

**PS - Práctica 2**

**Semestre 25-2**

**Prof: Dr. Orlando Muñoz Texzocotetla**

**Análisis gramatical**

En la práctica anterior se creó la microsintaxis que contiene las reglas para cada una de las palabras que conforman el lenguaje de programación que está diseñando cada equipo. En esta segunda práctica se usará esa microsintaxis para diseñar en Bison la gramática del correspondiente lenguaje de programación. Realice las actividades que se indican en los siguientes incisos.

1. Investigar el problema de ambigüedad: ***Dangling Else Problem***. Poner un ejemplo diferente del if-then-else e indicar cómo solucionar dicho problema. **Pregunta de examen.**

2. Indique la sintaxis y de al menos dos ejemplos para cada uno de los siguientes elementos de su lenguaje de programación:

1. Expresión, dentro de las expresiones que podemos incluir en cualquier lenguaje de programación tenemos las siguientes:
   1. Asignación: en C la asignación es Variable = Expresión
   2. Expresiones aritméticas: estás utilizan los operadores de suma, resta, producto, división y módulo.
   3. Expresiones relacionales: generadas con los operadores <, >, <=, >=, ==, !=
   4. Expresiones lógicas: sus operadores son: AND, OR y NOT

Para este punto hay que tomar en cuenta que podemos formar expresiones con los operadores mostrados en cada inciso, por ejemplo:

* Asignación: var = 45;
* Expresión aritmética: var = 4 + 5 \* (45 / 3);
* Expresión relacional: var = 54 > var2;
* Expresión lógica: var = !(a || b);

Pero también podemos tener expresiones que tienen más de un tipo de operador:

* var = (b \* a) > (45 \* c) || (b \* b + c) > 23;

1. Instrucciones. Dentro de las instrucciones a incluir tenemos:
   1. Imprimir en pantalla equivalente al printf de C (mejor hacerlo tipo Python).
   2. Leer variable equivalente a scanf de C
   3. Condicional simple, por ejemplo:

| if(a > 4){  printf("\nHola"); } |
| --- |

* 1. Condicional doble: este tiene el bloque principal y el bloque del else.
  2. Condicional múltiple equivalente al if-elseif
  3. Ciclo while
  4. Ciclo for

1. Comentarios de una sola línea
2. Comentarios de varias líneas
3. Cadenas de caracteres
4. Declaración de variables
5. Bloque principal del programa
6. **Agregue un elemento más a la sintaxis (y en caso necesario a la microsintaxis). Para esto hay que revisar qué le hace falta al lenguaje, y agregarlo tanto a la microsintaxis como a la gramática. Este valor agregado se tomará en cuenta para la calificación de la práctica.**

Tip: revisar la documentación de C o cualquier otro lenguaje de programación para hacer este inciso. Les dejo el link de la sintaxis de Pascal para que se den una idea:

[Documentación de Pascal](https://docs.pascal65.org/en/latest/langref/syntax/)

3. Escriba un programa de ejemplo con su gramática y su microsintaxis propuesta. El programa deberá tener todos los elementos del inciso anterior: declaración de variables, bloque principal de programa, ciclos, condicionales, expresiones, instrucciones para imprimir y leer variables, etc. Coloree la sintaxis, en notepad+ se le puede dar color a las instrucciones, números, cadenas de caracteres, etc.

**IMPORTANTE: La gramática si bien puede estar basada en un lenguaje de programación existente, debe tener elementos originales (EN CUANTO A LA SINTAXIS) y no ser una copia exacta de los lenguajes en que se basan (NO BASTA CON QUE LAS PALABRAS DE LA MICROSINTAXIS SEAN DIFERENTES, TAMBIÉN LA GRAMÁTICA DEBE SER LO MÁS ORIGINAL POSIBLE). Estos elementos originales se tomarán en cuenta para la calificación de la práctica. Entre más originalidad mayor calificación.**

4. Tomando en cuenta lo anterior escriba, en notación Backus-Naur o la que usamos en clase, la gramática completa de su lenguaje de programación.

5. Integre en Flex y Bison la microsintaxis y la gramática de su lenguaje de programación. Esta deberá aceptar cualquier fuente que tenga un programa bien escrito con la gramática propuesta, y si el programa tiene errores deberá indicar el error y la línea donde éste se encuentre.

Entregables: se entregarán los siguientes archivos

1. Reporte en PDF: esta debe incluir la documentación del lenguaje de programación con su sintaxis y gramática. REFERENCIAS.
2. Archivo de flex con el nombre **compilador.l**  (Microsintaxis)
3. Archivo de bison con el nombre **compilador.y**  (Gramática)
4. Archivo fuente con un programa de ejemplo con todos los elementos: fuente.txt
5. Archivo Makefile para crear los archivos y ejecutar los archivos necesarios.

**NO COMPRIMIR**

**SÓLO UN MIEMBRO DEL EQUIPO SUBE LOS ARCHIVOS, LOS DEMÁS NO HAGAN NADA DE SUBIR LOS ARCHIVOS NI DE COMENTAR NADA EN CLASSROOM.**

Links de ayuda

[C-- Language Specification](https://www2.cs.arizona.edu/~debray/Teaching/CSc453/DOCS/cminusminusspec.html)

[A Pascal grammar](https://www.freepascal.org/daily/doc/ref/refap1.html)