

# Prolog - Primeiros passos em Inteligência Artificial

Luiz Alexandre de Sousa Freitas; Mateus Augusto

# Prolog

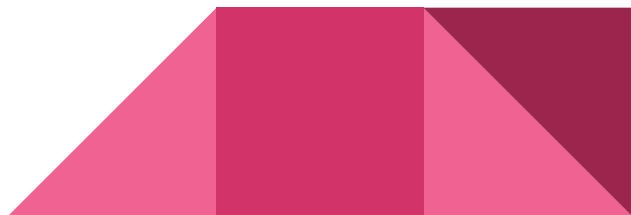


**Wikipedia:** “**Prolog (Programação Lógica)** é uma linguagem de programação que se enquadra no paradigma de Programação em Lógica Matemática. É uma linguagem de uso geral que é especialmente associada com a inteligência artificial e linguística computacional. Consiste numa linguagem puramente lógica.”





**SWI Prolog**



```

/** Operador de crossover */
cruzamento([], []).
cruzamento([Ind*_], [Ind]).
cruzamento([Ind1*_ , Ind2*_ | Resto], [NInd1, NInd2 | Resto1]):-
    gerar_pontos_cruzamento(P1, P2),
    prob_cruzamento(Pcruz), Pc is rand(1),
    ((Pc <= Pcruz, !, cruzar(Ind1, Ind2, P1, P2, NInd1),
      cruzar(Ind2, Ind1, P1, P2, NInd2))
    ;
    (NInd1=Ind1, NInd2=Ind2)),
    cruzamento(Resto, Resto1).

cruzar(Ind1, Ind2, P1, P2, NInd1):-
    sublista(Ind1, P1, P2, Sub1),
    tarefas(NumT),
    R is NumT-P2,
    rotate_right(Ind2, R, Ind21),
    elimina(Ind21, Sub1, Sub2),
    insere(Sub2, Sub1, P2, NInd1).
/*insere Sub2 em Sub1 a partir de P2 obtendo NInd1*/

```

# Hello World em Prolog



# Prolog

## **Desciclopédia: Hello World**

“Não existe programa Hello, World em Prolog porque não há lógica em dizer Alô para o mundo. Como consequência disto, não existe nenhum programa em Prolog porque o primeiro programa que todos escrevem em qualquer linguagem é o Hello, World e se ninguém escreve o primeiro programa não consegue escrever os demais, isso é lógico.”



Agora sim um Hello World em Prolog :D





# Característica

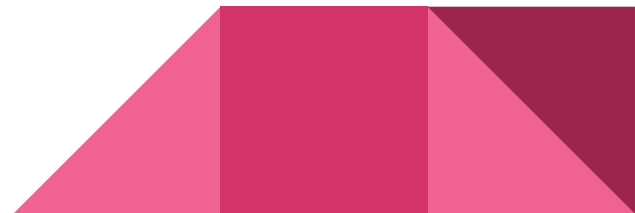
O Prolog é uma linguagem declarativa, significando que em vez de o programa estipular a maneira de chegar à solução, passo a passo, (como nas linguagens procedimentais ou imperativas), limita-se a fornecer uma descrição do problema que se pretende computar. Usa uma coleção base de dados de  *fatos*  e de relações lógicas ( *regras* ) que exprimem o domínio relacional do problema a resolver. Alguns conceitos fundamentais são unificação, recursão, e backtracking.



Um programa pode rodar num modo interativo, a partir de consultas (queries) formuladas pelo usuário, usando a base de dados (os 'fatos') e as regras relacionais (essencialmente implicações lógicas: se.. então), e o mecanismo de unificação para produzir (por uma cadeia de deduções lógicas) a solução.

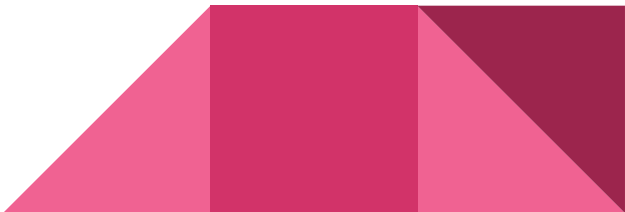


# Elementos da Linguagem



# Átomos

- Definição: cadeias compostas pelos seguintes caracteres:
  - Letras Maiúsculas: A,..., Z (“Variáveis”)
  - Letras Minúsculas: a,..., z (“Argumentos”)
  - Dígitos: 0, 1,..., 9
  - Caracteres especiais: \*, +, \_, -, >, <, =, :, , ~
- Composição de Átomos:
  - Cadeias começando com letras minúsculas. Ex.: x\_y, maria, curso\_de\_IA
  - Cadeias de caracteres especiais. Ex.: <---->, ::=
  - Cadeias de caracteres entre apóstrofos. Ex.: ‘Maria’, ‘casa branca’, ‘a’ (String)



# Números

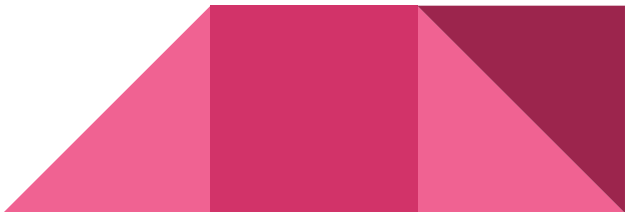
Números: 1, 2.45, -20

# Variáveis

Variáveis são declaradas da mesma forma que átomos, porém iniciando com uma letra maiúscula ou underscore. No ambiente Prolog uma variável não é um contêiner cujo valor pode ser atribuído (como ocorre nas linguagens imperativas). Seu comportamento é mais próximo de um padrão, que é incrementalmente especificado pela unificação.

Variáveis: são cadeias de letras, dígitos e o caractere “\_”.

Ex.: X, Mapa\_da\_mina, \_nome



# Predicados

- Propriedade ou qualidade característica; dote; prenda; virtude; atributo;
- Aquilo que numa oração se enuncia acerca do sujeito.

Ex: A Galinha bota ovo.



# Escopo

- O escopo de uma variável é a asserção (fato, regra, ou consulta) na qual aparece.
- O escopo de qualquer outro nome (constante, nome de função, ou nome de predicado) é todo o programa.

Isto significa que um nome de variável pode ser utilizado e reutilizado a vontade no programa para denotar variáveis diferentes, enquanto qualquer outra notação representa, ou é, o mesmo objeto para o programa todo.



**CONTRA FATOS**



**NÃO HÁ ARGUMENTOS**




# Fatos

Programar em Prolog é bem diferente de programar em uma linguagem procedimental. Em Prolog se fornece fatos e regras para uma base de dados; então se executam consultas ou (*queries*) a essa base de dados. A unidade básica do Prolog é o predicado, que é postulado verdadeiro. Um predicado consiste de uma cabeça e um número de argumentos.



# Regras

As regras são afirmações e questionamentos, dados aos objetos e seus relacionamentos, ou seja seriam como as Queries do bom e velho SQL .

Exemplo de representação de regras: Pedro gosto de todo mundo que gosta de vinho, ou seja, Pedro gosta de X se X gosta de vinho.

Em Prolog, regras consistem de uma cabeça e um corpo. A cabeça e o corpo são conectados pelo símbolo “:-” formado por dois pontos e hífen. O “:-” pronunciase “se”.

A dependência acima seria escrita como: `gosta(pedro, X) :- gosta(X, vinho).`



# Base de conhecimentos

- Estrutura de um fato  

Nome do predicado → gosta(joao, maria). ← Parâmetros
- Estrutura de uma regra  

gosta(joao, X) :-  
gosta(X, vinho),  
gosta(X, futebol). ← Conjunção

**UHUUUUL!!!!**



**AGORA VAMOS PARA A AULA  
PRÁTICA**

GERADORMEMES.COM

# Avaliação

Quando o interpretador recebe uma consulta, ele tenta encontrar predicados que se encaixam na consulta, sejam eles fatos diretos ou regras que possuem o termo consultado como conclusão.

```
irmaos(X,Y) :- filho(X,Z), filho(Y,Z).
```



# Negação

Tipicamente, uma consulta é avaliada como falsa no caso de não estar presente nenhuma regra positiva ou fato que dê suporte ao termo proposto. Isso é chamado hipótese do mundo fechado; assume-se que tudo o que é importante saber está na base de dados, de modo que não existe um mundo exterior que pode possuir evidências desconhecidas. Em outras palavras, se um fato não é conhecido ser verdadeiro (ou falso), assume-se que ele é falso.

```
inlegal(X) :- not(legal(X)).
```



# Fail

Exemplo (Ana gosta de todos os mamíferos exceto de gatos):

```
mamifero(X) :- gato(X).
```

```
mamifero(X) :- cachorro(X).
```

```
mamifero(X) :- rato(X).
```


```
gato(tom).
```

```
rato(jerry).
```

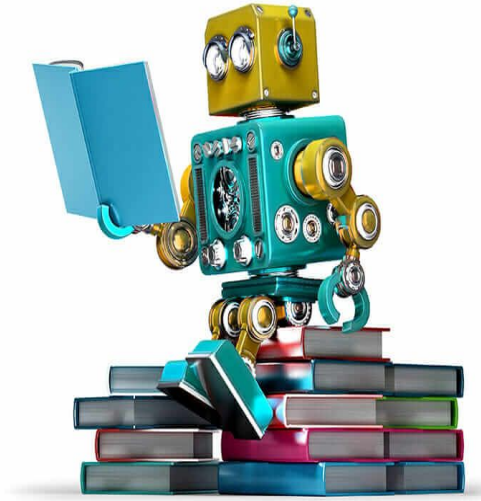
```
cachorro(spike).
```

```
gosta(ana,X) :- gato(X),!,fail.
```

```
gosta(ana,X) :- mamifero(X).
```



# Machine Learning





# Tipos de problemas e tarefas

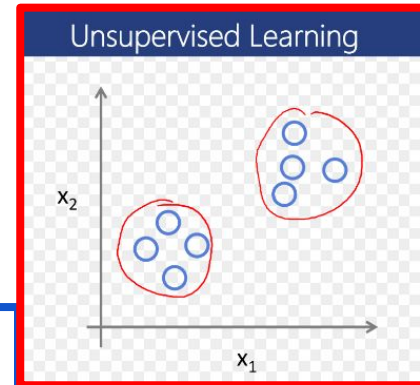
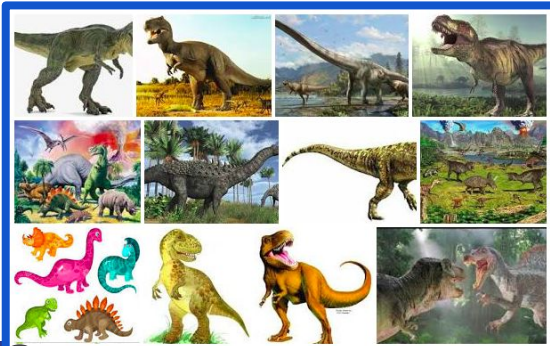
- **Aprendizado supervisionado:** São apresentadas ao computador exemplos de entradas e saídas desejadas, fornecidas por um "professor". O objetivo é aprender uma regra geral que mapeia as entradas para as saídas.

O que é um Dinossauro?



# Tipos de problemas e tarefas

- **Aprendizado não supervisionado:** Nenhum tipo de etiqueta é dado ao algoritmo de aprendizado, deixando-o sozinho para encontrar estrutura nas entradas fornecidas. O aprendizado não supervisionado pode ser um objetivo em si mesmo (descobrir novos padrões nos dados) ou um meio para atingir um fim.



# Tipos de problemas e tarefas

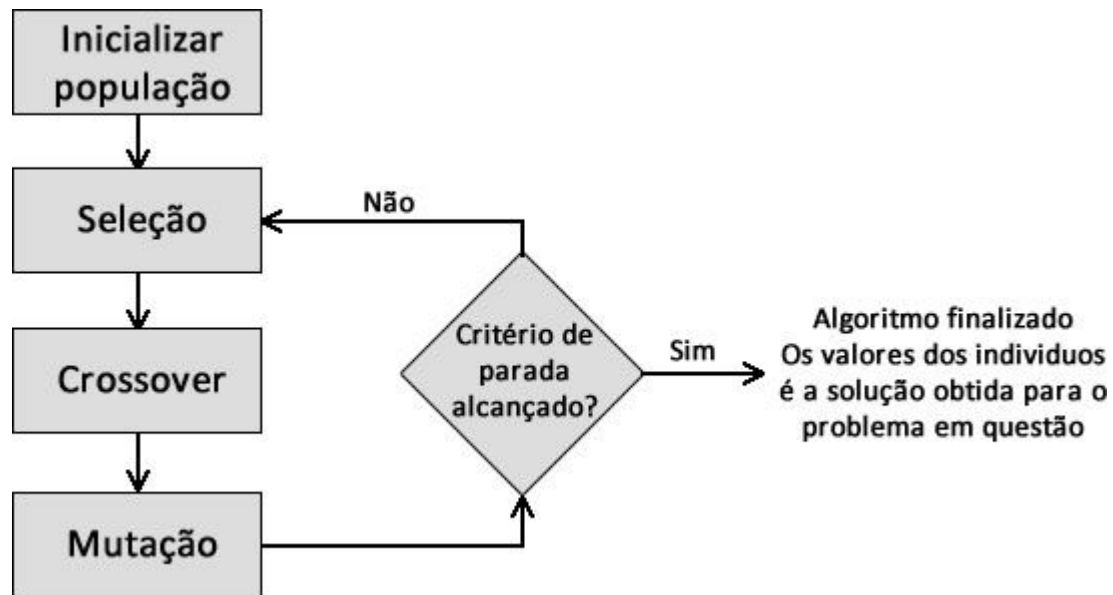
- **Aprendizado por reforço:** Um programa de computador interage com um ambiente dinâmico, em que o programa deve desempenhar determinado objetivo (por exemplo, dirigir um veículo). É fornecido, ao programa, feedback quanto a premiações e punições, na medida em que é navegado o espaço do problema. Outro exemplo de aprendizado por reforço é aprender a jogar um determinado jogo apenas jogando contra um oponente.

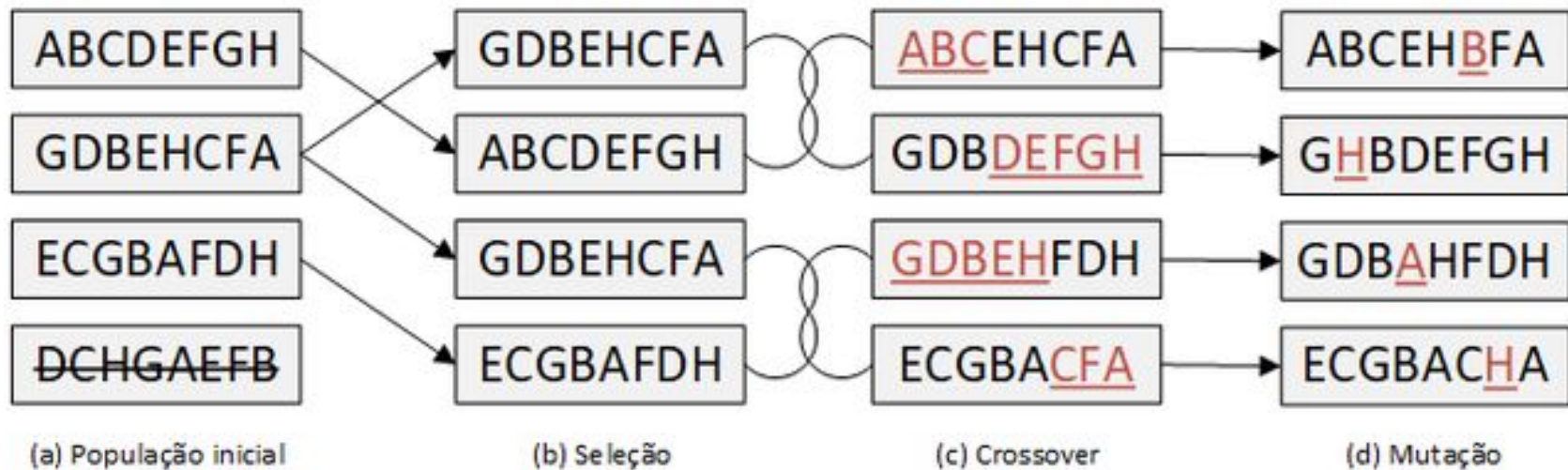


# Algoritmo Genético

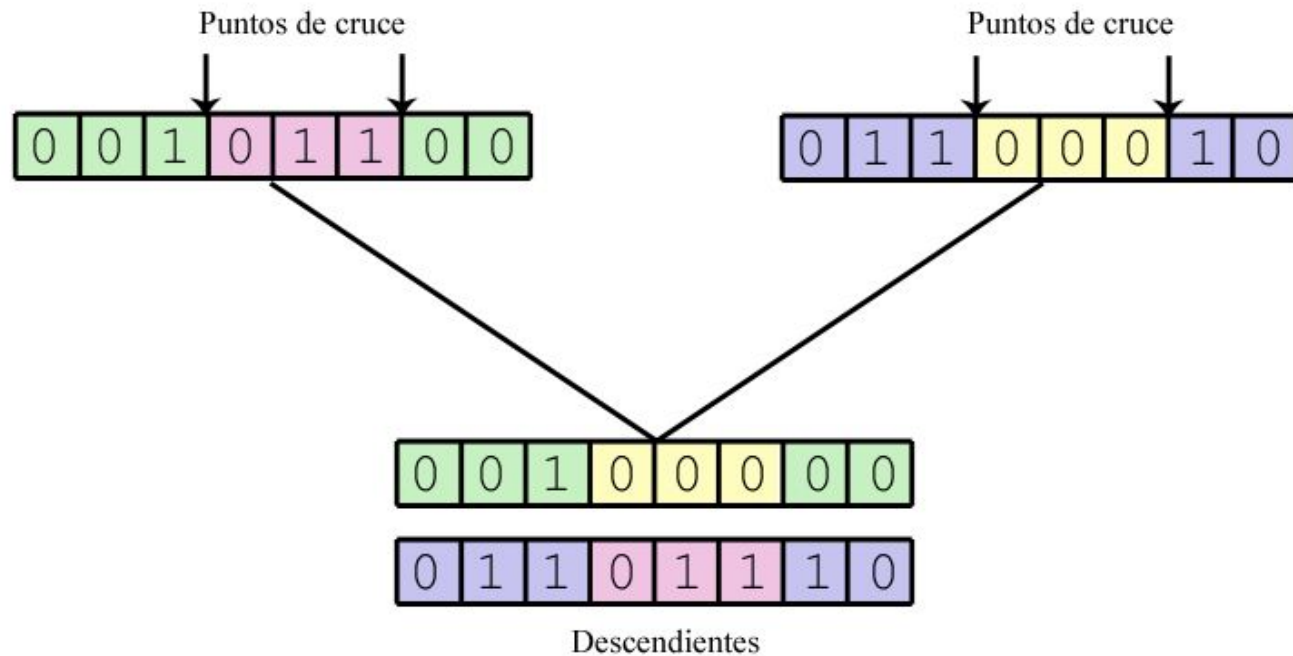
Wikipédia: Um **algoritmo genético (AG)** é uma técnica de busca utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e busca. Algoritmos genéticos são uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação (ou *crossing over*).







Fluxo básico do AG



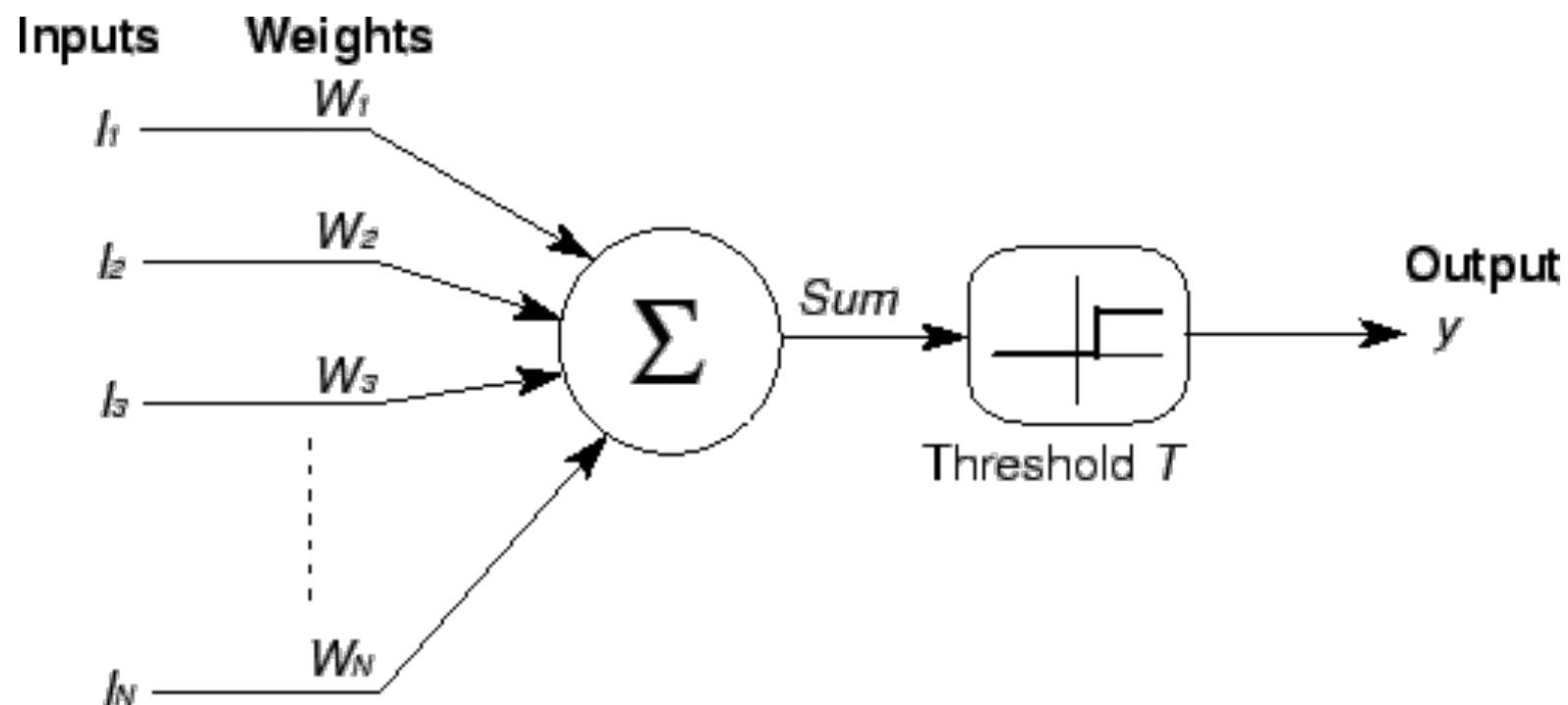
Modelo de Genoma clásico

# Redes Neurais Artificiais

Wikipédia: Em ciência da computação e campos relacionados, **redes neuronais artificiais (RNAs)** são modelos computacionais inspirados pelo sistema nervoso central de um animal (em particular o cérebro) que são capazes de realizar o aprendizado de máquina bem como o reconhecimento de padrões. Redes neurais artificiais geralmente são apresentadas como sistemas de "neurônios interconectados, que podem computar valores de entradas", simulando o comportamento de redes neurais biológicas.







# Neurônios e Sinapses

O primeiro tijolo na construção de uma rede neural são os neurônios.

Um neurônio é como uma função, aceita algumas entradas e devolve algumas saídas.

Existem diferentes tipos de neurônios. Nossa rede irá utilizar neurônios do tipo sigmoide, que levam qualquer número e o comprime para um valor entre 0 ou 1.

O círculo na imagem abaixo ilustra um neurônio do tipo sigmoide. Sua entrada é 5 e sua saída é 1. A seta é a sinapse, que conectam os neurônios as outras camadas da rede.



Exemplo



$$1 * 7 + 0 * 3 - 2 = 5$$

# Referências

<http://dcm.ffclrp.usp.br/~augusto/teaching/ia/IA-Prolog-Conceitos-Avancados.pdf>

<http://www.dei.isep.ipp.pt/~jtavares/ALGAV/TeoricoPraticas.htm>

<http://swi-prolog.org>

