

Procesamiento de Lenguaje Natural (PLN - NLP)

Lengua Natural → idioma, para comunicar ideas.
↳ Forma de comunicación



Discurso, escrita
↳ Hablar, Leer, escribir

Tres fases para la construcción de lenguaje

- Léxica → Palabras, símbolos, quedan bien definidos
- Sintáctica → Las frases siguen una gramática.
- Semántica → significado, palabras, frases.

Aplicaciones del PLN.

→ Resumir textos (Información No estructurada)

→ Abstractar información

→ Analizar sentimientos

→ Intención

Rabia @dio
Esperanza Amor
Depresión tristeza

Twitter

API



Python
+ etc.

Construcción de un lenguaje de Programación.

Fases:

Código Fuente

Compilados

interpretado.

- ps
- c
- cpp
- js
- java

```
def suma(x, y):  
    return x + y
```

```
print(suma(1, 2))
```

Compilador

Léxica

Gramática Regular

↓ Tokens

Sintáctica

Gramática libre al contexto

↓ Arbol de sintaxis

Semántica

Significado frases

↓ Arbol de sintaxis Abstracta

Generación Código Intermedio

↓
Optimización de Código

↓
Generación de Código máquina.

Byte Code: 1010.10

↓
Ejecución del Programa

```
def suma(x,y):
    return x+y

print(suma(1,2))
```

Lexica:

def → palabra reservada def
 suma → identificador
 (→ par-A
 x → id.
 , → coma
 y → id
) → par-C
 : → Dos Puntos
 return → palabra Reservada return
 print → palabra Reservada print
 (→ par-A
 suma → id
 (→ par-A
 1 → # Numero
 , → coma
 2 → # Numero
) → par-C
) → par-C

lexema	token
def	def
suma	Id.
(par-A

Gramática Regular

```
def suma(x,y):  
    return x+y
```

```
print(suma(1,2))
```

Ejemplo:

- 1.2
- 1.2
- 1.2 E +3
- 1.2 E -3

+: 1 a mucho
*: 0 a muchos

$Id = [a-zA-Z]$

$Numero = [0-9]$

Expresiones
regulares.

Reglas gramaticales para construir un número

$digito = [0-9]$

$Numero = -?[digito][.digito][e|E]?[digito]^*$

Parte Sintaxis = Seguir la Reglas de una gramática.

Construir una gramática.

I.J.C (Gramática independiente
del contexto)

$$I.J.C = (N, T, S, P)$$

N = Conjunto de No terminales

T = Conjunto de Terminales

S = simbolo de inicio

P = Reglas de la gramática

Ejemplo: gramática calculadora.

Expr

stat : expr NEWLINE
 | ID '=' expr NEWLINE
 | NEWLINE
 .

$\overline{\text{Expr}} = (N, T, S, P)$
 $N = \{\text{stat}, \text{expr}\}$
 $T = \{\text{ID}, \text{INT}, \text{NEWLINE}, \text{WS}\}$
 $S = \{\text{stat}\}$
 $P = \text{Reglas de Producción}$

Evaluar la gramática.

Var = 12
Var + 3

Stat
↓

Expr NewLine

↓
Expr + Expr
↓ ↓
ID INT
| |
var 3

{ \$1 + \$3 }

acción/significado
sumar

Lexema → token

Var → ID → Sintactica.

= → =
12 → INT
| n
Var → ID
+ → +
3 → INT

Stat
↙ ↘ Newline
ID = Expr Newline
↓ ↓
var 12

{ \$1 = \$3 }
↓
asignar.