A.I.A.D.P

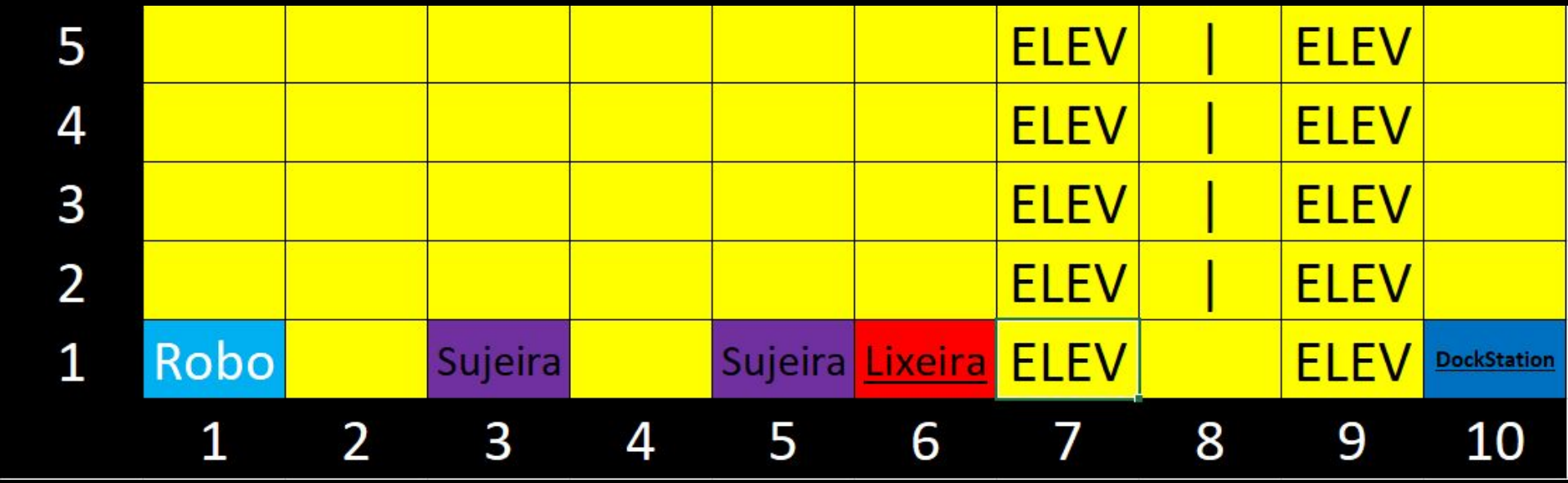
Agente Inteligente de Aspirador de Pó

Definição dos Estados

```
[Posicao,Sacola,Sujeiras]
 Posicao: p(X,Y) (Ex.: p(3,1))
 Sacola: Número Inteiro de 0 a 2
 Sujeiras: [P1,P2,P3,...]
```

Definição dos Estados

```
[p(1,1),0,[p(3,1), p(5,1)]
```


5
4
3
2
1
ELEV
|
Robo
Sujeira
Sujeira
Lixeira
DockStation
1
2
3
4
5
6
7
8
9
10

Modelagem

OS OBJETOS SÃO FATOS

Modelagem - Lixeira

```
lixeira(Pos)

lixeira(p(1,3)).

lixeira(p(10,5)).
```

Modelagem - Powerstation

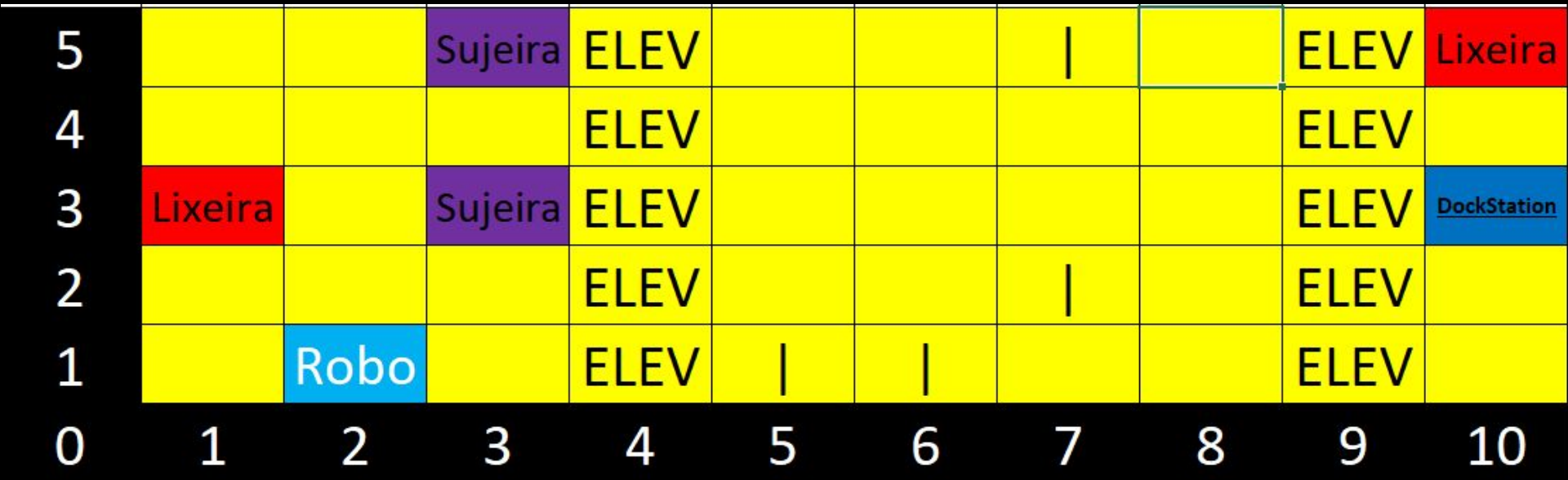
```
powerstation(Pos)

powerstation(p(10,1)).
```

Modelagem - Parede

```
parede(Pos)

  parede(p(5,1)).

  parede(p(6,1)).

  parede(p(7,2)).

  parede(p(7,5)).
```

Modelagem - Elevador

```
% é definido apenas a posição X
elevador(X)
  elevador(4).
  elevador(9).
```


5
4
3
2
1
0
1
2
3
4
5
6
7
8
9
10
Sujeira
ELEV
Lixeira
Lixeira
Sujeira
DockStation
Robo

Regras de Sucessão - Pegando sujeira

```
s([Pos, Sacola, Sujeiras], [Pos,Sacola2,Sujeiras2]):-
   pertence(Pos,Sujeiras),
retirar_elemento(Pos,Sujeiras,Sujeiras2),
   Sacola < 2,
   Sacola2 is Sacola + 1.
```

Regra Pertence

```
pertence(Elem,[Elem|_]).
pertence(Elem,[_|Cauda]) :-
    pertence(Elem,Cauda).
```

Regra Retirar Elemento

```
retirar_elemento(Elem,[Elem|Cauda],Cauda).
retirar_elemento(Elem,[Cabeca|Cauda],
[Cabeca|Resultado]) :-
   retirar_elemento(Elem,Cauda,Resultado).
```

Regras de Sucessão - Esvaziando sacola

```
s([Pos,Sacola,Sujeiras],[Pos,Sacola2,Sujeiras]) :-
  lixeira(Pos),
  Sacola > 0,
  Sacola2 is 0.
```

Regras de Sucessão - Andando na horizontal

```
s([p(X, Y), Sacola, Sujeiras],
  [p(SX, Y), Sacola, Sujeiras]) :-
      (SX is X + 1 ; SX is X - 1),
      pode_passar(p(X,Y),p(SX,Y)).
```

Regra Pode Passar

```
pode_passar(_,Pos2) :-
  not(parede(Pos2)),
  not(fora_do_mapa(Pos2)).
```

```
fora_do_mapa(p(X,Y)) :-
  X = 0;
  Y = 0;
  X = 11;
  Y = 6.
```

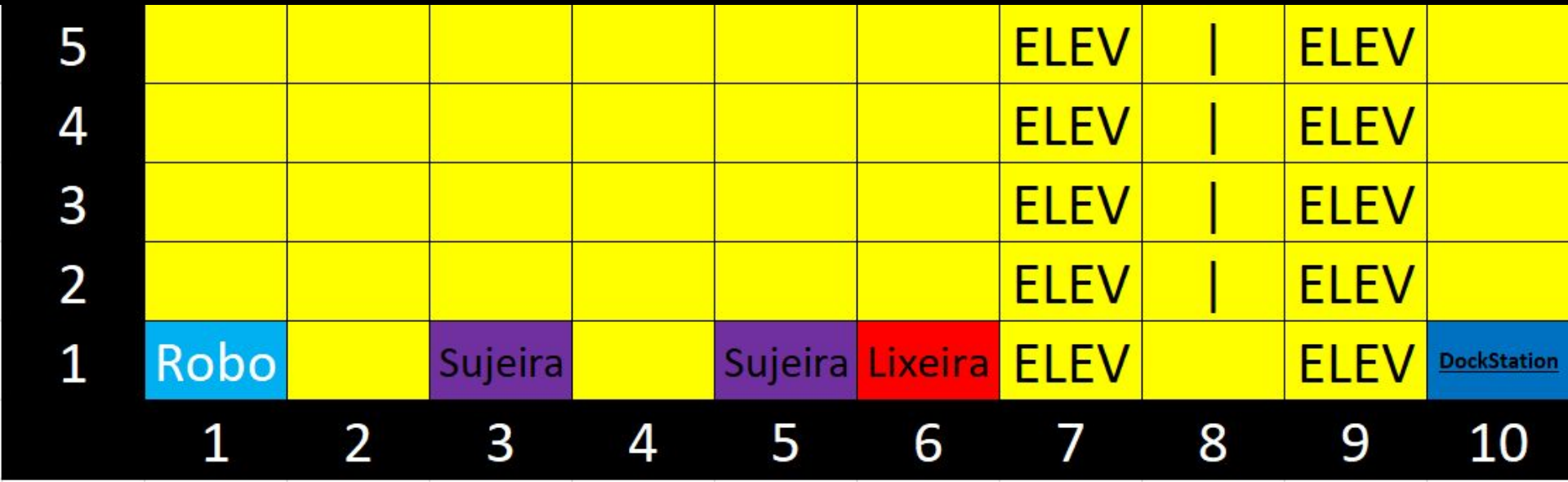
Regras de Sucessão - Subindo no Elevador

```
s([p(X,Y), Sacola,Sujeiras],
  [p(X,SY),Sacola,Sujeiras]) :-
      elevador(X),
      (SY is Y + 1; SY is Y - 1),
      not(fora_do_mapa(p(X,SY))).
```

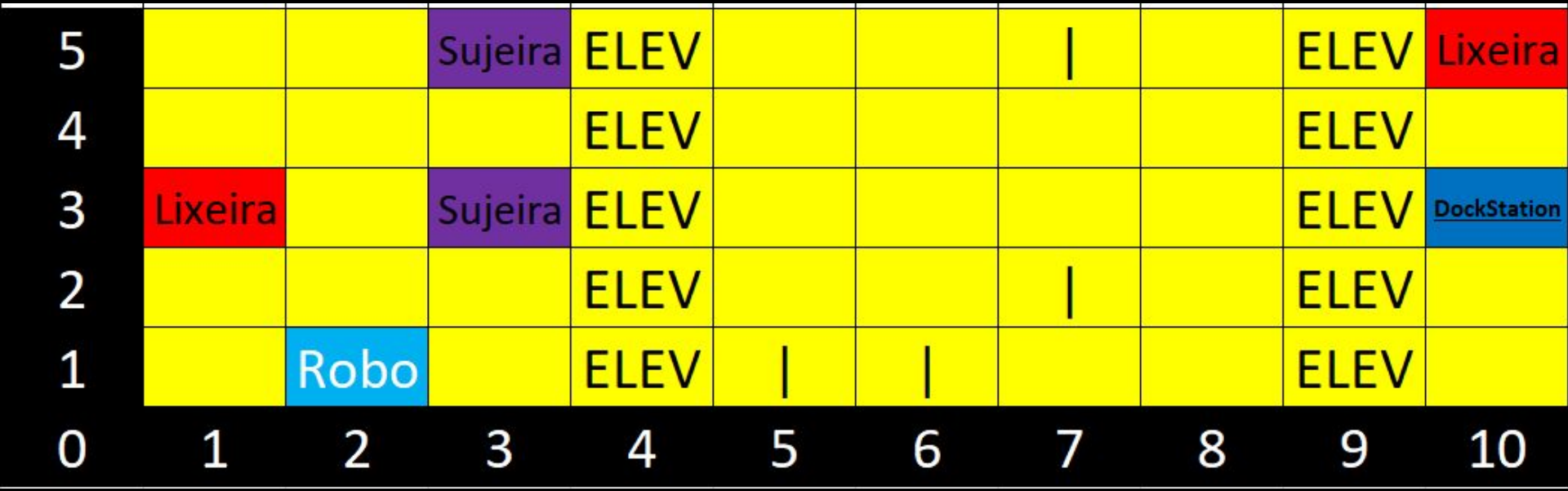
Meta

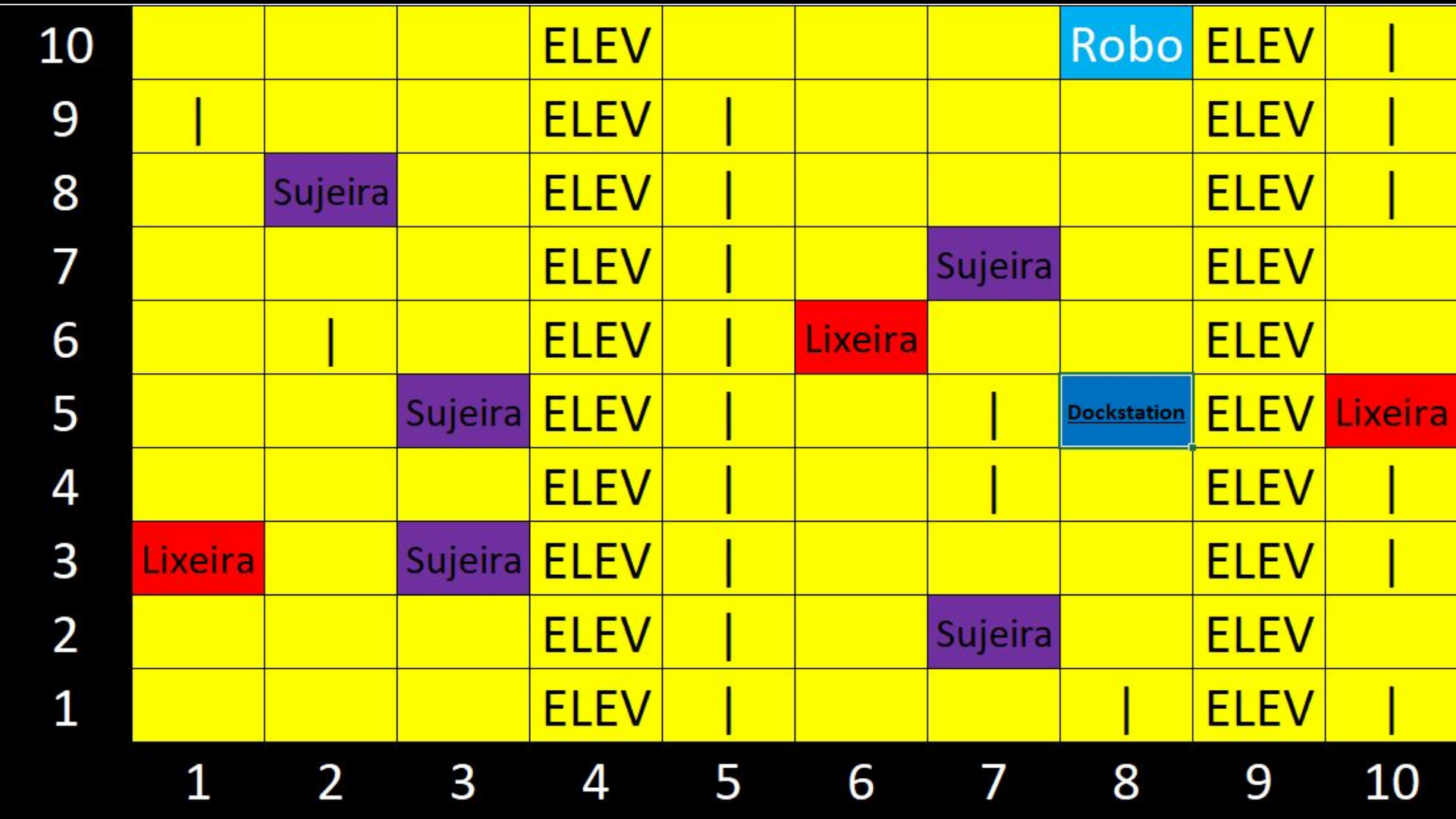
```
meta([Pos, 0, Lixos]) :-
  powerstation(Pos),
  Lixos = [].
```

Caso 0


5
4
3
2
1
ELEV
|
Robo
Sujeira
Sujeira
Lixeira
DockStation
1
2
3
4
5
6
7
8
9
10

Caso 1


5
4
3
2
1
0
Sujeira
ELEV
|
Lixeira
DockStation
Lixeira
Sujeira
Robo
1
2
3
4
5
6
7
8
9
10


10
9
8
7
6
5
4
3
2
1
ELEV
Robo
Sujeira
Lixeira
Dockstation
1
2
3
4
5
6
7
8
9
10

Referências

- Para produção desse projeto foram utilizados trechos de códigos disponibilizados nas aulas de Inteligência Artificial, Pelo Professor Murilo Naldi.
- Também foi consultada a documentação do swi-prolog, disponível em: http://www.swi-prolog.org/pldoc/index.html