



Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación

# PROYECTO DE FIN DE CICLO

## COMPILADORES

### Integrantes:

Jhonny Morocho  
Lorena Guadalima  
Jessica Correa  
Edhisson Sanmartin  
Santiago Rivera





Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación

## **TEMA:**

# **Analizador Sintáctico para reconocer las palabras Agudas.**

### **Objetivos:**

- 1. Identificar la gramática para el reconocimiento las palabras agudas.**
- 2. Representar la gramática propuesta mediante el análisis sintáctico.**
- 3. Proponer los pasos en la construcción de la solución.**



Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación





Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación

# Resultados

**Objetivo 1: Identificar la gramática para el reconocimiento las palabras agudas.**

# ANÁLISIS DE LAS PALABRAS AGUDAS.

## Palabras Agudas:

Son las polisílabas (de más de una sílaba) cuya última sílaba es tónica (sílaba tónica de una palabra es aquella sílaba que se pronuncia acentuada). De acuerdo a las reglas de acentuación de las palabras, la sílaba tónica puede o no llevar tilde): *menú, pared, amar, sutil*.

	AGUDAS
Sílaba tónica	La última
Llevan tilde	Cuando terminan en vocal, en -n, o en -s
Ejemplos	me- <b>nú</b> co-ra- <b>zón</b> com- <b>pás</b>

Orientación de uso	Ejemplos
Las palabras agudas llevan tilde en la sílaba tónica cuando terminan en <b>vocal</b> , <b>-n</b> o <b>-s</b> .	<b>Balón, papá, Aragón, París, café.</b>



Universidad  
Nacional  
de Loja

# ANÁLISIS DE LAS PALABRAS AGUDAS..



Carrera de Ingeniería en  
Sistemas / Computación

**Aguda** → <palabra>

**Palabra** → Compuesta <Sílabas>

**Sílabas** → tiene <acento>, termina <vocal> o <letra>

**Acento** → **café** | **bebé** | **además** | **balón** | **papá**

**Sílabas:** **C****a****f****é**

**Vocal** → **á,é,í,ó,ú**

**letra** → **l** | **s** | **n**



Universidad  
Nacional  
de Loja

# ANÁLISIS DE LAS PALABRAS AGUDAS..



Carrera de Ingeniería en  
Sistemas / Computación

## Gramática

**A → P**

**P → S**

**S → EVL**

**E → caf | beb | bal | pap | adem**

**V → á | é | í | ó | ú**

**L → λ | s | n**



Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación

# Resultados

**Objetivo 2: Representar la gramática propuesta mediante el Análisis Sintáctico.**





Universidad  
Nacional  
de Loja

# FUNCION PRIMEROS



Carrera de Ingeniería en  
Sistemas / Computación

*La Función de los PRIMEROS es el conjunto de terminales que pueden aparecer los primeros en las cadenas que se derivan de aquella que estamos analizando.*

## Reglas para el cálculo del conjunto PRIMERO

1. Si  $X$  es un símbolo que pertenece al conjunto de los terminales, entonces  $\text{PRIMERO}(X) = \{X\}$ .
2. Si  $X$  es  $\lambda$ , entonces  $\text{PRIMERO}(X) = \{\lambda\}$ .
3. Si  $X$  es un no terminal y  $X \rightarrow Y_1 Y_2 \dots Y_n$  se incluirá los que haya en  $\text{PRIMERO}(Y_1)$  y si de  $Y_1$  deriva  $\lambda$ , además de otros terminales, se incluirán estos terminales y en lugar de  $\lambda$ , lo que haya en  $\text{PRIMERO}(Y_2)$ , y así sucesivamente hasta llegar a  $Y_n$ . Solo se incluiría  $\lambda$ , si esta estuviera en todas las producciones que se derivan de  $Y_1 Y_2 \dots Y_n$ .
4. Repetir hasta que no se puedan añadir más terminales o  $\lambda$  a ningún conjunto PRIMERO.



Universidad  
Nacional  
de Loja

# FUNCION PRIMEROS



Carrera de Ingeniería en  
Sistemas / Computación

## Ejemplo:

**A** → **P**  
**P** → **S**  
**S** → **EVL**  
**E** → **caf | beb | bal | pap | adem**  
**V** → **á | é | í | ó | ú**  
**L** → **λ | s | n**

## Tabla Primeros:

P(A)	→	{ caf , beb , bal , pap , adem }
P(P)	→	{ caf , beb , bal , pap , adem }
P(S)	→	{ caf , beb , bal , pap , adem }
P(E)	→	{ caf , beb , bal , pap , adem }
P(V)	→	{ á , é , í , ó , ú }
P(L)	→	{ λ , s , n }

**Siguientes**



Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación

# **FUNCION SIGUIENTES**

## **Reglas para el cálculo del conjunto siguiente**

1. Si  $S$  es el símbolo inicial de la gramática, entonces  $\$$  está en  $SIGUIENTE(S)$ .
2. Si tenemos la producción  $A \rightarrow \alpha B \beta$ , se añaden a  $SIGUIENTE(B)$  todos los terminales que haya en  $PRIMERO(\beta)$  excepto el símbolo de la cadena vacía ( $\lambda$ ). En el caso de que apareciera  $\lambda$  en  $PRIMERO(\beta)$  entonces se elimina  $\lambda$  y se incluyen los terminales que se deriven de  $SIGUIENTE(A)$ .
3. Si tenemos una producción  $A \rightarrow \alpha B$  añadir a  $SIGUIENTE(B)$  los terminales que se deriven de  $SIGUIENTE(A)$ .
4. Repetir para todas las producciones en las que aparezca en el lado derecho el símbolo para el que estamos calculando el conjunto.

# OBTENCIÓN DE LOS SIGUIENTES

Ejemplo:

$A \rightarrow P$   
 $P \rightarrow S$   
 $S \rightarrow EVL$   
 $E \rightarrow \text{caf} \mid \text{beb} \mid \text{bal} \mid \text{pap} \mid \text{adem}$   
 $V \rightarrow \acute{a} \mid \acute{e} \mid \acute{i} \mid \acute{o} \mid \acute{u}$   
 $L \rightarrow \lambda \mid s \mid n$

Tabla Siguietes:

$S(A)$	$\rightarrow$	$\$$	1ra Regla
$S(P)$	$\rightarrow$	$\$$	1ra Regla
$S(S)$	$\rightarrow$	$\$$	1ra Regla
$S(E)$	$\rightarrow$	$\acute{a}, \acute{e}, \acute{i}, \acute{o}, \acute{u}$	3ra Regla
$S(V)$	$\rightarrow$	$s, n, \$$	3ra Regla
$S(L)$	$\rightarrow$	$\$$	2do Regla



- $S(A) \rightarrow \$$
- $A \rightarrow \alpha B$   
 $S(B) \rightarrow S(A)$
- $A \rightarrow \alpha B \beta$   
 $S(B) \rightarrow \underline{P}(\beta) \in \lambda$   
 $S(B) \rightarrow P(\beta) \cup S(A)$



Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación

# Tabla de Análisis Sintáctico LL(1)

La tabla de análisis sintáctico LL(1) es una matriz de dos dimensiones indexada por no terminales (N, columna de la izquierda en la tabla que hemos utilizado) y terminales (T, fila superior de la tabla, que incluye el símbolo \$).

Por tanto accedemos a la tabla utilizando  $M[N, T]$ .

En la tabla las entradas vacías son errores, que posteriormente cuando veamos la fase de tratamiento de errores se puede rellenar con llamadas a funciones o procedimientos que avisan del error.

# PRIMERS

P(A)	→	{ caf , beb , bal , pap , adem }
P(P)	→	{ caf , beb , bal , pap , adem }
P(S)	→	{ caf , beb , bal , pap , adem }
P(E)	→	{ caf , beb , bal , pap , adem }
P(V)	→	{ á , é , í , ó , ú }
P(L)	→	{ λ , s , n }

## Siguientes

S(A)	→	\$
S(P)	→	\$
S(S)	→	\$
S(E)	→	á,é,í,ó,ú
S(V)	→	s,n,\$
S(L)	→	\$

**A** → **P**  
**P** → **S**  
**S** → **EVL**  
**E** → **caf | beb | bal | pap | adem**  
**V** → **á | é | í | ó | ú**  
**L** → **λ | s | n**

## GRAMÁTICA

[illegible]



Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación



# Validar Cadenas Mediante Tabla LL(1)

Coincidencia	Pila	Entrada	Acción
	A\$	además\$	
	P\$	además\$	Emitir(A->P)
	S\$	además\$	Emitir(P->S)
	EVL\$	además\$	Emitir(S->EVL)
	ademVL\$	además\$	Emitir(E->adem)
adem	VL\$	ás\$	Relacionar adem
	áL\$	ás\$	Emitir(V->á)
á	L\$	s\$	Relacionar á
	s\$	s\$	Emitir(L->s)
s	\$	\$	Relacionar s





Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación

# Futuros pasos en la construcción de la solución.



# Entorno Web



Universidad  
Nacional  
de Loja

## 1. JavaScript :

**Librería (Flex.JS):** [GitHub - sormy/flex-js: FLEX.JS - fast lexer \(tokenizer, scanner\) for JavaScript inspired by FLEX lexer generator](https://github.com/sormy/flex-js)

## 2. Interfaces Bootstrap.



1859

unl

Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación



GRACIAS