

# Paradigmas de Programação

Prof. Maicon R. Zatelli

Prolog - Programação Lógica  
Problem Solving

Universidade Federal de Santa Catarina  
Florianópolis - Brasil

# Prolog

Prolog pode ser utilizado pra facilmente resolver alguns problemas de lógica.

## Prolog - O Problema dos Navios no Porto

Nesse problema temos 5 navios que estão ancorados em um porto. Cada um deles tem um destino, um horário de partida, uma carga, uma cor de bandeira e um país de origem distintos.

Sua tarefa é encontrar todas as informações a partir das dicas dadas.

O problema pode ser acessado em:

<https://rachacuca.com.br/logica/problemas/navios-no-porto/>

# Prolog - O Problema dos Navios no Porto

	Navio 1	Navio 2	Navio 3	Navio 4	Navio 5
Nacionalidade	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Saída	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Carregamento	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Chaminé	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Destino	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

O navio Grego sai às 6 da manhã e carrega Café.

~~O navio do meio tem a chaminé Preta.~~

O navio Inglês sai às 9 da manhã.

O navio Francês, que tem a chaminé Azul, está à esquerda do navio que carrega Café.

À direita do navio que carrega Cacaú está o navio que vai para Macau.

O navio Brasileiro está indo para Manila.

O navio que carrega Arroz está ancorado ao lado do navio com chaminé Verde.

O navio que vai para Santos sai às 5 da manhã.

O navio Espanhol sai às 7 da manhã e está à direita do navio que vai para Macau.

O navio com a chaminé Vermelha vai para Hamburgo.

O navio que sai às 7 da manhã está ao lado do navio que tem a chaminé Branca.

O navio do canto carrega Milho.

O navio com chaminé Preta sai às 8 da manhã.

O navio que que carrega Milho está ancorado ao lado do navio que carrega Arroz.

O navio que vai para Hamburgo sai às 6 da manhã.

## Demonstração ...

# Prolog - O Problema dos Navios no Porto

Implementando alguns fatos.

## Nacionalidades

```
nacionalidade(brasileiro).  
nacionalidade(espanhol).  
nacionalidade(grego).  
nacionalidade(ingles).  
nacionalidade(frances).
```

# Prolog - O Problema dos Navios no Porto

Implementando alguns fatos.

## Horários de Partida

```
saida(5).  
saida(6).  
saida(7).  
saida(8).  
saida(9).
```

# Prolog - O Problema dos Navios no Porto

Implementando alguns fatos.

## Carregamento

```
carregamento(arroz).  
carregamento(cacau).  
carregamento(cha).  
carregamento(milho).  
carregamento(cafe).
```

# Prolog - O Problema dos Navios no Porto

Implementando alguns fatos.

## Cor do Chaminé

```
chamine(azul).  
chamine(branca).  
chamine(verde).  
chamine(vermelha).  
chamine(preta).
```



# Prolog - O Problema dos Navios no Porto

Implementando alguns fatos.

## Destino

```
destino(hamburgo).  
destino(macau).  
destino(manila).  
destino(santos).  
destino(rotterdam).
```

# Prolog - O Problema dos Navios no Porto

Definimos cada navio como sendo um predicado na forma abaixo

```
navio(Nacionalidade, Saida, Carregamento, Chamine, Destino)
```

A solução terá a forma de uma lista contendo 5 navios, assim:

```
[  
  navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
  navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
  navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
  navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
  navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
]
```

## Prolog - O Problema dos Navios no Porto

Para implementar algumas das dicas dadas no problema, também precisaremos implementar algumas novas regras.

*O navio Francês, que tem a chaminé Azul, está à esquerda do navio que carrega Café.*

```
%X está à esquerda de Y (em qualquer posição à esquerda)  
aEsquerda(X,Y,Lista) :- nth0(IndexX,Lista,X), nth0(IndexY,Lista,Y),  
                        IndexX < IndexY.
```

*O navio Espanhol sai às 7 da manhã e está à direita do navio que vai para Macau.*

```
%X está à direita de Y, se Y está à esquerda de X  
aDireita(X,Y,Lista) :- aEsquerda(Y,X,Lista).
```

## Prolog - O Problema dos Navios no Porto

Para implementar algumas das dicas dadas no problema, também precisaremos implementar algumas novas regras.

*O navio que carrega Arroz está ancorado **ao lado** do navio com chaminé Verde.*

```
%X está à ao lado de Y  
aoLado(X,Y,Lista) :- nextto(X,Y,Lista);nextto(Y,X,Lista).
```

*O navio **do canto** carrega Milho.*

```
%X está no canto se ele é o primeiro ou o último da lista  
noCanto(X,Lista) :- last(Lista,X).  
noCanto(X,[X|_]).
```

# Prolog - O Problema dos Navios no Porto

Agora, podemos implementar a regra para computar a solução do problema, a qual terá o seguinte formato:

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    %Implementaremos as dicas aqui  
  
    ...  
  
    %Ao terminar as dicas, tentaremos completar a solução com  
    todas as possibilidades de valores para as variáveis  
  
    ...  
    .
```

## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    %O navio Grego sai às 6 da manhã e carrega Café.  
    member(navio(grego, 6, cafe, _, _), ListaSolucao),  
  
    ...  
    .
```

Nesta primeira dica, dizemos que um navio de nacionalidade grega, partindo às 6:00 e com carregamento de café faz parte da solução, ou seja, é membro da lista de navios da solução.

## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    %0 navio Grego sai às 6 da manhã e carrega Café.  
    member(navio(grego, 6, cafe, _, _), ListaSolucao),  
  
    %0 navio do meio tem a chaminé Preta.  
    Chamine3 = preta,  
  
    ...  
    .
```

Nesta segunda dica, dizemos que o navio que está na terceira posição da lista possui chaminé preta.

## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio Inglês sai às 9 da manhã.  
    member(navio(ingles, 9, _, _, _), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio de nacionalidade inglesa que sai às 9:00 faz parte da solução



## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %O navio Francês, que tem a chaminé Azul, está à esquerda  
    do navio que carrega Café  
    aEsquerda(navio(frances, _, _, azul, _),  
               navio(_, _, cafe, _, _), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio de nacionalidade francesa e com chaminé azul está à esquerda de um navio que carrega café.

# Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %À direita do navio que carrega Cacau está o navio  
    que vai para Macau.  
    aEsquerda(navio(_, _, cacau, _, _),  
               navio(_, _, _, _, macau), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que o navio que carrega cacau está à esquerda do navio que vai para Macau.

## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio Brasileiro está indo para Manila.  
    member(navio(brasileiro, _, _, _, manila), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio de nacionalidade brasileira e que vai para Manila faz parte da solução.

# Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio que carrega Arroz está ancorado ao lado  
do navio com chaminé Verde.  
    aoLado(navio(_, _, arroz, _, _),  
            navio(_, _, _, verde, _), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio que carrega arroz está ao lado de um navio com chaminé verde.

## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio que vai para Santos sai às 5 da manhã.  
    member(navio(_, 5, _, _, santos), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio que sai às 5:00 e vai para Santos faz parte da solução.

# Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %O navio Espanhol sai às 7 da manhã e está à direita  
    do navio que vai para Macau.  
    aDireita(navio(espanhol, 7, _, _, _),  
             navio(_, _, _, _, macau), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio de nacionalidade espanhola e que sai às 7:00 está à esquerda de um navio que vai para Macau.

## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio com a chaminé Vermelha vai para Hamburgo.  
    member(navio(_, _, _, vermelha, hamburgo), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio com chaminé de cor vermelha e que vai pra Hamburgo faz parte da solução.

# Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio que sai às 7 da manhã está ao lado do navio  
    %que tem a chaminé Branca.  
    aoLado(navio(_, 7, _, _, _),  
           navio(_, _, _, branca, _), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio que sai às 7:00 está ao lado de um navio com chaminé de cor branca.



## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio do canto carrega Milho.  
    noCanto(navio(_, _, milho, _, _), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio que carrega milho está na primeira ou última posição da minha solução.

## Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio com chaminé Preta sai às 8 da manhã  
    member(navio(_, 8, _, preta, _), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio com chaminé preta e que sai às 8:00 faz parte da solução.

# Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio que que carrega Milho está ancorado ao lado  
do navio que carrega Arroz.  
    aoLado(navio(_, _, milho, _, _),  
            navio(_, _, arroz, _, _), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio que carrega milho está ao lado de um navio que carrega arroz.

# Prolog - O Problema dos Navios no Porto

```
solucao(ListaSolucao) :-  
    ListaSolucao = [  
        navio(Nacionalidade1, Saida1, Carregamento1, Chamine1, Destino1),  
        navio(Nacionalidade2, Saida2, Carregamento2, Chamine2, Destino2),  
        navio(Nacionalidade3, Saida3, Carregamento3, Chamine3, Destino3),  
        navio(Nacionalidade4, Saida4, Carregamento4, Chamine4, Destino4),  
        navio(Nacionalidade5, Saida5, Carregamento5, Chamine5, Destino5)  
    ],  
  
    ...  
  
    %0 navio que vai para Hamburgo sai às 6 da manhã.  
    member(navio(_, 6, _, _, hamburgo), ListaSolucao),  
  
    ...  
    .
```

Nesta dica, dizemos que um navio que sai às 6:00 e vai pra Hamburgo faz parte da solução.

# Prolog - O Problema dos Navios no Porto

Uma vez finalizada as dicas, temos que testar todas as possibilidades de nacionalidades, horários de partida, carregamentos, chaminés, e destinos.

```
solucao(ListaSolucao) :-  
    ...  
  
    %0 navio que vai para Hamburgo sai às 6 da manhã.  
    member(navio(_, 6, _, _, hamburgo), ListaSolucao),  
  
    nacionalidade(Nacionalidade1),  
    nacionalidade(Nacionalidade2),  
    nacionalidade(Nacionalidade3),  
    nacionalidade(Nacionalidade4),  
    nacionalidade(Nacionalidade5),  
    todosDiferentes([Nacionalidade1, Nacionalidade2,  
                     Nacionalidade3, Nacionalidade4,  
                     Nacionalidade5]),  
  
    ...  
    .
```

## Prolog - O Problema dos Navios no Porto

A regra `todosDiferentes(L)` verifica se não há nenhum membro repetido em uma lista passada como parâmetro.

```
todosDiferentes([]).  
todosDiferentes([H|T]) :- not(member(H,T)), todosDiferentes(T).
```

Uma lista não contém números repetidos se a cabeça não ocorre na cauda e se na cauda também todos os elementos são diferentes.

## Prolog - O Problema dos Navios no Porto

Uma vez finalizada as dicas, temos que testar todas as possibilidades de nacionalidades, horários de partida, carregamentos, chaminés, e destinos.

```
solucao(ListaSolucao) :-  
    ...  
  
    saida(Saida1),  
    saida(Saida2),  
    saida(Saida3),  
    saida(Saida4),  
    saida(Saida5),  
    todosDiferentes([Saida1, Saida2,  
                      Saida3, Saida4,  
                      Saida5]),  
  
    ...  
    .
```

## Prolog - O Problema dos Navios no Porto

Uma vez finalizada as dicas, temos que testar todas as possibilidades de nacionalidades, horários de partida, carregamentos, chaminés, e destinos.

```
solucao(ListaSolucao) :-  
    ...  
  
    carregamento(Carregamento1),  
    carregamento(Carregamento2),  
    carregamento(Carregamento3),  
    carregamento(Carregamento4),  
    carregamento(Carregamento5),  
    todosDiferentes([Carregamento1, Carregamento2,  
                     Carregamento3, Carregamento4,  
                     Carregamento5]),  
  
    ...  
    .
```



## Prolog - O Problema dos Navios no Porto

Uma vez finalizada as dicas, temos que testar todas as possibilidades de nacionalidades, horários de partida, carregamentos, chaminés, e destinos.

```
solucao(ListaSolucao) :-  
    ...  
  
    chamine(Chamine1),  
    chamine(Chamine2),  
    chamine(Chamine3),  
    chamine(Chamine4),  
    chamine(Chamine5),  
    todosDiferentes([Chamine1, Chamine2,  
                      Chamine3, Chamine4,  
                      Chamine5]),  
  
    ...  
    .
```

## Prolog - O Problema dos Navios no Porto

Uma vez finalizada as dicas, temos que testar todas as possibilidades de nacionalidades, horários de partida, carregamentos, chaminés, e destinos.

```
solucao(ListaSolucao) :-  
    ...  
  
    destino(Destino1),  
    destino(Destino2),  
    destino(Destino3),  
    destino(Destino4),  
    destino(Destino5),  
    todosDiferentes([Destino1, Destino2,  
                     Destino3, Destino4,  
                     Destino5])  
    .
```

Esta é a última parte da solução!

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['navios.pl'].  
% navios.pl compiled 0.00 sec, 35 clauses  
true.  
  
?- solucao(L).
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['navios.pl'].  
% navios.pl compiled 0.00 sec, 35 clauses  
true.  
  
?- solucao(L).  
L = [navio(frances, 5, cacau, azul, santos),  
     navio(grego, 6, cafe, vermelha, hamburgo),  
     navio(brasileiro, 8, cha, preta, manila),  
     navio(ingles, 9, arroz, branca, macau),  
     navio(espanhol, 7, milho, verde, rotterdam)]
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['navios.pl'].
% navios.pl compiled 0.00 sec, 35 clauses
true.

?- solucao(L).
L = [navio(frances, 5, cacau, azul, santos),
     navio(grego, 6, cafe, vermelha, hamburgo),
     navio(brasileiro, 8, cha, preta, manila),
     navio(ingles, 9, arroz, branca, macau),
     navio(espanhol, 7, milho, verde, rotterdam)] ;
L = [navio(frances, 5, cha, azul, santos),
     navio(grego, 6, cafe, vermelha, hamburgo),
     navio(brasileiro, 8, cacau, preta, manila),
     navio(ingles, 9, arroz, branca, macau),
     navio(espanhol, 7, milho, verde, rotterdam)]
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['navios.pl'].
% navios.pl compiled 0.00 sec, 35 clauses
true.

?- solucao(L).
L = [navio(frances, 5, cacau, azul, santos),
     navio(grego, 6, cafe, vermelha, hamburgo),
     navio(brasileiro, 8, cha, preta, manila),
     navio(ingles, 9, arroz, branca, macau),
     navio(espanhol, 7, milho, verde, rotterdam)] ;
L = [navio(frances, 5, cha, azul, santos),
     navio(grego, 6, cafe, vermelha, hamburgo),
     navio(brasileiro, 8, cacau, preta, manila),
     navio(ingles, 9, arroz, branca, macau),
     navio(espanhol, 7, milho, verde, rotterdam)] ;
false.
```

# Sudoku

## Prolog - Sudoku

Para resolver o Sudoku em Prolog, seguimos a mesma estratégia do problema anterior. Tentamos atribuir valores para as casas em aberto utilizando todas as possibilidades e verificando se em cada linha, coluna, e quadrado menor temos todos os números distintos.

Para esta implementação, utilizaremos um tabuleiro 4x4.



## Prolog - Sudoku

Definimos o tabuleiro inicial e quais são os valores possíveis para nosso tabuleiro. Neste caso, os valores vão de 1 até 4.

```
tabuleiro([  
    [_ , _ , _ , 4],  
    [_ , 3 , _ , _],  
    [1 , _ , _ , _],  
    [_ , _ , 2 , _]  
]).
```

```
n(1).  
n(2).  
n(3).  
n(4).
```

# Prolog - Sudoku

Usaremos a regra `todosDiferentes` já definida anteriormente.

```
todosDiferentes([]).  
todosDiferentes([H|T]) :- not(member(H,T)), todosDiferentes(T).
```

## Prolog - Sudoku

Esta regra permitirá completar os valores restantes e verificar se todos são diferentes. Assim, dada uma lista com 4 elementos como parâmetro, complete os que faltarem, tentando todas as combinações, e mantendo a lista sem nenhum valor repetido.

```
completa([X1, X2, X3, X4]) :-  
    n(X1), n(X2), n(X3), n(X4),  
    todosDiferentes([X1, X2, X3, X4]).
```

## Prolog - Sudoku

A regra **solucao** retorna um tabuleiro solução para o problema do Sudoku. Assim, o primeiro passo é recuperar o estado inicial do tabuleiro, o qual está definido em nossos fatos.

```
solucao(TabuleiroSolucao) :-  
    TabuleiroSolucao = tabuleiro([  
        [X11, X12, X13, X14],  
        [X21, X22, X23, X24],  
        [X31, X32, X33, X34],  
        [X41, X42, X43, X44]  
    ]),  
  
    ...  
  
    .
```

# Prolog - Sudoku

Em seguida, deve-se resolver as linhas, colunas e quadrados menores.

```
solucao(TabuleiroSolucao) :-  
    TabuleiroSolucao = tabuleiro([  
        [X11, X12, X13, X14],  
        [X21, X22, X23, X24],  
        [X31, X32, X33, X34],  
        [X41, X42, X43, X44]  
    ]),  
  
    %linhas  
    ...  
    %colunas  
    ...  
    %quadrados menores  
    ...  
    .
```

# Prolog - Sudoku

Para cada linha, tente completar todos os valores.

```
solucao(TabuleiroSolucao) :-  
    TabuleiroSolucao = tabuleiro([  
        [X11, X12, X13, X14],  
        [X21, X22, X23, X24],  
        [X31, X32, X33, X34],  
        [X41, X42, X43, X44]  
    ]),  
  
    %linhas  
    completa([X11, X12, X13, X14]),  
    completa([X21, X22, X23, X24]),  
    completa([X31, X32, X33, X34]),  
    completa([X41, X42, X43, X44]),  
    ...  
    .
```

# Prolog - Sudoku

Para cada coluna, tente completar todos os valores.

```
solucao(TabuleiroSolucao) :-  
    TabuleiroSolucao = tabuleiro([  
        [X11, X12, X13, X14],  
        [X21, X22, X23, X24],  
        [X31, X32, X33, X34],  
        [X41, X42, X43, X44]  
    ]),  
    ...  
    %colunas  
    completa([X11, X21, X31, X41]),  
    completa([X12, X22, X32, X42]),  
    completa([X13, X23, X33, X43]),  
    completa([X14, X24, X34, X44]),  
    ...  
    .
```

# Prolog - Sudoku

Para cada quadrado menor, tente completar todos os valores.

```
solucao(TabuleiroSolucao) :-  
    TabuleiroSolucao = tabuleiro([  
        [X11, X12, X13, X14],  
        [X21, X22, X23, X24],  
        [X31, X32, X33, X34],  
        [X41, X42, X43, X44]  
    ]),  
    ...  
    %quadrados menores  
    completa([X11, X12, X21, X22]),  
    completa([X31, X32, X41, X42]),  
    completa([X13, X14, X23, X24]),  
    completa([X33, X34, X43, X44]).
```

E assim termina nossa solução!



# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['sudoku.pl'].  
% sudoku.pl compiled 0.00 sec, 2 clauses  
true.  
  
?- solucao(Tab).
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['sudoku.pl'].  
% sudoku.pl compiled 0.00 sec, 2 clauses  
true.  
  
?- solucao(Tab).  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,1,4,3], [4,3,2,1]])
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['sudoku.pl'].  
% sudoku.pl compiled 0.00 sec, 2 clauses  
true.  
  
?- solucao(Tab).  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,1,4,3], [4,3,2,1]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,3,4,1], [4,1,2,3]])
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['sudoku.pl'].  
% sudoku.pl compiled 0.00 sec, 2 clauses  
true.  
  
?- solucao(Tab).  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,1,4,3], [4,3,2,1]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,3,4,1], [4,1,2,3]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [4,1,2,3], [2,3,4,1]])
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['sudoku.pl'].  
% sudoku.pl compiled 0.00 sec, 2 clauses  
true.  
  
?- solucao(Tab).  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,1,4,3], [4,3,2,1]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,3,4,1], [4,1,2,3]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [4,1,2,3], [2,3,4,1]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [4,3,2,1], [2,1,4,3]])
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['sudoku.pl'].  
% sudoku.pl compiled 0.00 sec, 2 clauses  
true.  
  
?- solucao(Tab).  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,1,4,3], [4,3,2,1]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,3,4,1], [4,1,2,3]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [4,1,2,3], [2,3,4,1]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [4,3,2,1], [2,1,4,3]]) .
```

# Prolog - O Problema dos Navios no Porto

Testando a solução...

```
?- ['sudoku.pl'].  
% sudoku.pl compiled 0.00 sec, 2 clauses  
true.  
  
?- solucao(Tab).  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,1,4,3], [4,3,2,1]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [2,3,4,1], [4,1,2,3]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [4,1,2,3], [2,3,4,1]]) ;  
Tab = tabuleiro([[1,2,3,4], [3,4,1,2], [4,3,2,1], [2,1,4,3]]) .
```

Note que há muitas soluções possíveis a partir do estado inicial do tabuleiro.

## Prolog - Alguns Links Úteis

- `http://www.swi-prolog.org/pldoc/man?section=lists`
- `http://lpn.swi-prolog.org/lpnpage.php?pageid=online`
- `http://www.swi-prolog.org/pldoc/doc/_CWD_/index.html`



# Prolog

Ver atividade no Moodle