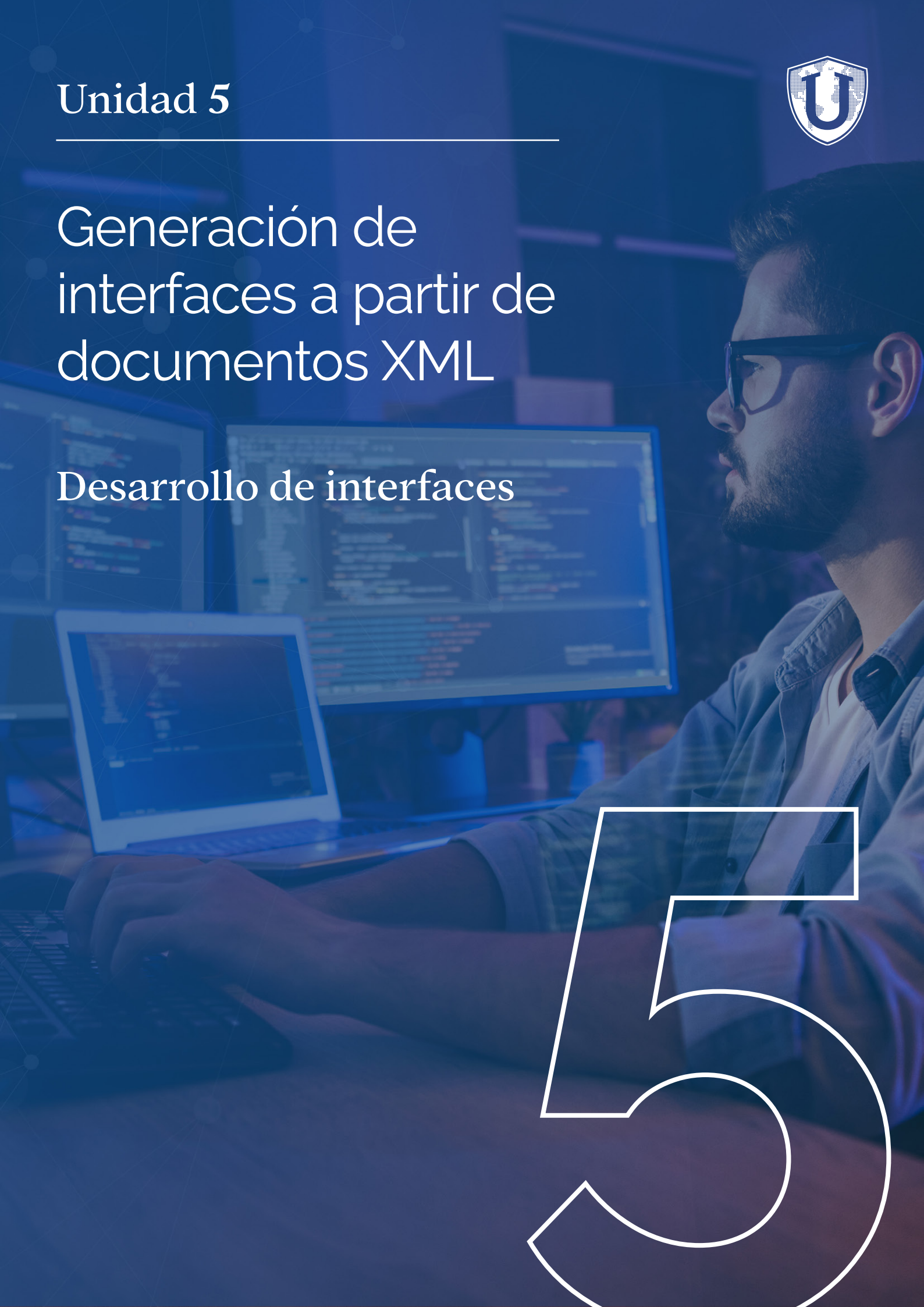
Unidad 5

Generación de interfaces a partir de documentos XML

Desarrollo de interfaces



Índice

Introducción

En este tema nos introduciremos en el lenguaje XML y sus variantes, así como en su uso para la creación de interfaces gráficas.

Distinguiremos el lenguaje XML de otros lenguajes derivados de él como son XHTML, MathML o SVG, de modo que conozcamos distintos lenguajes especializados para cada tipo de situación que funcionen bien con XML y por lo tanto no cause incompatibilidades.

Estudiaremos los tipos de componentes que conforman el lenguaje XML y sus funciones dentro de él:

- Etiquetas.
- Atributos.
- Valores.

Finalizaremos con el estudio de editores que nos permitirán trasformar nuestro lenguaje XML en una interfaz gráfica propia.

Al finalizar esta unidad

- Conoceremos los beneficios de la creación de interfaces con XML
- Estudiaremos los diferentes lenguajes derivados de XML
- Describiremos los distintos elementos del lenguaje XML
- Distinguiremos los distintos editores de XML que nos ayudarán en la creación de una interfaz.

5.1. Lenguajes de descripción de interfaces basados en XML

XML, eXtensible Markup Language, permite la compatibilidad entre diversos sistemas, lo cual lleva a un fácil y seguro intercambio de información. XML se puede implementar en múltiples plataformas. XML se puede emplear en: bases de datos, editores de texto, hojas de cálculo, etc.

En este tema se estudiará la creación de interfaces a partir de documentos XML, comenzando con el estudio de este último.

XML es un metalenguaje, y como tal se ha usado de base para construir muchos otros lenguajes, como veremos a continuación.

Los usos más conocidos de XML son:

Eficiencia y seguridad en el intercambio de información

Al ser un texto plano evita mucho de los problemas de formato, permitiendo llevar fácilmente información de una aplicación a otra.

Computación distribuida

Posibilidad de uso como medio para intercambiar información entre ordenadores a través de la web.

Información empresarial

Debido a la gran sencillez y versatilidad permite la estructuración de la información más acorde a las necesidades de la empresa.

5.1.1. XHTML

XHTML, eXtensible HyperText Markup Language, deriva de XML y posee grandes similitudes con HTML, pero posee la ventaja de tratarse de un lenguaje más robusto y, por lo tanto, más recomendado.

- Inicio obligatorio con la entrada `<!DOCTIPE>`.
- En `<html>` es obligatorio en el atributo *xmlns*.
- Uso obligatorio de:
  - `<html>`
  - `<head>`
  - `<title>`
  - `<body>`
- Elementos correctamente anidados.
- Elementos siempre cerrados.
- Siempre minúscula para los elementos y nombres de atributos.
- Citar los valores de atributos.
- Imposibilidad de minimizar atributos.

```
<!Doctype html>
<html xmlns= http://www.w3.org/1999/xhtml>
<head>
        <title>Título del documento</title>
</head>
<body>
        <h1>Título que aparecerá como cabecera</h1>
        <p>Párrafo de texto</p>
        <h2>Subtítulo en menor tamaño que el anterior</h2>
        <p>Párrafo de texto</p>
</body>
</html>
```

Imagen 1. Ejemplo de XHTML

5.1.2. GML

GML, Geography Markup Language, Se compone de marcas precedidas por dos puntos ":". Es posible determinar el tipo de formato deseado para cada línea. Se especializa en el almacenamiento de información geográfica.

```
:h1.Titulo del capítulo
:p.Párrafo de texto
```

5.1.3. MathML

MathML, Mathematical Markup Language, Es empleado por programas matemáticos como medio de intercambio de información. Por ejemplo, una ecuación como x + y = z2 quedaría de la siguiente forma:

```
<math>
    <mrow>
        <mi>x</mi>
        <mo>+</mo>
        <mi>y</mi>
        <mo>=</mo>
        <mi>z</mi>
        <mn>2</mn>
    </mrow>
</math>
```

Imagen 2. Ejemplo de MathML

5.1.4. RSS

RSS, Really Simple Syndication, es un lenguaje destinado a la difusión de información, por lo que se emplea en el anuncio de novedades o distribución de noticias mediante la sindicación de contenidos.

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
    <channel>
        <title>Titulo</title>
        <link>Link de la noticia</link>
        <description> Breve descripción de la noticia</description>
        <author>Nombre del autor</author>
    </channel>
</rss>
```

Imagen 3. Ejemplo de RSS

5.1.5. XSLT

XSLT, eXtensible Stylesheet Language for Transformations, es el recomendado para una hoja de estilo, por encima de CSS. XSLT incluye un conjunto de funciones que facilitarán el trabajo, como son:

- Añadir o eliminar elementos o atributos del archivo de salida.
- Reorganizar elementos.
- Realizar pruebas.
- Ocultar o mostrar elementos.

5.1.6. SVG

SVG, Scalable Vector Graphics. Muestra figuras geométricas vectoriales, imágenes como mapa de bits, así como texto.

```
<!DOCTYPE html>
<html>
<body>
<svg width="200" height="200">
   <circle cx="100" cy="100" r="40"
stroke="black" stroke-width="2"
fill="aqua" />
</svg>
</body>
</html>
```

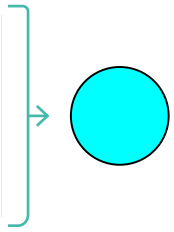


Imagen 4. Ejemplo SVG círculo

```
<!DOCTYPE html>
<html>
<body>
<svg width="200" height="300">
   <ellipse cx="100" cy="100" rx="40" ry="60"
stroke="green" stroke-width="1" fill="tomato" />
</svg>
</body>
</html>
```

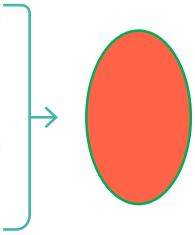


Imagen 5. Ejemplo SVG elipse

```
<!DOCTYPE html>
<html>
<body>
<svg width="200" height="300">
   <rect x="50" y="50" width="150" height="100"
fill="yellow" />
</svg>
</body>
</html>
```


Imagen 6. Ejemplo SVG rectángulo

```
<!DOCTYPE html>
<html>
<body>
<svg width="120" height="120">
   <polygon stroke="green" stroke-width="2"
fill="salmon" points="25,100 55,40 85,100" />
</svg>
</body>
</html>
```

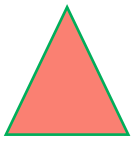


Imagen 7. Ejemplo SVG triángulo

5.2. El documento XML. Análisis y edición

XML es un fichero de texto plano que hace uso de etiquetas para la identificación de sus elementos, con el fin de que se conozca desde donde se deben extraer dichas etiquetas se debe incluir la siguiente línea.

```
<?xml version=”1.0” encoding=”utf-8”?>
```

Esta línea muestra dos elementos clave para el documento:

- `version`: muestra la versión XML que se está empleando.
- `encoding`: muestra el conjunto de caracteres empleados en el documento.

Una vez marcada la versión y el conjunto de etiquetas empleadas podemos empezar a trabajar en el documento empleando las etiquetas prefabricadas.

Es necesario saber que toda etiqueta debe poseer un cierre ya sea con una etiqueta de cierre o en la misma línea:

```
<etiqueta>_____</etiqueta>
<etiqueta _____ />
```

Otro punto necesario para el buen funcionamiento es mantener una jerarquía adecuada, cerrando las etiquetas en el momento adecuado para poder contener los elementos que se incluyen dentro de él. Igualmente, los valores deben incorporarse en sus lugares designados y correctamente demarcados, en caso contrario todo el código se puede echar a perder por un solo error, lo cual muestra la importancia de la jerarquía y la estructura en este tipo de lenguaje.

5.2.1. Etiquetas

Una de las mayores ventajas del XML es, aunque se deba seguir la estructura, la capacidad de poder definir tus propias etiquetas, lo que lleva a una organización más específica en cada documento y, por tanto, más precisa.

Con el fin de que dichas etiquetas puedan ser usadas correctamente es imprescindible seguir la estructura de manera muy rígida y rigurosa. La estructura arbórea debe ir acompañada de sus correspondientes marcas, en este caso los signos de menor que “`<`” y mayor que “`>`” son imprescindibles para delimitar las etiquetas creadas. Se debe recordar que el cierre de la etiqueta se debe marcar con la propia etiqueta con una barra “`/`”. El nombre de la etiqueta se escribirá acotado por los signos menos que y mayor que, y en el caso de cierre detrás de la barra.

Se debe tener en cuenta a la hora de escribir que el código XML es uno de los llamados case sensitive, lo que implica que son sensibles a las mayúsculas por lo que en los nombres de las marcas las mayúsculas también deben ser tomadas en cuenta.

Ejemplos de entradas y cierres.

En el caso de encontrarse en la misma línea se abriría, se escribe el texto necesario, y se cierra añadiendo la barra.

Apertura ↓ `<texto>` Hola Cierre ↓ `<texto>`

En el caso de entradas abiertas y cerradas en distintas líneas de código deben escribirse siempre a la misma altura de tabulación, añadiendo sangrías tabuladas en función de la subordinación de la línea. Esto permite visualizar las entradas y salidas de manera fácil y simple con el fin de poder ver y corregir los posibles errores.

```
<Asunto>
    <Texto_1>Hola</Texto_1>
    <Texto_2>Adiós</Texto_2>
</Asunto>
```

Cualquier marca que contenga espacios, un número como inicio, use una barra que no sea la de cierre o no siga las reglas anteriores no será válida.

Tipos de errores en XML

Incorrectas	Correctas	Causa
`<Texto></texto>`	`<Texto></Texto>`	Mayúsculas
`<1Texto></1Texto>`	`<Texto1></Texto1>`	Número inicial
`<Texto 1></Texto 1>`	`<Texto_1></Texto_1>`	Espacios
`<Texto/1></Texto/1>`	`<Texto_1></Texto_1>`	Uso de barra no final

Podemos encontrar otro tipo de etiqueta llamada etiqueta vacía, la cual se abre y cierra en una única etiqueta:

```
<Texto_1 />
```

Pueden contener atributos o valores, pero no contenido.

5.2.2. Atributos

Si un elemento contiene un elemento con una propiedad específica, como un valor asignado, se debe adjuntar en la etiqueta de apertura. La forman de añadirlo sería la siguiente: Nombre del elemento, espacio, atributo, signo de igual "=" y por último el valor o características entre comillas, si hubiese más atributos se añadirían con un espacio entre ellos. Como un ejemplo podemos ver los siguientes elementos.

```
<Texto Tipo=”12”>Hola</Texto>
<Plantas categoría=”flores”>Rosa</Plantas>
<Plantas categoría=”arbustos”/>
```

Estos atributos añaden información al XML, pero su abuso puede ser contraproducente.

NOTA

No es posible incluir en la misma etiqueta dos atributos con el mismo nombre de atributo, por lo que en casos como los apellidos se deben diferenciar con algo, como apellido1 y apellido2.

5.2.3. Valores

Los valores especifican a los atributos. Siempre poseen una misma construcción de `atributo="valor"`.

Los valores nos permiten dar significado a los atributos. En un registro de una persona seguramente aparecerán los valores como nombre, edad, etc., pero estos, por sí solos, no son más que una plantilla general vacía y sin valor, son un documento en bruto esperando a ser usado. Cuando se le añaden los valores esa plantilla pasa a ser una instancia concreta, ya no es un documento en blanco, es la ficha de una persona concreta identificada por los valores incluidos, y, por tanto, posee valor.

5.2.4. Eventos

Los eventos son acciones predefinidas que permiten la interacción entre la persona y el ordenador. Generalmente el XML no es el lenguaje empleado para la construcción de interfaces y, por lo tanto, no suele incluir eventos, pero es posible emplear el XML como base para una interfaz e incluir eventos. Los eventos más simples pueden ser la pulsación de un botón, donde se debe incluir un evento de respuesta.

- `MouseMove`: este evento se produce al colocar el puntero sobre un elemento.
- `MouseDown`: este evento se produce al pulsar los botones del ratón.
- `Change`: este evento se produce al cambiar el contenido del control.
- `Click`: este evento se produce al hacer clic con el botón izquierdo del ratón sobre el control.
- `GetFocus`: este evento se produce al poner el enfoque de sobre uno de los elementos de control, manteniéndolo en estado de espera para ejecución.
- `LostFocus`: este evento se produce al eliminar el enfoque sobre uno de los elementos de control, generalmente al pasarlo a otro elemento.

5.3. Herramientas libres y propietarias para la creación de interfaces de usuario multiplataforma

El lenguaje XML destaca por su sencillez en su escritura, haciéndolo que este sea un lenguaje accesible y fácil de usar. Existen diferentes editores con los que podemos trabajar este lenguaje como son:

- **Atom.** Herramienta multinivel que permite su uso tanto para profesionales como para novatos. Permite la personalización añadiendo lenguajes e interfaces gráficas no preestablecidas en el programa.

  Con el fin de facilitar el trabajo este se ejecuta sobre paneles que podemos modificar, pudiendo colaborar en tiempo real mediante la herramienta Teletype, es especialmente recomendada por esta última función.

- **Notepad++.** Reconoce múltiples lenguajes de programación, es gratuito y además apto para Linux y Windows, pudiendo incluso descargar su código fuente.

  Puede personalizarse y mediante el plugins XML Tools puede añadir opciones de validación para XML.

- **Adobe Dreamweaver CC.** Muestra directamente los resultados de la escritura del código XML en tiempo real, lo que nos permite modificar el documento a nuestro gusto y eliminar errores en tiempo real al tiempo que previsualizamos el resultado final. Es compatible tanto con Windows como con OS X.

- **Visual Studio Code.** Uno de los más empleados por sus grandes prestaciones. Es gratuito y compatible con Windows, Linux y macOS. Ofrece múltiples herramientas como el autocompletado de código, señalización de la sintaxis, interfaz personalizable y, mediante Live Share, la colaboración en tiempo real con otros programadores.

	Características de los editores			
	Atom	Notepad++	Dreamweaver	Visual Studio
Multinivel	✓	✓	✓	✓
Paneles	✓	✓	✓	✓
Control de versiones	✓			✓
Libre	✓	✓	✓	✓
WYSIWYG			✓	

5.4. Generación de código para diferentes plataformas

El empleo de XML elimina, o al menos reduce, uno de los mayores problemas para los desarrolladores, el intercambio de información entre aplicaciones o sistemas incompatibles.

El carácter portable de XML lo ha convertido en el lenguaje ideal para almacenar información tanto para la representación de los metadatos como para el intercambio de información. El empleo de XML se basa en tres grandes pilares:

- **Representar metadatos.** Permite una indexación y recuperación rápida y sencilla. Posee una gran capacidad de discriminación ya sea por contenido, elementos o atributos.
- **Medio para intercambio de información.** XML no solo es un lenguaje extendido, sino que muchos otros lenguajes derivan de él, haciendo de este un lenguaje compatible con muchos otros, además su uso no solo se restringe a bases de datos, ya que se puede emplear en otros propósitos como la visualización de sitios web o en la creación de mensajes para aplicaciones.
- **Almacenamiento.** Las bases de datos XML nativas son muy demandadas por su sencillez y versatilidad.

Uno de los elementos que permiten la construcción de una interfaz adecuada es el empleo de tablas, las cuales permiten la estructuración eficiente de la información y por tanto mejora la experiencia que la interfaz otorga. Estas tablas se pueden crear mediante XML y se ejecutan con una hoja de estilo como XSLT o CSS.

www.universae.com