

# **MANUAL DE USO**

# **DAMAS**

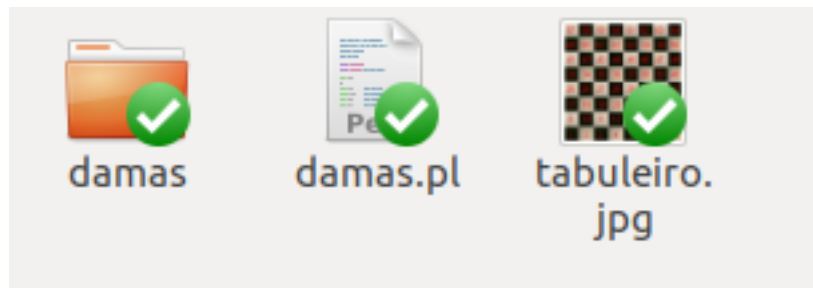
Jaine da Conceição Santos

# Sumário

1- Para Jogar.....	3
2- Para alimentar com agentes.....	4

## Para Jogar:

1. Faça o download do JPL 7.4. Pode ser encontrado em:  
<https://mvnrepository.com/artifact/jpl/jpl/7.4.0>
2. Adicione ao projeto java já existente (que representa a interface) na sua IDE, a biblioteca jpl-7.4.0.jar.
3. Modifique o LD\_LIBRARY\_PATH para conter os diretórios que contêm libjava.so, libjni.so e libjsig.so.
4. Busque os caminhos desses diretórios usando o seguinte comando: `dpkg --search libjsig.so libjava.so`
5. Faça o seu comando final seguindo o exemplo desse:  
`LD_LIBRARY_PATH=/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server:/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/ swipl`
6. Deixe o `codigo.pl` na mesma pasta que o pacote com as classes Java da interface estiver, juntamente com a imagem `tabuleiro.jpg`.



7. Depois, no terminal utilize o comando final para acessar o swipl:  
`LD_LIBRARY_PATH=/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server:/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/ swipl`
8. Posteriormente execute os comandos:  
`>consult(damas).`  
`>inicio.`

Agora é só jogar e se divertir! :D

**\*Na máquina que executou o projeto roda o sistema operacional Ubuntu 16.10, a versão do swipl instalada é a SWI-Prolog version 7.4.2 for amd64 e a versão do java é a 8.**

## Para alimentar com agentes:

Na implementação da jogada da máquina (*processa*) foi tentado utilizar o aspecto de agência que é um mecanismo simplificado para que o usuário alimente com sua lógica. Segue abaixo o predicado que permite o usuário alimentar com agentes.

```
processa([LB,LP,LD],'Preto',_, '1',0):-          %jogada da maquina
    interface([LB,LP,LD]),nl,
    pick_nums(PI1,LP,AUX,PF1),                    %lógica da maquina(aleatoria)
    ehDama(LD,PI1,D),
    ehJogadaValidaMaquina(D,E,[LB,LP,LD],Turno,X,PI1,PF1,C),
    estadoMaquina(E,X,Y,Turno,PF1),
    jpl_call('javax.swing.JOptionPane',showMessageDialog,[@null,'AmaquinaJogou'],_),
    trocarTurno(Turno,Turno2),
    processa(Y,Turno2,_, '1',C).
```

Assim, toda a lógica da máquina está no predicado *pick\_nums*, que na implementação feita para a extensão atual foi utilizada uma lógica aleatória. Logo, se alguém quiser entrar com uma nova lógica, basta remover *pick\_nums* e colocar seu novo predicado. Segue abaixo a imagem da implementação da lógica de *pick\_nums*.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  CALCULO DA JOGADA DO COMPUTADOR  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

chute(N,S):-repeat, S is random(N).                %jogada atual é aleatória
selecionaPF(1,PI1,PF1):- PF1 is PI1-7.
selecionaPF(2,PI1,PF1):- PF1 is PI1-9.
selecionaPF(3,PI1,PF1):- PF1 is PI1-18.
selecionaPF(4,PI1,PF1):- PF1 is PI1-14.
selecionaPF(5,PI1,PF1):- PF1 is PI1+7.
selecionaPF(6,PI1,PF1):- PF1 is PI1+9.
selecionaPF(7,PI1,PF1):- PF1 is PI1+18.
selecionaPF(8,PI1,PF1):- PF1 is PI1+14.

pick_nums(PI1,LP,AUX,PF1):-
    repeat, random_member(PI1,LP),chute(9,AUX),selecionaPF(AUX,PI1,PF1).
```