

# Processamento de Linguagem Natural

---

Prof. Dr. Silvio do Lago Pereira

Departamento de Tecnologia da Informação

Faculdade de Tecnologia de São Paulo

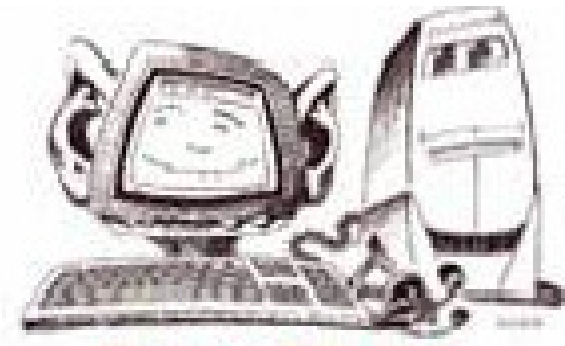


# Introdução

## Processamento de linguagem natural (PLN)

consiste no desenvolvimento de modelos computacionais para a realização de tarefas que dependem de informações expressas em uma língua natural.

- Alguns exemplos de aplicação:
  - tradução e interpretação de textos
  - busca de informações em documentos
  - interface homem-máquina (*chatterbots*)
- Aspectos da pesquisa em PLN:
  - **som**: fonologia
  - **estrutura**: morfologia e sintaxe
  - **significado**: semântica e pragmática





## Alguns aspectos da pesquisa em PLN

- **Fonologia:** reconhece os sons (fonemas) que formam as palavras de uma língua.
- **Morfologia:** reconhece uma palavra em termos de unidades básicas (morfemas).
- **Sintaxe:** define a estrutura de uma frase com base na forma como as palavras desta frase se relacionam entre si (categorias gramaticais).
- **Semântica:** associa significado às estruturas sintáticas, em função do significado das palavras que a compõem.
- **Pragmática:** adequa o significado de uma frase ao contexto em que ela é usada.

### Considerações

- PLN é uma vasta área de pesquisa que envolve diversas áreas do conhecimento.
- Para simplificar, abordaremos apenas alguns conceitos básicos de análise sintática.
- Mostraremos como usar conceitos básicos de linguagens formais para definir uma gramática capaz de gerar e reconhecer um conjunto restrito de frases em português.



# Gramáticas

- Uma linguagem é um conjunto de sentenças, formadas pela concatenação de símbolos.
- Linguagens formais são linguagens artificiais (tais como lógica proposicional ou Pascal) que podem ser matematicamente definidas, de forma rigorosa.
- Linguagens naturais (tais como português ou inglês) não são matematicamente definidas.
- Embora a correspondência não seja perfeita, podemos tratar linguagens naturais como tratamos linguagens formais.

## Uma gramática

é uma especificação matemática da estrutura das sentenças de uma linguagem.

- Formalmente, uma gramática é definida por:
- $S$ : o **símbolo inicial** da gramática ( $S \in \mathcal{N}$ )
- $\mathcal{T}$ : um conjunto de **símbolos terminais**, denotando palavras da linguagem (léxico).
- $\mathcal{N}$ : um conjunto de **símbolos não-terminais**, denotando componentes de sentenças.
- $\mathcal{R}$ : um conjunto de **regras de produção**, que especificam como símbolos não-terminais podem ser expandidos em símbolos não-terminais e terminais.



## Especificação de gramática usando diferença de listas

Considere a gramática a seguir:

$S = \{\text{frase}\}$

$T = \{\text{o, gato, rato, caçou}\}$

$\mathcal{N} = \{\text{frase, sujeito, predicado, artigo, substantivo, verbo}\}$

$\mathcal{R} = \{\text{frase} \rightarrow \text{sujeito, predicado};$   
     $\text{sujeito} \rightarrow \text{artigo, substantivo};$   
     $\text{predicado} \rightarrow \text{verbo, artigo, substantivo};$   
     $\text{artigo} \rightarrow [\text{o}];$   
     $\text{substantivo} \rightarrow [\text{gato}] \mid [\text{rato}];$   
     $\text{verbo} \rightarrow [\text{caçou}]\}$

De acordo com esta gramática:

- uma frase é um sujeito seguido de um predicado
- um sujeito é um artigo seguido de um substantivo
- um predicado é um verbo, seguido de um artigo, seguido de um substantivo
- um artigo é o símbolo terminal **o**
- um substantivo é o símbolo terminal **gato** ou **rato**
- um verbo é o símbolo terminal **caçou**



## Especificação de gramática usando diferença de listas

- Considere que frases são representadas por listas de palavras. Então, por exemplo, a frase “o gato caçou o rato” pode ser representada como  
`[o,gato,caçou,o,rato]`
- Reconhecendo o primeiro artigo em `[o,gato,caçou,o,rato]`, obtemos a lista `[gato,caçou,o,rato]`. Isto pode ser indicado em Prolog com o fato:  
`artigo([o|A],A).`
- Analogamente, reconhecendo o primeiro substantivo em `[gato,caçou,o,rato]`, obtemos a lista `[caçou,o,rato]`. Isto pode ser indicado com o fato:  
`substantivo([gato|A],A).`
- Neste ponto, é importante notar que, reconhecendo um artigo seguido de um substantivo, acabamos reconhecendo o sujeito da frase. Em Prolog, isto pode ser indicado pela regra:  
`sujeito(A,C) :- artigo(A,B), substantivo(B,C).`



## Especificação de gramática usando diferença de listas

- Após o reconhecimento do sujeito em [o, gato, caçou, o, rato], obtemos a lista [caçou, o, rato]. Reconhecendo o verbo nesta lista, obtemos a lista [o, rato]. Isto pode ser indicado com o fato:

`verbo([caçou|A],A).`

- O artigo em [o, rato] pode ser reconhecido pelo fato já declarado anteriormente e, para reconhecer este novo substantivo, podemos usar o fato:

`substantivo([rato|A],A).`

- Aqui também podemos notar que, para reconhecer o predicado da frase, é necessário reconhecer um verbo, seguido de um artigo e um substantivo. Isto pode ser indicado pela regra:

`predicado(A,D) :- verbo(A,B), artigo(B,C), substantivo(C,D).`

- Finalmente, para reconhecer uma frase, podemos usar a regra:

`frase(A,C) :- sujeito(A,B), predicado(B,C).`



# Reconhecimento de frases

## Exemplo 1. Especificação completa da gramática

```
frase(A,C) :- sujeito(A,B), predicado(B,C).  
sujeito(A,C) :- artigo(A,B), substantivo(B,C).  
predicado(A,D) :- verbo(A,B), artigo(B,C), substantivo(C,D).  
artigo([o|A],A).  
substantivo([gato|A],A).  
substantivo([rato|A],A).  
verbo([caçou|A],A).
```

## Exercício 1. Uso da gramática para reconhecimento de frases

Digite a gramática do Exemplo 1 e faça as consultas a seguir:

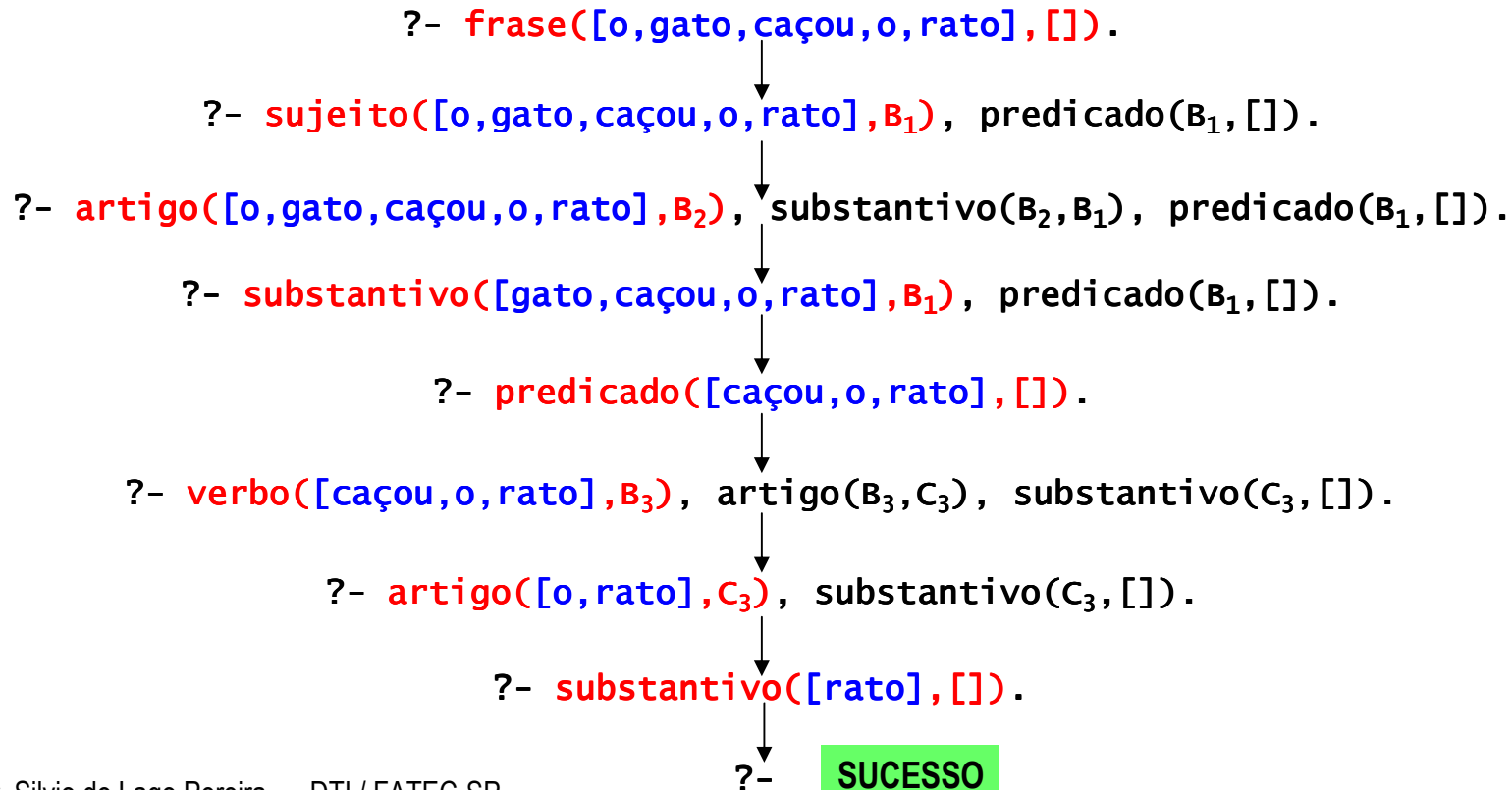
```
?- artigo([o,gato,caçou,o,rato],R).  
?- sujeito([o,gato,caçou,o,rato],R).  
?- frase([o,gato,caçou,o,rato],[]).  
?- frase([o,gato,rato,o,caçou],[]).  
?- frase([o,rato,caçou,o,gato],[]).  
?- frase([gato,caçou,rato],[]).
```





## Reconhecimento de frases

```
/* 1 */ frase(A,C) :- sujeito(A,B), predicado(B,C).  
/* 2 */ sujeito(A,C) :- artigo(A,B), substantivo(B,C).  
/* 3 */ predicado(A,D) :- verbo(A,B), artigo(B,C), substantivo(C,D).  
/* 4 */ artigo([o|A],A).  
/* 5 */ substantivo([gato|A],A).  
/* 6 */ substantivo([rato|A],A).  
/* 7 */ verbo([caçou|A],A).
```





## Reconhecimento de frases

```
/* 1 */ frase(A,C) :- sujeito(A,B), predicado(B,C).  
/* 2 */ sujeito(A,C) :- artigo(A,B), substantivo(B,C).  
/* 3 */ predicado(A,D) :- verbo(A,B), artigo(B,C), substantivo(C,D).  
/* 4 */ artigo([o|A],A).  
/* 5 */ substantivo([gato|A],A).  
/* 6 */ substantivo([rato|A],A).  
/* 7 */ verbo([caçou|A],A).
```

?- frase([o,gato,rato,o,caçou],[]).

?- sujeito([o,gato,rato,o,caçou],B<sub>1</sub>), predicado(B<sub>1</sub>,[]).

?- artigo([o,gato,rato,o,caçou],B<sub>2</sub>), substantivo(B<sub>2</sub>,B<sub>1</sub>), predicado(B<sub>1</sub>,[]).

?- substantivo([gato,rato,o,caçou],B<sub>1</sub>), predicado(B<sub>1</sub>,[]).

?- predicado([rato,o,caçou],[]).

?- verbo([rato,o,caçou],B<sub>3</sub>), artigo(B<sub>3</sub>,C<sub>3</sub>), substantivo(C<sub>3</sub>,[]).

FALHA



## Geração de frases

### Exemplo 1. A gramática completa

```
frase(A,C) :- sujeito(A,B), predicado(B,C).  
sujeito(A,C) :- artigo(A,B), substantivo(B,C).  
predicado(A,D) :- verbo(A,B), artigo(B,C), substantivo(C,D).  
artigo([o|A],A).  
substantivo([gato|A],A).  
substantivo([rato|A],A).  
verbo([caçou|A],A).
```

### Exercício 2. Uso da gramática para geração de frases

Faça as consultas a seguir, que geram todas as frases da linguagem definida pela gramática do Exemplo 1:

```
?- frase(F, []).  
?- forall( frase(F, []), writeln(F) ).
```



## Geração de frases

### Exemplo 1. A gramática completa

```
frase(A,C) :- sujeito(A,B), predicado(B,C).  
sujeito(A,C) :- artigo(A,B), substantivo(B,C).  
predicado(A,D) :- verbo(A,B), artigo(B,C), substantivo(C,D).  
artigo([o|A],A).  
substantivo([gato|A],A).  
substantivo([rato|A],A).  
verbo([caçou|A],A).
```

### Exercício 3. Ampliação do léxico da gramática

Altere a definição da gramática do Exemplo 1, adicionando o artigo **um** e o verbo **assustou**. Em seguida, faça as consultas a seguir:

```
?- frase(F, []).  
?- forall( frase(F, []), writeln(F) ).
```



## Notação DCG (*Definite Clause Grammar*)

- A linguagem Prolog oferece uma notação, denominada DCG, que facilita a especificação de gramáticas. Ao ser compilada, uma gramática em notação DCG é automaticamente transformada em uma gramática usando diferença de listas.

### Exemplo 2. Uma gramática em notação DCG

```
frase --> sujeito, predicado.  
sujeito --> artigo, substantivo.  
predicado --> verbo, artigo, substantivo.  
artigo --> [o].  
substantivo --> [gato] | [rato].  
verbo --> [caçou].
```

### Exercício 4. Análise do resultado da compilação da notação DCG

Digite a gramática do Exemplo 2, compile e faça a consulta a seguir:

```
?- listing.
```



## Categorias gramaticais

- Em PLN, é bastante comum o uso de termos técnicos para designar categorias gramaticais. Alguns destes termos são os seguintes:

**s** : sintagma sentencial (frase)

**sn** : sintagma nominal

**sv** : sintagma verbal

**det** : determinante (artigo)

**n** : nome (substantivo)

**adj** : adjetivo

**v** : verbo

**vi** : verbo intransitivo

**vt** : verbo transitivo

**adv** : advérbio

- A fim de simplificar a notação, usaremos estes termos na especificação das próximas gramáticas.



## Categorias gramaticais

### Exemplo 3. Gramática G1

```
s    --> sn, sv.  
sn   --> det, n, adj.  
sv   --> vi, adv.  
sv   --> vt, adv, sn.  
det  --> [o].  
n    --> [gato] | [rato].  
adj  --> [] | [gordo] | [magro].  
vi   --> [caçou] | [dormiu].  
vt   --> [caçou].  
adv  --> [] | [silenciosamente].
```

### Exercício 5. Teste da gramática G1

Gere e analise todas as frases da linguagem definida pela gramática G1.



## Dependência de contexto

A dependência de contexto ocorre quando uma palavra da frase deve concordar com outras palavras da mesma frase.

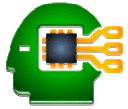
### Exemplo 4. Gramática G2

```
s    --> sn, sv.  
sn   --> det, n.  
sv   --> v, sn.  
det  --> [o].  
n    --> [gato] | [rato].  
v    --> [caçou].
```

### Exercício 6. Ampliação da gramática com gênero feminino

Amplie a gramática G2, adicionando o artigo a e os substantivos gata e rata. Em seguida, gere e analise todas as frases definidas pela gramática alterada.





## Dependência de contexto

### Exemplo 5. Gramática com concordância de gênero G3

```
s      --> sn, sv.
sn     --> det(G), n(G).
sv     --> v, sn.
det(m) --> [o].
det(f) --> [a].
n(m)  --> [gato] | [rato].
n(f)  --> [gata] | [rata].
v      --> [caçou].
```

### Exercício 7. Gramática com concordância de gênero

Gere e analise todas as frases definidas pela gramática G3.

### Exercício 8. Ampliação do léxico

Adicione os artigos indefinidos **um** e **uma** e gere as frases definidas pela gramática.



# Dependência de contexto

## Exemplo 6. Gramática G4

```
s      --> sn, sv.
sn     --> det(G), n(G).
sv     --> v, sn.
det(m) --> [o]      | [um].
det(f) --> [a]      | [uma].
n(m)   --> [gato]   | [rato].
n(f)   --> [gata]   | [rata].
v      --> [caçou].
```

## Exercício 9. Ampliação da gramática com plural

Amplie a gramática G4, adicionando:

- os artigos: os, uns, as, umas
- os substantivos: gatos, ratos, gatas, ratas
- o verbo: caçaram

Em seguida, gere e analise todas as frases definidas pela nova gramática.



## Dependência de contexto

### Exemplo 7. Gramática G5

```
s          --> sn(N), sv(N).
sn(N)      --> det(G, N), n(G, N).
sv(N)      --> v(N), sn(_).
det(m, s)  --> [o]      | [um].
det(m, p)  --> [os]     | [uns].
det(f, s)  --> [a]      | [uma].
det(f, p)  --> [as]     | [umas].
n(m, s)    --> [gato]   | [rato].
n(m, p)    --> [gatos]  | [ratos].
n(f, s)    --> [gata]   | [rata].
n(f, p)    --> [gatas]  | [ratas].
v(s)       --> [caçou].
v(p)       --> [caçaram].
```

### Exercício 10. Teste da gramática G5

Gere e analise todas as frases definidas pela gramática G5.

**Fim**

