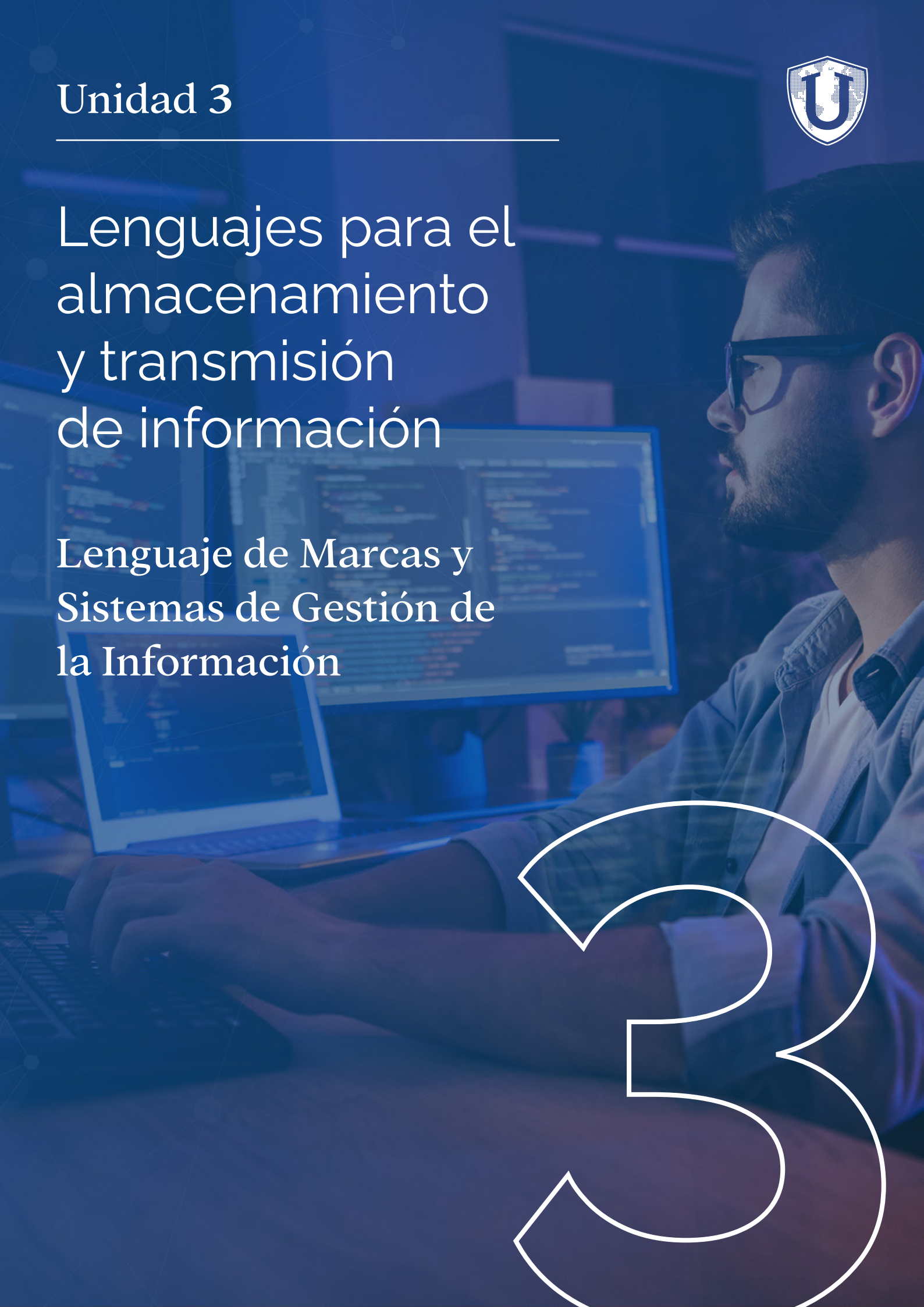
Unidad 3

# Lenguajes para el almacenamiento y transmisión de información

## Lenguaje de Marcas y Sistemas de Gestión de la Información



3

# Índice

# Introducción

En los últimos años, el uso de internet ha crecido increíblemente. La mayor parte de los usuarios navegan a través de internet intercambiando información los unos con los otros, y utilizan las páginas web, las cuales están hecha con lenguaje de marcas, facilitando así que todo fluya con normalidad y que el proceso sea transparente para esos usuarios.

Además, ciertos programas que instalamos en nuestro propio ordenador utilizan el lenguaje de marcas, tanto para configurar aplicaciones, almacenar información, etc. El lenguaje de marcado generalizado estándar SGML (standard generalized markup language) es el origen de estos lenguajes y está considerado como un metalenguaje.

Para la correcta visualización de documentos HTML, XML, etc. necesitamos de los navegadores web o browsers. Estos navegadores interpretan el código del lenguaje de marcas y nos presentan la página web como debería verse.

# Al finalizar esta unidad

- Nos habremos iniciado en el mundo de los lenguajes de marcas, identificando sus características más generales, reconociendo las ventajas que proporcionan, estudiando su clasificación y estructura e identificando los más relevantes.
- Conocerás el ámbito de aplicación de estos lenguajes y que, gracias a la inclusión de ejemplos reales de su uso, comprenderás la necesidad de su estudio.
- Habremos tratado de la forma más amena posible el contenido teórico, el cual nos ayudará en próximos capítulos como pilar a la hora de ir trabajando con los lenguajes de marcas.

# 3.1. Identificación de ámbitos de aplicación

Los lenguajes de marcas se pueden aplicar de maneras muy diversas, esto es debido a que posibilitan el intercambio de información entre aplicaciones de distinto tipo, sin importar las tecnologías usadas para crearlas ni el tipo de plataforma que utilicen.

Podemos generar vistas como WML, HTML o PDF desde un fichero XML. Para la especificación de datos de configuración, Java EE utiliza ficheros XML. Para la creación de servicios web, Visual Studio crea diferentes documentos que cuentan con una estructura XML. Esta estructura es utilizada tanto por Android Studio como por Windows Phone a la hora de guardar las vistas. Generalmente, XML es un formato que utilizan una gran cantidad de softwares, ya sea para guardar información o para su configuración. XML se puede utilizar tanto para frameworks de desarrollo, bases de datos, sistemas de publicación de contenido, etc.

# 3.2. XML: estructura y sintaxis

Las siglas de XML significan extensible markup language, en español lenguaje de marcas extensibles. Este lenguaje es considerado como un metalenguaje y surgió como método de resolución para aquellos problemas planteados por HTML a la hora de que muestren el significado de sus datos las etiquetas. Entre las ventajas de XML, una de ellas es que no es necesario contar con conocimientos de programación para generar un documento simple. Lo importante está en los intercambios de datos entre diferentes programas de forma segura, lo cual facilita la creación de etiquetas propias, reutilizar el contenido y la presentación de un diseño y estructuras independientes.

Más adelante veremos la estructura y sintaxis en detalle, pero ahora veremos un pequeño adelanto para poder familiarizarnos con la terminología.

Los documentos XML cuentan con un tipo de estructura jerárquica en árbol y cuentan con dos partes bien diferenciadas: el prólogo, que es la parte principal del documento y que puede estar formada por varias líneas o solo una, y el cuerpo.

En el ejemplo que vemos a continuación podemos observar que el prólogo cuenta con la primera línea, en la cual la versión de XML está declarada, además de la codificación que necesitamos utilizar para la representación de caracteres. Por otro lado, en tenemos el cuerpo, en el que encontramos la etiqueta <restaurante>, donde podemos encontrar los diferentes platos del establecimiento. En este ejemplo, podemos ver que se han añadido el plato junto con información relativa a sus ingredientes, su nombre y el tiempo de preparación.

```
<?xml version=”1.0” encoding”’iso-8859-“’?>
<restaurante>
     <plato>
     <ingredientes> Tomate, queso mozzarella y
     orégano</ingredientes>
     <tiempo> 30 minutos </tiempo>
     <nombre> Pizza margarita </nombre>
</plato>
</restaurante>
```

**RECUERDA**

ISO 8859-1 permite utilizar la letra ñ y los acentos.

Para la escritura de las etiquetas, hay que utilizar los signos (<) y (>) para encerrarlas, y suelen contar con una etiqueta de apertura y otra para el cierre. Asimismo, encontramos etiquetas vacías. En el siguiente ejemplo podemos ver una etiqueta <email> que cuenta con etiqueta de apertura y de cierre, y luego tenemos otra etiqueta vacía en la cual encontramos el elemento casado, cuyo atributo v tiene el valor “no”. Estos atributos aportan propiedades a los elementos y se encuentran integrados en la etiqueta de apertura.

```
<email>emaildeprueba@prueba.com</email>
<casado v=”no” />
```

# 3.3. Herramientas de edición

A continuación, se explican algunas de las herramientas básicas para editar páginas web, además de otro tipo de herramientas utilizadas para crear documentos XML. Como en otros softwares, encontramos herramientas offline, las cuales hay que instalar en nuestro ordenador para utilizarlas, y las online, que con tener acceso a internet y un navegador web podemos utilizarlas.

## 3.3.1. Bloc de notas Windows

Primero tenemos el Bloc de notas de Windows o también conocido como Notepad, este programa viene instalado en Windows y es un editor de texto plano. Los archivos creados con este programa tienen la extensión .txt y con él podemos codificar caracteres en UTF-8, Unicode y ANSI.

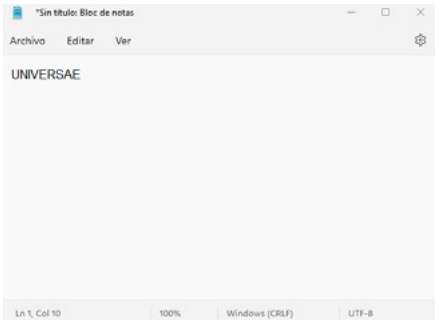



Imagen 1. Bloc de notas de Windows.

## 3.3.2. Gedit

Otro editor de texto es Gedit, este también se puede encontrar para Windows, pero, además, está hecho para el entorno de escritorio GNOME. Este programa suele utilizarse simplemente como un bloc de notas, así como entorno para el desarrollo. A parte de las características propias de los editores de texto plano, como pueden ser guardar, copiar, imprimir, etc. este editor también nos aporta facilidades como la numeración de las líneas de código, así como introducir colores asociados a los distintos elementos del programa.

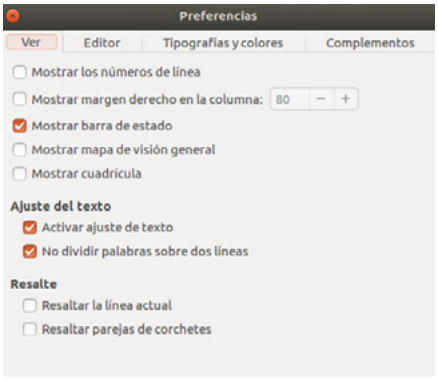



Imagen 2. Cómo activar el número de líneas en Gedit de Ubuntu.

## 3.3.3. Notepad++

Notepad++ es gratuito, tiene licencia GPL y está para Windows. Fue escrito en C++ y utilizando APIWin32. Con este editor podemos utilizar una gran variedad de lenguajes. Dependiendo del lenguaje utilizado, nos organiza el texto y aporta características como colorear el texto en función de la sintaxis del lenguaje, numerar la líneas, etc.

Destacan las siguientes características: autocompletado, buscar y reemplazar, ampliación y reducción de texto y resaltar la sintaxis.

# 3.4. Elaboración de documentos XML bien formados

Si queremos crear un documento XML correctamente, debemos respetar una serie de reglas:

- Por recomendación del W3C, la primera línea debe de ser donde indiquemos el tipo de versión con el que trabajaremos, es decir, debe ser la instrucción de procesamiento.
- La estructura de los documentos debe de ser jerárquica, con solo un elemento principal o raíz, el cual deberá ser el primero en abrir y último en cerrar.
- Debemos anidar de forma correcta las diferentes etiquetas, es decir, que se abran y se cierren en el orden correcto. Y todos aquellos elementos que no sean vacíos, deben de contar con sus correspondientes etiquetas de apertura y cierre.
- Los valores de los atributos se deben de entrecomillar, ya sea mediante comillas simples o dobles, y además no pueden existir dos atributos con el mismo nombre dentro del mismo elemento.
- Los documentos XML hacen distinción entre mayúsculas y minúsculas, por lo que, a la hora de escribir etiquetas, atributos y elementos, debemos tener cuidado con esto. Por ejemplo, una etiqueta escrita como <etiqueta> no se corresponderá con una etiqueta escrita como </Etiqueta>.
- Cuando escribamos elementos, atributos y etiquetas, debemos de fijarnos en que estos sigan una serie de reglas, ya que estos no pueden empezar por un número ni por la palabra XML. Y tampoco se pueden utilizar caracteres especiales reservados como (<) o (*).
- No podemos escribir los comentarios dentro de etiquetas.

Este sería un ejemplo de documento XML escrito correctamente:

```
<?xml version="1.0" encoding="utf-8" ?>
<ejemplo>
    <!-- vamos a incorporar a un usuario-->
    <usuario>
        <nombre> Pedro </nombre>
        <municipio cp="30009">Murcia</municipio>
        <tlf n="123456789"/>
        <correo> ejemplo@ejemplo.com</correo>
    </usuario>
</ejemplo>
```

Mediante herramientas específicas y un simple navegador, podemos verificar todo lo anteriormente visto en este apartado. En caso de querer verificar que un documento XML tiene una estructura correcta y no tiene errores, podemos hacerlo crearlo dentro de un editor de texto plano, luego se guarda con la extensión de XML y lo abrimos en un navegador. En caso de que se muestre la estructura jerárquica arborescente, significa que ese documento se ha creado correctamente.

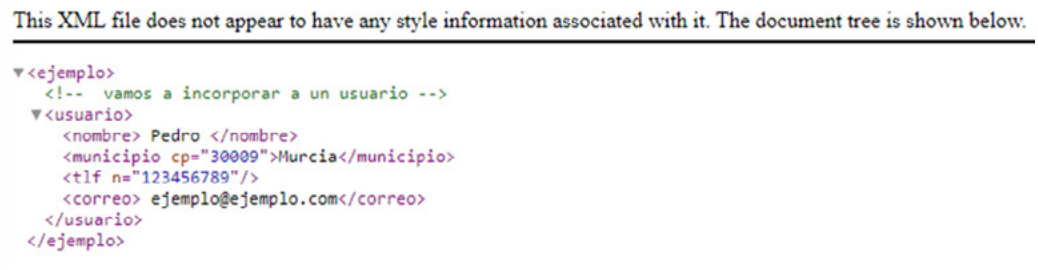



Imagen 3. Documento XML visualizado en navegador

Al documento anterior, le vamos a aplicar una serie de modificaciones, para ver el resultado que obtendríamos:

```
<?xml version=”1.0” encoding=”utf-8” ?>
<ejemplo>
        <!-- vamos a incorporar a un usuario-->
        <usuario>
                <nombre> Pedro </nombre>
                <municipio cp=”30009”>Murcia</Municipio>
                <tlf n=”123456789”/>
                <5correo> ejemplo@ejemplo.com</5correo>
        </usuario>
</ejemplo>
```

Como podemos ver, en el navegador se nos informa de que hay un error en la línea 6, ya que la etiqueta <municipio> no coincide con su correspondiente etiqueta </Municipio>.

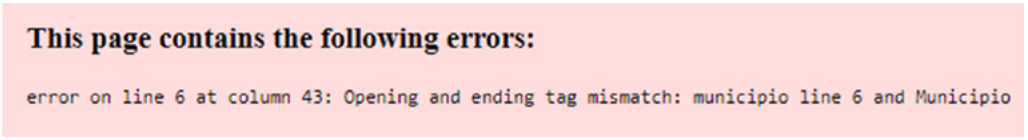



Imagen 4. Error mostrado en navegador al abrir el documento XML

Una vez arreglamos este error, volvemos a comprobar el documento en el navegador, y nos salta el siguiente error:


Imagen 5. Error mostrado en navegador al abrir el documento XML

Esto es debido a que nuestra etiqueta <5correo> comienza por un número, y esto, como hemos visto anteriormente, no es posible. Una vez corregido esto, obtenemos el resultado deseado.

También podemos usar XML Copy Editor para comprobar si nuestro documento está escrito correctamente.

# 3.5. Utilización de espacios de nombres en XML

Para evitar ambigüedades, usamos los espacios de nombres, estos mecanismos nos permiten diferenciar entre etiquetas que tengan la misma denominación al contar con varios documentos XML. La declaración es la siguiente:

```
<elemento xmlns:prefijo="URI">
```

Para utilizarlo, antepondremos el prefijo a aquellos atributos y elementos que pertenezcan a etiquetas ambiguas:

```
<prefijo:etiqueta> </prefijo:etiqueta>
```

En el siguiente ejemplo contamos con dos documentos XML diferentes en los que encontramos el elemento caña, pero con significados y estructuras diferentes. En el primero se refiere a una caña de pescar, y en el segundo ejemplo se refiere a una caña de azúcar.

### Documento XML 1:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<caña>
        <material>grafeno</material>
        <color>verde</color>
</caña>
```

### Documento XML 2:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<caña>
        <origen>Jamaica</origen>
        <tipo> Blanco </tipo>
</caña>
```

En caso de que queramos usar ambos elementos <caña> en un mismo documento, deberemos de establecer prefijos, en este caso c1 y c2:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<c1:ejemplo xmlns:c1="http…documentoXML1"
                xmlns:c2="http…documentoXML2">
        <c1:caña>
                <c1:material> grafeno</c1:material>
                <c1:color> verde </c1:color>
        </c1:caña>
        <c2:caña>
                <c2:origen> Jamaica</c2:origen>
                <c2:tipo> Blanco </c2:tipo>
        </c2:caña>
</c1:ejemplo>
```

www.universae.com