Inteligência Artificial – Adji-Boto

# Manual de Utilizador

## Índice

Conteúdo

[Manual de Utilizador 1](#_Toc532864350)

[Índice 1](#_Toc532864351)

[Acrónimos e convenções utilizadas 1](#_Toc532864352)

[Introdução 2](#_Toc532864353)

[Instalação 3](#_Toc532864354)

[Iniciar Programa 3](#_Toc532864355)

[Ver Tabuleiros Existentes 3](#_Toc532864356)

[Novos Tabuleiros 4](#_Toc532864357)

[Escrita Resultados 4](#_Toc532864358)

[Algoritmos de Procura 4](#_Toc532864359)

## Acrónimos e convenções utilizadas

Nó: Estrutura de dados que guarda em forma de lista o estado atual do tabuleiro e valores relacionados.

Lista:

## Introdução

Este manual tem como objetivo dar a conhecer a utilização do Adji-Boto em versão simplificada, desenvolvido em LISP.

O jogo contém um tabuleiro com 2 linhas e 6 colunas contendo várias peças em cada posição. Em cada jogada é escolhida uma posição para remover todas a peças da mesma e no sentido contrário aos ponteiros do relógio distribuir peça a peça em cada posição. No final se existirem 1, 3 ou 5 peças na última posição essas são recolhidas e saem permanentemente do tabuleiro.

Já o programa em si analisa um tabuleiro já pré-definido ou recebido pelo utilizador e escolhendo um algoritmo devolve a solução ótima do tabuleiro onde este se encontra com zero peças em todas as posições

Através das opções fornecidas pelos menus é possível:

- Resolver um tabuleiro (Pré-definido ou introduzido)

- Visualizar um tabuleiro

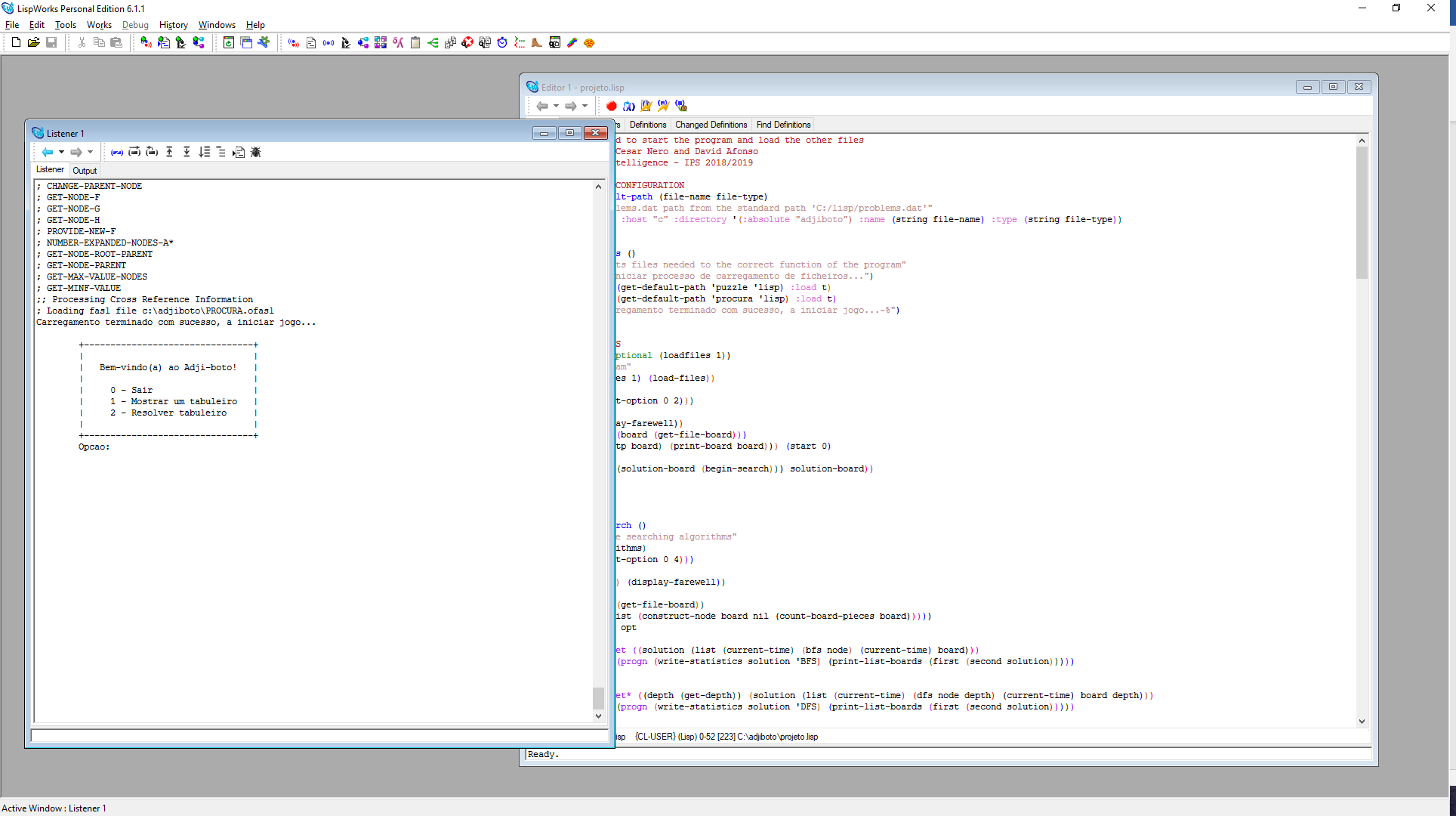
- Analisar os resultados de um tabuleiro depois de o mesmo se encontrar resolvido.

## Instalação

Para um normal funcionamento do programa é necessário haver um directório em ‘C:\adjiboto’ com o ficheiro ‘problemas.dat’ responsável pelos tabuleiros pré-definidos e respectivos ficheiros do programa ‘projeto.lisp’, ‘puzzle.lisp’ e ‘procura.lisp’.

## Iniciar Programa

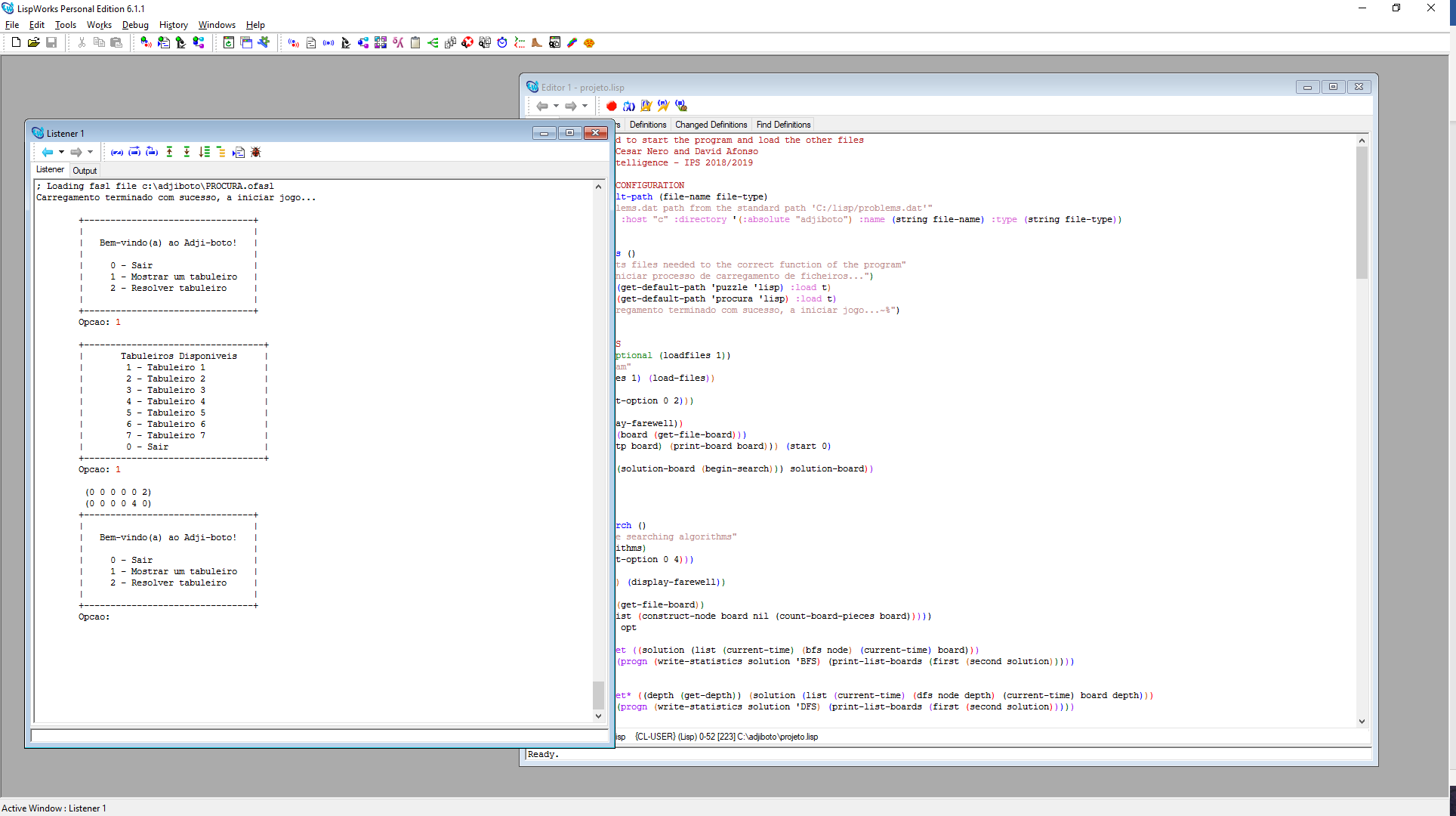
Para iniciar o programa é necessário a executar o ficheiro ‘projeto.lisp’ e de seguida a função (start). Ao executar a função o seguinte menu aparecerá:



De seguida poderá navegar pelos menus digitando o número pretendido e de seguida dando Enter.

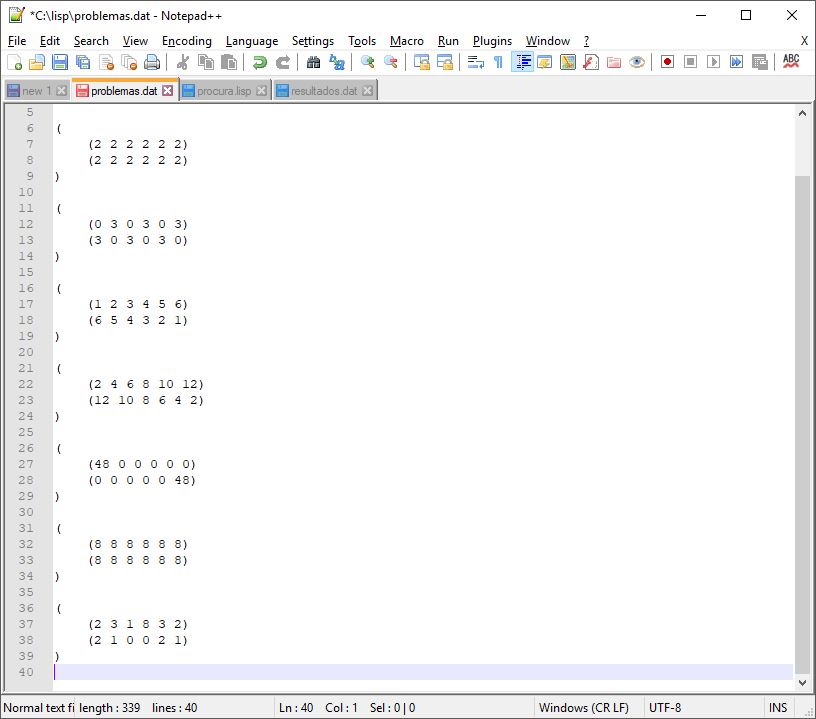
## Ver Tabuleiros Existentes

É possível os tabuleiros existentes escolhendo a opção 2 no menu inicial, que posteriormente irá apresentar uma lista de tabuleiros onde basta escolher um número de modo a visualizar o tabuleiro.



## Novos Tabuleiros

Em qualquer momento é possível adicionar novos tabuleiros sem necessidade de reiniciar o programa. Para isto basta modificar o ficheiro ‘problemas.dat’ e acrescentar no final um tabuleiro dando enter entre o tabuleiro anterior. Exemplo de um novo tabuleiro:



## Escrita Resultados

Após a resolução de um tabuleiro, irá aparecer na janela a resolução do tabuleiro, apresentado desde o nó inicial até ao final. Além disto será escrita estatísticas no ficheiro ‘resultados.dat’ que podem ser encontrados na mesma pasta onde se encontra o resto dos ficheiros ‘C:\adjiboto’.

O ficheiro de estatísticas poderá ser encontrado alguns valores como:

- Estado inicial

- Estado final

- Algoritmo utilizado

- Data de início e fim

- Número de nós gerados

- Número de nós expandidos

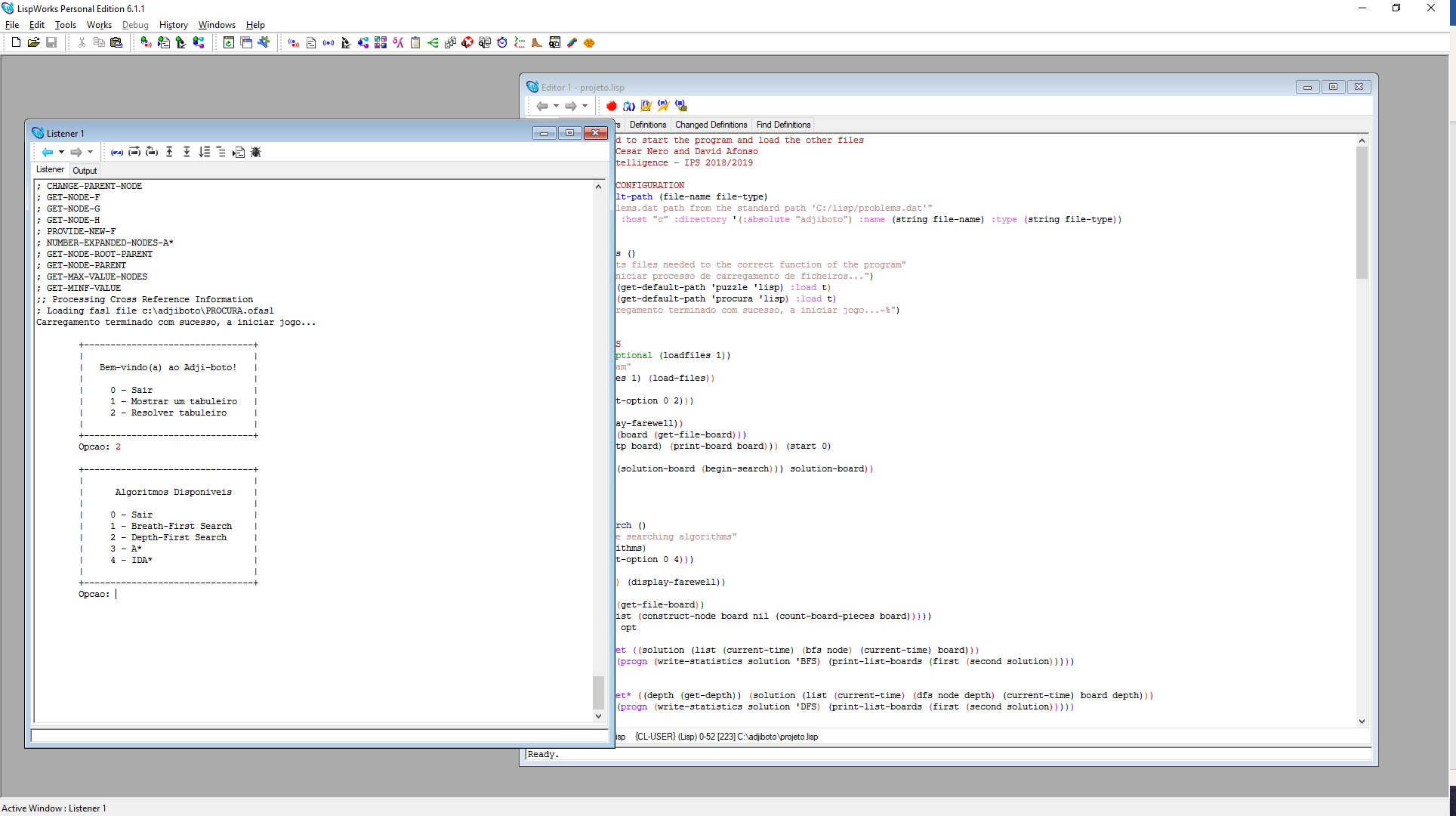
- Penetrância

- Comprimento da solução

- Fator de ramificação média

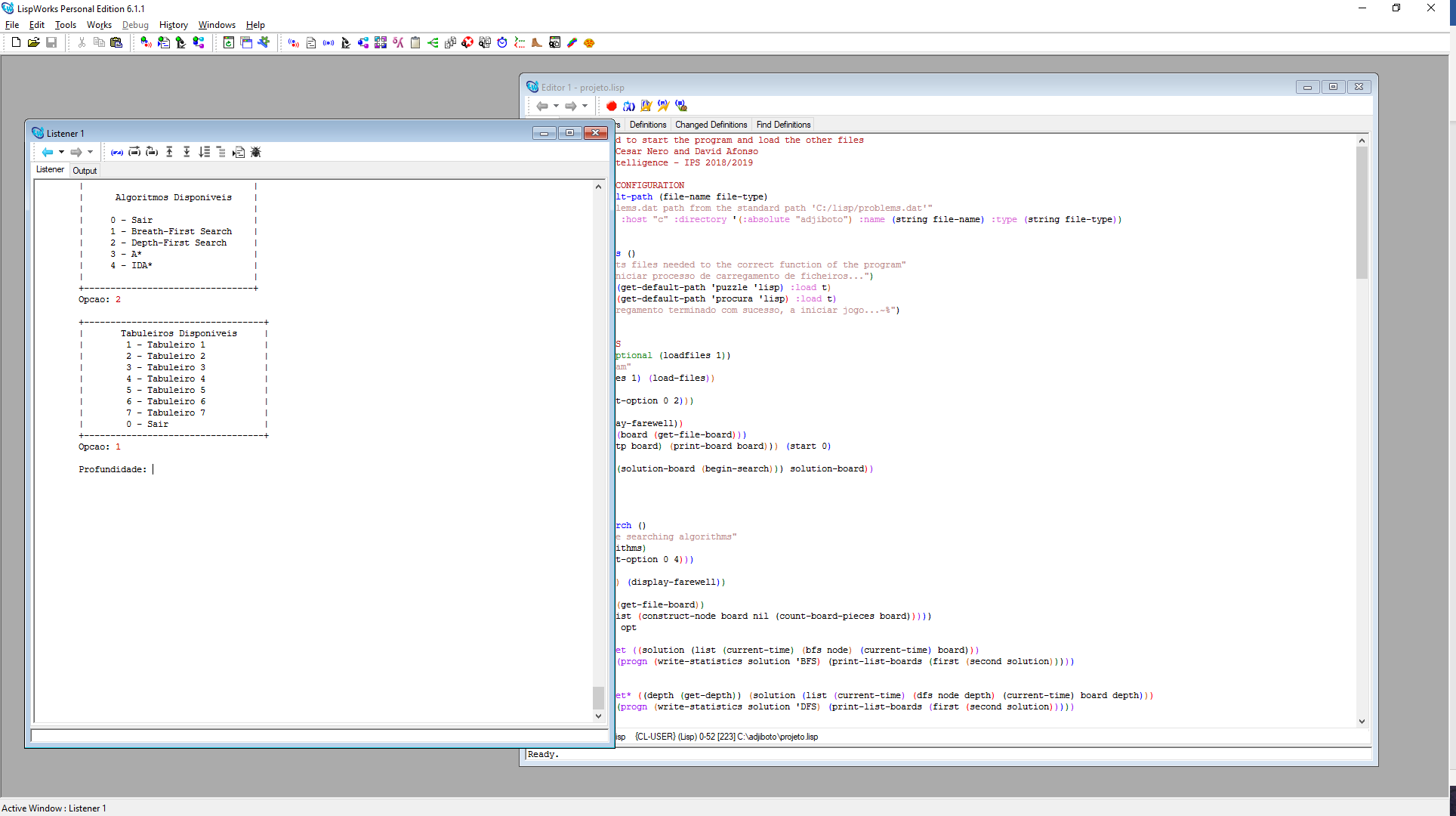
## Algoritmos de Procura

Após a escolha de um tabuleiro, é dada a opção de escolhe de um dos algoritmos disponíveis como demonstrado na imagem:



**Breath-First Search** – Algoritmo de procura em largura. Explora todo o nível da árvore antes de passar para o seguinte.

**Depth-First Search** – Algoritmo de procura em profundidade. Explora em primeiro lugar o primeiro nó até que seja atingida a profundidade máxima ou encontrada a solução. No caso se atingir a profundidade máxima desse nó passa o nó seguinte. Neste algoritmo é possível indicar a profundidade máximo após selecionar a opção.



**A\*** - Algoritmo de procura informada. Utiliza uma função heurística para calcular o respetivo valor do nó e vai sempre explorando o nó com o melhor valor heurístico de forma a garantir o melhor caminho até à solução. Neste algoritmo é possível utilizar a heurística fornecida no enunciado ou então a heurística criada.

**IDA\*** - Algoritmo de procura informada, mas este tendo memória limitada. À semelhança do A\* também utiliza uma heurística para calcular o valor do nó limiar e apenas expande quando o valor é abaixo desse limiar, no caso de não existir é definido um novo limiar com base no valor heurístico mínimo já existente.

Tanto para o **IDA\*** como o **A\*** são fornecidas duas funções heurísticas como demonstrado na figura abaixo:

