

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación.

IC5701 – Compiladores e Intérpretes.

Gramática de un lenguaje loco – Ciruelas Extendido

Profesor: Aurelio Sanabria.

Estudiante:

Tamara Nicole Rodríguez Luna.

17 de marzo del 2025.

Tabla de contenidos

Motivación.....	3
Análisis	4
Gramática EBNF.....	6
Ambiente estándar	8
Ejercicios	9

Motivación

Este lenguaje fue primeramente diseñado basándose en el lenguaje de programación Mico-Brujo y su nombre fue inspirado por el río Ciruelas que pasa junto a la Sede Interuniversitaria de Alajuela y se menciona en el libro “Marcos Ramírez” pues el personaje principal solía jugar en ese río.

En mi caso además de la lógica ya implementada en este lenguaje he propuesto diferentes gramáticas para la implementación de nuevas funciones a Ciruelas como lo son “como está la vara” y “movida” para la implementación de un switch-case, “juéguesela” y “tortón” que corresponden a un try-catch y “dele vuelta” que sería el equivalente a un for.

Además, en el ambiente estándar propuse “grítele” y “susúrrele” para cambiar un texto a mayúsculas y minúsculas respectivamente, también “déjelo_parejo” que recibe un número flotante y lo redondea.

Análisis

Ciruelas es un lenguaje interesante, pues al tener cierto conocimiento en los costarricenseños y la programación uno puede tener una buena idea de lo que está pasando solo leyéndolo sin ningún contexto adicional. Aunque creo que las Bombas a como están definidas en la gramática EBNF es difícil de determinar cuándo van a cerrar, entonces no podría poner un comentario a la izquierda de una línea de código que quiero ejecutar, esto puede ser una ventaja para tener código más ordenado con respecto a la documentación interna, pero también puede ser bastante incómodo para algunas personas.

Hay términos costarricenses que yo desconocía antes de ver este lenguaje y al verlos en acción, aún en un lenguaje de programación me puedo hacer a la idea de cómo podría usarlos en una oración en un ambiente informal.

Siento que fue pensado haciendo una fusión entre algunas ventajas de Python que declara variables sin tanto rollo como:

```
var_1 = 0
```

Ciruelas hace:

```
var_1 metale 0
```

Pero con las ventajas que implica C o C++ al delimitar bien sus funciones pues una función en esos lenguajes se ve como:

```
int suma (int a, int b){  
    return (a+b);  
}
```

En Ciruelas se ve como

```
mae suma (a, b){  
    resultado metale a hechele b
```

sarpe resultado

}

Claro que hay puntos de mejora en claridad, tal vez metale y/o hechele no son las mejores palabras para definir a una suma o una asignación, pues en el ejemplo anterior siento que se presta para confusión.

Gramática EBNF

Programa ::= (Comentario|Asignación|Función)*Principal

Principal ::= (Comentario)?(jefe|jefa)mae{Instrucción+}

Instrucción ::= (Repetición

|Bifurcación

|Asignación

|Retorno

|Error

|Comentario

|TryCatch

|SwitchCase)

Función ::= (Comentario)?mae Identificador (Parámetros){Instrucción+}

Comentario ::= Bomba: (\w(\s\w)*?)?

SwitchCase ::= como está la vara(Valor){Case+(Sino)?}

Case ::= movida Valor{Instrucción+}

TryCatch ::= juéguesela{Instrucción+}tortón{Instrucción+}

Repetición ::= (For|While)

For ::= de vuelta(Asignación/Condición/Asignación){Instrucción+}

While ::= upee(Condición){Instrucción+}

Asignación ::= Identificador metale

(Literal|ExpresiónMatemática|Invocación)

ExpresiónMatemática ::= (Expresión)|Literal|Identificador

Expresión ::= ExpresiónMatemática Operador ExpresiónMatemática

Operador ::= (hechele|quitele|chuncherequee|desmadeje)

Invocación ::= Identificador(Parámetros)

Parámetros ::= Valor(/Valor)+

Retorno ::= sarpe Valor?

Error ::= safis Valor

Bifurcación ::= DiaySi(Sino)?

DiaySi ::= diay siii (Condición){Instrucción+}

Sino ::= sino ni modo {Instrucción+}

Condición ::= Comparación((divorcio|casorio)Comparación)?

Comparación ::= Valor Comparador Valor

Valor ::= (Identificador|Literal)

Comparador ::= (cañazo

|poquitico

|misma vara

|otra vara

|menos o igualitico

|más o igualitico)

Literal ::= (Número|Texto|ValorVerdad)

Número ::= (Entero|Flotante)

Texto ::= ~[a-zA-Z0-9_ áéíóöúüñÁÉÍÓÖÚÜÑ]+~

ValorVerdad ::= True | False

Entero ::= (-)?[0-9]+

Flotante ::= (-)?[0-9]+.[0-9]

Identificador ::= [a-z][a-zA-Z0-9_áéíóöúüñÁÉÍÓÖÚÜÑ]*

Ambiente estándar

<code>hacer_menjunje(texto1, texto2)</code>	Junta dos textos y retorna 1
<code>viene_bolita(texto, indice)</code>	Retorna el carácter en la pos índice
<code>trone(texto)</code>	(retro) Calcula el largo de un string
<code>sueltele(texto)</code>	Suelta el texto en la pantalla
<code>echandí_jimenez()</code>	le pide al usuario que suelte algo (un texto)
<code>grítele(texto)</code>	pone el texto en mayúsculas
<code>susúrrele(texto)</code>	pone el texto en minúsculas
<code>déjelo_parejo(flotante)</code>	recibe un número flotante y lo redondea

Ejercicios

Bomba: Ejercicio 1

Bomba: Calcular vuelto

```
mae calcule_vuelto(precio / pago){  
    diay siii(precio poquitico 0){  
        safis ~Precio no puede ser negativo~  
    }  
    diay siii(pago poquitico 0){  
        safis ~Pago no puede ser negativo~  
    }  
    diay siii(pago poquitico precio){  
        safis ~Pago debe ser mayor o igual al precio~  
    }  
    resultado metale pago quitele precio  
    sarpe resultado  
}  
jefe mae {  
    resultado metale calcule_vuelto(5000, 20000)  
    sueltele(resultado)  
    sarpe 0  
}
```

Bomba: Ejercicio 2

Bomba: División

```
mae division(a / b){  
    juéguesela{  
        resultado metale a desmadeje b  
    }tortón{  
        safis ~b no puede ser 0~  
    }  
    sarpe resultado  
}  
jefe mae {  
    resultado metale promedio_notas(90 / 85 / 70)  
    sueltele(resultado)  
    sarpe 0  
}
```

Bomba: Ejercicio 3

Bomba: Costo total

```
mae calcule_total(precioNeto / iva){  
    impuesto metale precioNeto chuncherequee iva  
    precio_total metale precioNeto hechele impuesto  
    sarpe precio_total  
}
```

```

jefe mae {
    resultado metale calcule_total(5000 / 0.13)
    sueltele(~El total a pagar es ~)
    sueltele(resultado)
    sarpe 0
}
Bomba: Ejercicio 4
Bomba: Cifrador BDPQ
mae cifradorBDPQ(texto){
    Bomba: Recibe un texto en minúsculas y cambia las B por D y viceversa,
además de las P por Q y viceversa
    largo_texto metale trome(texto)
    resultado metale ~~
    dele vuelta(i metale 0 / i poquitico largo_texto / i metale i hechele 1){
        caracter metale viene_bolita(texto / i)
        diay siii(caracter misma vara ~b~){
            caracter metale ~d~
        }
        diay siii(caracter misma vara ~p~){
            caracter metale ~q~
        }
        diay siii(caracter misma vara ~q~){
            caracter metale ~p~
        }
        resultado metale hacer_mejunje(resultado / caracter)
    }
    sarpe resultado
}
Bomba: Ejercicio 5
Bomba: Contador de mae

mae contador_mae(texto){
    Bomba: Recibe un texto en minúsculas cuenta las veces que aparece la
palabra mae
    contador metale 0
    largo_texto metale trome(texto)
    Bomba: Hasta largo del texto menos 2 porque necesitamos al menos 3 letras
    dele vuelta(i metale 0 / i poquitico largo_texto quitele 2 / i metale i hechele 1){
        caracter metale viene_bolita(texto / i)
        diay siii(caracter misma vara ~m~){
            caracter2 metale viene_bolita(texto / i hechele 1)
            diay siii(caracter2 misma vara ~a~){
                caracter3 metale viene_bolita(texto / i hechele 2)
                diay siii(caracter3 misma vara ~e~){

```

```

        contador metale contador hechele 1
    }
}
}
sarpe contador
}

```

Ejercicio de Switch Case

Bomba: Ejercicio 6 con switch case

```

mae seleccionar_tarifa_envío(destino / peso){
    como está la vara(destino){
        movida ~San José~{
            costoBase metale 1500
            costoPorKilo metale 300
        }
        movida ~Alajuela~{
            costoBase metale 2000
            costoPorKilo metale 350
        }
        movida ~Limón~{
            costoBase metale 3500
            costoPorKilo metale 500
        }
        movida ~Puntarenas~{
            costoBase metale 4500
            costoPorKilo metale 600
        }
        sino ni modo {
            costoBase metale 5000
            costoPorKilo metale 700
        }
    }
    resultado metale costoBase hechele (costoPorKilo chuncherequee peso)
    sarpe resultado
}

```