Reporte del Área: Matemáticas

Descripción: sdsdsssssssssssssssssssssssssssssssssss

Encargado: Marcos

Recomendación: Jesus angel2

Descripción: ddddddddddddddddddddddddddd

Recomendación: Miguel

Descripción: dsfasasdasdasd

Reporte del plan de acción: Miguel Reprueba

Descripción: Miguel reprobo dos materias

Recomendación	Categoría	Estado	Fecha de término
Miguel	Matemáticas	En progreso	2019-12-17

Evidencias:

UNIVERSIDAD DEL ESTADO DE SONORA

"El saber de mis hijos hará mi grandeza"

Licenciatura en Ciencias de la Computación

Lenguajes de Programación

Maestra: Guadalupe Cota Ortiz

Gramática de Ruby

Equipo:

Yitzhak David Gutiérrez Moya

Jesus Angel Carmona Alvarez

José Alberto Ozuna Madrid

Marcos Abraham Sánchez Galindo

Diciembre 2019

Ruby

Ruby es un lenguaje de programación, interpretado, de a alto nivel y de propósito general. Fue diseñado a mediados de 1990 por Yukihiro Matsumoto en Japón. Ruby es un lenguaje dinámico y usa un recolector de basura, este soporta varios paradigmas de programación. La idea de ruby surgió cuando Yukihiro Matsumoto quería usar un lenguaje que fuera completamente orientado a objetos y sea fácil de usar como un “scrpting language”, los lenguaje que estaban a su disposición eran Perl y python estos al no poner cumplir con lo que el quería decidió crear uno.

Semántica

Ruby es orientada a objetos esto es cada valor es tratado como un objeto como las clases y también lo que en otros lenguajes son los tipos primitivos como los enteros, booleanos y null. Las variables siempre tienen referencias a objetos. Cada función es un método y estos siempre son llamados a partir de un objeto. A pesar de ser diseñado como un lenguaje completamente orientado a objetos este también soporta otro tipo de paradigmas de programación como la programación funcional, por procedimientos. La sintaxis de ruby es similar a Perl ,su semántica es similar a Smalltalk, pero difiere mucho a la de python.

Sintaxis

La sintaxis de ruby es similar a la de Perl y python. La definición de las clases y los métodos están definidas por palabras claves. Los saltos de linea son significativos y son interpretados como el final de una sentencia, para esto también se puede usar el punto y coma. A diferencia de python la indentacion no es significativa. Una diferencia entre ruby y python es que ruby cuando se declara una instancia de una variable estas siempre son privadas y solo se pueden acceder por métodos otorgados por la clase, esto es para mantener la idea que en la programación orientada a objetos uno no puede acceder a variables por fuera del objeto si no se debe mandar un mensaje a este para poder obtener su valor.

Para que se usa

El framework mas utilizado para ruby es ruby on rails, hay aplicaciones muy populares que usan ruby como por ejemplo AirBnB, Hulu, Github y Goodreads. Ruby es un lenguaje sencillo de aprender con el cual se puede utilizar para crear una aplicación sencilla y ruby on rails se utiliza para crear paginas web.

Gramática

Z→ ET | ETZ | TET | TETZ

T→ ; | \n

E→ F | J | K | B | D

F→ v = H

J→ M | Q

H→ D | “Y” | A | N | X

Y→ L | G | U | LY | GY | UY

A→ L | LA

D→ getsUchomp | getsUto_f

K→ ifSthenZend

B→ print”V” | printA

N→ G | GN

U→ T | | . | > | , | : | == | < | %

Q→ whileSZend

M→ forain(N..N)Zend

S→ AUN

V→ Y | #{A} | Y#{A}V | #{A}V | Y#{A}

X→ v+N | v-N | v/N | v*N

L→ a | b | c | … | z

G→0 | 1 | 2 | 3 | 4 | 5 | 6| 7 | 8 | 9

<grupo_sentencias> ::= <sentencia><símbolo_de_termino>

| <sentencia><símbolo_de_termino><grupo_sentencias>

| <símbolo_de_termino><sentencia><símbolo_de_termino>

| <símbolo_de_termino><sentencia><símbolo_de_termino><grupo_sentencias>

<simbolo_de_termino> ::= ‘;’ | ‘\n’

< sentencia > ::= <asignacion_simple> | <ciclo> | <if> | <salida_consola> | <entrada_consola>

<asignacion_simple> ::= ‘v’ ‘=’ <valor>

<ciclo> ::= <for> | <while>

<valor>::= <entrada_consola> | ‘ ” ’ <cadena> ‘ ” ‘ | <palabra> | <numero>| <operacion>

<cadena>::= <letra> | <digito> | <simbolo> | <letra><cadena> | <digito><cadena>

| <simbolo><cadena>

<palabra>::= <letra> | <letra><palabra>

<entrada_consola> ::= “gets”<simbolo>”chomp” | “gets”<simbolo>”to_f”

<if>::= “if” <comparación> “then” <grupo_sentencias> “end”

<salida_consola>::= ”print” ‘ “ ‘ <formato> ‘ “ ‘ | print <palabra>

<numero> ::= <digito> | <digito><numero>

<simbolo>::= <simbolos_de_termino> | ‘ ‘ | ‘.’ | ‘>’ | ‘,’ | ‘:’ | ‘==’ | ‘<’ | ‘%’

<while>::=’while’<comparacion><grupo_de_sentencias>’end’

<for>::=‘for’ ‘a’ ’in’ ‘(‘<numero>’..’<numero>’)’<grupo_sentencias>’end’

<comparacion>::=<palabra><simbolo><numero>

<formato>::= <cadena> | ‘#{‘ <palabra> ’}’ | <cadena> ‘#{‘ <palabra> ’}’ <formato>

| ‘#{‘ <palabra> ’}’ <formato> | <cadena> ‘#{‘ <palabra> ’}’

<operacion>::= ‘v’ ‘+’<numero> |‘v’ ‘-’<numero> |‘v’ ‘/’<numero> |‘v’ ‘*’<numero>

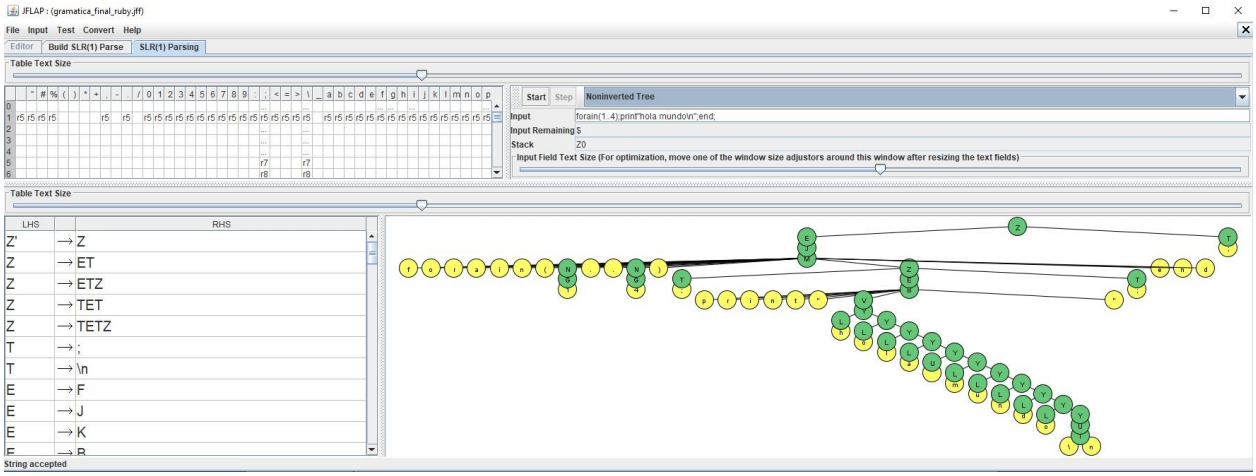
<letra> ::= a | b | c . . . | z

<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6| 7 | 8 | 9

Ejemplos:

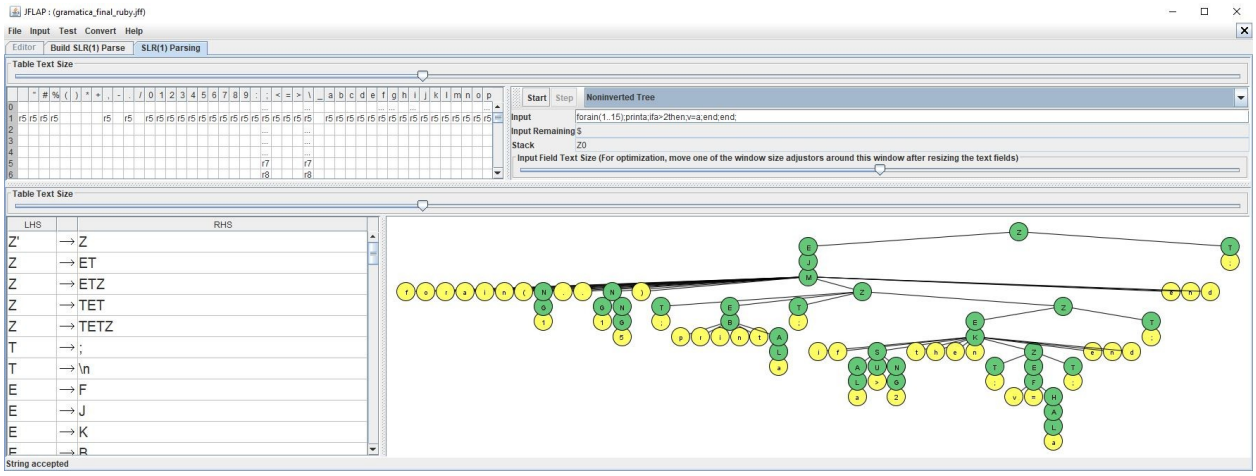
Ejemplo 1:

```
for a in (1..4)
        print "hola mundo\n"
end
```



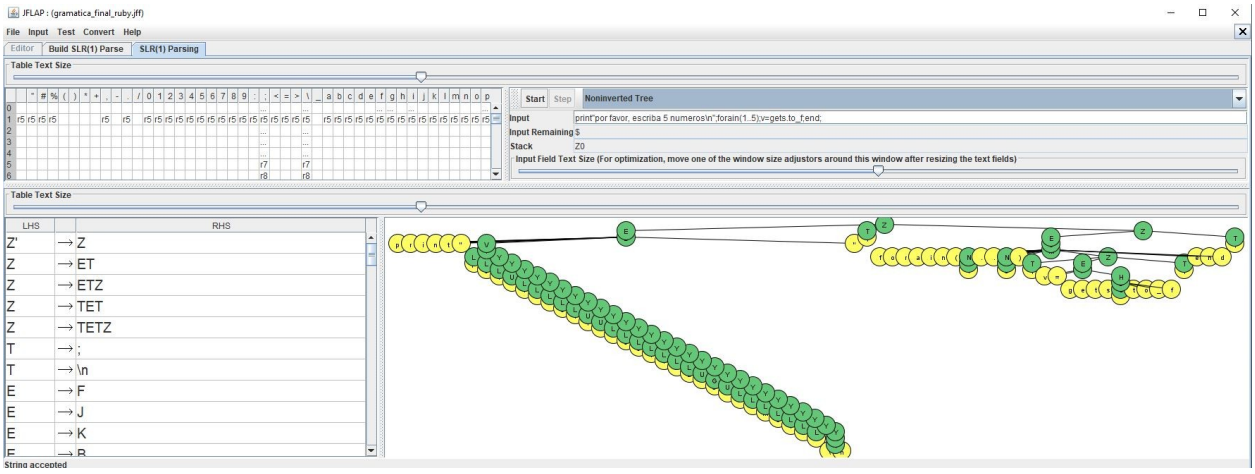
Ejemplo 2:

```
for a in (1..15)
        print a
        if a>2 then
                v=a
        end
end
```



Ejemplo 3:

```
print "por favor, escriba 5 numeros\n"
for a in (1..5)
        v=gets.to_f
end
```



Ejemplo 4:

print "por favor, escriba su nombre: "

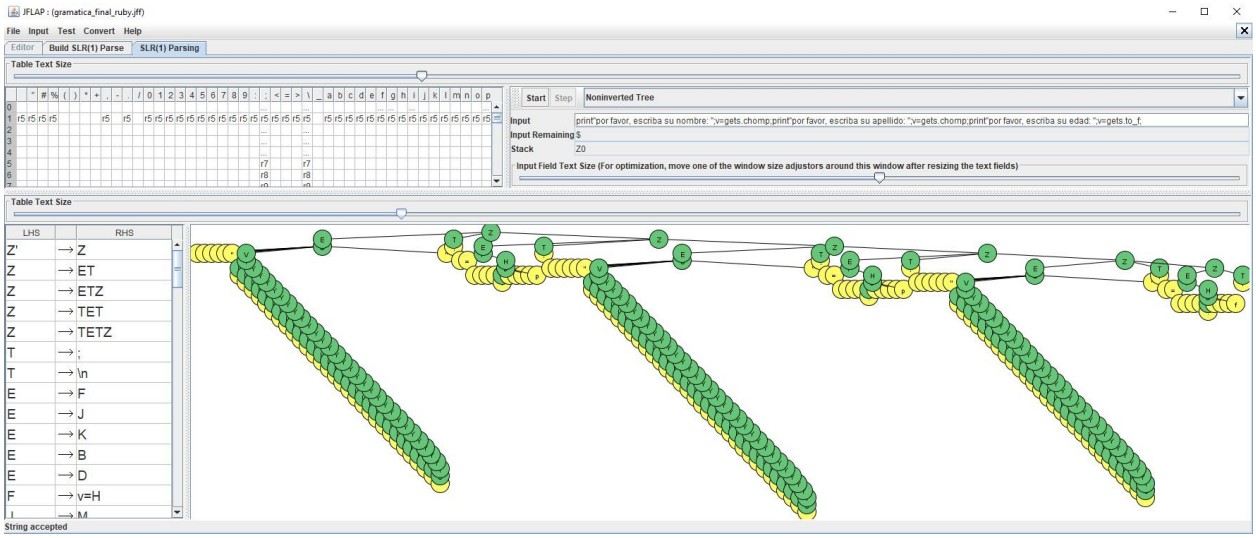
v=gets.chomp

print " por favor, escriba su apellido: "

v=gets.chomp

print " por favor, escriba su edad: "

v=gets.to_f;



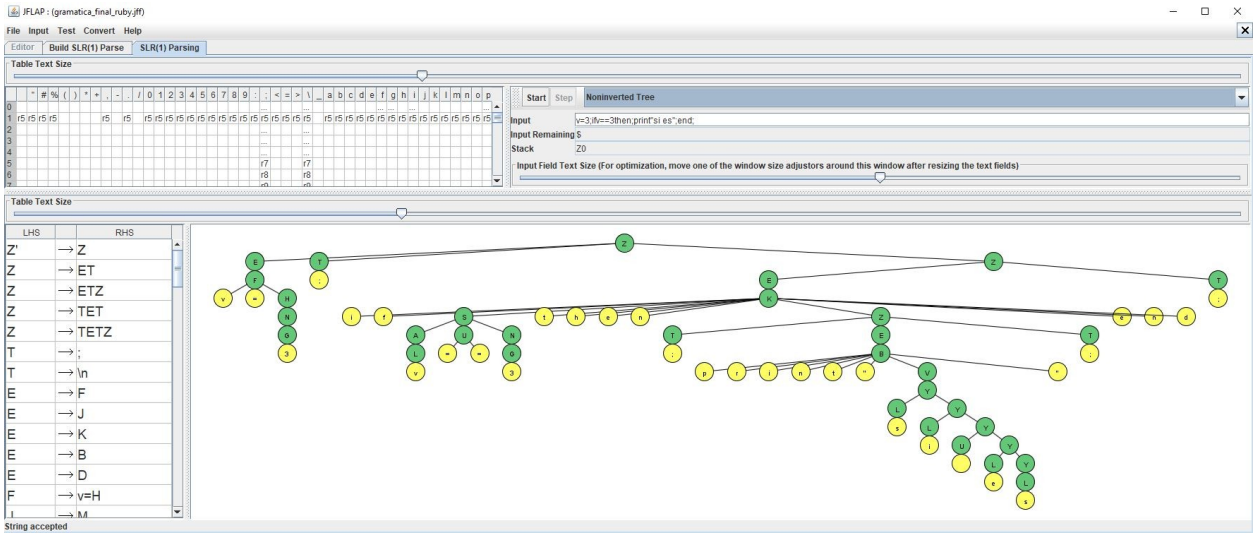
Ejemplo 5:

v=3

If v==3 then

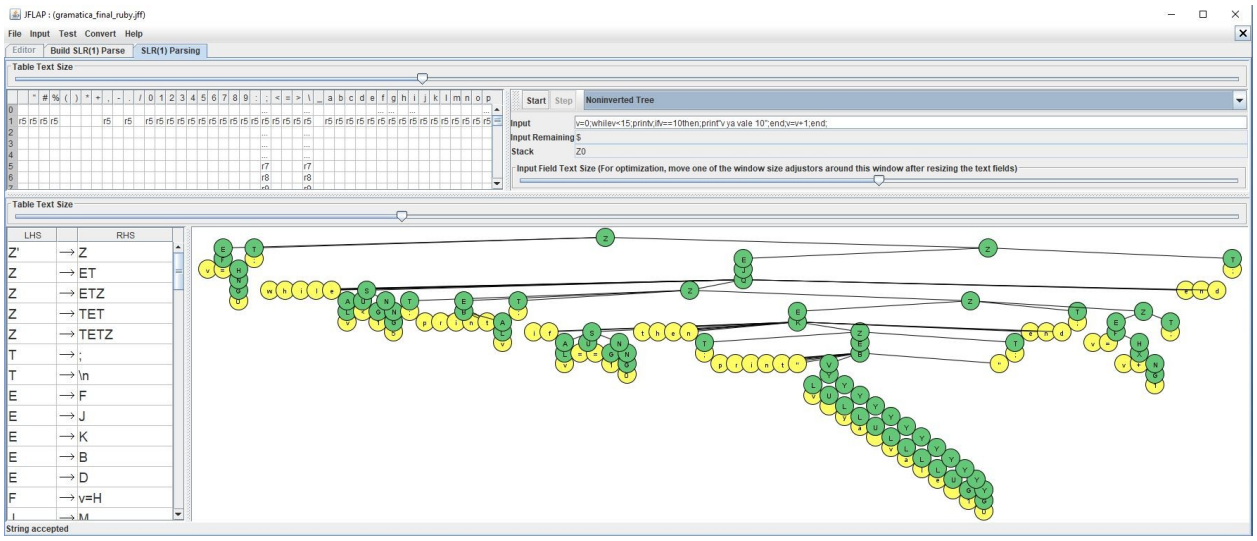
Print “si es”;

End



Ejemplo 6:

```
v = 0
while v < 15
    print v
    if v == 10 then
        print "v ya vale 10"
    end
    v = v + 1
end
```

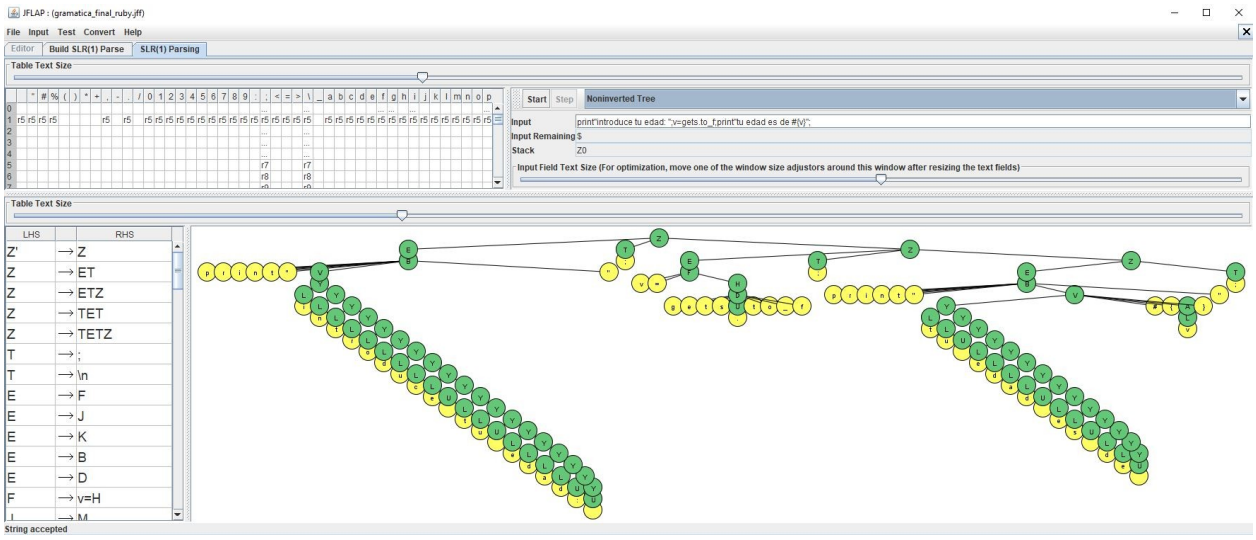


Ejemplo 7:

Print "introduce tu edad: "

V=gets.to_f

Print "tu edad es de #{v}"

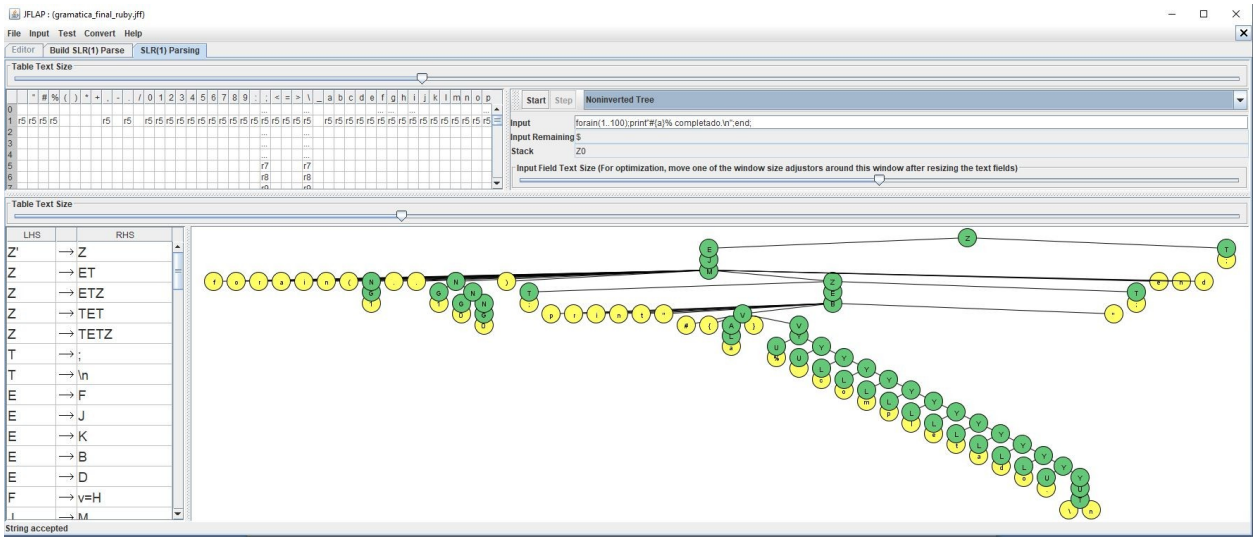


Ejemplo 8:

For a in (1..100)

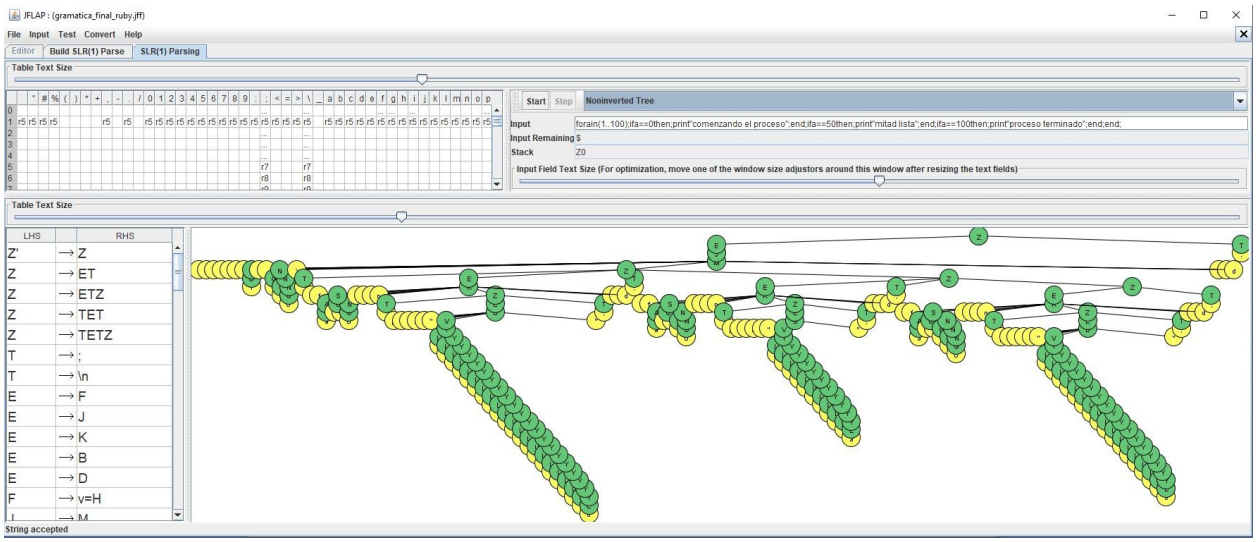
Print "#{a}% completado.\n"

End



Ejemplo 9:

```
For a in (1..100)
        If a==0 then
                Print "comenzando el proceso"
        End
        If a==50 then
                Print "mitad lista"
        End
        If a==100 then
                Print "proceso terminado"
        End
End
```



Ejemplo 10:

```
Print "introduzca un numero"
V=gets.to_f
If v<51 then
        Print "derrota"
End
If v>50 then
        Print "victoria"
End
```