Parte 1

**1.¿Qué es XML?**

XML (Extensible Markup Language) es un lenguaje de marcado de texto diseñado para almacenar y transportar datos de forma estructurada. Es un estándar de la W3C que permite a los desarrolladores crear sus propios elementos y atributos personalizados para describir la información contenida en un documento.

2. ¿Cuál es el propósito del elemento raíz en un documento XML?

El elemento raíz en un documento XML es el elemento principal que contiene todos los demás elementos del documento. Su propósito es proporcionar un punto de referencia para todos los demás elementos en el documento, y permite a los procesadores de XML verificar que el documento esté bien formado.

3. ¿Cuál es la diferencia entre un elemento vacío y un elemento de inicio/fin en XML?

Un elemento vacío en XML es un elemento que no contiene ningún contenido, mientras que un elemento de inicio/fin es un elemento que contiene contenido. Por ejemplo, el elemento "<img>" es un elemento vacío, ya que no contiene ningún texto o datos, mientras que el elemento "<p>" es un elemento de inicio/fin, ya que puede contener texto u otros datos.

4. ¿Cómo se define un atributo en un elemento XML?

Un atributo en un elemento XML se define como una característica adicional del elemento que proporciona información adicional sobre él. Por ejemplo, un elemento "<img>" puede tener un atributo "src" que especifica la URL de la imagen a mostrar. Un atributo se define dentro del elemento, entre los símbolos "<" y ">", y se escribe en la forma nombre="valor".

5. ¿Cuál es el propósito de la declaración XML?

La declaración XML es una línea de texto que se coloca al principio de un documento XML y que especifica la versión del lenguaje XML que se está utilizando en el documento. ¿Por ejemplo, la declaración “<?xml versi="1.0"?>" indica que el documento está utilizando la versión 1.0 del lenguaje XML. La declaración XML también puede especificar la codificación de caracteres utilizada en el documento, para que los procesadores de XML sepan cómo interpretar los caracteres contenidos en el documento.

Parte 2

-Identifique las características generales de los lenguajes de marcas.

Lenguajes de marcas

Los lenguajes de marcas son lenguajes de programación que se utilizan para describir y estructurar la información contenida en un documento. Estos lenguajes utilizan etiquetas o marcas para identificar y definir diferentes elementos dentro del documento, como pueden ser títulos, párrafos, imágenes, tablas, etc.

Características generales

Los lenguajes de marcas utilizan etiquetas o marcas para identificar y definir diferentes elementos en un documento.

Estas etiquetas suelen estar incluidas entre símbolos "<" y ">", y pueden contener atributos que proporcionan información adicional sobre el elemento.

Los lenguajes de marcas permiten la creación de elementos y atributos personalizados, lo que permite a los desarrolladores describir la información contenida en el documento de manera precisa y detallada.

La estructura y el contenido de un documento escrito en un lenguaje de marcas deben seguir una sintaxis específica, y deben cumplir con ciertas reglas y estándares para ser considerados válidos.

- Reconocer las ventajas que proporciona el tratamiento de información.

Ventajas del tratamiento de información

Los lenguajes de marcas permiten a los desarrolladores estructurar y organizar la información de un documento de manera clara y precisa.

Los lenguajes de marcas facilitan la interoperabilidad entre diferentes sistemas, ya que permiten a los procesadores de datos interpretar la información contenida en el documento de manera uniforme.

Los lenguajes de marcas son adecuados para la creación de documentos que se deben compartir y utilizar en diferentes plataformas y dispositivos, ya que pueden ser leídos por diferentes programas y aplicaciones.

- Clasifique los lenguajes de Marcas e identifique a los más relevantes.

Clasificación de los lenguajes de marcas

Los lenguajes de marcas pueden clasificarse de diferentes maneras, dependiendo del criterio utilizado. Una posible clasificación sería la siguiente:

Lenguajes de marcas estáticos: son aquellos lenguajes que se utilizan para describir documentos estáticos, que no cambian una vez que han sido creados. Ejemplos de lenguajes de marcas estáticos son HTML, XML, SGML y LaTeX.

Lenguajes de marcas dinámicos: son aquellos lenguajes que se utilizan para describir documentos dinámicos, que pueden cambiar y actualizarse de manera automática. Ejemplos de lenguajes de marcas dinámicos son JavaScript, PHP y ASP.

Lenguajes de marcas orientados a presentación: son aquellos lenguajes que se utilizan para describir la apariencia y el diseño de un documento, como pueden ser colores, fuentes, tamaños de letra, etc. Ejemplos de lenguajes de marcas orientados a presentación son CSS y XSL.

Lenguajes de marcas orientados a datos: son aquellos lenguajes que se utilizan para describir y estructurar datos de manera uniforme, de forma que puedan ser utilizados por diferentes aplicaciones y sistemas. Ejemplos de lenguajes de marcas orientados a datos son RDF y JSON.

Lenguajes de marcas más relevantes

Algunos de los lenguajes de marcas más relevantes y utilizados actualmente son:

HTML (HyperText Markup Language): es un lenguaje de marcas estático que se utiliza para describir la estructura y el contenido de una página web. HTML permite a los desarrolladores crear elementos como enlaces, imágenes, tablas, formularios, etc.

El ya mencionado anteriormente:

XML (eXtensible Markup Language): es un lenguaje de marcas estático que se utiliza para describir y estructurar datos de manera uniforme. XML permite a los desarrolladores crear sus propios elementos y atributos personalizados para describir la información contenida en un documento.

CSS (Cascading Style Sheets): es un lenguaje de marcas orientado a presentación que se utiliza para describir la apariencia y el diseño de una página web. CSS permite a los desarrolladores especificar colores, fuentes, tamaños de letra, etc. de manera uniforme para todos los elementos de una página web.

JavaScript: es un lenguaje de marcas dinámico que se utiliza para agregar interactividad y funcionalidades a una página web. JavaScript permite a los desarrolladores crear animaciones, validaciones de formularios, acciones al hacer clic en un botón, etc.

-Ámbitos de aplicación de los lenguajes de marcas

Los lenguajes de marcas tienen una amplia gama de aplicaciones en diferentes ámbitos, como pueden ser:

Desarrollo web: los lenguajes de marcas como HTML, XML, CSS y JavaScript son utilizados en la creación de páginas web, ya que permiten describir y estructurar el contenido de una página de manera precisa y detallada.

Almacenamiento y transporte de datos: los lenguajes de marcas como XML y JSON son utilizados en la creación de archivos que contienen datos estructurados, ya que permiten describir y organizar los datos de manera clara y uniforme. Estos archivos pueden ser utilizados para intercambiar información entre diferentes sistemas y aplicaciones.

Publicación electrónica: los lenguajes de marcas como HTML, XML y LaTeX son utilizados en la creación de libros, revistas, periódicos y otros documentos que se deben publicar en formato electrónico. Estos lenguajes permiten describir el contenido y la apariencia del documento de manera precisa y detallada.

Búsqueda y procesamiento de información: los lenguajes de marcas como XML y RDF son utilizados en la indexación y el procesamiento de información en diferentes bases de datos y sistemas de búsqueda. Estos lenguajes permiten describir y estructurar los datos de manera uniforme, lo que facilita su procesamiento por parte de las aplicaciones.

-Ámbitos específicos de aplicación de un lenguaje de marcas

Un ejemplo de un ámbito específico de aplicación de un lenguaje de marcas sería la publicación de libros electrónicos. En este caso, se puede utilizar un lenguaje de marcas como ePub, que es un estándar de la International Digital Publishing Forum (IDPF) que permite describir el contenido y la apariencia de un libro electrónico de manera uniforme.

ePub utiliza una combinación de HTML, CSS y JavaScript para describir el contenido del libro, como pueden ser capítulos, párrafos, imágenes, tablas, etc. Además, ePub permite a los desarrolladores especificar la apariencia del libro, como pueden ser colores, fuentes, tamaños de letra, etc.

Esto permite a los desarrolladores crear libros electrónicos que pueden ser leídos en diferentes dispositivos y plataformas, y que mantienen su apariencia y contenido de manera uniforme en todas ellas.

Ejercicio 1

El siguiente texto puede marcarse con XML de la siguiente manera:

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

<pedido>

<destinatario>

<nombre>Juan Delgado Martínez</nombre>

</destinatario>

<articulo>

<nombre>Bicicleta A2023</nombre>

</articulo>

<direccion>

<calle>Barco</calle>

<numero>4</numero>

<piso>3</piso>

<puerta>A</puerta>

</direccion>

<fecha>19-5-2000</fecha>

</pedido>

En este caso, se han creado elementos y atributos personalizados que permiten identificar y describir la información contenida en el texto de manera detallada. Esto permite realizar búsquedas de información según los campos especificados, como puede ser la búsqueda de pedidos realizados por un determinado destinatario, o la búsqueda de pedidos que contengan un determinado artículo.

Ejercicio 2

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

El siguiente documento XML representa la estructura y los datos que se muestran en el diagrama:

<persona>

<nombre>Maria</nombre>

<apellido>Gonzalez</apellido>

</persona>

En este caso, se ha creado un elemento "<persona>" que contiene dos elementos hijos: "<nombre>" y "<apellido>". Estos elementos contienen los valores "Maria" y "Gonzalez", respectivamente, que representan la información que se muestra en el diagrama.

Ejercicio 3

El siguiente documento XML representa la estructura y los datos que se muestran en el diagrama:

<?xml version="1.0"?>

<libro>

<capitulo>

En un lugar de

<negrita>

<negrita>la mancha</negrita>

</negrita>

de cuyo nombre

</capitulo>

</libro>

En este caso, se ha creado un elemento "<libro>" que contiene un elemento hijo "<capitulo>". Dentro de este elemento, se han creado dos elementos "<negrita>" que contienen el texto que se muestra en el diagrama. Estos elementos se han anidado de manera que se refleja la estructura que se muestra en el diagrama.

Ejercicio 4

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

<Carta desayunos>

<desayuno>

<nombre>Gofres Belgas</nombre>

<precio>5.95</precio>

<descripcion>Dos de nuestros famosos Gofres belgas con abundante sirope</descripcion>

<calorias>650</calorias>

</desayuno>

<desayuno>

<nombre>Gobres Belgas con fresas</nombre>

<precio>7.95</precio>

<descripcion>Ligeros gofres belgas cubiertos de fresas y nata montada</descripcion>

<calorias>900</calorias>

</desayuno>

<desayuno>

<nombre>Gofres Belgas con frutas del bosque</nombre>

<precio>8.95</precio>

<descripcion>Ligeros gofres belgas cubiertos con frutas del bosque y nata montada</descripcion>

<calorias>900</calorias>

</desayuno>

<desayuno>

<nombre>Tostada Francesa</nombre>

<precio>4.50</precio>

<descripcion>Dos gruesas rebanadas de nuestro pan francés</descripcion>

<calorias>600</calorias>

</desayuno>

<desayuno>

<nombre>Desayuno de la casa</nombre>

<precio>6.95</precio>

<descripcion>Dos huevos, bacon o salchicha, tostada y patatas fritas</descripcion>

<calorias>950</calorias>

</desayuno>

</Carta desayunos>