

# **Aspectos básicos de los lenguajes de marcas y sistemas de gestión empresarial.**

---



## **Caso práctico**

María y Félix son los fundadores y propietarios de una asesoría legal y empresarial, que tiene su sede en Cantabria, con oficinas en los municipios más importantes de la región.

María, licenciada en Derecho, ejercía como abogada especializada en derecho laboral y representaba a alguna empresa, además de particulares en su propio despacho situado en Torrelavega. Tenía una red informática cliente-servidor sobre un sistema operativo Windows y trabajaba con una base de datos de documentos jurídicos.

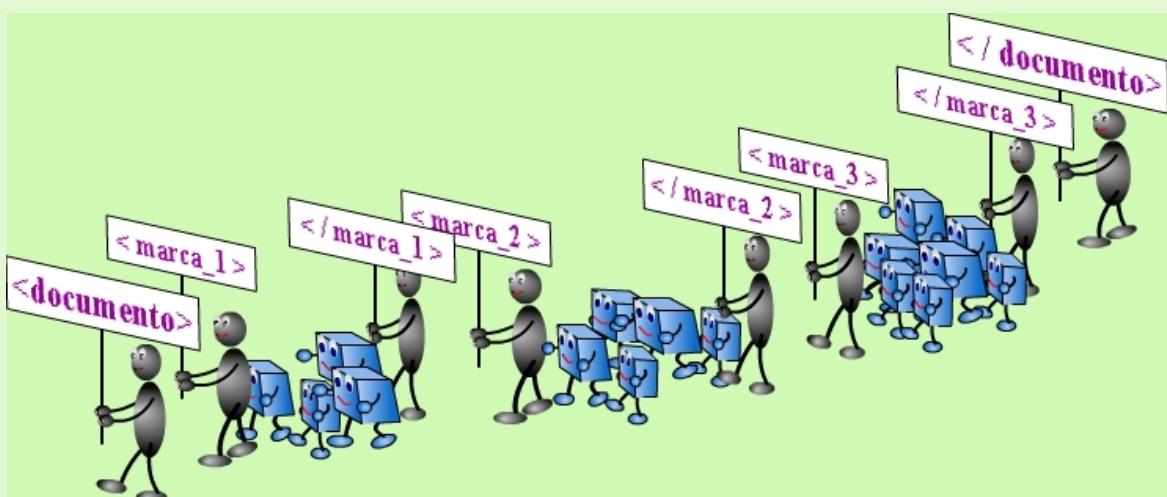
Félix, diplomado en Ciencias Empresariales, había creado una asesoría empresarial, ubicada en Santander, que básicamente, se encargaba de la contabilidad de varias  PYMES. También tenía una red cliente-servidor, pero ésta bajo un sistema Linux con software libre de contabilidad.

Ambos eran amigos y un día en que habían estado hablando de sus respectivos trabajos, decidieron que sus ingresos podían aumentar sustancialmente si, además de mantener sus respectivas carteras de clientes, se unían y formaban una sociedad que ofreciese a las empresas asesoría legal y empresarial de forma conjunta.

Desde el principio, la idea de asociarse fue un éxito. Al cabo de dos años el volumen del negocio se había extendido y se hizo imprescindible el intercambio de comunicación entre ambos.

Dado que trabajaban con sistemas informáticos diferentes se planteaba el problema de cómo podían compartir información sobre los clientes comunes manteniendo la infraestructura informática con la que trabaja cada uno.

Consultaron el problema a Juan, un técnico en administración de sistemas informáticos en red, y éste les dijo que no había ningún problema de interconexión si los ficheros que manejaban se ajustaban a un formato estándar conocido como  XML. Según lo que Juan les dijo, generar documentos con dicho estándar apenas requiere conocimientos previos de informática, por tanto era una solución que parecía perfecta para su problema.



Ana Polo Arozamena. Elementos ([CC BY-NC-ND](#))



# 1.- Definición y clasificación de lenguajes de marcas.

Los "lenguajes de marcas" (o lenguajes de marcado) sirven para **codificar un documento** donde, junto con el texto, se **incorporan etiquetas**, marcas o anotaciones **con información adicional** relativa a la estructura del texto o su forma de presentarlo. El lenguaje de marcas es el que especifica cuáles serán las etiquetas posibles, donde deben colocarse y el significado que tendrá cada una de ellas. Permiten hacer explícita la estructura de un documento, su contenido semántico o cualquier otra información lingüística o extralingüística que se quiera hacer patente.

Todo lenguaje de marcas está definido en un documento denominado  DTD. En él se establecen las marcas, los elementos utilizados por dicho lenguaje y sus correspondientes etiquetas y atributos, su sintaxis y normas de uso.



## Ejemplo

Aspecto de un documento realizado con un lenguaje de marcas:

```
<noticia>
    <fecha>25/09/2015</fecha>
    <lugar>Almería</lugar>
    <titular>Inicio del curso 2015/2016</titular>
    <desarrollo>El centro de estudios presenta un nuevo curso escolar...</c
</noticia>
```

Aunque en la práctica, en un mismo documento pueden combinarse varios tipos diferentes de lenguajes de marcas, éstos se pueden clasificar en tres grupos:

- ✓ **Orientados a presentación:** Son los utilizados generalmente por los procesadores de texto y codifican cómo ha de presentarse el documento, es decir, define el formato del texto. Por ejemplo: indicando que una determinada palabra se debe presentar en negrita, o que se debe dejar un espacio entre caracteres determinado. Generalmente, las marcas se ocultan al usuario, lo que permite obtener un efecto  WYSIWYG.
- ✓ **De procedimientos:** Orientados también a la presentación pero, en este caso, dentro de un marco procedural que permite definir macros (secuencias de acciones), es decir, el programa que representa el documento debe interpretar el código en el mismo orden en que aparece. Entre los ejemplos más comunes, encontramos TeX, LaTeX y Postscript.
- ✓ **Descriptivos o semánticos:** Este tipo no define qué se debe hacer con un trozo o sección del documento, sino que por el contrario, las marcas sirven para indicar qué es esa información, es decir, describen qué es lo que se está representando. Describen las diferentes partes pero sin especificar cómo deben representarse.

Algunos ejemplos de lenguajes de marcado agrupados por su ámbito de utilización son:

- ✓ Documentación electrónica
  - ◆ **RTF** (Rich Text Format): Formato de Texto Enriquecido, fue desarrollado por Microsoft en 1987. Permite el intercambio de documentos de texto entre distintos procesadores de texto.
  - ◆ **TeX**: Su objetivo es la creación de ecuaciones matemáticas complejas.
  - ◆ **Wikitexto**: Permite la creación de páginas wiki en servidores preparados para soportar este lenguaje.
  - ◆ **DocBook**: Permite generar documentos separando la estructura lógica del documento de su formato. De este modo, dichos documentos, pueden publicarse en diferentes formatos sin necesidad de realizar modificaciones en el documento original.
- ✓ Tecnologías de internet
  - ◆ **HTML, XHTML**: (Hypertext Markup Language, eXtensible Hypertext Markup Language): Su objetivo es la creación de páginas web.
  - ◆ **RSS**: Permite la difusión de contenidos web
- ✓ Otros lenguajes especializados
  - ◆ **MathML** (Mathematical Markup Language): Su objetivo es expresar el formalismo matemático de tal modo que pueda ser entendido por distintos sistemas y aplicaciones.
  - ◆ **VoiceXML** (Voice Extended Markup Language) tiene como objetivo el intercambio de información entre un usuario y una aplicación con capacidad de reconocimiento de habla.
  - ◆ **MusicXML**: Permite el intercambio de partituras entre distintos editores de partituras.

# Autoevaluación

**Los lenguajes de marcas se utilizan para:**

- Dar formato a los documentos de texto.
- Definir la estructura de los datos de un documento.
- Permitir el intercambio de ficheros entre diferentes aplicaciones y plataformas.
- Todas las anteriores.

## 2.- Evolución de los lenguajes de marcas.

A finales de los años 60, para poder introducir anotaciones dentro de documentos electrónicos, surgen unos lenguajes informáticos, distintos de los lenguajes de programación, orientados a la gestión de información. Con el desarrollo de los editores y procesadores de texto surgen los primeros lenguajes informáticos especializados en tareas de descripción y estructuración de información: **los lenguajes de marcas**. Paralelamente, también, surgen otros lenguajes informáticos orientados a la representación, almacenamiento y consulta eficiente de grandes cantidades de datos: lenguajes y sistemas de bases de datos.

Los lenguajes de marcas surgieron, inicialmente, como lenguajes formados por el conjunto de códigos de formato que los procesadores de texto introducen en los documentos para dirigir el proceso de presentación ( impresión) mediante una impresora. Como en el caso de los lenguajes de programación, inicialmente estos códigos de formato estaban ligados a las características de una máquina, programa o procesador de textos concreto y, en ellos, inicialmente no había nada que permitiese al programador (formateador de documentos en este caso) abstraerse de las características del procesador de textos y expresar de forma independiente a éste la estructura y la lógica interna del documento.



### Ejemplo

Código de marcas anterior a GML. Las etiquetas son de invención propia.

Dado el siguiente documento:

```
<times 14><color verde><centrado> Este texto es un ejemplo para mostrar la util  
<color granate><times 10><cursiva>Para realizar este ejemplo se utilizan etiqu  
Las partes importantes del texto pueden resaltarse usando la  
<negrita>negrita</negrita>, o el <subrayar>subrayado</subrayar></times 10></color>
```

Al imprimirlo se obtendría:

**Este texto es un ejemplo para mostrar la utilización primitiva de las  
marcas**

*Para realizar este ejemplo se utilizan etiquetas de nuestra invención. Las partes importantes del texto  
pueden resaltarse usando la **negrita**, o el **subrayado***

Posteriormente, se añadieron como medio de presentación a la pantalla. Los códigos de estilo de visualización anteriores ya no aparecen, y se emplean otros medios para marcados, diferentes a la inclusión a mano de cadenas formateadoras, por lo que ahora ese proceso se automatiza y es suficiente con pulsar una combinación de teclas, o un botón, para lograr los resultados requeridos. Aunque esto es sólo una abstracción, para su uso interno, las aplicaciones siguen utilizando marcas para delimitar aquellas partes del texto que tienen un formato especial.

Este marcado estaba exclusivamente orientado a la presentación de la información, aunque pronto fueron conscientes de las posibilidades del marcado y se le dieron nuevos usos que resolverían una gran variedad de necesidades, apareció el formato generalizado.

## 2.1.- El origen: GML y SGML.

Uno de los problemas que se conocen desde hace décadas en la informática es la **falta de estandarización** en los formatos de información usados por los distintos programas.

Para resolver este problema, en los años sesenta IBM encargó a Charles F. Goldfarb la construcción de un sistema de edición, almacenamiento y búsqueda de documentos legales. Tras analizar el funcionamiento de la empresa llegaron a la conclusión de que para realizar un buen procesado informático de los documentos había que establecer un formato estándar para todos los documentos que se manejaban en la empresa. Con ello se lograba gestionar cualquier documento en cualquier departamento y con cualquier aplicación, sin tener en cuenta dónde ni con qué se generó el documento. Dicho formato tenía que ser válido para los distintos tipos de documentos legales que utilizaba la empresa, por tanto, debía ser flexible para que se pudiera ajustar a las distintas situaciones.

El formato de documentos que se creó como resultado de este trabajo fue  **GML**, cuyo objetivo era describir los documentos de tal modo que el resultado fuese independiente de la plataforma y la aplicación utilizada.

El formato GML evolucionó hasta que en 1986 dio lugar al estándar ISO 8879 que se denominó  **SGML**. Éste era un lenguaje muy complejo y requería de unas herramientas de software caras. Por ello su uso ha quedado relegado a grandes aplicaciones industriales.



### Ejemplo

Documento SGML sencillo:

```
<email>
    <remitente>
        <nombre>Peter</nombre>
        <apellido>Pan</apellido>
    </remitente>
    <destinatario>
        <direccion>campanilla@paisdenuncajamas.com</direccion>
    </destinatario>
    <asunto>Paseo</asunto>
    <mensaje>¿Te apetece dar una vuelta?</mensaje>
</email>
```

## 2.2.- La popularización: HTML.

---

En 1989/90 Tim Berners-Lee creó el World Wide Web y conociendo SGML, se encontró con la **necesidad de organizar, enlazar y compatibilizar gran cantidad de información procedente de diversos sistemas**. Para resolverlo, a partir de la sintaxis SGML, creó un lenguaje de descripción de documentos llamado  **HTML**, siendo una combinación de dos estándares ya existentes:

- ✓  ASCII: Es el formato que cualquier procesador de textos sencillo puede reconocer y almacenar. Por tanto es un formato que permite la transferencia de datos entre diferentes ordenadores.
- ✓ SGML: Lenguaje que permite dar estructura al texto, resaltando los títulos o aplicando diversos formatos al texto.

**HTML es una versión simplificada de SGML**, ya que sólo se utilizaban las instrucciones absolutamente imprescindibles. Era tan fácil de comprender que rápidamente tuvo gran aceptación logrando lo que no pudo SGML, siendo un rotundo éxito en la World Wide Web. **HTML se convirtió en un estándar general para la creación de páginas web**. Además, tanto las herramientas de software como los navegadores que permiten visualizar páginas HTML son cada vez mejores.

El HTML es hoy día el tipo de documento más empleado en el mundo. Su sencillez era tal que cualquier persona podía escribir documentos en este formato, sin apenas necesidad de conocimientos de informática. Esta fue una de las razones de su éxito, pero también condujo a un cierto caos. El crecimiento exponencial de la web en los años 90 produjo documentos en cantidades ingentes pero mal estructurados, problema agravado aún más por la falta de respeto por los estándares, por parte de diseñadores web y fabricantes de software.



# Ejemplo

Documento HTML:

```
<html>
  <head>
    <title>Ejemplo de código HTML</title>
  </head>
  <body bgcolor="#ffffff">
    <p><b>1 de Octubre de 2015</b></p>
    <p><b>Bienvenido al modulo de "Lenguajes de Marcas y Sistemas de Gestión de Información"</b></p>
    <p>En este curso aprenderás, entre otras cosas:<br/>
      <ul>
        <li>Las ventajas que ofrece XML </li>
        <li>La creación de documentos bien formados </li>
        <li>La creación de DTD</li>
      </ul>
    </p>
  </body>
</html>
```

Al publicarlo en un navegador, obtendríamos el siguiente resultado:

**1 de Octubre de 2015**

**Bienvenido al modulo de "Lenguajes de Marcas y Sistemas de Gestión de Información"**

En este curso aprenderás, entre otras cosas:

- Las ventajas que ofrece XML
- La creación de documentos bien formados
- La creación de DTD

## 2.3.- La madurez: XML.

Como respuesta a problemas surgidos en torno al HTML, el  W3C establece, en 1998, el estándar internacional  XML, un lenguaje de marcas puramente estructural que **no incluye ninguna información relativa al diseño**, que permite crear etiquetas adaptadas a las necesidades (de ahí lo de "extensible"). Está convirtiéndose con rapidez en estándar para el intercambio de datos en la Web. A diferencia de HTML las etiquetas indican el significado de los datos en lugar del formato con el que se van a visualizar los datos.

XML es un  metalenguaje **caracterizado** por:

- ✓ Permitir definir etiquetas propias.
- ✓ Permitir asignar atributos a las etiquetas.
- ✓ Utilizar un esquema para definir de forma exacta las etiquetas y los atributos.
- ✓ La estructura y el diseño son independientes.

En realidad XML es un conjunto de estándares relacionados entre sí y que son:

- ✓ **XSL**, eXtensible Style Language. Permite definir hojas de estilo para los documentos XML e incluye capacidad para la transformación de documentos.
- ✓ **XML Linking Language**, incluye Xpath, Xlink y Xpointer. Determinan aspectos sobre los enlaces entre documentos XML.
- ✓ **XML Namespaces**. Proveen un contexto al que se aplican las marcas de un documento de XML y que sirve para diferenciarlas de otras con idéntico nombre válidas en otros contextos.
- ✓ **XML Schemas**. Permiten definir restricciones que se aplicarán a un documento XML. Actualmente los más usados son las DTD.



### Ejemplo

Documento XML:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE biblioteca>
<biblioteca>
    <ejemplar tipo="libro" isbn="978-2-7460-4958-1" edicion="1" paginas="320">
        <titulo>XML práctico</titulo>
        <editorial>Ediciones Eni</editorial>
        <autor>Sebastien Lecomte</autor>
        <autor>Thierry Boulanger</autor>
        <autor funcion="traductor">Ángel Belinchon Calleja</autor>
        <prestamos>
            <lector socio="0223" inicio="13/05/2014" devolucion="15/05/2014">
                <lector socio="0065" inicio="13/07/2015" devolucion="15/07/2015">
            </prestamos>
        </ejemplar>
        <ejemplar tipo="revista" paginas="51" fecha="01/11/2009">
            <titulo>Todo Linux 101. Virtualización en GNU/Linux</titulo>
            <editorial>Studio Press</editorial>
            <autor>Teresa López</autor>
            <autor>Andrés Lozano</autor>
            <prestamos>
                <lector socio="0211" inicio="11/05/2015" devolucion="15/05/2015">
            </prestamos>
        </ejemplar>
    </biblioteca>
```

Al publicarlo en un navegador, obtendríamos el siguiente resultado:

```
- <biblioteca>
- <ejemplar tipo="libro" isbn="978-2-7460-4958-1" edicion="1" paginas="347" fecha="01/04/2011">
  <titulo>XML practico</titulo>
  <editorial>Ediciones Eni</editorial>
  <autor>Sebastien Lecomte</autor>
  <autor>Thierry Boulanger</autor>
  <autor funcion="traductor">Ángel Belinchon Calleja</autor>
- <prestamos>
  <lector socio="0223" inicio="13/05/2014" devolucion="15/05/2014">Pedro López</lector>
  <lector socio="0065" inicio="13/07/2015" devolucion="15/07/2015">Ali Méndez</lector>
</prestamos>
</ejemplar>
- <ejemplar tipo="revista" paginas="51" fecha="01/11/2009">
  <titulo>Todo Linux 101. Virtualización en GNU/Linux</titulo>
  <editorial>Studio Press</editorial>
  <autor>Teresa López</autor>
  <autor>Andrés Lozano</autor>
- <prestamos>
  <lector socio="0211" inicio="11/05/2015" devolucion="15/05/2015">Víctor Rodríguez</lector>
</prestamos>
</ejemplar>
</biblioteca>
```

Víctor Gómez. Resultado del ejemplo en un navegador ([CC0](#))

## 2.4.- Comparativa XML-SGML.

### Comparativa XML - SGML

XML	SGML
✓ Su uso es sencillo.	✓ Su uso es muy complejo.
✓ Trabaja con documentos bien formados, no exige que estén validados.	✓ Sólo trabaja con documentos válidos.
✓ Facilita el desarrollo de aplicaciones de bajo coste.	✓ Su complejidad hace que las aplicaciones informáticas para procesar SGML sean muy costosas.
✓ Es muy utilizado en informática y en más áreas de aplicación.	✓ Sólo se utiliza en sectores muy específicos.
✓ Compatibilidad e integración con HTML.	✓ No hay una compatibilidad con HTML definida.
✓ Formateo y estilos fáciles de aplicar.	✓ Formateo y estilos relativamente complejos.
✓ No usa etiquetas opcionales.	

### Autoevaluación

¿Cuáles son las características comunes de XML y SGML?

- Guardan el formato de un documento.
- Guardan la estructura lógica de los documentos.
- Guardan los documentos en el formato universal txt.
- Guardan el formato de los documentos independientemente de la plataforma.



## Para saber más

La recomendación de XML publicada por el W3C es pública y accesible en:

<http://www.w3.org/TR/REC-xml/>

## 2.5.- Comparativa XML-HTML.

### Comparativa XML-HTML

XML	HTML
✓ Es un perfil de SGML.	✓ Es una aplicación de SGML.
✓ Especifica cómo deben definirse conjuntos de etiquetas aplicables a un tipo de documento.	✓ Aplica un conjunto limitado de etiquetas sobre un único tipo de documento.
✓ Modelo de  hiperenlaces complejo.	✓ Modelo de hiperenlaces simple.
✓ El navegador es una plataforma para el desarrollo de aplicaciones.	✓ El navegador es un visor de páginas.
✓ Fin de la guerra de los navegadores y etiquetas propietarias.	✓ El problema de la 'no compatibilidad' y las diferencias entre navegadores ha alcanzado un punto en el que la solución es difícil.



### Ejemplo

#### Fichero XML.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE libro>
<libro>
    <titulo>XML practico </titulo>
    <autor>SebastienLecomte</autor>
    <autor>Thierry Boulanger</autor>
    <editorial>Ediciones Eni</editorial>
    <isbn>978-2-7460-4958-1</isbn>
    <edicion>1</edicion>
    <paginas>347</paginas>
</libro>
```

Al interpretar este fichero XML con un navegador, se obtendría:

– <b><libro></libro></b>

```
<b><libro></libro></b>
<titulo>XML practico </titulo>
<autor>SebastienLecomte</autor>
<autor>Thierry Boulanger</autor>
<editorial>Ediciones Eni</editorial>
<isbn>978-2-7460-4958-1</isbn>
<edicion>1</edicion>
<paginas>347</paginas>
</libro>
```

Víctor Gómez. Resultado del ejemplo en un navegador ([CC0](#))

**Fichero HTML.**

```
<html>
  <head>
    <title>Libro</title>
  </head>
  <body>
    <h3>XML práctico</h3><br>
    <p>Autores: Sebastien Lecomte, Thierry Boulanger</p>
    <ul>
      <li>Editorial: Ediciones Eni</li>
      <li>ISBN: 978-2-7460-4958-1</li>
      <li>Edición: 1 </li>
      <li>Páginas: 347</li>
    </ul>
  </body>
</html>
```

Al interpretar este fichero HTML con un navegador, se obtendría:

## XML práctico

Autores: Sebastien Lecomte, Thierry Boulanger

- Editorial: Ediciones Eni
- ISBN: 978-2-7460-4958-1
- Edición: 1
- Páginas: 347

Víctor Gómez. Resultado del ejemplo en un navegador ([CC0](#))

### 3.- Etiquetas, elementos y atributos.

Los lenguajes de marcas utilizan una serie de etiquetas especiales intercaladas en un documento de texto sin formato. Dichas etiquetas serán posteriormente interpretadas por los intérpretes del lenguaje y ayudan al procesado del documento.

Existen tres términos comúnmente usados para describir las partes de un documento de lenguajes de marcas: **etiquetas**, **elementos** y **atributos**.

Una **etiqueta** (tag) es que se escriben encerradas entre símbolos menor que (<) y mayor que (>). Normalmente, se utilizan dos etiquetas: una de inicio y otra de fin para indicar que ha terminado el efecto que queríamos presentar. La única diferencia entre ambas es que la de cierre lleva una barra inclinada "/" antes del código.

texto que heredará las características de la etiqueta

Los **elementos** representan estructuras mediante las que se organizará el contenido del documento o acciones que se desencadenan cuando el navegador interpreta lo interpreta. Constan de la etiqueta de inicio, la etiqueta de fin y de todo aquello que se encuentre entre ambas.

Algunos elementos no tienen contenido, y se denominan elementos vacíos (no deben llevar etiqueta de fin).

Un **atributo** es un par nombre-valor, que se encuentra dentro de la etiqueta de inicio de un elemento e indican diferentes propiedades que pueden llevar asociados los elementos.



#### Ejemplo

Por ejemplo, en HTML

Hola a todos. **Este texto se ve en negrita.**

Al interpretarlo en un navegador se verá así (Con el fondo de la página en amarillo debido al atributo *bgcolor*):

**Hola a todos. Este texto se ve en negrita.**

Víctor Gómez. Resultado del ejemplo en un navegador ([CC0](#))

# Autoevaluación

¿Cuál de las siguientes líneas es correcta?

- Texto en cursiva*
- Texto en cursiva*
- Texto en cursiva*
- Texto en cursiva*

## 4.- Herramientas de edición.

---

Para trabajar en XML es necesario editar los documentos y luego procesarlos, por tanto tenemos dos tipos de herramientas:

### ✓ Editores XML

Una característica de los lenguajes de marcas es que se basan en la utilización de ficheros de texto plano por lo que basta utilizar un editor de texto para construir un documento XML, es importante que el editor no permita formatear el texto, para que no introduzca "código basura" en nuestro documento.

Para crear documentos XML complejos e ir añadiendo datos es conveniente usar algún software de edición XML. Estos nos ayudan a crear estructuras y etiquetas de los elementos usados en los documentos, colorean las etiquetas para diferenciarlas más cómodamente y además algunos incluyen ayuda para la creación de otros elementos como DTD, hojas de estilo CSS o XSL, ... El W3C ha desarrollado un editor de HTML, XHTML, CSS y XML gratuito cuyo nombre es Amaya, pero existen otros gratuitos como son: *Notepad++*, *Sublime Text*, *Netbeans*, etc...

### ✓ Procesadores XML

Para interpretar el código XML se puede utilizar cualquier navegador. Los procesadores de XML permiten leer los documentos XML y acceder a su contenido y estructura. Un procesador es un conjunto de módulos de software entre los que se encuentra un parser o analizador de XML que comprueba que el documento cumple las normas establecidas para que pueda abrirse. Estas normas pueden corresponderse con las necesarias para trabajar sólo con documentos de tipo válido o sólo exigir que el documento esté bien formado, primeros se conocen como validadores y los segundos como no validadores. El modo en que los procesadores deben leer los datos XML está descrito en la recomendación de XML establecida por W3C.

Para publicar un documento XML en Internet se utilizan los procesadores XSLT, que permiten generar archivos HTML a partir de documentos XML.

Puesto que XML se puede utilizar para el intercambio de datos entre aplicaciones, hay que recurrir a motores independientes que se ejecutan sin que nos demos cuenta. Entre estos destacan "XML para Java" de IBM, JAXP de Sun, etc

# Autoevaluación

**Para crear documentos XML es necesario:**

- Software especializado para la tecnología XML.
- Herramientas de validación de XML.
- Un bloc de notas y un navegador.
- Al menos, un editor XML.

## 5.- XML

---



### Caso real

Suponemos dos librerías (A y B) y una Editorial que provee libros. Cada una de estas librerías tiene un software diferente para gestionar sus libros, pero la finalidad es la misma, que es almacenar y gestionar sus libros, pero cada una de ellas está introduciendo "a mano" todos los datos de cada uno de ellos, con la consiguiente pérdida de tiempo, ya que son los mismos datos...

Si la editorial ya dispone de los datos principales de los libros que provee, ¿por qué volver a introducir esa información en cada librería?. Lo más lógico es que la editorial les entregara un fichero o les indicara un lugar donde importar y actualizar esos datos, de modo que todas las librerías tendrían siempre la información actualizada instantáneamente.

El problema radica en que el software que posee cada librería es completamente diferente, por lo que sería necesario convertir los datos que proporciona la editorial de forma individual para cada librería, debido a que estamos ante distintos sistemas que organizan la misma información.

La solución es sencilla y consiste en almacenar los datos de una forma tan simple, que cualquier programa pueda interpretarla: XML.

Por ejemplo, para almacenar una información típica de un libro:

- ✓ Título: Don Quijote de la Mancha
- ✓ Autor: Miguel de Cervantes
- ✓ Género: Burlesque
- ✓ Editorial: Francisco de Robles
- ✓ Personajes: Sancho Panza, Dulcinea del Toboso, Rocinante.
- ✓ ISBN: 999-9-99-999999-9

vamos a utilizar etiquetas que ilustren su contenido y así identificar cada elemento. Hay que recordar que el nombre de las etiquetas se escriben sólo con caracteres, sin tildes, sin espacios ni otros símbolos. De este modo, tendremos la información del libro con una estructura y sintaxis de XML:

```
<libro>
  <titulo>Don Quijote de la Mancha</titulo>
  <autor>Miguel de Cervantes</autor>
  <genero>Burlesque</genero>
  <editorial>Francisco de Robles</editorial>
  <reparto>
    <personaje>Sancho Panza</personaje>
    <personaje>Dulcinea del Toboso</personaje>
    <personaje>Rocinante</personaje>
  </reparto>
  <isbn>999-9-99-999999-9</isbn>
</libro>
```

Más adelante veremos cómo construir un XML como el anterior, paso por paso.

**XML** significa *eXtensible Markup Language*, es decir, **Lenguaje de Marcado Extendido**, es lenguaje de etiquetas que proviene del **SGML** y está regulado por el **W3C**. Son **creadas por el programador**, que **estructuran y guardan de forma ordenada la información**. No representa datos por sí mismo, **sólo organiza la estructura**.

Su importancia radica en que **permite compartir datos entre diferentes equipos y aplicaciones de una manera segura, fiable y sencilla**. El hecho de que diversos equipos y aplicaciones puedan generar y leer fácilmente archivos en este formato, lo convierten en una herramienta muy común para el envío de información a través de la Web.

Cabe decir que se suele confundir HTML y XML, y para aclararlo diremos, en principio, que HTML está diseñado para mostrar datos en nuestras pantallas, mientras que XML sirve para almacenar y compartir datos, no para ser mostrados. Un dato de un XML lo mostrará un HTML que accederá a él, pero eso lo veremos en unidades posteriores.

El XML ahorra tiempos de desarrollo y proporciona ventajas, dotando a webs y a aplicaciones de una forma realmente potente de guardar la información. Además, se ha convertido en un formato universal que ha sido asimilado por todo tipo de sistemas operativos y dispositivos móviles.

## 5.1.- Estructura y sintaxis

---

Al igual que en HTML un documento XML es un documento de texto, en este caso con **extensión ".xml"**, compuesto de **parejas de etiquetas, estructuradas en árbol**, que describen una función en la organización del documento, que puede editarse con cualquier editor de texto y que es interpretado por los navegadores Web.

Las **características básicas** de XML son:

- ✓ Dado que XML se concibió para trabajar en la Web, es directamente compatible con protocolos que ya funcionan, como HTTP y los URL.
- ✓ Todo documento que verifique las reglas de XML está conforme con SGML.
- ✓ No se requieren conocimientos de programación para realizar tareas sencillas en XML.
- ✓ Los documentos XML son fáciles de crear.
- ✓ La difusión de los documentos XML está asegurada ya que cualquier procesador de XML puede leer un documento de XML.
- ✓ El marcado de XML es legible para los humanos.
- ✓ El diseño XML es formal y conciso.
- ✓ XML es extensible, adaptable y aplicable a una gran variedad de situaciones.
- ✓ XML es orientado a objetos.
- ✓ Todo documento XML se compone exclusivamente de datos de marcado y datos carácter entremezclados.



[Wikipedia Commons](#). Logo XML (CC0)

El proceso de creación de un documento XML pasa por varias etapas en las que **el éxito de cada una de ellas se basa en la calidad de la anterior**. Estas etapas son:

- ✓ Especificación de requisitos.
- ✓ Diseño de etiquetas.
- ✓ Marcado de los documentos.

El marcado en XML son etiquetas que se añaden a un texto para estructurar el contenido del documento. Esta información extra permite a los ordenadores "interpretar" los textos. El marcado es todo lo que se sitúa entre los caracteres "<" y ">" o "&" y ";"

Los datos carácter son los que forman la verdadera información del documento XML.

El marcado puede ser tan rico como se quiera. Puede ser interesante detectar necesidades futuras y crear documentos con una estructura fácilmente actualizables.

Los documentos XML **pueden tener comentarios**, que no son interpretados por el interprete XML. Estos se incluyen entre las cadenas "", pueden estar en cualquier posición en el documento salvo:

- ✓ Antes del prólogo.
- ✓ Dentro de una etiqueta.

Los documentos XML **están formados por dos partes: prólogo y ejemplar**. Los vemos a continuación.

## 5.1.1.- El prólogo.

---

El prólogo **debe preceder al ejemplar del documento**. Su inclusión facilita el procesado de la información del ejemplar. El prólogo está dividido en dos partes: la **declaración XML** y la **declaración del tipo de documento**.

- ✓ **1.- La declaración XML:** Es la primera línea del documento, de no ser así se genera un error que impide que el documento sea procesado.

El hecho de que sea opcional permite el procesamiento de documentos HTML y SGML como si fueran XML, si fuera obligatoria éstos deberían incluir una declaración de versión XML que no tienen. Esta declaración permite indicar de forma explícita que el documento es de tipo XML.

El prólogo puede tener **tres funciones**:

◆ **Declaración la versión de XML usada para elaborar el documento**

Para ello se utiliza la etiqueta:

En este caso indica que el documento fue creado para la versión 1.0 de XML.

◆ **Declaración de la codificación empleada para representar los caracteres.**

Determina el conjunto de caracteres que se utiliza en el documento. Para ello se escribe:

En este caso se usa el código iso-8859-1 (Latin-1) que permite el uso de tildes o caracteres como la ñ. Otro de los códigos a emplear es el  UTF-8 ( unicode) Sin entrar en detalles, diremos que UTF-8 soporta más caracteres y permite la visualización correcta de éstos en más sistemas o lugares que iso-8859-1. A no ser que por algún motivo no sea posible el uso de UTF-8, la recomendación es siempre utilizar UTF-8, que recordemos es la codificación de caracteres seleccionada por defecto. Hay que tener precaución en el momento de guardar el documento en el editor de código, por ejemplo, si usamos Sublime Text, cuando guardemos habrá que hacerlo según la codificación elegida.

• Los códigos más importantes son:

### **Estándares ISO y códigos de país más importantes**

Estándar ISO	Código de país
UTF-8 (Unicode)	Conjunto de caracteres universal
ISO -8859-1 (Latin-1)	Europa occidental, Latinoamérica
ISO -8859-2 (Latin-2)	Europa central y oriental
ISO -8859-3 (Latin-3)	Sudoeste de Europa
ISO -8859-4 (Latin-4)	Países Escandinavos, Bálticos
ISO -8859-5	Cirílico
ISO -8859-6	Árabe
ISO -8859-7	Griego
ISO -8859-8	Hebreo
ISO -8859-9	Turco
ISO-8859-10	Lapón. Nórdico, esquimal
EUC-JP oder Shift_JIS	Japonés

◆ **Declaración de la autonomía del documento**

Informa de si el documento necesita de otro para su interpretación. Para declararlo hay que definir el prólogo completo:

En este caso, el documento es independiente, es decir, que en su interior contiene toda la información relevante para su interpretación, de no ser así el atributo standalone hubiese tomado el valor "no".

Standalone Indica al procesador XML si un documento es independiente (standalone="yes") o se basa en información de fuentes externas, es decir, si depende de declaraciones de marca externas como una DTD externa (standalone="no").

Dicho de otro modo: Especifica si la validez del documento depende de otro documento externo, bien una DTD o un esquema, en cuyo caso el valor es no, o si depende de una DTD incluida en el mismo documento, en cuyo caso se especifica yes.

- ✓ 2.- **La declaración del tipo de documento**, define qué tipo de documento estamos creando para ser procesado correctamente. Toda declaración de tipo de documento comienza por la cadena:

## 5.1.2.- El ejemplar.

Es la parte más importante de un documento XML, ya que **contiene los datos reales del documento**. Está formado por elementos anidados.

En realidad, **el ejemplar es el elemento raíz de** un documento XML, y éste se nombrará igual que la declaración de tipo de documento (!DOCTYPE).



### Ejemplo

Sea el siguiente código XML

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE libro>
<libro>
    <titulo>XML práctico</titulo>
    <autoria>
        <autor>Sebastien Lecomte</autor>
        <autor>Thierry Boulanger</autor>
    </autoria>
    <editorial>Ediciones Eni</editorial>
    <isbn>978-2-7460-4958-1</isbn>
    <edicion>1</edicion>
    <paginas>347</paginas>
</libro>
```

El ejemplar es el elemento , que a su vez está compuesto de los elementos , , , y .



## Debes conocer

En el siguiente enlace encontrarás una tabla con los caracteres ASCII, el nombre HTML, y el número HTML de cada uno de ellos que te será muy útil a la hora de realizar documentos en HTML y XML.

<http://ascii.cl/es/codigos-html.htm>



# Atención

Es recomendable establecer un criterio y mantenerlo en todo el documento, por ejemplo que las etiquetas vayan escritas **siempre en minúsculas**.

Por otro lado, para una visualización óptima del documento, **se anidarán los elementos indentando o tabulando el código**.

## Código NO indentado (Incorrecto):

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<biblioteca>
  <titulo>XML para todos</titulo>
  <editorial>Ediciones Eni</editorial>
  <autoria>
    <autor>Sebastien Lecomte</autor>
    <autor>Thierry Boulanger</autor>
  </autoria>
</biblioteca>
```

## Código indentado (Correcto):

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<biblioteca>
  <titulo>XML para todos</titulo>
  <editorial>Ediciones Eni</editorial>
  <autoria>
    <autor>Sebastien Lecomte</autor>
    <autor>Thierry Boulanger</autor>
  </autoria>
</biblioteca>
```

Por último, es recomendable **anidar grupos de datos relacionados**, es decir, en el ejemplo anterior hemos anidado dentro de , ya que el documento queda más limpio y ordenado visualmente agrupando elementos con información relacionada.

## **5.1.2.1.- Elementos.**

---

**Todos los datos** de un documento XML **han de pertenecer a un elemento** del mismo.

Los elementos son los distintos **bloques de información** que permiten definir la estructura de un documento XML. Están **delimitados por una etiqueta de apertura y una etiqueta de cierre**. A su vez los elementos pueden estar formados por otros elementos.

**Los nombres de las etiquetas han de ser autodescriptivos**, lo que facilita el trabajo que se hace con ellas, es decir, que ilustren su contenido. Por ejemplo, si estamos con datos relativos a un libro, una etiqueta no debería ser , ya que es demasiado ambiguo, deberíamos utilizar etiquetas como , , , etc...

**La formación de elementos ha de cumplir ciertas normas** para que queden perfectamente definidos y que el documento XML al que pertenecen pueda ser interpretado por los procesadores XML sin generar ningún error fatal. Dichas reglas son:

- ✓ En todo documento XML debe existir **un elemento raíz, y sólo uno**.
- ✓ **Todos los elementos tienen una etiqueta de inicio y otra de cierre**. En el caso de que en el documento existan elementos vacíos, se pueden sustituir las etiquetas de inicio y cierre por una de elemento vacío. Ésta se construye como la etiqueta de inicio, pero sustituyendo el carácter ">" por "/>". Es decir, puede sustituirse por , que se considera una etiqueta de inicio y de cierre a la vez.
- ✓ Al anidar elementos hay que tener en cuenta que **no puede cerrarse un elemento que contenga algún otro elemento que aún no se haya cerrado**.
- ✓ **Los nombres de las etiquetas de inicio y de cierre de un mismo elemento han de ser idénticos**, respetando las mayúsculas y minúsculas. Pueden ser cualquier cadena alfanumérica que no contenga espacios, sin tildes y que no comience ni por el carácter dos puntos, ":"; ni por la cadena "xml" (ni ninguna de sus versiones en que se cambien mayúsculas y minúsculas: "XML", "XmL", "xML",...).
- ✓ El contenido de los elementos **no puede contener la cadena "]]>"** por compatibilidad con SGML. Además **no se pueden utilizar directamente los caracteres mayor que, >, menor que, <, ampersand, &, dobles comillas, ", y apostrofe, '**. En el caso de tener que utilizar estos caracteres se sustituyen por las siguientes cadenas:

Carácter	Cadena
>	>
<	<

Carácter	Cadena
&	&
"	"

Carácter	Cadena
'	'

- ✓ **Para utilizar caracteres especiales**, como £, ©, ®,... hay que usar las expresiones &#D; o &#H; donde D y H se corresponden respectivamente con el número decimal o hexadecimal correspondiente al carácter que se quiere representar en el código UNICODE. Por ejemplo, para incluir el carácter de Euro, €, se usarían las cadenas € o €



## Atención

Es habitual cometer un típico error cuando desarrollamos un documento XML y es que cuando tengamos que incluir varios ítems de un mismo tipo, utilicemos en el nombre de la etiqueta "\_1", "\_2", etc... Esto es incorrecto en XML.

Veamos un ejemplo erróneo para una receta de cocina:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE receta>
<receta>
    <nombre>Hornado</nombre>
    <tipo>Plato Fuerte</tipo>
    <ingredientes>
        <ingrediente_1 cantidad="1">Pierna de cerdo sin piel</ingrediente_1>
        <ingrediente_2 cantidad="20">Dientes de ajo machacados</ingrediente_2>
        <ingrediente_3 cantidad="1" unidad="litro">cerveza</ingrediente_3>
        <ingrediente_X cantidad="..." unidad="litro">...</ingrediente_X>
    </ingredientes>
    <preparacion>
        <paso_1>Con la ayuda de un cuchillo fino, haga incisiones...</paso_1>
        <paso_2>Mezcle en un plato la sal, la pimienta y el comino...</paso_2>
        <paso_3>Introduzca los dedos en los orificios que hizo con el cuchillo.
        <paso_X>...</paso_X>
    </preparacion>
</receta>
```

Como hemos comentado, estamos usando etiquetas diferentes para identificar elementos con las mismas características, por lo que debemos mantener el mismo nombre de la etiqueta y si queremos identificar una numeración para los pasos de la receta, por ejemplo, lo haremos con un atributo de la siguiente forma:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE receta>
<receta>
    <nombre>Hornado</nombre>
    <tipo>Plato Fuerte</tipo>
    <ingredientes>
        <ingrediente cantidad="1">Pierna de cerdo sin piel</ingrediente>
        <ingrediente cantidad="20">Dientes de ajo machacados</ingrediente>
        <ingrediente cantidad="1" unidad="litro">cerveza</ingrediente>
        <ingrediente cantidad="..." unidad="litro">...</ingrediente>
    </ingredientes>
    <preparacion>
        <paso orden="1">Con la ayuda de un cuchillo fino, haga incisiones...</paso>
        <paso <code>orden="2" >Mezcle en un plato la sal, la pimienta y el comino</paso>
        <paso <code>orden="3">Introduzca los dedos en los orificios que hizo con el cuchillo y coloque la pierna en la mezcla de especias. Deje reposar por 1 hora.</paso>
        <paso>...</paso>
    </preparacion>
</receta>
```

También sería incorrecto usar un único elemento para todos los pasos y separarlos por comas, o para todos los ingredientes. **Hay que intentar subdividir la información en varios elementos de forma que luego pueda ser tratada de forma individual** por otras aplicaciones.

## 5.1.2.2.- Atributos.

Las etiquetas pueden tener atributos, que permiten definir propiedades a los elementos de un documento. Los atributos no pueden organizarse en ninguna jerarquía o estructuras en árbol, no pueden contener ningún otro elemento o atributo, no pueden contener valores múltiples y no reflejan ninguna estructura lógica. En definitiva, los atributos no se podrán extender fácilmente para futuros cambios.

Un elemento puede tener varios atributos, pero no estar vacío. El atributo irá entre comillas dobles o simples, de la siguiente forma:

...

No se debe utilizar un atributo para contener información susceptible de ser dividida, deben utilizarse para proporcionar una información adicional sobre el elemento.



### Ejemplo básico

Suponemos que queremos incluir un dato de calidad, de 0 a 10, a un producto, lo haremos como atributo.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<catalogo>
    <producto calidad="10">
        <nombre>Silla</nombre>
    </producto>
</catalogo>
```

¿Quiere esto decir que no podremos incluirlo como elemento? La respuesta es que sí que podremos, de la siguiente forma:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<catalogo>
  <producto>
    <nOMBRE>Silla</nOMBRE>
    <calidad>10</calidad>
  </producto>
</catalogo>
```

Entonces, ¿cómo saber si una propiedad es mejor indicarla como elemento o como atributo? La respuesta es que siempre que podamos, debemos utilizar elementos, dejando los atributos para información correspondiente a metadatos, es decir, información que aclare información acerca del elemento en su conjunto. Por ejemplo, si vamos a crear un documento con datos acerca de un equipo, podríamos utilizar los siguientes atributos:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<equipo>
  <jugador>
    <nOMBRE>Alfredo</nOMBRE>
    <demarcacion>Delantero</demarcacion>
    <altura unidad="cm">178</altura>
    <peso unidad="kg">80</peso>
  </jugador>
</equipo>
```

Otro ejemplo con información acerca de los alumnos de un centro educativo:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<centro_educativo>
  <alumno sexo="Varón" fecha_nacimiento="05/06/1990">
    <nOMBRE>Pablo</nOMBRE>
    <apellido>Pérez</apellido>
    <telefono tipo="Móvil">6666666666</telefono>
    <direccion tipo="Calle">Policar, 34</direccion>
  </alumno>
</centro_educativo>
```

Resumimos diciendo que, en general, **intentaremos evitar el uso de atributos o procurar no abusar de ellos**. Normalmente los utilizaremos para metadatos o información que no sea relevante para los datos, por ejemplo: *Lolo*. Se puede ver como una manera de incorporar características o propiedades a los elementos: *Esto es un día que Patxi y Josu van paseando...* . También se pueden usar para especificar las unidades de medida, por ejemplo: 178.

Lo que si es recomendable que **una vez elegido un estilo, mantenerlo dentro de todo el documento XML**.



## Ejemplo

A continuación un ejemplo que usa varios atributos:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE biblioteca >
<biblioteca>
    <ejemplar tipo="libro" isbn="978-2-7460-4958-1" edicion="1" paginas="347"
        <titulo>XML para todos</titulo>
        <editorial>Ediciones Eni</editorial>
        <autoria>
            <autor>Sebastien Lecomte</autor>
            <autor>Thierry Boulanger</autor>
            <autor funcion="traductor">Angel Belinchon Calleja"></autor>
        </autoria>
        <historial>
            <prestamo id="003" inicio="25/09/2015" fin="29/09/2015">Gonzalo Terral</prestamo>
            <prestamo id="025" inicio="01/10/2015" fin="05/10/2015">Rosa Yoel</prestamo>
        </historial>
    </ejemplar>
</biblioteca>
```

Al abrir el documento anterior en un navegador obtenemos:

- ✓ El nombre de la etiqueta de cada elemento aparece coloreado en ciruela.
- ✓ Los nombres de los atributos en negro.
- ✓ Los valores de cada atributo en azul.

```
- <biblioteca>
  - <ejemplar tipo="libro" isbn="978-2-7460-4958-1" edicion="1" paginas="347">
    <titulo>XML para todos</titulo>
    <editorial>Ediciones Eni</editorial>
    - <autoria>
      <autor>Sebastien Lecomte</autor>
      <autor>Thierry Boulanger</autor>
      <autor funcion="traductor">Angel Belinchon Calleja"></autor>
    </autoria>
    - <historial>
      <prestamo id="003" inicio="25/09/2015" fin="29/09/2015">Gonzalo Terral</prestamo>
      <prestamo id="025" inicio="01/10/2015" fin="05/10/2015">Rosa Yoel</prestamo>
    </historial>
  </ejemplar>
</biblioteca>
```

Como se observa en el ejemplo, los atributos se definen y dan valor dentro de una etiqueta de inicio o de elemento vacío, a continuación del nombre del elemento o de la definición de otro atributo siempre separado de ellos por un espacio. Los valores del atributo van precedidos de un igual que sigue al nombre del mismo y tienen que definirse entre comillas simples o dobles.

Los nombres de los atributos han de cumplir las mismas reglas que los de los elementos, y no pueden contener el carácter menor que, <.



# Autoevaluación

¿Cuáles son los errores del siguiente documento XML?

```
<?XML version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE biblioteca >
<biblioteca>
    <ejemplar tipo='libro' isbn='978-2-7460-4958-1' edicion=paginas='347
        <titulo>XML para todos</titulo>
        <editorial>Ediciones Eni</editorial>
        <autoria>
            <autor>Sebastien Lecomte</autor>
            <autor>Thierry Boulanger</autor>
            <autor funcion='traductor'>Angel Belinchon Calleja'></aut
        </autoria>
        <historial>
            <prestamo id='003' inicio='25/09/2015' fin='29/09/2015'>
                <prestamo id='025' inicio='01/10/2015' fin='05/10/2015'>
            </historial>
        </ejemplar>
    </biblioteca>
```

- Utiliza mayúsculas en la definición de la versión XML.
- No utiliza el código de caracteres adecuado.
- Los valores de los atributos no están entre comillas dobles.
- Hay algún atributo vacío.
- La etiqueta no está cerrada.
- Se usan guiones en el contenido de un atributo.

[Mostrar retroalimentación](#)

## 6.- Documentos XML bien formados.

---

Se pueden hablar de documentos "**bien formados**" y documentos "**válidos**", ambos son documentos XML correctos.

Los documentos bien formados son aquellos que son sintácticamente correctos, es decir, que cumplen con las reglas de creación de documentos XML expuestas en apartados previos, por ejemplo, usar caracteres válidos para el nombre de una etiqueta y que las etiquetas deben cerrarse correctamente.

Sin embargo, los documentos válidos son aquellos que, además de estar bien formados, cumplen los requisitos de una definición de estructura que veremos más adelante (DTD, XML Schema,...).

Por lo tanto, para que un documento esté bien formado, **deben verificar las reglas sintácticas que define la recomendación del W3C para el estándar XML**. Esas normas básicas son:

- ✓ El documento **ha de tener definido una declaración XML en el prólogo**:



éstos serán los valores por defecto si no se incluye el prólogo, pero recordamos que es muy recomendable incluirlo, ya que algunos navegadores nos devolverán errores si no lo hacemos.

- ✓ **Existe un único elemento raíz para cada documento**: es un solo elemento en el que todos los demás elementos y contenidos se encuentran anidados.
- ✓ Los elementos se organizan entre sí en **estructura jerárquica y no se permite el solapamiento de los elementos**.
- ✓ Hay que **cumplir las reglas sintácticas del lenguaje XML para definir los distintos elementos y atributos** del documento. Los resumimos a continuación:
  - ◆ El nombre de los elementos pueden tener como primer carácter [A-Z], [a-z] y "\_", y para el resto de caracteres, además de los citados: [0-9], "-" y "."
  - ◆ **Las etiquetas de apertura y de cierre tienen que ser idénticas**, XML es sensible a las mayúsculas y minúsculas, por lo que, por ejemplo, ... sería incorrecto. Recordemos que se permiten elementos vacíos.
  - ◆ Los **valores de los atributos se escribirán siempre entre comillas dobles o simples**. Si queremos incluir un atributo que incluya alguno de esos caracteres, se hará de la siguiente forma: ":" ("") y '()' . Existen atributos reservados que no podremos usar, salvo para la finalidad en la que están reservados: "xml:", "xml:lang", "xml:space: default | preserve" y "xml:id"
  - ◆ **Los comentarios en XML** se escribirán así:

# Autoevaluación

¿Está "bien formado" el siguiente documento XML?

```
<destinatario>Víctor</destinatario>
<remitente>Fran</remitente>
<asunto>
    <contenido>Mañana quedamos para ver el clásico</contenido>
</asunto>
```

- Verdadero  Falso



## Para saber más

**Los espacios de nombres permiten definir la pertenencia de los elementos y los atributos de un documento XML al contexto de un vocabulario XML.** De este modo se resuelven las ambigüedades que se pueden producir al juntar dos documentos distintos, de dos autores diferentes, que han utilizado el mismo nombre de etiqueta para representar cosas distintas.

Los espacios de nombres también conocidos como XML namespaces, permiten dar un nombre único a cada elemento, indexándolos según el nombre del vocabulario adecuado además están asociados a un URI que los identifica de forma única.

En el documento, **las etiquetas ambiguas se sustituyen por otras en las que el nombre del elemento está precedido de un prefijo**, que determina el contexto al que pertenece la etiqueta, seguido de dos puntos, esto es:

Esta etiqueta se denomina "nombre cualificado". Al definir el prefijo hay que tener en cuenta que no se pueden utilizar espacios ni caracteres espaciales y que no puede comenzar por un dígito.

Antes de poder utilizar un prefijo de un espacio de nombres, para resolver la ambigüedad de dos o más etiquetas, es necesario declarar el espacio de nombres, es decir, asociar un índice con el URI asignado al espacio de nombres, mediante un atributo especial `xmlns`. Esto se hace entre el prólogo y el ejemplar de un documento XML y su sintaxis es la siguiente:

://.../

Los espacios de nombres tienen una recomendación en XML:

<http://www.w3.org/TR/REC-xml-names/>

# Autoevaluación

**Los espacios de nombres permiten...**

- Utilizar etiquetas idénticas para estructurar distintos tipos de información de texto.
- Estructurar la información de un documento XML cuando proviene de varios documentos.
- Asignar varias etiquetas a una misma información.
- Definir etiquetas en otros documentos.

## 7.- Sistemas de gestión empresarial.



### Caso práctico

Félix ha disfrutado de un merecido descanso en un pueblecito costero y a su vuelta a su puesto de trabajo ha solicitado una reunión con Juan para plantear su nueva idea.

Mientras tanto Juan se ha estado informando sobre la utilidad de los ERP, ha leído que son sistemas informáticos que pueden mejorar notablemente la actividad comercial de una empresa. Además, se pueden usar desde los dispositivos móviles, con lo que no es imprescindible estar en un despacho con un ordenador conectado a una red local determinada para poder consultar la información alojada en ellos.

El mayor problema que ve Juan, es la posible necesidad de formar a los trabajadores de la empresa para que puedan utilizar esta herramienta, así como la integración de la misma con el resto de programas que se usan, los que casi pueden reducirse a una suite ofimática. Según se va informando del tema, cada vez encuentra menos inconvenientes ya que no parece que la utilización de estos recursos requieran una gran formación por parte de los usuarios y, por otro lado, la mayoría de los ERP del mercado, si no todos, permiten la integración las suites informáticas. Esto significa que los usuarios podrán seguir usando los mismos programas que hasta ahora.

La reunión se plantea del siguiente modo:

- ✓ Comenzará por exponer **cuales son los distintos flujos de información que hay en la empresa**, haciendo especial hincapié en aquellos que se usan en los procesos más importantes.
- ✓ Despues pasará a **analizar los distintos subsistemas de información que existen** y su relación con los flujos de información. Incidiendo en que si se optimizan los flujos de información entre los subsistemas, se logrará mejorar la gestión y el control de la información.



[Let Ideas Compete](#). Félix en el puerto. (CC BY-NC-ND).

Para estudiar este apartado lo primero que hemos de hacer es pararnos a pensar cómo fluye la información dentro de la empresa, es decir, es necesario conocer todos sus recursos empresariales y gestionarlos de forma eficaz para poder ser eficiente en todos los procesos productivos. Si lo meditamos con calma veremos que lo hace de varias formas posibles:

- ✓ Entre los empleados de la empresa.
  - ✓ Entre los empleados y la empresa.
  - ✓ Entre la empresa con sus clientes y proveedores.

Estos **flujos de información** pueden ser de dos tipos:

- ✓ Informales y no estructurados.
  - ✓ Formales y estructurados, que se centran en información acerca de procesos críticos de la empresa.

Para facilitar este flujo de información es conveniente tener instalado un Sistema de Información. Pero, **¿qué es un Sistema de Información?**

Un sistema de información son subsistemas dentro de la empresa que facilitan la transferencia de información. Los elementos de los que constan se clasifican en: físicos, humanos y normas y protocolos.

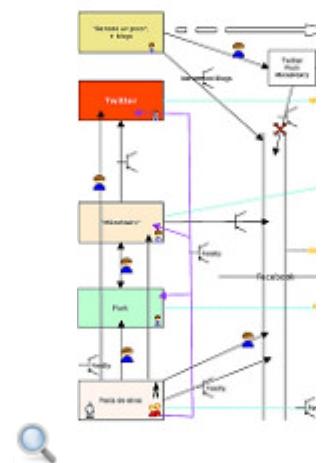
Denominamos **Sistema de Información** a un conjunto organizado de elementos relacionados orientados al tratamiento y administración de información.

Un sistema de información facilita el conocimiento propio de la empresa para mejorar la planificación, la gestión y el control.

¿Qué es lo que la empresa obtiene del conocimiento y la gestión de los flujos de información?

Principalmente obtiene ventajas competitivas, que mejoran su eficiencia, la calidad del producto, el servicio ofrecido a los clientes y facilitan la captación de clientes.

La automatización de los flujos de información, asociándolos a los flujos físicos de producción de materiales, cambió sustancialmente cuando aparecieron los sistemas de información ERP y CRM, en los que se integran las aplicaciones que soportan los diferentes procesos de la empresa.



Galería de fotosmak. *Flujos de información.* (CC BY-NC-ND)

La rápida expansión de Internet ha incrementado las interacciones entre empresas y entre éstas y sus clientes, provocando la modificación de los modelos de negocio tradicionales. La proliferación del uso de dispositivos portátiles ha facilitado la existencia de empleados no ubicados en una oficina fija, lo que ha obligado a buscar soluciones para integrar a los empleados móviles en los flujos de información de la empresa.

## 7.1.- ERP.



### Caso práctico



[Let Ideas compete. Félix pensando.](#) ([CC BY-NC-ND](#))

Durante la reunión Félix y María escuchan atentamente las explicaciones de Juan, pero no logran ver el beneficio que les aportará el uso de un **ERP**.

Juan pasa a explicarles que un **ERP** está formado por diversas partes integradas cuya funcionalidad es totalmente independiente de las demás. Les expone que en realidad la información estaría guardada en una única base de datos, a la que accedería el **ERP** a medida que los usuarios lo demandasen. La gran ventaja que Juan ve en todo esto, es que todos los usuarios comparten la misma información garantizando que los datos con los que trabajan siempre estén actualizados.

Cuando una empresa es pequeña, es posible que podamos asumir todas las tareas necesarias para mantener su funcionamiento, pero si ésta se expande suficientemente, será necesario tener controlados todos los recursos empresariales y para ello tendremos que recurrir a algún sistema automatizado de gestión y administración.

**Éstos sistemas se llaman ERP, que son sistemas de gestión de información que se caracterizan por ser una aplicación en la que hay varias partes integradas y se especializan en manejar todos los datos relevantes para la continuidad de la empresa.** Estas partes son de diferente uso, por ejemplo: producción, ventas, compras, pedidos, nóminas, etc.

Un sistema de  CRM es un ERP que se centra en la relación con los clientes que tiene una empresa, es decir, información de contacto orientada a ventas.

Los objetivos principales de los sistemas ERP son:

- ✓ Optimizar los procesos empresariales.
- ✓ Acceder a la información confiable y precisa.
- ✓ Permitir compartir información entre los componentes de la organización.
- ✓ Eliminar los datos y operaciones innecesarias.

Las características que distinguen a un ERP de cualquier otro software empresarial, es que deben de ser sistemas integrales, modulares y adaptables. Además, un ERP se caracteriza por:

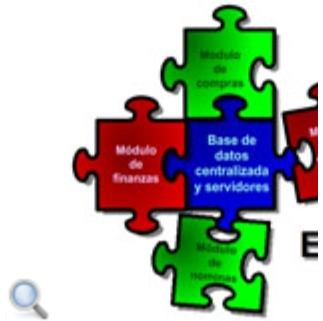
- ✓ Ser un programa con **acceso a una base de datos**.
- ✓ Sus **componentes interactúan** entre sí.
- ✓ Los **datos** deben ser consistentes, completos.

En ocasiones son sistemas complejos y difíciles de implantar, debido a que necesitan un desarrollo personalizado para cada empresa a partir del paquete inicial. Estas adaptaciones suelen encargarse a las consultorías.

La consultoría en materia de ERP puede ser de dos tipos:

- ✓ **Consultoría de negocios.** Estudia los procesos de negocio de la compañía, y evalúa su correspondencia con los procesos del sistema ERP para poder personalizarlo, y de este modo ajustarlo a las necesidades de la organización.
- ✓ **Consultoría técnica.** Conlleva el estudio de los recursos tecnológicos existentes, en ocasiones implica la programación del sistema, obtener determinados informes.

En la actualidad, y debido a la amplia implantación de las intranets en las empresas, la mayoría de los sistemas ERP tienen interfaz web, lo que aporta la ventaja de permitir el acceso al ERP a través del navegador web.



Víctor Gómez. Partes de un ERP.  
(Dominio público)

# Autoevaluación

**Marca aquellas afirmaciones que sean ciertas para un ERP:**

- Es un sistema formado por varias aplicaciones que interactúan entre sí.
- La empresa ha de escoger entre los diversos ERP que existen, el que se adapte a su modo de trabajo.
- Para implementar un ERP es necesario tener asociada una base de datos centralizada.
- Un ERP está formado por diversos módulos.

[Mostrar retroalimentación](#)

## 7.1.1.- Características.

---

Existen tres características principales que distinguen a un sistema de gestión empresarial:

✓ **Integración:**

Un sistema  ERP integra todos los procesos de la empresa, de tal forma que las considera como una serie de áreas que se relacionan entre sí, para conseguir una mayor eficiencia reduciendo tiempo y costes.

En un sistema  ERP los datos se ingresan una sola vez, formando una base de datos centralizada y facilitando el flujo de información entre los diferentes módulos.

✓ **Modularidad:**

Cada módulo del sistema  ERP se corresponde con un área funcional de la empresa. Gracias a una base de datos centralizada, estos módulos comparten información entre sí, facilitando la adaptabilidad, personalización e integración. Es habitual que cada módulo utilice un software específico para su funcionalidad.

✓ **Adaptabilidad:**

Aunque las dos características anteriores; integración y modularidad facilitan la adaptabilidad a las necesidades de cada empresa, algunas veces para abaratar costes se utiliza una solución más genérica y se modifican algunos de los procesos para alinearlos al sistema  ERP.

Gracias a la modularidad y capacidad de integración de las funcionalidades un sistema  ERP es fácilmente adaptable a las necesidades de cada empresa, permitiendo una total configuración. Éstos módulos se interconectan entre sí, de forma que una única herramienta  ERP puede dar servicio a diferentes empresas, modificando el conjunto de módulos activos y sus relaciones.



Víctor Gómez. Diagrama de módulos en un ERP ([CCO](#))

Éstos módulos permitirán almacenar, buscar, mostrar y representar cada proceso interno de la empresa, pudiendo componerse un **ERP** de los siguientes módulos activos básicos:

- ✓ **Ventas / Marketing:** Interfaz pública que interactúa con los clientes, pedidos, estrategias de ventas, precios, promociones, publicidad, etc.
- ✓ **Finanzas:** Es la base del **ERP**, donde se almacena cada transacción, facilitando las auditorías.
- ✓ **Inventario / Logística:** Stock y flujos de entrada y salida.
- ✓ **Recursos humanos:** Gestión de personal, nóminas, productividad, incentivos, beneficios, etc.
- ✓ **Producción:** El núcleo que se encarga de los movimientos físicos de los artículos, planificación de materiales, etc.

El listado anterior es básico y se puede complementar con **otros módulos**, por ejemplo: proyectos, planificación de ventas, mantenimiento, gestión de cambios de producto, configuración de productos a medida, gestión de relaciones con los clientes ( **CRM**), controles de planta y de almacenes, seguridad y planes de contingencia, cuadros de mando, gestión de archivos, gestión comercial, proveedores, etc.

No todos los trabajadores accederán al **ERP** de la misma forma, ya que cada grupo tendrá su **rol**, que será supervisado por un administrador, por lo que según el puesto a desempeñar, se activarán o desactivarán sus módulos asociados.



## Para saber más

👉 **CRM** (*Customer Relationship Management* o Sistemas de información globales hacia clientes), se originan como consecuencia de una aplicación específica de 📊 **ERP**. Están enfocados a mantener, crear, y potenciar las relaciones con los clientes de una empresa, apoyando así a las políticas de marketing de una empresa.

En la actualidad los sistemas globales de 📊 CRM se pueden dividir en: aplicaciones electrónicas para los canales de distribución de la empresa, centros de atención telefónica, autoservicio hacia los clientes, gestión electrónica de las actividades que afectan a clientes y ventas.

Entre sus principales características se pueden destacar que facilitan tomar decisiones en tiempo real, incrementando la rentabilidad del cliente gracias a que obtienen una información muy útil a partir de datos complejos, es decir pueden identificar fácilmente a los clientes que compran o que no están interesados y actuar en consecuencia.

## 7.1.2.- Ventajas e inconvenientes.

Contar con un sistema  ERP personalizado, permite a la empresa tener integradas diferentes utilidades que le facilitan la gestión de la información. Permitiendo entre otras cosas, administrar interdependencias de los recibos de materiales, de los productos estructurados en el mundo real, de los cambios de la ingeniería y de la revisión y la mejora, y la necesidad de elaborar materiales sustitutos, etc. La seguridad de los ordenadores está incluida dentro del ERP, para proteger a la organización contra el espionaje industrial y la malversación.

### Ventajas del uso de los ERP:

- ✓ Aumento de la información que la empresa posee sobre sus actuales y potenciales clientes: Los ERP que incluyen  CRM aportan beneficios relacionados con la administración de los clientes de la empresa, y pueden incluir un control de calidad en los productos finales, que garantiza que no hay problemas no resueltos, lo que le permite dirigir la oferta hacia sus deseos y necesidades, aumentando así el grado de satisfacción y optimizando su ciclo de vida.
- ✓ Aumento de las ventas.
- ✓ Permiten resolver problemas derivados del tratamiento de la información con los sistemas anteriores.
- ✓ Aumentan la eficiencia operativa.
- ✓ Facilitan el acceso de la información y constituyen una mejora en las herramientas para el tratamiento de la misma.
- ✓ Reducción de costes empresariales, especialmente los relacionados con las operaciones de las tecnologías de información y comunicación en la empresa.
- ✓ Permiten mayor facilidad en la configuración de los sistemas de la empresa.
- ✓ Mejoran el entorno de integración de todas sus acciones.



[Skotan.](#), Cara feliz. (CC0)

Muchos de los problemas que tienen las compañías con el ERP, son debidos a la inversión inadecuada para la formación del personal, y una falta de políticas corporativas que afectan al modo en que se obtienen y actualizan los datos del ERP.



[SunKing2.](#), Cara triste. (CC0)

### Limitaciones y obstáculos del uso de los ERP:

El ERP ha de ser utilizado y realizado por personal capacitado. La instalación del ERP es muy costosa. Los ERP son vistos como sistemas muy rígidos, y difíciles de adaptar al modo de trabajo de las empresas. Son sistemas que sufren problemas de "cuello de botella", es decir, todos los usuarios del sistema se pueden ver afectados por la ineficiencia en uno de los departamentos participantes. Altos costes de las modificaciones del ERP una vez implantado.



## Reflexiona

¿Te has parado a pensar alguna vez en los efectos negativos que tiene la falta de normalización en los procedimientos que se realizan en las pequeñas empresas? Y ¿la falta de interés que suelen tener los empleados de las mismas en establecer un protocolo de comportamiento que los formalice?

## Autoevaluación

**Marca en las siguientes afirmaciones cuál es una ventaja para un ERP:**

- Solo personal capacitado debe utilizarlo.
- En la actualidad la instalación es asequible a cualquier empresa.
- Son sistemas sencillos de manejar en el que el error de un usuario no afecta al resto.
- Es un sistema en el que están integradas varias de las utilidades que usa la empresa.

## 7.1.3.- ERP de software libre.

---

Dentro de los  ERP de software libre existe una gran cantidad de aplicaciones que aportan soluciones a nivel de planificación empresarial. Entre ellas, podemos destacar **Openbravo**, que es una iniciativa de origen español y **OpenERP**, ahora denominado **Odoo** de origen belga y que se caracteriza por tener una gran cantidad de módulos disponibles.



**Openbravo** es una aplicación de código abierto de planificación de recursos empresariales. Utiliza una arquitectura cliente/servidor web y está escrita en Java. Se ejecuta sobre un servidor web, y ofrece soporte para las bases de datos Oracle y PostgreSQL.

Openbravo consta de dos versiones:

- ✓ **Openbravo Community Edition**, libre y gratuita desde la que no se puede acceder a los módulos comerciales. Versión bajo licencia OBPL
- ✓ **Openbravo Network Edition**, versión bajo licencia OBCL que proporciona actualizaciones de código y en la que sí se pueden instalar los módulos comerciales.

Openbravo nos ofrece dos soluciones:

- ✓ **Suite de comercio Openbravo**: Solución de comercio para minoristas
- ✓ **Suite de negocio Openbravo (ERP)**: Solución global para empresa. Nosotros nos centraremos en este producto.

En el siguiente enlace se pueden ver las diferentes alternativas que se disponen: [Suite de comercio y negocio Openbravo](#)

**Odoo (Anteriormente llamado OpenERP)** resuelve problemas complejos haciendo uso de soluciones sencillas. Está escrita en Phyton y la base de datos utilizada es PostgreSQL



## Sabías que...

OpenERP fue creado en el año 2005 por un joven informático belga llamado Fabien Pinckaers, conocido como el nuevo Bill Gates. Muchos se sorprendieron de que creara un programa de estas características y lo dejara de forma gratuita en Internet, mientras otras empresas venden sus productos a precios desorbitantes.

En el siguiente enlace se muestra el por qué del cambio de nombre de OpenERP a Odoo, también conoceremos la respuesta a algunas cuestiones interesantes.

[OpenERP cambia su nombre a Odoo a partir de la versión 8.0.](#)

El modelo de negocio de Odoo está basado en los servicios prestados en torno al software, tiene colaboradores en muchos países y su facturación se mide en millones de euros.



## Para saber más

Enlace a la página web oficial del software Openbravo, con información y descargas de la aplicación:

[Página oficial Openbravo](#)

A continuación te proporcionamos el enlace al wiki de Openbravo. En ella podrás consultar toda la información que necesites sobre este ERP:

[Wiki sobre Openbravo](#)

Enlace a la página web oficial del software Odoo, con información y descargas de la aplicación:

[Página oficial Odoo](#)

En la siguiente página puedes acceder a documentación sobre la versión actual de Odoo:

[Documentación de usuario Odoo.](#)

[Documentación técnica sobre Odoo.](#)

## 7.1.4.- Instalación.

Para realizar la instalación de un paquete de gestión empresarial, primero tenemos que **definir cuáles son las necesidades que debe cubrir el software** y buscar aquel que mejor se ajuste a nuestras necesidades.

En general, las tareas implicadas en un proceso de instalación e implantación de un  ERP son las siguientes:

- ✓ **Diseño de la instalación.** Previo a la instalación deberá realizarse un estudio de las necesidades de la empresa y cómo serán resueltas por la aplicación ERP: tablas que es preciso adaptar, datos, formularios e informes que se requieren, etc.
- ✓ **Instalación de equipos servidores y clientes.** Será necesaria la instalación, revisión y/o actualización del hardware de la empresa, de manera que cumpla los requisitos mínimos necesario.
- ✓ **Instalación del software.** Instalación tanto de la aplicación ERP como del software que ésta necesite para su correcto funcionamiento.
- ✓ **Adaptación y configuración del programa.** Una vez instalado, será necesario la configuración del software y su adaptación a la empresa cliente.
- ✓ **Migración de datos.** Este proceso es de gran importancia para la empresa, pues los datos son imprescindibles para su buen funcionamiento: clientes y proveedores, contabilidad, facturación, son datos muy importantes y de gran volumen. En ocasiones serán necesarios procesos manuales si no existe forma de automatizar el traspaso de datos del sistema antiguo a la nueva aplicación ERP.
- ✓ **Realización de pruebas.** La instalación del software puede conllevar un periodo de transición en el que coexistirá la solución ERP con la solución antigua de gestión de la empresa. Este periodo de transición comprenderá la realización de pruebas del nuevo aplicativo, y la migración de datos del software antiguo al nuevo. En cualquier caso, será necesario verificar mediante las pruebas necesarias que la solución ERP funciona correctamente y los resultados obtenidos son satisfactorios.
- ✓ **Documentación del sistema.** En esta fase se deben realizar los documentos y manuales necesarios y su puesta a disposición ante la organización, mediante los medios de difusión interna que disponga: tablón de anuncios, correo electrónico, Intranet, etc.
- ✓ **Formación de usuarios.** Esta etapa comprende la formación de los usuarios sobre la utilización del ERP, que podrá conllevar una formación inicial para los responsables del proyecto y una formación para los usuarios finales.



Galería marcusrg. Descarga. (CC BY)

En la mayoría de los casos los sistemas ERP correrán sobre una **plataforma cliente-servidor**, aunque esto no quita que el sistema pueda estar integrado en un **Servidor Web**, o que pueda utilizar **tecnología SaaS**.

Independientemente del sistema operativo que tengamos instalado en nuestra empresa, o que decidamos utilizar para implementar nuestro ERP Windows o Linux, básicamente siempre tenemos que **tener en cuenta lo siguiente**:

- ✓ Disponer de una máquina con prestaciones de servidor donde inicialmente instalaremos nuestro ERP.
- ✓ Instalar nuestra base de datos y conectarla con nuestro ERP.
- ✓ Instalar los módulos necesarios ERP que hayamos decidido adquirir .
- ✓ Configurar los diferentes clientes para que accedan al servidor y que puedan realizar sus peticiones al sistema ERP.

Nuestro sistema **se basa en una base de datos** en la que se irán almacenando los datos, y desde la que se irán generando los informes que requerimos al sistema, por lo que previamente la habremos creado en nuestro sistema.

Hoy en día, lo más habitual es incorporar el ERP dentro de la intranet de la empresa. En este caso, necesitamos tener activo un servidor web con soporte para bases de datos, y el lenguaje de script de servidor en el que se haya codificado la aplicación que vayamos a instalar. La instalación de este último tipo de sistemas se realiza también a través de un navegador web siguiendo los pasos que indica el sistema.

Los tipos de instalación en un sistema ERP/ CRM dependerán de la plataforma en la que se van a instalar los programas y del ERP con el que trabajemos. Los más habituales son los siguientes:

- ✓ **Instalación mediante máquina virtual.** La aplicación y programas necesarios para su funcionamiento se proporcionan en una máquina virtual lista para ejecutarla. Esta opción no es apta para un entorno de producción, y se utiliza normalmente para hacer una primera evaluación del producto.
- ✓ **Instalación de paquetes bajo entorno gráfico.** En este caso las aplicaciones se instalan mediante el entorno gráfico del sistema operativo, utilizando asistentes que instalan y resuelven las dependencias entre paquetes. Este tipo de instalación puede utilizarse en entornos de producción, pero hay que tener en cuenta que los paquetes pueden no estar actualizados a la última versión.
- ✓ **Instalación personalizada.** Si queremos instalar una versión más reciente de la aplicación, existe la posibilidad de descargarnos los paquetes fuente desde la página web que los contenga, e instalarlos mediante comandos. Ello permite un mayor control sobre los programas que se instalan y las dependencias entre ellos, aunque suele ser un proceso más complejo que la instalación bajo entorno gráfico.
- ✓ **No instalar y acceder a la aplicación on-line.** Algunos ERP incluyen la posibilidad de acceder a demostraciones online del producto, de manera que no es necesario instalar nada, ya que nos conectamos a un servidor en Internet que tiene todos los datos y programas de la aplicación. Esta opción es utilizada por los proveedores de ERP que ofrecen el servicio SaaS.

En **Odoo** podemos realizar diferentes tipos de instalaciones, desde paquetes autoinstalables, donde no tendremos que realizar prácticamente nada (se nos instalan y configuran todos los componentes y referencias necesarias), hasta instalaciones paso a paso, donde tendremos que instalar y configurar cada uno de los componentes.

Desde la versión 8.0, donde pasa a denominarse Odoo, **el servicio de acceso de cliente de escritorio no existe** y su **acceso es a través de cliente web**, es decir, se accede al software vía navegador web y la información se almacena en una base de datos. Esta es la forma de acceso a la mayoría de los ERP actualmente.

## Autoevaluación

Para realizar la instalación de un ERP en nuestro sistema necesitamos tener una base de datos creada para esta función así como un servidor web con soporte para bases de datos.

- Verdadero.
- Falso.



### Para saber más

En los dos siguientes enlaces puedes consultar los términos de la licencia de Openbravo ERP:

[Términos de Licencia de Openbravo ERP](#)

En el siguiente enlace puedes consultar los términos de la licencia del ERP de software libre Odoo:

[Términos de licencia de Odoo](#)

## 7.1.5.- Personalización.

Tras el proceso de instalación del ERP viene la personalización, para adecuarlo al entorno de la empresa que va a utilizarlo. Este proceso incluye aspectos como incorporar el logo de la compañía, o dar de alta a los usuarios con los permisos adecuados, además de configurar el sistema de avisos que proporciona la aplicación, o la compatibilidad con herramientas de correo.

Los sistemas ERP que existen hoy en el mercado permiten la incorporación de distintos tipos de módulos predefinidos que facilitan la personalización del paquete.

Un **módulo** es un programa que se realiza para cubrir una determinada función de la aplicación. Existen unos módulos básicos que se pueden cargar automáticamente durante la instalación inicial del sistema, y otros que pueden instalarse posteriormente desde el mismo programa o desde la web del ERP.

La integración de estos módulos, que complementan la aplicación base, puede realizarse tanto en el momento inicial de la instalación de la aplicación, como en un proceso posterior de ampliación de la misma. Los recursos que proporcionan estos módulos son muy variados, desde la creación de informes avanzados a servicios de comunicación para plataformas móviles (tablet-pc o smartphones), pasando por la interconexión con el paquete ofimático utilizado en la empresa. Hoy en día, la mayoría de los sistemas ERP cuentan con un módulo CRM integrado, o con la posibilidad de incorporarlo como un módulo añadido.

Los módulos que se incorporan pueden estar prediseñados por el fabricante de la aplicación base, ser módulos programados por terceros para ese software, o incluso ser un programa solicitado a medida por la empresa en la que se instala la aplicación.

Para realizar la instalación de módulos hay que entrar en el ERP con un usuario que tenga permisos de administración, además hay que usar el cargador de módulos que suelen tener estos paquetes en la sección de administración o aplicaciones.

A continuación un listado de los **módulos más importantes** de un ERP:



Víctor Gómez. Módulos en Odoo. (Dominio público)

- ✓ **Gestión contable y financiera:** El módulo contable recoge y automatiza todas las operaciones contables de la compañía, centralizándolas para su consulta, publicación o control. Este módulo debe estar integrado con los módulos de compras y ventas para evitar duplicidades en la introducción de datos, y para poder disponer siempre de la información en tiempo real. De esta forma no sólo realizaremos la gestión contable financiera, sino que podremos obtener automáticamente facturas de clientes y proveedores desde los pedidos de venta y compra, o desde los albaranes de salida y entrada, respectivamente.
- ✓ **Compras, ventas y almacén:** El módulo de compras y ventas registra todas las operaciones de solicitudes de presupuestos a proveedor, recepción de precios y creación de pedidos de compra, con indicación de cualquier información relevante como precios, plazos de entrega, etc. y del mismo modo con las ventas. El módulo de Almacén permite gestionar las existencias de productos en almacén.
- ✓ **Facturación:** El módulo de facturación se encarga de la generación de todo tipo de datos que tienen que ver con la facturación de productos y servicios a los clientes: facturas de venta, albaranes, tarifas, etc. Son contempladas diversas formas de cobro y pago de un cliente, como son contado, transferencia, pagaré, giro, etc.
- ✓ **Gestión de personal:** Este módulo lleva a cabo la planificación y realización de las nóminas de los empleados, así como las altas, bajas, contratos, control de horarios y datos de personal, además de un sistema de remuneraciones para el pago de los empleados, con la inclusión de comisiones por ventas.
- ✓ **Gestión de las relaciones con el cliente:** La gestión de las relaciones con los clientes o CRM (Customer Relationship Management) permite registrar todo lo relativo a la relación comercial con los clientes o posibles clientes: desde la creación de una ficha de cliente hasta la determinación de los productos que le interesan, histórico de pedidos, las ventas realizadas, contactos mantenidos o la facturación de productos o servicios.

## Autoevaluación

**Escoge las afirmaciones correctas:**

- Al instalar un ERP siempre se instalan todos los módulos que hay disponibles para él.
- En el proceso de instalación de un ERP tenemos que decidir todos los módulos que queremos instalar ya que luego no podremos modificarlo.
- Una empresa puede instalar en su ERP tanto módulos preprogramados como módulos hechos de encargo para ajustarse a sus necesidades.
- La integración de módulos puede hacerse en cualquier momento.

[Mostrar retroalimentación](#)



## Para saber más

Para mayor información sobre los tipos de módulos de OpenERP puedes consultar el siguiente enlace:

[Tipos de módulos en Odoo / OpenERP](#)

En la página web de la empresa Openbravo tienes información sobre los módulos de su ERP:

[Presentación Módulos en Openbravo](#)

## 7.1.6.- Seguridad: planificación, usuarios y roles.



### Caso práctico

Una vez que les convence de la utilidad de los **ERP**, **María y Félix le preguntan por la seguridad de los datos**, ya que en su empresa es imprescindible garantizar la privacidad de la información que manejan, y el hecho de que todos compartan la misma información, induce a creer que todos los usuarios del sistema van a poder acceder a la información.

**Juan** les cuenta que ese era uno de los puntos débiles que podría tener su propuesta, pero que después de haberlo estudiado y meditado no hay ningún problema.

Los ERP tienen integradas una serie de medidas de seguridad, que se orientan a que haya **distintos perfiles de usuario** con diferentes niveles de acceso a la información, lo que hace que cada usuario sólo acceda a la información que necesita para realizar su trabajo. Además el modo de este acceso es también el imprescindible para cada usuario. Es decir, un usuario que sólo necesita consultar cierta información, podrá consultarla, pero no modificarla.

La conexión entre distintos equipos se realiza utilizando protocolos de seguros.

Se garantizan las operaciones que se realizan sobre los datos.



[Jonny goldstein](#). Juan trabajando. (CC BY)

Para **aumentar la seguridad** de nuestro sistema, hay que analizar los riesgos a los que se somete, clasificándose en dos tipos:

- ✓ **Riesgos físicos:** Cuando falla algún componente electrónico en nuestro sistema informático (avería de discos duros, memorias, etc.). Pueden fallar principalmente por agresiones externas, tales como altas temperaturas, incendios, inundaciones, explosiones o robos)
- ✓ **Riesgos lógicos:** Cuando no hay una política adecuada en los sistemas informáticos, accesos no autorizados, bugs, errores en el sistema operativo o en el software, intrusiones externas, etc.

Las **medidas de seguridad** que se implementan en estos sistemas se basan, principalmente, en los siguientes aspectos:



TheCX. Seguridad. ([CC BY-NC-ND](#))

- ✓ **Niveles de acceso configurables para los usuarios según su rol:** En función de las tareas que deba realizar, el usuario debe contar con una serie de políticas que le permitan acceder a determinados datos, quedando algunos de ellos reservados para usuarios con un nivel de toma de decisión más elevado.
- ✓ **Auditoría de cada transacción:** Se controla cada envío de datos, lo que garantiza las operaciones realizadas.
- ✓ **Soporte para la conexión segura mediante https:** Para garantizar la seguridad de la comunicación entre los equipos cliente y el servidor en el que está instalada la aplicación, se realiza con un protocolo de comunicación seguro que no permite espiar el canal de comunicación. Se emplea este tipo de comunicación segura en el proceso de autenticación de usuarios.

Parte de la configuración de la seguridad del sistema es consecuencia de una buena asignación del rol de **cada uno de los usuarios del sistema, ya que esto garantiza que cada uno de los usuarios sólo tiene acceso a la información que necesita para realizar su trabajo**. Este trabajo hay que llevarlo a cabo con un usuario administrador.

Los paquetes básicos de los ERP y CRM suelen tener varios tipos de **usuarios posibles**:



Aeioux. The user. ([CC BY-NC](#))

- ✓ **Administrador**, es un tipo de usuario que tiene todos los privilegios, tanto en lo referente al acceso a la información como a las tareas de gestión del sistema, (creación de usuarios, integración de módulos, modificación del aspecto del sistema, etc.).
- ✓ **Usuario normal**, es un usuario que no tiene privilegios de administración, pero tiene todos los privilegios respecto a la información almacenada.
- ✓ **Usuario de grupo**, se crean para recibir el correo entrante para distribución.
- ✓ **Usuario de portal**, permite al usuario acceder a los portales creados en el entorno pero no a la aplicación.

Hay aplicaciones en las que los usuarios no pueden eliminarse directamente, hay que hacerlo desde la base de datos, aunque siempre permiten desactivarlos.

Además de los perfiles de usuarios es interesante conocer los diferentes **roles** que hay definidos para los usuarios de una aplicación. **Un rol define ciertos privilegios a la hora de realizar tareas específicas.**

**Las características de los roles son:**

- ✓ Podemos considerar que un rol es un grupo particular de privilegios.
- ✓ Un rol solo tiene validez cuando está asignado a algún usuario.
- ✓ Un usuario puede tener asignados varios roles, en ese caso prevalece el rol más restrictivo.
- ✓ Los cambios realizados en los roles no son efectivos hasta que no se inicia una nueva sesión.
- ✓ Cuando un rol niega el acceso a un módulo se pierde la posibilidad de ver cualquier subpanel del mismo.

## Autoevaluación

**La seguridad de los ERP y CRM se basa en los siguientes aspectos:**

- Controlar cada una de las transacciones de datos.
- Utilizar el protocolo http seguro.
- Establecer distintos perfiles de usuario.
- El administrador del sistema ha de confirmar las operaciones que realizan los usuarios antes de que estas sean válidas.

[Mostrar retroalimentación](#)

## Autoevaluación

**Combinando los distintos tipos de usuarios con los diferentes roles podemos crear usuarios con diferentes niveles de seguridad.**

- Falso.
- Verdadero.

## Anexo I.- Enlaces de interés

---



### Para saber más

A continuación se indican las URL de algunos videos explicativos que pueden ser de utilidad para ampliar o aclarar conceptos básicos relacionados con la Unidad:

- ✓ Introducción a XML: <https://youtu.be/bCd2xaQrTAo>
- ✓ Mi primer documento XML: <https://youtu.be/LijEw3Std1Y>
- ✓ Documentos XML bien formados: <https://youtu.be/s1RgyhO9F2I>
- ✓ Introducción a XML Namespaces: <https://youtu.be/YGFcD3-WO6c>
- ✓ Cómo se usa XML Namespaces: <https://youtu.be/2Hw-4QDySiA>
- ✓ Resumen teórico y práctico: <https://www.abrirllave.com>

# Condiciones y términos de uso de los materiales

Materiales desarrollados inicialmente por el Ministerio de Educación, Cultura y Deporte y actualizados por el profesorado de la Junta de Andalucía bajo licencia Creative Commons BY-NC-SA.



Antes de cualquier uso leer detenidamente el siguiente [Aviso legal](#)

## Historial de actualizaciones

Versión: 03.00.00	Fecha de actualización: 14/06/19	Autoría: Víctor Javier Gómez Arques
-------------------	----------------------------------	-------------------------------------

**Ubicación:** todas las asignaturas

**Mejora (tipo 1):** mas ejercicios resueltos, o ejercicios resueltos paso a paso en todas las asignaturas

temas de todas las asignaturas mejor explicados para gente que empieza desde 0 y no tiene conocimientos previos del modulo

**Ubicación:** Módulo

**Mejora (tipo 3):** En relación con el módulo, están muy lejos de lo que se pide en tareas y exámenes, se debería añadir video explicativos de los mismos profesores, y los contenidos menos técnicos y más prácticos, los alumnos debemos andar buscando solución en la web, que tampoco es que sea todo de buen nivel.

Resumiendo hacer un contenido más práctico y menos técnico.

**Ubicación:** anexo 1

**Mejora (tipo 1):** Sería de gran ayuda incluir la web <https://www.abrirllave.com> porque incluye teoría y práctica de XML, XSD, DTD,HTML.

Todo su contenido es muy resumido y directo, con ejemplos fáciles de entender en cada uno de los apartados.

**Ubicación:** 5.2

**Mejora (tipo 1):** Leyéndome los apuntes he visto que este párrafo contiene 2 veces el mismo texto.

En este caso se usa el código iso-8859-1 (Latin-1) que permite el uso de tildes o caracteres como la ñ. Otro de los códigos a emplear es el UTF-8 (Unicode, es un código de caracteres que permite el tratamiento informático de textos de múltiples lenguajes y disciplinas técnicas, ya que representa todos los caracteres conocidos para cualquier lengua. Es compatible con el código ASCII). Sin entrar en detalles, diremos que UTF-8 soporta más caracteres y permite la visualización correcta de éstos en más sistemas o lugares que iso-8859-1. A no ser que por algún motivo no sea posible el uso de UTF-8, la recomendación es siempre utilizar UTF-8, que recordemos es la codificación de caracteres seleccionada por defecto. Hay que tener precaución en el momento de guardar el documento en el editor de código, por ejemplo, si usamos Sublime Text, cuando guardemos habrá que hacerlo según la codificación elegida.). Sin entrar en detalles, diremos que UTF-8 soporta más caracteres y permite la visualización correcta de éstos en más sistemas o lugares que iso-8859-1. A no ser que por algún motivo no sea posible el uso de UTF-8, la recomendación es siempre utilizar UTF-8, que recordemos es la codificación de caracteres seleccionada por defecto. Hay que tener precaución en el momento de guardar el documento en el editor de código, por ejemplo, si usamos Sublime Text, cuando guardemos habrá que hacerlo según la codificación elegida.

**Ubicación:** En la sección 2.0,2.1,2.2,2.3,2.4 y 2.5

**Mejora (tipo 1):** Tienen información mal estructurada para el aprendizaje, en prácticamente todas las secciones mezclan datos de XML, SGML, HTML y GML, en lugar de tratarlos por separado y orden histórico o relevante en la actualidad.

Esto hace que sea un caos organizar que datos son relevantes para un determinado lenguaje porque todo se presenta mezclado y hace muy difícil asociar la información que recibes con un determinado lenguaje.

También sobran ejemplos de GML y faltan ejemplos completos de XML que no parezcan recortes de un periódico y ejercicios de práctica con solución más amplios o completos para aprender con más detalle lo que sí puntúa en las tareas.

**Ubicación:** 2.1.

**Mejora (tipo 1):** El apellido de Charles F. es Goldfarb no Goldfab.

**Ubicación:** Toda la unidad

**Mejora (tipo 3):** Unificar la unidad 7 de Sistemas de gestión empresarial a un punto dentro de la unidad 1.

La unidad 1 pasará a llamarse "Unidad 1. Aspectos básicos de los lenguajes de marcas y sistemas de gestión empresarial" y tendrá 16 horas.

**Ubicación:** 3.

**Mejora (tipo 1):** en el ejemplo, no está cerrada la etiqueta body: /body>

**Ubicación:** 5.

**Mejora (tipo 1):** Faltan etiquetas por cerrar.

reparto> debería cerrarse como /reparto>

**Ubicación:** 5.3.2

**Mejora (tipo 1):** Cuando dice El atributo irá entre comillas dobles o simples, de la siguiente forma:

nombreEtiqueta=valor>.../nombreEtiqueta>

es incorrecto, debería ser nombreEtiqueta atributo=valor>... /nombreEtiqueta>

**Ubicación:** Mapa conceptual

**Mejora (Mapa conceptual):** Actualización con la fusión de las unidades 1 y 7

**Ubicación:** Orientaciones alumnado

**Mejora (Orientaciones del alumnado):** Actualización con la fusión de las unidades 1 y 7

**Versión:**  
**02.01.00**

**Fecha de actualización:**  
**06/10/15**

**Autoría:** Víctor Javier Gómez  
Arques

**Ubicación:** Apartado 5.3.2

**Mejora (tipo 1):** El ejemplo de código no refleja correctamente el resultado, hay que sustituir la imagen del navegador.

**Ubicación:** Apartado 7

**Mejora (tipo 2):** Inclusión de dos vídeos explicativos acerca del espacio de nombres de dominio

**Ubicación:** 5.1.- El prólogo.

**Mejora (tipo 2):** La definición de standalone es incorrecta:

Informa de si el documento necesita de otro para su interpretación. Para declararlo hay que definir el prólogo completo:

```
?xml versión="1.0" encoding="iso-8859-1" standalone="no" ?>
```

En este caso, el documento es independiente, de no ser así el atributo standalone hubiese tomado el valor yes.

Lo correcto sería:

Informa de si el documento necesita de otro para su interpretación. Para declararlo hay que definir el prólogo completo:

```
?xml
```

**Ubicación:** 5.1.- El prólogo.

**Mejora (tipo 2):** La definición de standalone es incorrecta:

Informa de si el documento necesita de otro para su interpretación. Para declararlo hay que definir el prólogo completo:

```
?xml versión="1.0" encoding="iso-8859-1" standalone="no" ?>
```

En este caso, el documento es independiente, de no ser así el atributo standalone hubiese tomado el valor yes.

Lo correcto sería:

Informa de si el documento necesita de otro para su interpretación. Para declararlo hay que definir el prólogo completo:

```
?xml
```

Versión: 02.00.00

Fecha de actualización: 27/09/15

Autoría: Víctor Javier Gómez Arques

**Ubicación:** Toda la unidad

**Mejora (tipo 3):** Esta unidad de introducción se debe iniciar desde la necesidad del uso de los lenguajes de marcas y en concreto de XML, para que el alumnado pueda ubicarse en un entorno, de modo que no se encuentre perdido. Las fases de explicación deben realizarse con una progresión más sutil e incluir más ejemplos que ilustren lo explicado. Esta unidad sirve de base para entender el concepto XML y la experiencia me dice que no es suficiente para el éxito en la comprensión de las siguientes unidades.

**Ubicación:** Vista previa del cuestionario

**Mejora (tipo 1):** Pregunta 17

La etiqueta permite determinar el formato de todas las columnas agrupadas. ¿Verdadero o falso?

Seleccione una:

Verdadero

Falso

Solucion:

La etiqueta colgroup permite determinar el formato de todas las columnas agrupadas.

¿Verdadero o falso?

Seleccione una:

Verdadero

Falso

**Ubicación:** 7.- Capas o cajas

**Mejora (tipo 3):** En esta unidad falta la inclusión del uso de capas, que son lo que se realizan la mayoría de las páginas web. El uso de marcos o frames no responde a la realidad actual.

**Ubicación:** Vista previa del cuestionario

**Mejora (tipo 1):** Pregunta 17

La etiqueta permite determinar el formato de todas las columnas agrupadas. ¿Verdadero o falso?

Seleccione una:

Verdadero

Falso

Solucion:

La etiqueta colgroup permite determinar el formato de todas las columnas agrupadas.

¿Verdadero o falso?

Seleccione una:

Verdadero

Falso

**Ubicación:** 7.- Capas o cajas

**Mejora (tipo 3):** En esta unidad falta la inclusión del uso de capas, que son lo que se realizan la mayoría de las páginas web. El uso de marcos o frames no responde a la realidad actual.

**Ubicación:** No especificada.

**Mejora (tipo 1):** Falta añadir más ejemplos prácticos resueltos con hojas de estilos, y usar las capas o elementos div para crear la estructura de una página web, en lugar de los marcos o frameset que llevan desfasados muchos años.

Versión:  
01.02.01

Fecha de actualización:  
02/10/14

Autoría: Víctor Javier Gómez  
Arques

Títulos y descripción de videos explicativos. Enlaces en nuevas ventanas.

<b>Versión:</b> <b>01.01.01</b>	<b>Fecha de actualización:</b> <b>02/10/14</b>	<b>Autoría:</b> Víctor Javier Gómez Arques
Inclusión de más ejemplos y aclaración de conceptos.		
<b>Versión:</b> 01.00.00		<b>Fecha de actualización:</b> 28/04/14
Versión inicial de los materiales.		