



Apellidos, Nombre: \_\_\_\_\_ Calificación: \_\_\_\_\_

## PRÁCTICA

Se desea construir un lenguaje de características similares al HTML utilizado para la transferencia de documentos en Internet. Nuestro lenguaje, al que llamaremos UML (*Universitas Markup Language*) queda definido mediante la siguiente gramática:

```
pagina      : listaTextos
              ;
listaTextos : /* Epsilon */
              | listaTextos texto
              ;
texto       : listaPalabras
              | tag
              | '/'
              | ':'
              | FTP
              | HTTP
              | TELNET
              | GOPHER
              ;
listaPalabras : CHARACTER
               | palabraReservada
               | listaPalabras CHARACTER
               | listaPalabras palabraReservada
               ;
palabraReservada : NUMERO
                  | '.'
                  ;
tag               : NEGRITA listaTextos CIERRA_NEGRITA
                  | CURSIVA listaTextos CIERRA_CURSIVA
                  | REFERENCIA protocolo path fichero ':' listaTextos CIERRA_REFERENCIA
                  ;
protocolo        : FTP servidor
                  | HTTP servidor
                  | TELNET servidor
                  | GOPHER servidor
                  | /* Epsilon */
                  ;
servidor         : NUMERO '.' NUMERO '.' NUMERO '.' NUMERO
                  ;
path             : '/'
                  | path listaPalabras '/'
                  ;
fichero         : listaPalabrasRegistrar
                  ;
listaPalabrasRegistrar : CHARACTER
                          | '.'
                          | NUMERO
                          | listaPalabrasRegistrar CHARACTER
                          | listaPalabrasRegistrar NUMERO
                          | listaPalabrasRegistrar '.'
                          ;
```

en la que se tiene la siguiente relación entre token y expresión regular:

Lexema	Token
"<b>"	NEGRITA
"</b>"	CIERRA_NEGRITA
"<i>"	CURSIVA
"</i>"	CIERRA_CURSIVA
"<a>"	REFERENCIA
"</a>"	CIERRA_REFERENCIA
"ftp://"	FTP
"http://"	HTTP
"telnet://"	TELNET
"gopher://"	GOPHER

Responder a las siguientes cuestiones:

a) ¿Cuáles de las siguientes cadenas se pueden reducir a **servidor** y por qué sí o por qué no?

- 2222.57.0.64554
- 512.32.55
- 150.214..122

b) ¿Cuáles de las siguientes cadenas se pueden reducir a **fichero** y por qué sí o por qué no?

- Hola.jpg
- hola:hola.jpg
- hola@hola.jpg
- hola<a>hola.jpg

c) ¿Cuáles de las siguientes cadenas se pueden reducir a **path** y por qué sí o por qué no?

- /uno/ftp://dos/tres/
- /uno/dos

d) Cuando se introduce esta gramática en PCYACC (LALR(1)) se producen conflictos reduce/reduce en la regla:

```
tag : REFERENCIA protocolo path fichero ':' listaTextos CIERRA_REFERENCIA
```

- ¿Cuál es el motivo concreto de tales conflictos?
- Supuesto que tenemos plena libertad para modificar la gramática, ¿cómo se podrían solucionar tales conflictos?

Como puede observarse en la gramática, se reconocen los **tag** anidados, como p.ej.:

<b> Esto es <b> un ejemplo </b> de tag anidado </b>

Construir los programas Lex y Yacc asociados a la gramática anterior que realicen las siguientes acciones:

- e) Producir un eco (copia del fichero de entrada en la salida) en el que se hayan eliminado los tags anidados (sólo los lexemas de los tokens **NEGRITA**, **CURSIVA**, **REFERENCIA** y sus respectivos cierres). Ver ejemplo. El anidamiento de los tags de tipo **REFERENCIA**, se considera un error semántico y se debe visualizar un mensaje de error cualquiera.
- f) Una vez realizado el eco anterior, debe visualizarse un resumen del contenido del fichero UML, resumen que debe contener:
- A qué protocolos se ha hecho referencia.

- A qué ficheros se ha hecho referencia.
- A qué servidores se ha hecho referencia.

Por último, responder a la siguiente cuestión:

g) Los no terminales `listaPalabrasRegistrar` y `listaPalabras` ¿reconocen los mismos lexemas? En caso afirmativo, ¿qué necesidad hay de tener dos no terminales que reconocen los mismos lexemas?

### Ejemplo de entrada:

```
Este es un texto sin nada
<b> Esto es una prueba de negrita </b>
<i> Esto es una prueba de cursiva </i>
<a>http://150.214.108.97/pantera /tau ceti, 23/:catunambu.jpg: Este es el otro texto </a>
<a>http://150.214.108.97/pantera /tau ceti, 23/:catunambu222.jpg: Este es el otro texto </a>
<a>http://205.123.65.21/pantera /tau ceti, 23/:catunambu.jpg: Este es el otro texto </a>
Este es un texto sin nada

Esto es un texto ftp:// comprometido
<b>esto es <b> para que quite <b> los internos </b> no? </b></b>
<b> Ahora esta bien </b>
<i>esto es <i> para que quite <i> los internos </i> no? </i></i>
<i> Ahora esta bien </i>
<a>/:Alfanovaftp: //Este es un texto cualquiera <a>/:mal :</a></a>
Y sigo por aqui
```

### y la salida correspondiente:

```
Este es un texto sin nada
<b> Esto es una prueba de negrita </b>
<i> Esto es una prueba de cursiva </i>
<a>http://150.214.108.97/pantera /tau ceti, 23/:catunambu.jpg: Este es el otro texto </a>
<a>http://150.214.108.97/pantera /tau ceti, 23/:catunambu222.jpg: Este es el otro texto </a>
<a>http://205.123.65.21/pantera /tau ceti, 23/:catunambu.jpg: Este es el otro texto </a>
Este es un texto sin nada

Esto es un texto ftp:// comprometido
<b>esto es  para que quite  los internos  no? </b>
<b> Ahora esta bien </b>
<i>esto es  para que quite  los internos  no? </i>
<i> Ahora esta bien </i>
<a>/:Alfanovaftp: //Este es un texto cualquiera No se pueden anidar los links.:mal :</a>
Y sigo por aqui
Protocolos referenciados:
  HTTP
Ficheros referenciados:
  catunambu.jpg
  catunambu222.jpg
  Alfanovaftp
  mal
Servidores referenciados:
  150:214.108.97
  205:123.65.21
```