29, 31, 33
específica _____ 9, 23, 39, 48
especificando _____ 10, 11, 13, 24, 25, 26, 28
especificar _____ 10, 27
específico _____ 10, 11, 14, 15, 16, 17, 18, 19, 20, 21, 23, 28
esquema _____ 11, 28, 44
establece _____ 27, 28, 35
estableciendo _____ 27, 28
estándar _____ 27, 28, 44, 45
estilo _____ 11, 39
estructura _____ 4, 8, 10, 22, 23, 24, 25, 26, 27, 28, 39, 44, 48
estructurada _____ 8, 34
estructurados _____ 4, 5, 44, 45, 47
estructurar _____ 36, 44, 45
estructuras _____ 22, 34
etc 9, 23, 27
etiquetas _____ 22, 36
explicación _____ 24, 28
extra _____ 7, 13, 24, 25, 28, 31
extraer _____ 34, 35, 39

F

facebook _____ 34
facilita _____ 36, 39, 44, 45
filtrar _____ 4, 39, 40
forma _____ 8, 39
formato _____ 4, 5, 22, 36, 38, 39, 48
formatos _____ 4, 45
freeformatter _____ 35, 38
fuentes _____ 46
functions _____ 46
futura _____ 5, 8

G

galleta _____ 21
garantizando _____ 26, 44
garantizar _____ 22, 27
genera _____ 38, 39, 40
general _____ 2, 4, 9, 46
generar _____ 4, 34
gestión _____ 3, 4, 8, 44, 45
gestionar _____ 4, 47
gluten _____ 8, 40, 43
grandes _____ 4, 45

H

hago _____ 36, 38
herramienta _____ 45
herramientas _____ 5, 44, 45
hijos _____ 24, 28
horario _____ 9, 30
horarios _____ 4, 9, 11, 23, 24, 29, 30
html _____ 35, 38
https _____ 35, 38, 41, 45, 46

I

identificador _____ 8, 12, 14
identificar _____ 40

ilimitado _____ 29, 30, 31, 32
importancia _____ 44
incluye _____ 5, 33, 39, 40
incluyendo _____ 4, 8, 27, 29, 40
información _____ 4, 5, 8, 9, 11, 23, 34, 35, 44, 45, 48
ingrediente _____ 7, 10, 13, 24, 25, 31, 44
ingredientes 3, 4, 6, 7, 8, 9, 10, 12, 13, 16, 17, 18, 19, 20, 23, 24, 25, 28, 31, 40, 43
instagram _____ 34
instalar _____ 41
iterar _____ 34, 44

J

jerarquía _____ 9, 45
jerárquica _____ 8, 26, 27, 44
junto _____ 38, 39

L

leche _____ 21
lista _____ 6, 7, 8, 12, 13, 14, 24, 43
listado _____ 10, 40

LI

llamado _____ 11, 29, 30, 31, 32, 33, 34

L

localidad _____ 4, 9, 11, 23, 24, 29
luego _____ 36, 38, 43, 45

M

manera _____ 4, 8, 27, 29, 34, 39, 40, 47
mango _____ 21
manipular _____ 40, 45
mapa _____ 47, 48
marco _____ 26, 28
mayor _____ 43
mediante _____ 4
menor _____ 43
mental _____ 47, 48
menú 2, 4, 5, 8, 14, 16, 17, 18, 19, 20, 21, 25, 26, 32, 34, 36, 37, 38, 39
modelo _____ 22, 27
molida _____ 21
mostrando _____ 35, 36, 38, 43
muestra _____ 7, 9, 10, 35, 39, 43
múltiples _____ 29, 32, 33, 45

N

necesario _____ 41
nombre 8, 9, 10, 12, 13, 14, 23, 24, 25, 31, 32, 33, 36, 38, 39, 40, 42, 43
nombres _____ 26, 34, 37
nueva _____ 36, 38, 42
número _____ 3, 23, 27, 28, 29, 30, 31, 32, 35, 40, 43

O

objetivo _____ 4, 8, 22, 26, 39
obligatorio _____ 32, 33
obligatorios _____ 27, 28, 29, 31, 32, 33, 34
oficial _____ 45, 46
ofrece _____ 24, 27
ofrecidos _____ 4, 33
opcionales _____ 27, 33
opciones _____ 36, 39
ordenar _____ 4

Pedro Manuel García Álvarez
Actividad Evaluable 1.3 (tecnología XML)

org 41, 45, 46
organizar _____ 23, 32, 39

P

página _____ 4, 8, 37, 38, 39
paso _____ 44
pequeño _____ 20, 21
permite _____ 4, 27, 34, 39
permitidos _____ 22, 27
permitiendo _____ 7, 29, 32, 33, 34, 36, 39, 40, 45
pizza _____ 10, 24, 36, 38, 39, 43
pizzas _____ 2, 3, 4, 5, 7, 9, 10, 15, 23, 24, 36, 37, 38, 39, 40, 43
pizzería _____ 9, 11, 23, 24, 27, 28, 35, 38, 40, 42, 48
pizzería 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28, 33, 34, 35, 36, 37, 39, 40, 43, 48
poder _____ 41
posibles _____ 27, 30, 32
postres _____ 4, 5, 9, 10, 20, 21, 23, 36, 43
prácticas _____ 46
precio _____ 9, 10, 13, 14, 20, 21, 23, 24, 25, 27, 31, 33
precios _____ 4, 5, 10, 15, 16, 17, 18, 19, 20, 26, 36, 37, 38, 39, 48
presenta _____ 5, 8, 28, 36, 37, 38, 39, 48
presentación _____ 11, 36, 37, 39, 48
presentar _____ 34, 35
presentes _____ 10, 32
previamente _____ 22, 44
primer _____ 2, 3, 34, 35, 42
principal _____ 4, 9
principales _____ 8, 23
procesamiento _____ 5, 44
proceso _____ 3, 34, 44, 48
producto 4, 5, 9, 10, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 26, 27, 32, 33, 35, 36, 39, 40, 43, 48
productos 2, 4, 8, 9, 10, 23, 24, 26, 27, 28, 32, 33, 34, 35, 36, 39, 40, 43, 48
proporcionó _____ 44, 46
proyecto _____ 2, 4, 5, 44, 45, 46, 47
puede _____ 27, 29, 30, 31, 32, 33
pueden _____ 5, 22, 27, 28, 32

Q

queso _____ 14, 21, 43, 44
quesos _____ 6, 12

R

raíz 9, 11, 23, 24, 28
realizar _____ 4, 39, 44, 45
redes _____ 22, 24, 26, 28, 34
reglas _____ 22, 27, 28, 44, 45, 48
relacionadas _____ 4, 44, 46, 48
relacionados _____ 24, 40
relaciones _____ 5, 27
relevantes _____ 4, 8
repetirse _____ 27, 28
representa _____ 23, 31, 32, 39
resolvió _____ 44, 45
restricciones _____ 26, 27, 48
resultado _____ 36, 43
resultados _____ 41
resumen _____ 35, 36
robusto _____ 44, 45

S

salsas _____ 6, 12
sección _____ 22, 29, 30, 31, 32, 34, 42
secuencia _____ 28, 29, 30, 31, 32, 33, 34
según _____ 7, 8, 10, 14, 43
segundo _____ 2, 36, 38
semana _____ 9, 11, 23, 30
ser 4, 8, 28, 31, 32
siguiente _____ 27, 44
siguiientes _____ 8, 9, 10
sistema _____ 4, 8
sistemas _____ 5, 45
sitio _____ 22, 35, 38, 41
sociales _____ 22, 24, 26, 28, 34

solamente _____ 24, 44
solución _____ 44
sucursal _____ 9, 11, 23, 29
sucursales _____ 4, 8, 9, 23, 24, 27, 28, 29

T

tabla _____ 2, 6, 7, 8, 36, 37, 38, 39
tamaño _____ 10, 24, 33
tamanos _____ 10, 23, 24, 33
tamaño _____ 10, 20, 21, 24, 33
tamaños _____ 5, 9, 10, 15, 16, 17, 18, 19, 20, 23, 24, 26, 36, 37, 38, 39
tecnologías _____ 4, 5, 44, 45, 46, 47
telefono _____ 9, 23, 29
teléfono _____ 4, 9, 11, 24
tener _____ 22, 27
texto _____ 23, 27, 30
tipificación _____ 28
tipo 2, 10, 23, 27, 28, 29, 30, 31, 32, 33, 34, 35, 43, 48
tipos _____ 22, 27, 39, 44
toda _____ 8, 9, 23
trabajar _____ 39, 44, 45
trabajo _____ 3, 44, 45, 47, 48
transforma _____ 36, 37, 48
transformación _____ 4, 35, 36, 44, 45, 48
transformaciones _____ 4, 34, 44, 46
transformar _____ 34, 36, 38, 44, 47
transformer _____ 35, 38
través _____ 4, 39
tres 38, 39
twitter _____ 34

U

ubicaciones _____ 5, 9
unbounded _____ 29, 30, 31, 32
único _____ 8, 9, 10, 12, 14, 23, 28, 31, 32
uso 34, 46
usuarios _____ 36, 39
utiliza _____ 28, 30, 36
utilizada _____ 46
utilizado _____ 46
utilizados _____ 2, 6, 9, 23, 26
utilizando _____ 4, 35, 37, 44, 48
utilizar _____ 8, 39

V

vacio _____ 25
valida _____ 28, 48
validación _____ 4, 5, 11, 26, 27, 28, 46
validaciones _____ 27, 44
validar _____ 4, 44, 45, 46
validez _____ 44, 45, 46
valores _____ 5, 27, 30
value _____ 34, 35
variedad _____ 5, 12, 36
varios _____ 4, 43, 44
veces _____ 27, 28, 29, 30, 31, 32, 33
ventana _____ 36, 38
verduras _____ 6, 12
visual _____ 36, 37
visualizar _____ 39, 41
volumenes _____ 4, 45

W

web _____ 4, 8, 22, 26, 34, 35, 38, 39, 41, 45
www _____ 35, 38, 45, 46

X

xml _____ 35, 38, 40, 42, 48
xpath _____ 46
xsd _____ 48
xsl _____ 34, 35, 38, 48