



Sintaxis y Semántica del Lenguaje

Trabajo Práctico Integrador: Diseño e implementación de Lexer y Parser

Ciclo lectivo: 2022

Equipo docente:

Director de Cátedra

Nombre y Apellido: Ing. Gabriela P. TOMASELLI

Categoría docente: Profesora Asociada

e-mail: gabriela.tomaselli@gmail.com

Docentes

Ing. Gabriela P. TOMASELLI
gabriela.tomaselli@gmail.com

e-mail:

Ing. Rodrigo VIGIL

e-mail: rodrigovigil@gmail.com

Ing. Nicolas G. Tortosa

e-mail: nicotortosa@gmail.com

Ing. Juliana Torre

e-mail: julitorre025@gmail.com



Contenido

Indice de Contenido

1. Introducción 1	3
Competencias	3
Objetivo	3
2. Elaboración y entrega	3
3. ¿Qué es RSS?	4
a) Componentes léxicos o tokens	4
Componentes léxicos o tokens. estructura básica de un documento RSS	4
Etiquetas para estructura del documento	4
b) Etiquetas	5
versión XML:	5
versión RSS:	6
channel	6
Item	6
Image	6
Link	6
4. Control de errores	7
5. Modos de ejecución del intérprete	8
• Modo interactivo	8
• Ejecución desde un archivo	8
6. Documentación del trabajo	8
7. Presentación y Criterios de evaluación	9
Documentación: 40 %	9
Funcionamiento del intérprete (software): 60 %	9
8. Ejemplos	10
Básico con elementos obligatorios	10
Completo con elementos opcionales	10
9. Enlaces Interés	11



1. Introducción 1

Competencias

El presente trabajo práctico integrador pretende desarrollar las siguientes competencias de la asignatura:

Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes de programación.

Capacidad de diseño, y habilidad instrumental necesaria para llevar a cabo proyectos de cierta envergadura, cuya complejidad exige la utilización de conocimientos adquiridos en diversas asignaturas a lo largo de la carrera.

Aptitud y actitud para trabajo en grupo.

Objetivo

Utilizando algún lenguaje de programación o algún generador de lexer y parser para determinado lenguaje de programación deberán construir los analizadores léxico y sintáctico que permita analizar, validar y transformar un archivo en formato RSS.

Puntualmente la utilidad construida recibe un archivo en formato RSS y deberá indicar si esta bien construido (adecuado al estándar, sin errores) de otra manera indicar los errores; adicionalmente transformar el contenido del documento en un documento HTML válido.

El analizador lexicográfico es un módulo que recibe una secuencia de caracteres que componen el programa a analizar y lo convierte lógicamente en una secuencia de tokens. El analizador sintáctico recibe la secuencia de tokens que le entrega el analizador lexicográfico y verifica que la secuencia pueda ser generada por la gramática del lenguaje.

Hay dos grupos de herramientas que se pueden usar para generar los analizadores:

1) Se utilizan expresiones regulares y autómatas finitos para el análisis lexicográfico y la técnica LALR para el análisis sintáctico. Ejemplos de esto son lex y yacc, que generan código C o C++, o JLex y CUP, que generan código Java. flex y bison son implementaciones libres y gratuitas de lex y yacc.

2) El otro utiliza la técnica LL(k) tanto para el análisis léxico como para el sintáctico, generando parsers descendentes recursivos. Ejemplos son JavaCC, que genera código Java, y ANTLR, que está escrito en Java pero puede generar código Java, C++ python o C#. ANTLR se puede conseguir en

<http://www.antlr.org/>.

https://en.wikipedia.org/wiki/Comparison_of_parser_generators

2. Elaboración y entrega

• Modo de realización del trabajo

El trabajo se debe realizar de forma grupal en grupos de hasta cuatro (4) alumnos, donde cada integrante deberá presentar una parte del trabajo durante la presentación final.

Cada grupo estará representado por un delegado, encargado de subir las entregas parciales y el trabajo final al campus virtual, enviará consultas o dudas, etc.

Cada grupo tendrá un docente designado como tutor encargado del seguimiento del trabajo, responder inquietudes.

• Presentaciones.

Se prevén tres instancias de presentación:

- . 1er entrega: Primera documentación del proyecto y gramática a generar.

Domingo 24 de Abril de 2022



. 2da entrega: Presentación del lexer o scanner que reconozca los tokens del lenguaje.

Domingo 29 de Mayo de 2022

. 3er entrega final: Presentación de trabajo completo. Lexer y parser, incluye la presentación del trabajo ante la clase con una exposición de no más de 20 minutos.-

Domingo 3 de Julio de 2022

• Metodología de entregas parciales y final.

o Un archivo comprimido deberá ser “subido” a la tarea correspondiente dentro del curso en el campus virtual.

o Dicho archivo comprimido deberá contener:

directorio *doc*. Documentación del trabajo (véase el apartado nº 6)

directorio *src*. Archivos fuente del proyecto (flex, bison, python, archivos de C (“.c”, “.h”))

directorio *bin*. Archivos binarios listos para ser ejecutados. (lexer y scanner)

directorio *prueba*. Archivos de ejemplo de pseudocódigo con la extensión “.e”

3. ¿Qué es RSS?

RSS significa Really Simple Syndication (sindicación realmente simple sería una pobre traducción ;)

Con RSS es posible distribuir contenido web actualizado desde un sitio web a otros sitios web en todo el mundo.

RSS permite navegar rápidamente por canales de noticias y sus actualizaciones, dado que define, estandariza y facilita la forma de compartir y distribuir el contenido (titulares y contenido) de cualquier sitio web-

Los archivos RSS se pueden actualizar automáticamente

RSS está escrito en XML. por ello debe cumplir:

- Como todo documento XML se usa para describir, almacenar y transmitir o transportar información .
- Todos los elementos deben tener una etiqueta de cierre.
- Los elementos distinguen entre mayúsculas y minúsculas.
- Los elementos deben estar correctamente anidados.
- Los valores de los atributos siempre deben estar entre comillas dobles.

a) Componentes léxicos o tokens

Componentes léxicos o tokens. estructura básica de un documento RSS

Etiquetas para estructura del documento

Para crear un archivo RSS la estructura básica es la siguiente:



```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">

<channel>
  <title>RSS de la cátedra de Sintaxis y Semántica de Lenguajes </title>
  <link>https://frre.cvg.utn.edu.ar/course/view.php?id=399</link>
  <description>Sintaxis y Semántica de Lenguajes de la U.T.N.
  F.R.Resistencia. </description>

  <item>
    <title>Planificacion 2022</title>
    <link>https://</link>
    <description>Planificacion de catedra, con cronograma de clases y
    evaluaciones</description>
  </item>

  <item>
    <title>Guia de Trabajos practicos</title>
    <link>https://</link>
    <description>Guía de ejercicios propuestos a resolver en clase
    practica</description>
  </item>

  <item>
    <title>Enunciado TPI</title>
    <link>https://wl</link>
    <description>Trabajo práctico integrador</description>
  </item>

</channel>

</rss>
```

Donde el significado de los elementos es el siguiente:

- La primera línea del documento, la declaración XML.
- La siguiente línea es la declaración RSS.
- La siguiente línea contiene el elemento <channel>. Este elemento se utiliza para describir la fuente RSS, título, url, última actualización, imagen, etc.
- Dentro de <channel> hay uno o más elementos <item>, que son los que definen un artículo o "historia".
- Finalmente, las dos últimas líneas cierran los elementos <channel> y <rss>.

Tipos de Datos:

- Número
 - Se utilizarán números enteros,
 - Todos ellos serán tratados conjuntamente como números.
- Cadena
 - Estará compuesta por letras, números, signos de puntuación, caracteres especiales,
 - Aparecen dentro de los principales elementos , títulos, descripción, categoría, etc.
"Ejemplo de cadena"
"Ejemplo de cadena con salto de línea \n y tabulador \t"
- URL
 - Los únicos caracteres permitidos en URL son letras, números, guión medio, guión bajo y punto. además de los caracteres reservados: # , / , : , & , ? , =

b) Etiquetas

Importante, Todas las etiquetas son sensibles a mayúsculas y debe respetarse como tal. Los el valor de los



atributos deben encerrarse entre comillas dobles.

versión XML:

- RSS es un subtipo de archivo XML, por tanto debe indicarse la versión que se utiliza y codificación.
- El elemento <xml>, aparece por única vez, y no tiene etiqueta de cierre. `<?xml version="1.0"?>`
- Define la versión XML y como atributo (opcional), la codificación de caracteres utilizada en el documento. En este caso, el documento se ajusta a la especificación 1.0 de XML y utiliza el conjunto de caracteres UTF-8.
 - `<?xml version="1.0" encoding="UTF-8"?>`

versión RSS:

- Etiqueta que Identifica que se trata de un documento RSS y que versión utiliza (en este caso, RSS versión 2.0).
- RSS surgió en 1997 como **scriptingNews 0.90** pero utilizaremos la versión 2.0 por ser simple y estricta.
- Es el contenedor para todas las demás etiquetas y tiene su propia etiqueta de cierre.

channel

- El elemento <channel>, aparece por única vez, tiene tres elementos secundarios obligatorios:
 - <title>: define el título del canal (p. ej., página de inicio de la cátedra)
 - <link>: define el hipervínculo al canal, *ver etiqueta link más abajo.*
 - <description>: descripción del canal (por ejemplo, RSS de la cátedra de Sintaxis y Semántica de lenguajes)
- Además posee otros elementos secundarios opcionales, de los cuales vamos a destacar e implementar los siguientes:
 - <category>: define una categoría para agrupamiento
 - <copyright>: informa sobre los autores y derechos o restricciones.
 - <image>: una imagen que describe al canal. *Ver etiqueta de image más abajo.*
- *Nota: Otros elementos opcionales no se solicitan en el trabajo, pero si quieren pueden implementarlos.*

Item

- Cada elemento <channel> puede tener **uno o más** elementos <item>.
- Cada elemento <item> define un artículo o "historia" en el feed RSS.
- El elemento <item> tiene tres elementos secundarios obligatorios:
 - <title>: define el título del elemento (por ejemplo, Enunciado TPI)
 - <link>: define el hipervínculo al elemento (por ejemplo, <https://www.>) *ver etiqueta link más abajo.*
 - <description>: describe el elemento (p. ej., Enunciado del TPI 2022)
- Además posee otros elementos secundarios opcionales, de los cuales vamos a destacar e implementar los siguientes:
 - <category>: define una categoría para agrupamiento
- *Nota: Otros elementos opcionales no se solicitan en el trabajo, pero si quieren pueden implementarlos.*



Image

- Se puede incluir una imagen por canal.
- El elemento `<image>` tiene tres elementos secundarios obligatorios:
 - `<url>` : Define la URL a la imagen, (GIF, JPEG or PNG) *ver etiqueta link más abajo.*
 - `<title>` : Define el texto a mostrar si la imagen no puede ser cargada
 - `<link>` : Define el hipervínculo al elemento. *ver etiqueta link más abajo.*
- Además posee otros elementos secundarios opcionales, de los cuales vamos a destacar e implementar los siguientes:
 - `<height>`: define altura de la imagen. máximo es 400
 - `<width>`: define ancho de la imagen. máximo es 144

Link

- El elemento `<link>` permite incluir una URL.
- Una URL (**Uniform Resource Location**) es la dirección concreta de un recurso en Internet.
- La sintaxis completa de una URL es la siguiente:

protocolo://dominio:puerto/ruta#fragmento

Se detalla el significado de todos los elementos

- Los únicos caracteres permitidos en URL son letras, números, guión medio, guión bajo y punto. además de los caracteres reservados: `#`, `/`, `:`
- El **protocolo o esquema de red**. Hace referencia al nombre de protocolo de red necesario para poder alcanzar el recurso al que hace referencia la URL. Protocolos habituales son:
 - **http://** (para recursos de la web)
 - **https://** (para recursos de la web contenidos en un servidor seguro)
 - **ftp://** (recursos contenidos en un servidor de ficheros)
 - **ftps://** (recursos contenidos en un servidor de ficheros seguro)

Tras el protocolo se indican dos puntos, tras los cuales normalmente se indican dos barras para indicar la máquina.

- **Nombre de dominio**. Nombre completo en Internet de la máquina (o la red) que posee el recurso en forma de nombre de dominio. no distinguen entre mayúsculas y minúsculas.
- **Puerto**. Opcional. Puerto por el que se debe conectar con el servidor para obtener el recurso. Si no se indica (que es lo habitual) se toma el puerto por defecto. Por ejemplo en *http* el puerto por defecto es el *80*. Si queremos usar uno en particular se indica tras el servidor poniendo dos puntos y el puerto.
- **Ruta**. Opcional. Indica el recorrido dentro de la máquina remota que hay que hacer a través de los directorios para llegar al recurso que queremos. Se pone después del servidor. Ejemplos:
 - */index.html* Accede a la página index.html situada en el directorio raíz.
 - */imagenes/paisajes/foto001.jpg* Accede a la imagen foto001.jpg dentro del directorio paisajes dentro, a su vez, del directorio imagenes.
- **Localizador interno**. Opcional. Va detrás del símbolo `#` y sirve para indicar un identificador (que puede ser más o menos complejos) que permita localizar o seleccionar una parte concreta del recurso destino de la URL.



4. Control de errores

El intérprete deberá controlar errores:

- Léxicos:
 - Etiquetas faltantes o mal escritas.
 - Utilización de símbolos no permitidos.
 - Etc.
- Sintácticos:
 - etiquetas que aparecen en lugares no debidos,
 - Ocurrencia o
 - Elementos con atributos incompatibles.
 - Etc.
 - Observación : Se valorará la utilización de “reglas de producción de control de errores” que no generen conflictos.
- De ejecución
 - archivo de entrada inexistente o con una extensión incorrecta.
 - Etc.

5. Modos de ejecución del intérprete

El intérprete se podrá ejecutar de dos formas diferentes:

• **Modo interactivo**

- o Se escribira el archivo manualmente desde un terminal de texto
- o Se utilizará el carácter de fin de archivo para terminar la ejecución: *Control + D*

• **Ejecución desde un archivo**

- o Se interpretarán las sentencias de un archivo pasado como argumento desde la línea de comandos
- o El archivo deberá tener la extensión “.rss”
rss.exe ejemplo.rss

6. Documentación del trabajo

Se deberá elaborar un informe técnico relatado con sus palabras el trabajo desarrollado, consistente de un vídeo resumiendo y promocionado el trabajo junto a un documento de texto con las siguientes características:

- Portada
 - o Título del trabajo desarrollado
 - o Nombre y apellidos de las personas que forman el grupo
 - o Nombre de la asignatura
 - o Nombre de la carrera
 - o Primer cuatrimestre
 - o Curso académico: 2022
 - o Universidad y regional
 - o Lugar y fecha
- Índice
 - o Las páginas deberán estar numeradas.
- Introducción
 - o Breve descripción del trabajo realizado y de las partes del documento.
 - o Descripción de cómo se implementó la solución.
 - o Información y requerimientos de software para ejecutar y recompilar el tp (versiones de compiladores, herramientas, plataforma, etc).
- Lenguaje de pseudocódigo
 - o Se corresponde con el apartado no 3 de este documento



Componentes léxicos

Sentencias

o *Observación: Si se ha ampliado la cantidad de etiquetas entonces se deberá indicar en este apartado.*

- Gramática

- o Descripción de cada regla de la gramática libre de contexto a utilizar.

- # Símbolos de la gramática

- # Símbolos terminales (componentes léxicos)

- # Símbolos no terminales

- #Reglas de producción de la gramática

- # Se valorará la inclusión de gráficos explicativos.

- Análisis léxico

- o Descripción del archivo del lexer para definir y reconocer los componentes léxicos.

- Análisis sintáctico:

- o Descripción del archivo del parser utilizado para definir la gramática libre de contexto.

- Funciones auxiliares

- o Se deben indicar y describir las funciones auxiliares que se hayan codificado.

- Modo de obtención del intérprete

- o Nombre y descripción de cada archivo utilizado

- Modo de ejecución del intérprete

- o Interactiva

- o A partir de un archivo

- Ejemplos

- o Al menos se deben proporcionar dos ejemplos.

- o Se valorará la cantidad, originalidad y complejidad de los ejemplos propuestos.

- o También se puede incluir el ejemplo propuesto por el docente y/o otros grupos.

- Conclusiones:

- o Reflexión sobre el trabajo realizado.

- o Puntos fuertes y puntos débiles del intérprete desarrollado.

- Bibliografía o referencias web.

- Anexos

- o Se podrían incluir aquellos anexos que se consideren oportunos para mejora la calidad de la documentación

7. Presentación y Criterios de evaluación

Documentación: 40 %

Se tendrá en cuenta lo indicado en el apartado no 6.

- o El código elaborado deberá estar documentado.

- o Se valorará la inclusión de gráficos o figuras.

- o Se valorará la cantidad, originalidad y complejidad de los ejemplos propuestos.

- o También se valorará:

- # la acentuación,

- # la corrección ortográfica

- # y la calidad y claridad de la redacción.

- o Conocimiento que cada integrante del grupo aporte durante la exposición final del trabajo.

Funcionamiento del intérprete (software): 60 %

- o El intérprete deberá:

- # funcionar correctamente tanto de forma interactiva como ejecutando las

instrucciones desde los archivos de ejemplo .



Indicar como salida si el análisis fue exitoso (archivo correctamente codificado, sin errores) , en otro caso indicar los errores existentes (indicando tipo de error , numero de línea y cadena que genero el error)

Traducir el documento, generando un archivo de texto HTML ,transformando algunas etiquetas RSS (XML) en etiquetas HTML. Formateado de la siguiente manera:

- Nombre de archivo: igual al archivo fuente RSS pero con extensión HTM o HTML.
- Título del canal : encerrados entre tags: <H1>
- Descripción de canal: párrafo común: <p>
- Enlace del canal: encerrado entre tags: <a> de enlace.
- Títulos del ítem : encerrados entre tags: <H3>
- Descripción de ítem: párrafo común: <p>

en particular, deberá ejecutar correctamente el ejemplo propuesto por el grupo y dos ejemplos más: uno al azar de los presentados por otros grupos y otro ejemplo elaborado por la cátedra.

#Ejemplo utilizado por la cátedra: " "

o Se valorará

- # la completitud del lenguaje de pseudocódigo.
- # La calidad en el diseño del lenguaje y la gramática.
- # El control de errores.
- # La ampliación de elementos del lenguaje.

Observación:

Además, se valorará la asistencia a clase de prácticas y la resolución de dificultades encontradas durante la elaboración del trabajo.

8. Ejemplos

Básico con elementos obligatorios

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">

<channel>
  <title>RSS de la cátedra de Sintaxis y Semántica de Lenguajes </title>
  <link>https://frre.cvg.utn.edu.ar/course/view.php?id=399</link>
  <description>Sintaxis y Semántica de Lenguajes de la U.T.N. F.R.Resistencia. </description>

  <item>
    <title>Planificacion 2022</title>
    <link>https://</link>
    <description>Planificacion de catedra, con cronograma de clases y evaluaciones</description>
  </item>

  <item>
    <title>Guia de Trabajos practicos</title>
    <link>https://frre.cvg.utn.edu.ar/mod/resource/view.php?id=43544</link>
    <description>Guía de ejercicios propuestos a resolver en clase practica</description>
  </item>

  <item>
    <title>Enunciado TPI</title>
    <link>https://wl</link>
    <description>Enunciado del Trabajo práctico integrador</description>
  </item>
</channel>
</rss>
```



Completo con elementos opcionales

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
  <title>RSS de la cátedra de Sintaxis y Semántica de Lenguajes </title>
  <link>https://frre.cvg.utn.edu.ar/course/view.php?id=399</link>
  <description>Sintaxis y Semántica de Lenguajes de la U.T.N. F.R.Resistencia. </description>
  <category>Educacion</category>
  <copyright>2022 UTN. FRRe. Licencia Creative Commons. Atribución-No Comercial-Compartir Igual (CC BY-NC-SA) </copyright>
  <image>
    <url>https://frre.cvg.utn.edu.ar/pluginfile.php/29750/theme_snap/coverimage/1584391474/cours
e-image.gif</url>
    <title>encabezado imagen SSL</title>
    <link>https://frre.cvg.utn.edu.ar/course/view.php?id=399</link>
    <height>250 </height>
    <width>120 </width>
  </image>

  <item>
    <title>Planificacion 2022</title>
    <link>https://</link>
    <description>Planificacion de catedra, con cronograma de clases y evaluaciones</description>
  <category>Planificacion</category>
  </item>

  <item>
    <title>Guia de Trabajos practicos</title>
    <link>https://frre.cvg.utn.edu.ar/mod/resource/view.php?id=43544&redirect=1</link>
    <description>Guía de ejercicios propuestos a resolver en clase practica</description>
    <category>Practica</category>
  </item>

  <item>
    <title>Enunciado TPI</title>
    <link>https://wl</link>
    <description>Enunciado del Trabajo práctico integrador</description>
  </item>
</channel>
</rss>
```

9. Enlaces Interés

- Descripción RSS: https://www.w3schools.com/xml/xml_rss.asp
- RSS validador: <https://validator.w3.org/feed/>
- Visualizador Rápido : <https://codebeautify.org/rssviewer>