

4.4.1.1. *Introducción a SGML****SGML como estándar de anotación textual***

El estándar SGML (*Standard Generalised Markup Language*) merece un apartado especial dentro de la lista de los distintos sistemas de marcaje de textos electrónicos, y es por ello por lo que le dedicamos una sección especial dentro de los métodos de anotación. Las ventajas ya comentadas anteriormente de poseer material textual en formato electrónico podrían reducirse a dos: (i) la posibilidad de manejar los textos mediante ordenadores, y (ii) la versatilidad que representa el tener textos en formato interpretable por (en principio) cualquier equipo informático, y, en consecuencia, cualquier usuario y/o investigador que posea un ordenador. Si bien estas dos características por sí mismas ya justifican todo esfuerzo humano, académico y (¿por qué no?) económico destinado a la informatización de material textual, el hecho de disponer de una herramienta de trabajo que sea capaz de organizar esos textos y que pueda enlazarse con otros programas de manejo textual ampliaría enormemente el horizonte del análisis textual computarizado. Así pues, SGML nace con la intención de convertirse en el estándar informático de la anotación textual.

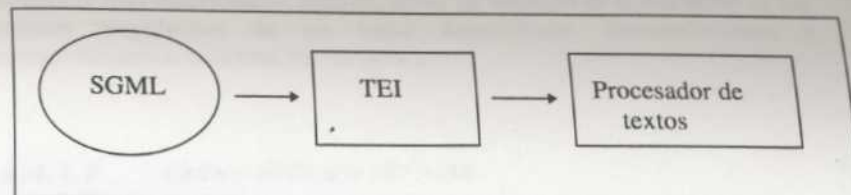
Hemos presentado en el párrafo anterior SGML como un proyecto de estandarización de la informatización textual encaminado al manejo y organización de textos. Dicha caracterización debe interpretarse como una meta que va mucho más allá de los objetivos concretos de SGML, los cuales se limitan al papel etiquetador de los aspectos más formales de un texto. Vendrán después lenguajes adicionales que complementarán los fines meramente formales de SGML (HTML, TEI, TEI Lite, etc).

SGML vs procesadores de textos

En un momento de batallas comerciales por adueñarse del usuario medio de procesadores de texto, SGML no puede vencer ni convencer a las compañías mundiales de software, las cuales en ningún momento admitirían un estándar que, por un lado, pusiera en peligro sus productos actuales, y, por otro, igualara los resultados reales de los procesadores propios y ajenos a la hora de editar, presentar e imprimir un texto. De hecho, SGML ni siquiera se plantea como una alternativa a los procesadores comerciales. Al contrario,

SGML simplemente quiere hacerse hueco en el terreno de las "recomendaciones" de anotación de elementos textuales formales, los cuales muchas veces son tan inservibles en cuanto a la maquetación definitiva de un texto como incapaces de conseguir los brillantes resultados de cualquiera de los paquetes de software existentes en el mercados. El adjetivo "formal", utilizado como calificador de los fines de SGML, cobra un nuevo significado, pues no se refiere al aspecto externo que debe tener un elemento textual determinado, sino a la clasificación con la que dicho segmento textual se debe asociar desde un punto de vista textual externo. Ilustremos lo anterior: convencionalmente, escribimos en cursiva aquellas palabras que no pertenecen al idioma del texto en el que estamos escribiendo. Así, deberíamos utilizar cursivas en todas las ocasiones en las que escribamos, por ejemplo, la voz "software" (su equivalente en español es "soporte lógico" — véase Gómez Guinovart y Lorenzo Suárez 1994). [En este trabajo no lo hacemos, pues hemos preferido "adoptar" dicha palabra en el glosario terminológico habitual de este volumen, lo cual descarta la consideración de "software" como extranjerismo.] Una anotación tipo SGML de "software" en ningún momento asignará un marcador CURSIVA a la palabra "software", aunque sí añadirá información adicional que la califique como "palabra extranjera" e incluso podrá indicar que en el texto original la palabra está enfatizada tipográficamente. Igualmente, el estándar SGML nunca pretenderá indicar que el segmento "Capítulo 2: Protocolos de anotación textual" debe aparecer en negrilla, en tipo Times New Roman de tamaño 16. Al contrario, SGML marcará dicho segmento como "título de un capítulo", y, posiblemente, actualizará un contador virtual de capítulos con el valor '2'. En definitiva, el que utilice SGML como herramienta de marcaje de textos no estará preocupado en absoluto por el aspecto final del texto electrónico que está editando, pues para ello existen fabulosos estándares comerciales o no (formatos Microsoft Word —Microsoft Corporation—, WordPerfect —WordPerfect Corporation—, PostScript —Adobe System Incorporated—, etc). El usuario de las recomendaciones SGML estará ocupado en determinar qué elementos del texto son títulos, subtítulos, extranjerismos, párrafos, citas bibliográficas, nombres propios (de personas, de lugares), gráficas, dibujos, etc. El Gráfico 2 retrata muy esquemáticamente el proceso hasta aquí descrito:

GRÁFICO 2: DE SGML A LA PRESENTACIÓN REAL



Podemos pues afirmar que SGML es una norma preocupada por la semántica del texto y no por su estética, en donde "semántica" se refiere al conjunto de conceptos formales que podemos identificar en un segmento textual (en palabras de Abaitua, "etiquetado descriptivo"), y en donde "estética" alude a la representación externa de cada uno o algunos de esos conceptos (lo que Abaitua denomina "etiquetado procedimental"). Una vez que consigamos separar la semántica y la estética en un texto, podríamos asociar los resultados de una con los de la otra. En otras palabras, podríamos disponer de una herramienta de software, más parecida a un procesador de textos o a un paquete de autoedición que a un intérprete SGML, que traduzca el marcaje SGML de una palabra o conjunto de palabras que indique "palabra extranjera" y lo exprese en cursiva, en definitiva, de una aplicación que establezca vínculos entre SGML y estándares de presentación formal de textos.

Sintaxis de referencia concreta

En el párrafo anterior hemos hecho alusión a qué se entiende por "dimensión formal" del marcaje SGML. Dediquemos unas líneas ahora a otro componente más del nombre SGML: SGML como lenguaje. Pensamos que no sería caer en sensacionalismo si afirmamos que la filosofía de SGML no es ser un lenguaje de anotación. Es más, los recursos gráficos que utiliza SGML para marcar un texto son perfectamente redefinibles por el propio sistema. La que aquí utilizaremos para presentar el estándar, la cual es prácticamente usada por toda la comunidad científica, no es más que una propuesta de formalización de contenidos SGML, denominada *Reference Concrete Syntax* (RCS), esto es, "una" sintaxis de expresión de aquello que busca expresar el anotador de SGML. Más adelante entraremos en más detalles de cómo actúa la RCS en un texto a anotar siguiendo las pautas de SGML.

SGML como "sistema" flexible

Preferiríamos, pues, caracterizar SGML como un *sistema* de marcaje de textos. Como tal sistema, incluirá un conjunto de conceptos que debemos o podemos utilizar; dispondrá de unas reglas de uso, las cuales (añadiremos) serán férreas y poco flexibles una vez definidas; tendrá unos objetivos muy concretos; será, sin embargo, flexible según las necesidades de cada usuario. No deberá extrañarnos que, al igual que el uso del estándar COCOA se adaptaba a lo que queríamos anotar en cada momento, el marcaje SGML de un texto puede variar muchísimo de un usuario a otro, tanto cuantitativa como cualitativamente. En concreto, un usuario de SGML puede hacer uso de n conceptos de marcaje de un texto, frente a otro que utiliza, por ejemplo, $n+3$ conceptos. El segundo usuario puede llegar a subclasificaciones conceptuales muy estrictas y exhaustivas de cada uno de los conceptos textuales que emplea, mientras que el primero se queda en el nivel más superficial de clasificación. Ambos están haciendo uso correcto de la norma SGML. Los resultados de ambos son interpretables por un intérprete/visor/editor/analizador/gestor o *parser* de SGML, el cual los comprobará automáticamente, esto es, sin necesidad de actuación humana más que para subsanar posibles errores humanos. En dicha comprobación el *parser* separa el documento textual y los datos del etiquetado SGML, y se cuida de que la anotación sea coherente con el DTD (*Document Type Definition*, analizado más adelante) y correcta.

La "normativa" SGML

Una vez que hemos descartado la consideración de SGML como estrictamente un lenguaje de marcaje textual, podríamos intentar definir el concepto SGML como una normativa o conjunto de normas estándar que permite a estudiosos codificar ciertos aspectos de un texto de un modo universal. El número de normas a disposición del investigador está recogido en el informe del estándar ISO 8879 de 1986. De ellas, podemos hacer uso de un conjunto reducido, o incluso ampliarlas según los intereses de cada operación de marcaje. SGML es, en definitiva, una norma estricta aunque flexible a los distintos fines de los investigadores. Otras normas, igualmente establecidas como estándares de proyección universal, como TEI (o su versión reducida TEI Lite), a la que nos referiremos más adelante, desarrollan la filosofía SGML y la centran en disciplinas determinadas. TEI representa la dimensión lingüística de la normativa general SGML, y bien podrá definirse como "lenguaje" de marcaje textual. TEI goza de la generalidad inspirada por

los principios de anotación textual SGML, aunque constreñida por un horizonte muy determinado, esto es, servir de vehículo de codificación de los valores *lingüísticos* de un texto determinado. Desarrollaremos y ejemplificaremos la norma TEI en §4.4.2.

4.4.1.2. Generalidades de SGML

El estándar SGML hace uso de cuatro conceptos básicos, que ejemplificaremos más adelante mediante la normativa concreta TEI: la "entidad" de marcaje, el "elemento" de marcaje, los "atributos" del elemento de marcaje y el "tipo de documento". Hemos preferido mantener la terminología inglesa (*markup entity*, *element*, *attribute* y *document type*) en su traducción más literal a pesar de su falta de transparencia. Pasemos a reproducir las definiciones que ofrecen de estos conceptos los manuales introductorios a SGML.

Entidad de marcaje

Una "entidad" de marcaje es un objeto concreto del texto: una línea, una palabra, un conjunto de caracteres, una tabla, un gráfico, un dibujo, una nota a pie de página, el propio documento completo, etc. Estos objetos pueden estar alojados físicamente en otro ordenador, unidad de disco, CD-ROM, etc. La utilidad de este concepto reside en el hecho de que cada entidad posee una "referencia" o nombre, que sustituirá en todo momento al objeto representado.

Por ejemplo, en TEI, siguiendo las convenciones de la RCS anteriormente mencionada, puedo representar el objeto textual "ñ", no existente en el juego alfabético de la lengua inglesa, como **ñ**. La codificación de la palabra "añorar" en un marcaje SGML sería:

ñorar

Lo mismo podríamos hacer con otras cadenas de un solo carácter, como las vocales acentuadas. Reproducimos en la siguiente tabla las equivalencias de otros signos gráficos especiales frecuentes en español, tal y como figuran en la norma ISO 8879 (1986), la cual incluye un listado de todos los conjuntos internacionales de entidades formadas por un único carácter.

entidad	etiqueta	entidad	etiqueta
á	á	Á	Á
é	é	É	É
í	í	Í	Í
ó	ó	Ó	Ó
ú	ú	Ú	Ú
ü	ü	Ü	Ü
ñ	ñ	Ñ	Ñ
ç	ç	Ç	Ç

Elemento de marcaje

Un “elemento” de marcaje en SGML es un concepto abstracto que sirve para denominar una realidad abstracta (denominada “objeto abstracto”) en el texto. Nombres, fechas, párrafos, páginas, notas a pie, etc. son objetos que no aparecen nombrados como tales en el texto, esto es, que no son “objetos concretos” (no son “entidades”), y que, sin embargo, pertenecen a la imagen que tenemos del texto. En SGML, mediante “elementos” representamos tales objetos pertenecientes a la realidad virtual del texto.

A modo de ejemplo, en RCS, los elementos de marcaje están delimitados por los siguientes caracteres: ‘<’, ‘>’, ‘</’ y ‘>’. Los dos primeros codifican la “etiqueta de apertura”, y los dos últimos la “etiqueta de cierre”. Por “etiqueta” debemos entender la denominación de un objeto abstracto del texto, esto es, un elemento. Entre otras, en TEI, existe la etiqueta **name**, que sirve para marcar nombres propios. El marcaje de *le decía don Antonio a Augusto aquella tarde* sería: [Todos los ejemplos están tomados de *Niebla*, a no ser que se indique lo contrario.]

```
le de&aacute; a don <name>Antonio</name> a <name>Augusto</name>
```

Atributo de marcaje

El concepto “atributo de un elemento de marcaje” en SGML se emplea para designar categorías de información adicional dependientes de un elemento de marcaje. Los “atributos” tienen un nombre y un valor. En RCS, a los nombres de los atributos no se les asigna delimitadores específicos, pues simplemente se separan del nombre del elemento de marcaje mediante un espacio en blanco. Por el contrario, los valores sí se introducen mediante un signo (= en RCS).

Por ejemplo, el elemento de marcaje que se utiliza para designar el objeto abstracto ‘página’ en TEI recibe la etiqueta **pb**. El elemento **pg** aislado simplemente nos indicará que entre <pb> y </pb> hay una página de texto. Esta información podría completarse si añadiésemos el número de página. Esto se realiza, en SGML, mediante el atributo **n**, que se asocia, en este ejemplo, con un valor numérico cualquiera. Así, nos referiríamos a la página 6 mediante el marcaje:

```
<pb n=6>
... Aquí estaría el contenido textual de la página 6 ...
</pb>
```

Ilustraremos la combinación de elementos y atributos más adelante en la sección dedicada a TEI y TEI Lite.

Tipo de documento

Uno de los conceptos más importantes y más dificultosos de codificar de SGML es el llamado “tipo de documento”. En el “tipo de documento” tendremos que indicar qué entidades y elementos estamos utilizando en la anotación del texto, qué atributos están asociados con cada elemento, qué valores pueden tener los atributos de los elementos de marcaje, así como las posibles reglas de asociación entre atributos y elementos. Esta sintaxis o gramática del texto conforma el “tipo de documento”. Gracias a la información ahí incorporada, el intérprete SGML puede no sólo manejar correctamente el texto, sino comprobar que la codificación interna de las entidades, elementos y atributos es correcta. En caso contrario, nos avisará del posible error, indicando la falta de coherencia entre las reglas definidas en el tipo de documento y el marcaje empleado en el texto. Además, la existencia de un tipo de documento separado y a la vez conectado electrónicamente con el documento textual facilita la comprensión de todas las etiquetas utilizadas en el marcaje del texto por cualquier usuario o lector. En el sistema COCOA se sugiere que las etiquetas estén explicadas al principio del documento dentro de un comentario u observación. La diferencia con SGML reside en el hecho de que en esta última normativa todo documento normalmente dispone de un tipo de documento (o más), de tal forma que el intérprete SGML pueda manejar correctamente el texto. Si toda la explicación, definición de entidades, elementos, atributos, valores de atributos, etc., así como las reglas de combinación de todos esos conceptos (la “gramática” del marcaje) están codificadas electrónicamente en un archivo especial, cualquier

persona que disponga del texto, y, por tanto, de su tipo de documento, podrá seguir las convenciones del editor a la perfección.

El DTD y sus "declaraciones"

Toda la información a la que hacemos alusión en el párrafo anterior se incluye en un componente de SGML, el denominado DTD o *Document Type Definition*. El DTD de un marcaje específico está formado por "declaraciones" (declaraciones de elementos de marcaje, de atributos, etc). En RCS, cada declaración está delimitada por los caracteres '<!' y '>'. Los componentes de una declaración son:

(i) especificador del tipo de declaración (**DOCTYPE** para la declaración del tipo del documento que sigue entre corchetes o en otro fichero, **ELEMENT** para la declaración de un elemento, **ATTLIST** para la declaración de los posibles atributos de un elemento, **ENTITY** para la declaración de una entidad mediante una cadena de texto más reducida, **NOTATION** para la declaración del sistema bajo el cual se debe interpretar datos no SGML, etc):

```
<!DOCTYPE (...) >
<!ELEMENT (...) >
<!ATTLIST (...) >
<!ENTITY (...) >
<!NOTATION (...) >
```

Existe un tipo especial de declaración que no se ajusta a esta sintaxis. Nos referimos a la llamada "declaración de contenido". En este caso, el anotador simplemente quiere introducir un comentario personal en el marcaje del documento. La información contenida en una declaración de contenido será desconocida para aquel que esté interesado en exclusivamente el texto final y que no revise la anotación en sí. Ya que en una declaración de contenido no existe especificador, el anotador de SGML simplemente incluye el comentario entre pares de dos guiones seguidos ('--'):

```
<!-- Este es el espacio reservado para cualquier comentario del
anotador -- >
```

Añadamos para finalizar que la convención de los dos pares de guiones puede utilizarse prácticamente en todo momento en una anotación SGML para introducir observaciones personales. El *parser* nunca procesará aquellas cadenas alfanuméricas comprendidas entre '--' y '--':

```
<!ENTITY nieb "Niebla de Miguel de Unamuno" --este es el título de
la obra-- >
```

La única restricción aplicable es el hecho de que dentro de una declaración de contenido o de un comentario (no incluido en una declaración) no se puede introducir otra.

(ii) una cadena de un máximo de ocho caracteres que expresa el nombre del concepto del que estamos estableciendo la declaración (el nombre del tipo de documento en el caso de una declaración de tipo de documento, el nombre del elemento en una declaración de elemento, el de un elemento más su(s) atributo(s) en la declaración del conjunto de atributos, el de una entidad, el de un tipo de datos no SGML, etc.): [El especificador de la declaración y el nombre del concepto de la declaración estarán separados únicamente por un espacio en blanco.]

```
<!DOCTYPE novela >
<!ELEMENT pb (...) >
<!ATTLIST pb n (...) >
<!ENTITY nieb "Niebla de Miguel de Unamuno" >
<!NOTATION jpg SYSTEM "L-View" >
```

Estas declaraciones implican, respectivamente, que el tipo de documento es **novela** (todas las declaraciones del mismo DTD que apareciesen a continuación de esta estarían encerradas entre corchetes), que el sistema de marcaje del texto correspondiente a este DTD incluye el elemento **pb**, el cual puede tener asociado el atributo **n**, que la entidad **nieb** se corresponde con la cadena *Niebla de Miguel de Unamuno*, y que aquellos objetos con anotación del tipo **jpg** deben ser leídos mediante el sistema (aplicación, en este caso) *L-View*.

(iii) en el caso de declaraciones de elementos, un tercer componente, consistente en la indicación de cuáles de las etiquetas delimitadoras (apertura y cierre) son necesarias, mediante el uso del guión '-' cuando son obligatorias y la letra 'o' cuando no lo son. Estos dos indicadores estarán separados por un espacio en blanco intermedio, así como por sendos espacios de separación con respecto a los segundos y cuartos constituyentes de la declaración. A modo de ejemplo, la siguiente declaración

```
<!ELEMENT pb - - (...) >
```

indica que se utilizará el elemento **TEI pb**, el cual deberá estar comprendido entre una etiqueta de apertura y una etiqueta de cierre, ambas obligatorias (indicadores '-' y '-'). Por el contrario, la declaración

```
<!ELEMENT name - o (...) >
```

no obliga al uso de una etiqueta de cierre (indicador 'o') y sí al de una de apertura (indicador '-').

(iv) en declaraciones de listas de atributos, un cuarto componente, consistente en el tipo de valor que puede tener el atributo. La codificación del tipo de valor del atributo está separada del componente segundo de la **ATTLIST** mediante un espacio en blanco. Veremos a continuación dos clases de contenidos de atributos:

· contenidos predefinidos: Muy posiblemente un atributo determinado de un elemento de marcaje sólo puede tener un número concreto de valores. Por ejemplo, si definimos una lista de atributos del elemento **datos** que incluya el atributo **sexo**, expresaremos en la declaración que los valores posibles de dicho atributo serán exclusivamente **femenino** y **masculino**, y que, además, ambos son excluyentes, esto es, el valor del atributo **sexo** ha de ser bien **masculino** bien **femenino**. El operador que expresa una relación disyuntiva entre valores (más adelante veremos que también entre elementos de marcaje en una declaración de **ELEMENT**) es la barra vertical '|'. La lista de los valores "a elegir" aparecerá entre paréntesis. La declaración quedaría definida de este modo:

```
<!ATTLIST datos sexo (femenino|masculino) (...) >
```

· contenidos no predefinidos: Imaginémonos que el atributo del elemento de marcaje en cuestión puede tener por valor cualquier cadena de caracteres. Utilizaremos en la declaración el comodín **CDATA**, que sustituye cualquier valor alfanumérico. Pongamos un ejemplo: si el atributo **n** del elemento **pb** (página) puede tener cualquier valor (por ejemplo, 35, A-2, II.345, 14bis, etc.), una posible declaración de la lista de atributos de **pb** podría ser la siguiente:

```
<!ATTLIST pb n CDATA (...)>
```

La cadena de caracteres definida por el comodín **CDATA** (semejante al **#PCDATA** en la definición de declaraciones de elementos) admite tanto caracteres numéricos como alfabéticos, es decir, cadenas alfanuméricas en general.

El tipo de valor **NUMBER** restringe la clase de caracteres asociados con un atributo concreto, y exige que esos valores sean exclusivamente numéricos. A modo de ejemplo, en la declaración de la lista de atributos del elemento **datos** arriba introducida, podríamos incluir el valor **edad**, el cual sólo podría expresarse mediante una cadena numérica. La declaración global de los atributos de **datos** sería la que sigue:

```
<!ATTLIST datos sexo (femenino|masculino) (...)
              edad NUMBER (...) >
```

En esta declaración constan las siguientes reglas de sintaxis: (i) el elemento de marcaje **datos** puede tener dos atributos: **sexo** y **edad**; (ii) el atributo **sexo** sólo puede tener uno de los dos valores siguientes: **femenino** o **masculino**; y (iii) el atributo **edad** debe estar asociado con un valor numérico. Un ejemplo correcto de un segmento de texto que incluya el marcaje del elemento **datos** sería el siguiente:

```
<datos>
  <sexo>femenino</sexo>
  <edad>29</edad>
</datos>
```

Por el contrario, el intérprete SGML indicaría que en el ejemplo siguiente existen dos errores de gramática, de DTD, o de sintaxis:

```
<datos>
  <sexo>var&oactute;n</sexo>
  <edad>adolescente</edad>
</datos>
```

En primer lugar, el atributo **sexo** del elemento **datos** está asociado con el valor *varón*, que está excluido de la lista de los posibles valores de **sexo** en la declaración del DTD. En segundo lugar, el valor del atributo **edad** no es numérico, sino alfabético (*adolescente*), lo cual estaría rechazado por la declaración de la **ATTLIST** de **datos** anteriormente definida.

(v) en declaraciones de elementos de marcaje, un cuarto y último componente, consistente en el llamado "modelo de contenido". El modelo de contenido aparece siempre entre paréntesis ('(', ')'), separado del tercer constituyente de la declaración mediante un espacio en blanco. Los

componentes del modelo de contenido son sub-elementos que bien obligatoria bien opcionalmente aparecen en el interior del elemento de la declaración. Cuando son opcionales en la declaración del elemento de marcaje están seguidos por un signo de interrogación '?', mientras que si son obligatorios no están seguidos directamente por ningún carácter. Un elemento comodín en SGML que se puede incluir en el modelo de contenido de un elemento de marcaje es **#PCDATA** (*Parsed Character Data*), el cual se utiliza para expresar que cualquier cadena de caracteres puede aparecer como contenido de un elemento de marcaje particular. Ilustremos todos los conceptos aquí descritos mediante la siguiente declaración del elemento **TEI pb**:

```
<!ELEMENT pb - - (#PCDATA)>
```

Como ya sabemos, el elemento de marcaje **pb** está delimitado obligatoriamente por una etiqueta de apertura y por otra de cierre, y en su interior podemos encontrar cualquier cadena de caracteres. Esta declaración permitiría la existencia de una página compuesta por, por ejemplo, un carácter, dos líneas de caracteres, la combinación de caracteres alfanuméricos y un gráfico, etc., lo cual es deseable a la hora de caracterizar el concepto "página" en un programa intérprete SGML.

Definamos a continuación un nuevo elemento, **apellidos**, el cual debe incluir información sobre los dos apellidos de una persona. En otras palabras, el elemento **apellidos** asociado con el individuo Augusto Pérez Bueno contendrá las voces **Pérez** y **Bueno**, las cuales son, a su vez, **nombres**. Una posible declaración del elemento **apellidos** sería

```
<!ELEMENT apellidos - - (nombre, nombre)>
```

Para que esta declaración pueda ser utilizada en el caso de personas con un único apellido, podríamos especificar que la segunda aparición de **nombre** es opcional:

```
<!ELEMENT apellidos - - (nombre, nombre?)>
```

lo cual se puede indicar con mayor simplicidad mediante el signo '+' añadido a los componentes del modelo de contenido. El signo '+' indica que el elemento al que sigue puede repetirse una o varias veces:

```
<!ELEMENT apellidos - - (nombre+)>
```

Con los componentes que conocemos podríamos intentar una declaración más compleja. Indiquemos, pues, la declaración completa correspondiente al elemento de marcaje **identidad**, el cual estará formado por únicamente el nombre de una persona, por el nombre y su apellido, o por el nombre y sus apellidos. Procedamos a expresar las declaraciones de los elementos que emplearemos en la declaración general de **identidad**, esto es, las declaraciones de nombre y apellidos. La primera puede estar formada por cualquier cadena de caracteres:

```
<!ELEMENT nombre - - (#PCDATA)>
```

La segunda, que hace uso del elemento de marcaje **nombre**, puede estar formada por uno o más nombres:

```
<!ELEMENT apellidos - - (nombre+)>
```

El elemento **identidad** podría tener entonces la siguiente declaración:

```
<!ELEMENT identidad - - (nombre, apellidos?)>
```

Esta declaración permitiría los siguientes contenidos del elemento **identidad**: *Augusto*, *Augusto Pérez* y *Augusto Pérez Bueno*, tal y como se deseaba en el enunciado de la tarea. Otra posible solución haría uso del signo '*', que indica que el elemento al que sigue puede aparecer ninguna, una o varias veces, esto es, cero veces o más. Gracias a este signo, podríamos prescindir de la declaración del elemento **apellidos**, y utilizar únicamente el elemento **nombre** a la hora de sugerir la declaración del elemento de marcaje **identidad**. A modo de ejemplo, la declaración

```
<!ELEMENT nombre - - (#PCDATA)>
<!ELEMENT identidad - - (nombre, nombre*)>
```

define, en primer lugar, el elemento de marcaje **nombre** como la cadena de caracteres comprendida entre las etiquetas obligatorias **<nombre>** y **</nombre>**, y, en segundo lugar, el elemento **identidad** como el conjunto de un **nombre** obligatorio y uno o varios opcionales. Esta declaración es tan flexible como la anterior de **identidad**, aunque más económica, pues no requiere la declaración especial de **apellidos**. Una pregunta para terminar: ¿cuál sería el efecto de la declaración **<!ELEMENT identidad - - (nombre+)>?**

Sabemos que el modelo de contenido puede establecer un conjunto de elementos de marcaje obligatorios, opcionales, una serie infinita de uno de ellos, o incluso una serie de cero o más apariciones de un elemento. Introduzcamos ahora un operador más en el modelo de contenido. Se trata del operador lógico "o" (la barra '|' entre los elementos del modelo de contenido, que ya habíamos introducido en la declaración de los posibles atributos de un elemento de marcaje), que nos permitirá seleccionar entre una serie cerrada de elementos de marcaje. Pongamos un ejemplo: utilicemos el elemento código (**codigo**), que nos servirá para codificar a una persona mediante un número o nombre. Hagamos una declaración de **codigo** tal que permita asignar la etiqueta **codigo** al DNI (elemento **dni**) de la persona, a su NIF (elemento **nif**), o a su nombre y apellido(s). Una posible declaración del elemento **codigo** sería:

```
<ELEMENT nombre - - (#PCDATA)>
<ELEMENT identidad - - (nombre+)>
<ELEMENT dni - - (#PCDATA)>
<ELEMENT nif - - (#PCDATA)>
<ELEMENT codigo - - (dni|nif|identidad)>
```

El contenido de **codigo** será bien el **dni**, a su vez compuesto por una cadena determinada de caracteres, bien el **nif**, también consistente en una cadena cualquiera de caracteres, o bien el elemento **identidad**, formado por uno o varios nombres. Esta relación disyuntiva entre los posibles elementos de marcaje definidores de **codigo** se expresa, como ya hemos indicado, mediante la barra '|'. Si no fuese necesario definir en el texto los contenidos estrictos de **dni**, **nif** e **identidad**, la declaración del elemento **codigo** pudiera ser la siguiente:

```
<ELEMENT codigo - - (#PCDATA)>
```

que coincidiría, a efectos prácticos, con la propuesta anterior de declaración.

(vi) En las declaraciones de la lista de atributos posibles de un elemento de marcaje, el quinto y último lugar está ocupado por el valor por defecto del atributo. Este valor por defecto puede bien estar definido en la declaración mediante una cadena concreta de caracteres, bien obviarse, o bien especificarse como valor de inclusión obligatoria. En el primero de los casos, se especifica claramente el valor por defecto. En el segundo, se utiliza el comodín **#IMPLIED**, y, en el tercero, la expresión **#REQUIRED**.

Ejemplifiquemos todo lo anterior mediante la siguiente declaración de atributos:

```
<!ATTLIST datos sexo (femenino|masculino) masculino
edad NUMBER #REQUIRED
direccion CDATA #IMPLIED>
```

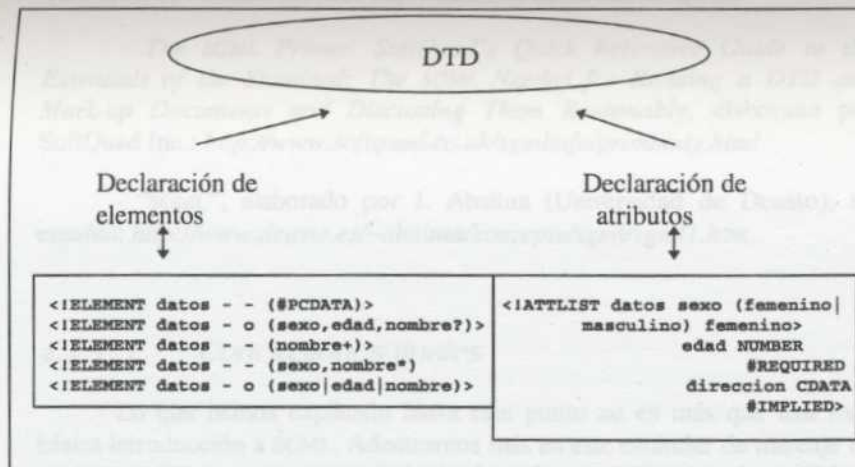
El atributo **sexo** del elemento de marcaje **datos** puede tener únicamente los valores **femenino** y **masculino**. En el caso de que no se especifique ningún valor, se entenderá que este es **masculino**. El atributo numérico **edad** deberá tener obligatoriamente un valor; de lo contrario, el intérprete SGML emitirá un informe de error. Finalmente, la declaración **ATTLIST** permite que el atributo **direccion**, que puede estar asociado con cualquier valor alfanumérico, no esté explícitamente definido en el texto. La siguiente porción de texto ilustra la definición de un elemento **datos**, así como sus atributos:

```
Estos son los datos de Augusto Péscute;rez Bueno:
<datos>
  <sexo></sexo>
  Tiene <edad>34</edad> a&ntilde;os
  Vive en <direccion>Vigo, Avda. Camelias</direccion>
</datos>
```

En dicho texto, el intérprete SGML asume que el sexo de *Augusto Pérez Bueno* es **masculino**, pues así fue indicado en la declaración del atributo **sexo**; el valor (correcto) de **edad** es 34, y el de **direccion** es *Vigo, Avenida Camelias*. Este último podría obviarse sin mayores problemas.

En el siguiente gráfico realizamos un resumen de todo lo explicado hasta el momento con respecto a la forma de expresar las declaraciones de elementos y atributos en el DTD:

GRÁFICO 3: RESUMEN DE LAS DECLARACIONES DE ELEMENTOS Y ATRIBUTOS



4.4.1.3. Breves referencias sobre elementos electrónicos relacionados con SGML.

En este apartado, nos limitaremos a recomendar ciertos paquetes de distribución pública y gratuita (normalmente, disponibles a través de Internet), destinados a un público principiante en el mundo SGML, lugares en los que se puede encontrar material textual marcado mediante el estándar SGML. Información actualizada muy completa sobre aplicaciones SGML puede encontrarse en la WWW: <http://www.sil.org/sgml/publicSW.html>.

· Gestores SGML (programas que permiten visionar un texto marcado según el estándar SGML, así como escribir marcaje siguiendo dicha normativa): El gestor SGML más utilizado, sin duda, es *Panorama* de SoftQuad. Puede encontrarse la versión gratuita limitada de *Panorama Viewer* en <http://www.softquad.co.uk>. También gratuitos son los gestores SGML de James Clark (*sgmls*, *nsgmls*, *SP parser toolkit*, etc), disponibles desde <http://www.jclark.com/sp/index.htm> [más información sobre gestores en <http://www.sil.org/sgml/publicSW.html>]. El BNC posee su propio intérprete de SGML, a saber, SARA (véase Aston y Burnard 1997).

· Algunas bibliotecas electrónicas SGML y TEI (textos en inglés y español) [más información en Referencias]:

- *Oxford Text Archive* (ver referencias finales)
- *The SGML Repository* en Oslo, Noruega (Departamento de Informática): <ftp://ftp.ifi.uio.no/pub/sgml>
- *The SGML Project Exeter* en Exeter, Gran Bretaña: <ftp://info.ex.ac.uk/pub/sgml>
- UIC: <ftp://ftp-tei.uic.edu/pub/tei>
- Grupos de noticias y listas de discusión electrónica sobre SGML- TEI:
 - grupo de noticias comp.text.sgml
 - *SGML-L*: Para suscribirse, envíese el mensaje **subscribe sgml-1** a listserv@vm.urz.uni-heidelberg.de
 - *SGML list*: Para suscribirse, envíese el mensaje **join sgml** a mailbase@mailbase.ac.uk
 - *SGML-FAQ-L*: Para suscribirse, envíese el mensaje **SUB SGML-FAQ-L** a listserv@listserv.aol.com
 - *TEI-L*: Para suscribirse, envíese el mensaje **subscribe tei-1** a listserv@uicvm.uic.edu

4.4.1.4. Para saber más sobre SGML (información en línea disponible en Internet)

Destacaremos las siguientes publicaciones electrónicas, parte de las cuales hemos utilizado para elaborar esta mínima introducción a SGML. Por razones de versatilidad, rigor y adecuación al lector novel, listamos a continuación cuatro trabajos que consideramos clásicos, atendiendo a criterios de preferencia personal, de accesibilidad y de conveniencia:

- "A Gentle Introduction to SGML", capítulo 2 de *TEI P3 Guidelines*, elaborado por el comité TEI:
 - en formato HTML: <http://sable.ox.ac.uk/ota/teip3sg>, o desde el servidor WWW de UIC <http://www-tei.uic.edu/orgs/tei/sgml/teip3sg/index.html>,
 - en formato de texto: vía *ftp*, de UIC: <ftp://ftp-tei.uic.edu/pub/tei/doc/p3sg.doc>
 - en formato de hipertexto: desde el servidor de *Electronic Book Technologies (EBT)*: <http://www.ebt.com/usrbooks/teip3/1336> o de la Universidad de Virginia (UVA): <http://etext.virginia.edu/bin/tei-tocs>

· *What is SGML and How Does It Help?*, elaborado por Lou Burnard (Oxford): en formato HTML: <http://sable.ox.ac.uk/ota/teiedw25/>

· *The SGML Primer. SoftQuad's Quick Reference Guide to the Essentials of the Standard: The SGML Needed for Reading a DTD and Mark-up Documents and Discussing Them Reasonably*, elaborado por SoftQuad Inc.: <http://www.softquad.co.uk/sgmlinfo/primbody.html>

· "SGML", elaborado por J. Abaitua (Universidad de Deusto), en español: <http://www.deusto.es/~abaitua/konzeptu/sgml/sgml1.htm>

4.4.1.5. Conclusiones finales

Lo que hemos explicado hasta este punto no es más que una muy básica introducción a SGML. Adentrarnos más en este estándar de marcaje no tendría sentido en este manual dedicado a las posibilidades de análisis y codificación automatizados de textos. Por otro lado, la enorme complejidad de explorar el diseño de un DTD que contemple un buen número de posibilidades de marcaje precisaría un volumen completo. De hecho, TEI o *Text Encoding Initiative* surge como una salida a la necesidad de disponer de un DTD flexible y estándar, específico para la anotación de textos desde la perspectiva de un lingüista. Nos detendremos en la próxima sección en los principios básicos de TEI y su versión abreviada TEI Lite.

4.4.2. TEI

4.4.2.1. Introducción a TEI

Historia de TEI

Como ya hemos indicado en §4.4.1, SGML ha supuesto la creación de un estándar de anotación textual que permite un alto grado de flexibilidad por parte del usuario. Sin embargo, el desarrollo de elementos de marcaje y sus atributos manejados por el editor del texto implica la realización de un DTD personal para cada ocasión. Desde esta perspectiva, la existencia de un estándar de anotación tenía que ir acompañada por la de un estándar de DTD aplicable a la anotación de cualquier material textual. Surge así TEI

(*Text Encoding Initiative*) como una propuesta universal de marcaje textual acorde con la filosofía que inspiró la propuesta de SGML.

Fue en el congreso de la *Association for Computers and the Humanities* (ACH) de 1987, celebrado en Poughkeepsie (Nueva York), cuando se planteó la necesidad de crear un estándar terminológico de anotación SGML. En este congreso se enunciaron los llamados "principios de Poughkeepsie", que más adelante darían lugar a la propuesta TEI.

En 1990, un grupo de estudiosos publicaron el primer borrador de TEI, denominado documento TEI-P1. A este le seguiría el TEI-P2, cuya versión definitiva vio la luz en 1993. Tras numerosas revisiones, la mayoría provenientes de instituciones ajenas al comité TEI, se publicó en Chicago y Oxford simultáneamente la primera introducción oficial a TEI (ya no un borrador) en 1994, también llamada TEI-P3, *Guidelines for Electronic Text Encoding and Interchange*.

Versatilidad de TEI-SGML

Las normas TEI sirven para anotar información no sólo textual, sino sonora y visual también. Todos aquellos estudiosos que se acojan al estándar TEI podrán intercambiar dicha información fácilmente. Como hemos indicado anteriormente, TEI hace uso de la filosofía SGML, de ahí que cualquier herramienta de software que sea capaz de procesar textos preparados para SGML podrá también manejar textos anotados bajo TEI sin dificultad alguna.

TEI Lite

Ya que en este volumen no podemos hacer otra cosa que introducir muy brevemente el modelo TEI, en las secciones que siguen nos limitaremos a una versión reducida, más que suficiente para los propósitos habituales (así esperamos) de los lectores de este manual.

4.4.2.2. Introducción a TEI Lite

La porción de las *Guidelines* generales de TEI a la que nos referiremos se conoce con el nombre de TEI Lite. [Una versión de TEI aún más reducida que TEI Lite es la denominada "Bare Bones TEI". Más información puede encontrarse en el documento TEI U6 "Bare Bones TEI: A

Very Very Small Subset of the TEI Encoding Scheme", escrito por C.M. Sperberg-McQueen, disponible en <http://www.uic.edu/orgs/tei/intros/teiu6.html>.) Centros de textos electrónicos como el *Oxford Text Archive* de la Universidad de Oxford (Gran Bretaña) o los centros de las Universidades de Virginia y Michigan (Estados Unidos de Norteamérica) utilizan TEI Lite para codificar sus textos o para traducir los marcajes originales de sus archivos. Es más, los propios documentos de TEI guardan escrupulosamente los principios de TEI Lite.

Los objetivos de TEI Lite, tal y como aparecen en la información publicada en la WWW <http://www-tei.uic.edu/orgs/tei/intros/teiu5.html> (documento TEI U5), son los siguientes:

- TEI Lite debería incluir la mayoría de las etiquetas del grupo básico de TEI, ya que este contiene elementos relevantes potencialmente para todos los tipos de textos y de trabajos relacionados con la edición de textos.

- TEI Lite debería ser capaz de manejar adecuadamente una razonable variedad de textos, y de conseguir detalles de marcaje ya utilizados en materiales disponibles (por ejemplo, en el *Oxford Text Archive*).

- TEI Lite debería ser útil tanto para la creación de nuevos textos anotados como para el marcaje de textos ya existentes.

- TEI Lite debería ser utilizable por una amplia gama de software SGML ya disponible.

- TEI Lite debería poderse derivar del DTD completo de TEI mediante los mecanismos descritos en las *Guidelines* de TEI.

- El conjunto de normas de TEI Lite debería ser lo más reducido y simple que sea posible, siempre y cuando se cumplan los objetivos anteriores.

Adentrémonos ahora en las entrañas de TEI Lite, con el único propósito de disponer de una visión general e introductoria de la filosofía de la normativa TEI. Finalicemos esta sección con la exposición de los principios básicos de TEI Lite. Más adelante, en §§ 4.4.2.3, 4.4.2.4 y 4.4.2.5, mediante resúmenes esquemáticos, introduciremos algunas de las convenciones más empleadas de la norma.

Principios básicos de TEI Lite

Un documento anotado bajo la norma TEI Lite, esto es, que se adapte al DTD de TEI Lite (selección del DTD general de SGML/TEI) [DTD de TEI Lite incluido en Apéndice 1], necesita obligatoriamente los siguientes elementos: una indicación de la versión utilizada de TEI, la descripción bibliográfica del texto, la autoría del marcaje, y, obviamente, el texto en sí.

En TEI Lite, se emplea el elemento **TEI.x** (donde **x** es la versión de TEI utilizada) para referirse al objeto abstracto constituido por el texto anotado y todos sus detalles. En consecuencia, todo texto anotado siguiendo las normas de, por ejemplo, la versión 2 de TEI, ha de empezar con la etiqueta de apertura **<TEI.2>** y ha de terminar con la de cierre **</TEI.2>**. Los detalles de la procedencia del texto original, así como de la autoría de la anotación, están incluidos en el elemento **teiHeader**, al que nos referiremos más adelante. Finalmente, el texto, junto con toda la anotación, constituye el elemento **text**. De un modo más gráfico, la estructura de un texto anotado bajo el estándar TEI Lite (o el propio TEI) sería la siguiente [A pesar de que la filosofía SGML es contraria a la elaboración de etiquetas inalterables y apoya la definición individualizada por cada DTD, por las razones expuestas en la introducción, en este manual adoptaremos las convenciones más generalizadas y operativas de textos marcados mediante TEI en el mundo anglosajón.]:

```
<TEI.2>
  <teiHeader></teiHeader>
  <text></text>
</TEI.2>
```

Colección de textos

Sería posible agrupar varios textos en un mismo elemento. Para ello, TEI propone el nombre **teiCorpus**, que comprende una colección de textos anotados bajo el estándar TEI. El elemento **teiCorpus** no sustituye al elemento **TEI.x** ya revisado, sino que ocupa un lugar superior en la jerarquía de elementos. Es más, las *Guidelines* permiten que el corpus de textos TEI tenga una cabecera común (elemento **teiHeader** ya introducido). Imaginémosnos un corpus de cuentos infantiles. El esqueleto del marcaje sería el siguiente:

```
<teiCorpus>
  <teiHeader>
    [Aquí información sobre el título de la colección, datos del marcaje, etc.]
```

```

</teiHeader>
<TEI.2>
  <text>
    [Aquí aparecería el cuento número 1]
  </text>
</TEI.2>
<TEI.2>
  <text>
    [Aquí aparecería el cuento número 2]
  </text>
  (etc.)
<teiCorpus>

```

4.4.2.3. Principales etiquetas de TEI Lite

(a) la cabecera de TEI Lite (elemento **teiHeader**)

Un texto TEI Lite está formado por una cabecera mediante la cual el texto (elemento **text**) queda plenamente identificado. En ella especificaremos datos como quién transcribió el texto electrónico, edición del original, detalles de la codificación, etc. Cualquier persona que disponga de la cabecera de un texto electrónico tendrá la información necesaria para situar correctamente dicho texto.

Con el fin de clasificar ordenadamente los datos referenciales del texto electrónico, cada **teiHeader** está formada por varias secciones, de las cuales sólo la primera es obligatoria. A continuación expondremos un sucinto resumen de las divisiones más relevantes de la cabecera TEI:

(i) descripción bibliográfica del texto: elemento **fileDesc**. Posibles subelementos:

titleStmt	título y autoría [véase <title> , <author> , entre otros, en §4.4.2.5]
editionStmt	detalles de la edición [véase <edition>]
publicationStmt	detalles de la publicación [véase <publisher> , <pubPlace> , <date>]
sourceDesc	procedencia bibliográfica [véase <bibl>]

(ii) metodología de la transcripción: elemento **encodingDesc**. Algunos subelementos relevantes:

<projectDesc>	comentario sobre la finalidad de la codificación del corpus [elemento similar: <samplingDesc>]
<refsDecl>	comentario con información específica sobre (algunas de) las etiquetas empleadas y sus referentes

(iii) caracterización bibliográfica del material textual: elemento **profileDesc**. Algunos subelementos importantes:

<creation>	información sobre el origen del texto [véase <date> , <name> , <place>]
<textClass>	clasificación del texto mediante alguna tipología bibliográfica [véase <list>]

(iv) historial de la transcripción: elemento **revisionDesc**. [véase **<date>**, **<item>**]

(b) el texto TEI Lite (elemento **text**)

Un texto TEI Lite está formado obligatoriamente por el elemento **body**, que contiene el material textual en sí. El elemento **body** puede estar precedido por una sección en la que se incluirían títulos, dedicatorias, índices iniciales, prefacios, etc. Dicha sección, que iría delante del elemento **body**, recibe el nombre de **front**. Igualmente, **body** puede ir seguido del elemento **back**, en el que el editor transcribirá índices finales, apéndices, listados bibliográficos, etc. Recordemos que tanto **front** como **back** son elementos opcionales.

La estructura anteriormente reproducida en el párrafo introductorio de esta sección quedaría de la siguiente forma:

```

<TEI.2>
  <teiHeader></teiHeader>
  <text>
    <front></front>
    <body></body>
    <back></back>
  </text>
</TEI.2>

```

El elemento **body** puede estar sustituido por el elemento **group**, en aquellos casos en los que hay varios textos asociados con la misma cabecera TEI (**teiHeader**), misma sección introductoria (**front**) y misma sección final (**back**). Cada uno de los componentes del elemento **group** (elementos **text**) pueden disponer de sus propios **front** y **back**. Ilustremos lo anterior mediante el libro imaginario *Lingüística de corpus*, formado por dos monografías, cada una de las cuales posee su propio índice y listado de referencias bibliográficas:

```
<TEI.2>
  <teiHeader>
    Lingüística de corpus
  </teiHeader>
  <text>
    <front>
      Índice del libro
    </front>
    <group>
      <text>
        <front>
          Índice de monog. 1
        </front>
        <body>Cap. 1</body>
        <back>
          Referencias de monog. 1
        </back>
      </text>
      <text>
        <front>
          Índice de monog. 2
        </front>
        <body>Cap. 1</body>
        <back>
          Referencias de monog. 2
        </back>
      </text>
    </group>
    <back>Referencias del libro</back>
  </text>
</TEI.2>
```

El elemento front

Elementos típicos de **front** (véanse elementos generales en §4.4.2.5):

- <div0> primer nivel de subdivisión del **front**
- <div1> segundo nivel de subdivisión del **front**

<div2> tercer nivel de subdivisión del **front**, etc.

El elemento body

Nos limitaremos a presentar un listado de elementos habituales que pueden aparecer en **body**, así como sus atributos más utilizados (véanse igualmente elementos generales en §4.4.2.5):

Elementos (grupo primero):

- <l> línea de verso (no necesita etiqueta de cierre)
- <lg> unidad formal formada por versos (stanza, etc.)
- <p> párrafo (no necesita etiqueta de cierre)
- <div0> primer nivel de subdivisión del **body**
- <div1> segundo nivel de subdivisión del **body**
- <div2> tercer nivel de subdivisión del **body**, etc.

Elementos (grupo segundo):

- <head> cabecera de cada una de las divisiones explicadas en el grupo primero de elementos. Las cabeceras, como en el caso de las *header*, contienen títulos, epígrafes, etc.
- <trailer> elementos textuales especiales que aparecen a continuación de las divisiones a las que nos hemos referidos en el primer grupo de elementos, normalmente compuesto por títulos finales, agradecimientos, etc.

Elementos (grupo tercero):

- <sp> intervención de un personaje en una pieza teatral
- <speaker> nombre del personaje que habla [atributo: **who**]
Atributo de <speaker> (véanse atributos universales en §4.4.2.4):

who El valor de este atributo contiene un código que permite reconocer al personaje que interviene en la pieza teatral.

Ejemplifiquemos los elementos de este último grupo mediante el siguiente fragmento de *Niebla* (cap. XXXIII) [adaptación a las convenciones teatrales]:

```
Augusto
    ¡Aquí estoy otra vez!
Miguel
    ¿A qué vienes?

<sp who=Augusto><speaker>Augusto</speaker>
<p>Aquíacute; estoy otra vez</sp>
<sp who=Miguel><speaker>Miguel</speaker>
<p>A quéacute; vienes</sp>
```

<stage> acotaciones teatrales [atributo: **type**]

El elemento **back**

Elementos típicos de **back** (véanse elementos generales en §4.4.2.5):

<div0> primer nivel de subdivisión del **back**

<div1> segundo nivel de subdivisión del **back**

<div2> tercer nivel de subdivisión del **back**, etc.

4.4.2.4. Una selección de atributos universales en TEI Lite

id El valor consiste en un código de identificación de la unidad correspondiente al elemento asociado con este atributo. A modo de ejemplo, si quisiésemos identificar la primera subdivisión de la unidad textual bajo análisis como "NIEBLA01", de tal forma que dicho segmento textual pudiese ser citado como "NIEBLA01" en futuras anotaciones dentro del documento, definiríamos la codificación del siguiente modo, utilizando el atributo general TEI **id**:

```
<div0 id=NIEBLA01></div0>
```

n

El valor consiste en un código numérico que sirve para numerar la unidad indicada por el elemento dentro de una posible lista. Por ejemplo, el capítulo primero del documento estaría representado por la siguiente anotación:

```
<div0 type=chapter n='1'></div0>
```

type

tipo; valores, que aparecerán entre comillas simples cuando coincidan con la terminología exacta de la división del documento [por ejemplo, **<div0 type='chapter'></div0>**]:

volume (volumen), **book** (libro), **chapter** (capítulo), **poem** (poema), **part** (sección), **stanza**, **act** (acto), **scene** (escena), etc. en el caso de divisiones textuales tipo **div** o unidades formales marcadas mediante el elemento **lg**

entrance (entrada), **exit** (salida), **setting** (escenario), etc. cuando el elemento asociado con el atributo es **stage**

abbreviated (abreviado), **main** (principal), **subordinate** (ligado a divisiones textuales mayores), etc. cuando el elemento asociado con el atributo es, por ejemplo, **title**

spoken (estilo directo), **thought** (monólogo), etc. en el caso de las citas (elemento **q**)

4.4.2.5. Una selección de elementos generales en TEI Lite

<abbr> abreviatura [atributos: **expan**]

Atributos:

expan expresión completa correspondiente a la abreviatura

<add> inserción de texto realizada por el autor, escriba, anotador, corrector, etc. [atributos: **hand**, **place** (véase **<note>**)] [elemento opuesto: **<gap>**]

Atributos:

hand autor de la inserción

<address> dirección postal, en la que cada una de las líneas de información pertenecen a elementos **<addrLine>** [elementos asociados: **<addrLine>**]

<addrLine> cada una de las líneas de una dirección postal [véase **<address>**]

<appendix> apéndice

<author> autor [véase **<bibl>**]

<bibl> referencias bibliográficas [incluye normalmente **<author>**, **<date>**, **<editor>**, **<publisher>**, **<pubPlace>**, **<title>**]

<bibliography> referencias bibliográficas, normalmente compuestas por elementos **<bibl>** [elementos asociados: **<bibl>**]

<corr> forma corregida en un pasaje [atributos: **sic**, **resp**] [véase **<sic>** como elemento similar desde la perspectiva opuesta]

Atributos:

sic forma errónea original

resp responsable (editor, transcriptor) de la corrección

<date> fecha [atributos: **value**]

Atributos:

value normalización de la fecha mediante (normalmente) estándares como aaaa-mm-dd, aaaa-mm o aaaa (en donde 'a' indica un

carácter del dígito que expresa el año, 'm' uno del mes y 'd' uno del día):

Me casé; el <date value='1997-08-30'>30 de agosto de 1997</date>

**** elemento superfluo [atributos: **type**]:

Este es un ejejemplo

<docDate> fecha de publicación [véase **<docImprint>**]

<docImprint> datos bibliográficos incluidos en una página de títulos que acompañan a los títulos propiamente dichos [véase **<titlePage>**, **<docTitle>**, **<publisher>**, **<pubPlace>**, **<docDate>**]

Ejemplo de un **<docImprint>**: 488395

<docImprint>
 <publisher>Ediciones Cásacute;tedra</publisher>
 <pubPlace>Madrid</pubPlace>
 <docDate>1983</docDate>
 </docImprint>

<docTitle> título completo del documento, normalmente formado por varias líneas de títulos (título/s principal/es y subtítulo/s [véase **<titlePage>**, **<titlePart>**, **<docImprint>**])

<editor> editor [véase **<bibl>**]

<emph> segmento tipográficamente destacado dentro del documento por razones de énfasis lingüístico [elemento similar **<hi>**]

<figure> gráfico

<foreign> segmento escrito en un idioma distinto del general del documento [atributos: **lang**]

Atributo:

lang idioma utilizado en el objeto textual anotado mediante el elemento de marcaje **foreign**; valores: **fra** (francés), **deu** (alemán), **lat** (latín), etc.

- <gap>** omisión de texto [atributos: **desc**, **resp** (véase **<corr>**), **reason**] [elemento opuesto **<add>**] [elemento similar ****]
- Atributos:
- desc** descripción de la omisión:
- ```
<gap desc='continuacion de la cita'>
```
- reason** causa de la omisión:
- ```
<gap reason='mancha de tinta en el original'>
```
- <gloss>** definición de otro objeto textual [atributos: **target**]
- Atributo:
- target** identificación del objeto textual referido por el elemento **gloss**
- <glossary>** glosario
- <hi>** segmento tipográficamente destacado dentro del documento, sin interés de especificar la razón de la distinción gráfica [atributos: **rend**] [elemento similar **<emph>**]
- Atributos:
- rend** modo tipográfico mediante el que el segmento está destacado:
- ```
<hi rend='sc'>tei</hi> (TEI está en versalita)
```
- <index>** índice
- <item>** cada uno de los componentes de una lista [véase **<list>**]
- <l>** línea de verso (no necesita etiqueta de cierre si coincide con una línea real en el documento con anotación)
- <lb>** corte de línea
- <list>** lista de componentes o *items* [atributos: **type**; valores específicos: **"ordered"** (los *items* están numerados mediante dígitos), **"bulleted"** (los *items* están ordenados

mediante letras consecutivas), **"unordered"** (lista no ordenada o no numerada)] [véase **<item>**]

- <lg>** unidad formal formada por versos (stanza, etc.) [atributo: **type**]
- <milestone>** corte de una unidad textual [atributo: **type**]. Los elementos **pb** y **lb** constituyen casos particulares de **milestone**, que bien pueden sustituirse por anotaciones más complejas como:
- ```
<milestone type=pagina>
<milestone type=linea>
```
- <name>** nombre propio [atributos: **value**, **type**; valores específicos: **person** (persona), **place** (lugar), **organization** (organización), **ship** (barco), **county** (provincia), etc.]
- <note>** nota [atributos: **type**, **resp**, **place**]
- Atributos (véanse atributos universales en §4.4.2.4):
- resp** responsable de la nota; valores: **author** (autor del documento o división), **editor** (del documento o división)
- place** lugar de la anotación en el texto original; valores: **interlinear** (entre líneas), **left** (izquierda), **right** (derecha), **foot** (a pie-de página), **end** (al final), **margin** (al margen)
- <num>** número [atributos: **type**; valores específicos: **cardinal**, **ordinal**, **fraction** (fracción), **percentage** (porcentaje)]
- <p>** párrafo (no necesita etiqueta de cierre si coincide con un párrafo real en el documento con anotación)
- <pb>** corte de página
- <ptr>** llamada a una referencia textual dentro del mismo documento, creada mediante **<ref>** [atributos: **target** (see **<gloss>**)]

Incluimos un ejemplo del uso de estos dos elementos a continuación:

Sobre el concepto "parsing", véase;ase especialmente la <ref target=sec223>sección 2.2.3</ref>
(...)
En la <ptr target=sec223>tendrás información sobre "parsing".

<publisher> editorial [véase <docImprint>, <bibl>]

<pubPlace> lugar de publicación [véase <docImprint>, <bibl>]

<q> cita [atributos: **who**, **type**]

<ref> define una referencia textual en el documento, que puede ser llamada mediante <ptr> posteriormente [atributos: **target** (véase <gloss>) más arriba, **type**] [elemento similar: **seg**] [véase <ptr>]

<s> oración principal [atributos: **type**; valores específicos: **declarative** (declarativa), **interrogative** (interrogativa), etc.]

<sic> marca de texto supuestamente incorrecto [atributos: **corr**, **resp** (véase <corr>)] [véase <corr> como elemento equivalente desde la perspectiva opuesta]

Atributos:

corr corrección del texto erróneo

<table> tabla (disposición en filas y columnas)

<term> término que pertenece a una jerga determinada

<time> hora [atributos: **value**]

Atributos:

value normalización de la hora mediante estándares como hh:mm:ss, hh:mm o hh (donde 'h' representa un dígito de la hora, 'm' de los minutos y 's' de los segundos):

Me casé; a la <time value="13:00">una de la tarde</time>

<title> título [atributos: **level**, **type**] [véase <bibl>]

Atributos (véanse atributos universales en §4.4.2.4):

level tipo de documento que recibe el título marcado mediante el elemento **title**; valores: **m** (de una monografía), **s** (de una serie), **j** (de una publicación periódica), **u** (de un documento no publicado), **a** (de un artículo), etc.

<titlePage> página que contiene títulos [véase <docTitle> y <docImprint>]

<titlePart> cada una de las secciones tipográficas en las que se divide un título o <docTitle> [atributos: **type**; valores específicos: **'main'** (título principal), **'sub'** (subtítulo)] [véase <titlePage>, <docTitle>]

Por ejemplo, una título con la siguiente apariencia podría ser anotado mediante el código TEI incluido más abajo:

Análisis textual computarizado Una introducción a TACT

```
<titlePage>
<docTitle>
<titlePart type='main'>Análisis textual
computarizado</titlePart>
<titlePart type='sub'>Una introducción a
TACT</titlePart>
</docTitle>
</titlePage>
```

<unclear> transcripción incierta de una secuencia [atributos: **reason** (véase <gap>), **resp** (véase <corr>)]

En Apéndice 2 incluimos un esquema de los elementos y atributos descritos anteriormente.