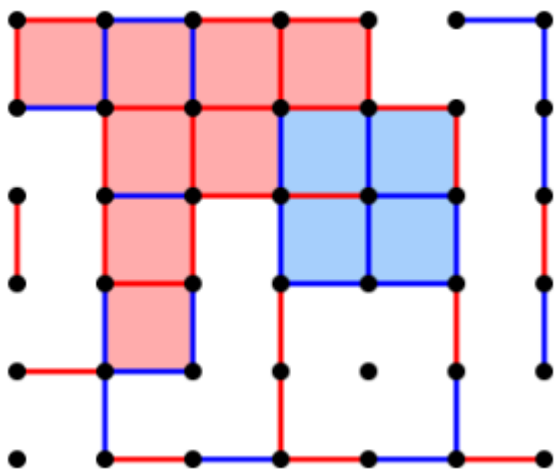


Manual de Utilizador



Diogo Dias - 202001673, Rafael Maria - 202001443

Docentes: Joaquim Filipe, Filipe Mariano

Acrónimos

- BFS - Breath first Search
- DFS - Depth first Search

Identificação do projeto

Neste Documento, iremos abordar como o utilizador deverá preparar o sistema e utilizar o software, desenvolvido para a 1ª fase do projeto de Inteligência Artificial de 2022/2023 da Licenciatura de Engenharia Informática do Instituto Politécnico de Setúbal.

Introdução

O seguinte programa foi desenvolvido no âmbito da disciplina de Inteligência Artificial.

Têm como objetivo desenvolver o jogo "Dots and Boxes", de forma a aplicarmos diversos algoritmos, tais como o DFS, BFS, A* e o IDA* para obter a resolução ideal no problema proposto.

Instalação do Ambiente

Para conseguirmos utilizar o software, devemos instalar uma IDE para permitir utilizar código Lisp, bem como mostrar a funcionalidade do Programa. Nós recomendamos a utilização da IDE LispWorks, pois é a IDE utilizada no ensino da Linguagem Lisp.

Nota: Esta instalação foi realizada num Windows 10 de 64-bit, caso seja para instalar em outras plataformas, seguir as instruções do website referido no 1º ponto da instalação.

1. Para a instalação do LispWorks devem começar pelo download do installer através do seguinte link. Escolhendo o ficheiro para realizar upload com base no seu Sistema Operativo.

Downloading LispWorks® Personal Edition

Please provide us with some information about yourself.

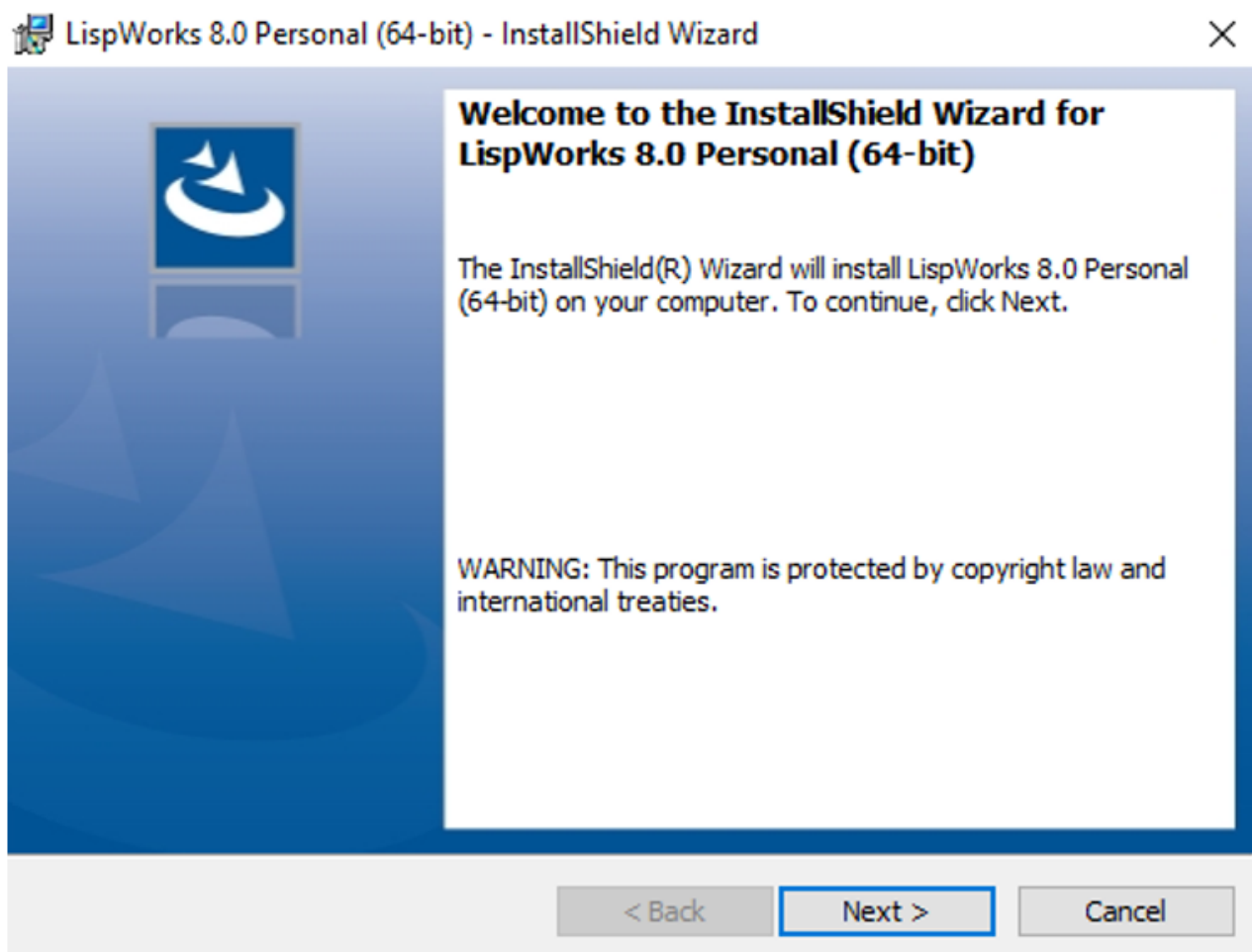
Download LispWorks for *	Please choose platform ▼
Your name *	Please choose platform
Organization	Windows (32-bit)
Email address *	Windows (64-bit)
Subscribe to the LispWorks user group ?	Macintosh (64-bit)
(You will receive a confirmation message)	x86/x86_64 Linux (32-bit)
How long is it since you first used Lisp?	x86_64 Linux (64-bit)
How did you hear about LispWorks?	ARM Linux (32-bit)
What will you use LispWorks for?	ARM Linux (64-bit)
	FreeBSD (32-bit)
	FreeBSD (64-bit)
	x86/x64 Solaris (32-bit)
	x86/x64 Solaris (64-bit)

Any other comments you would like to make?

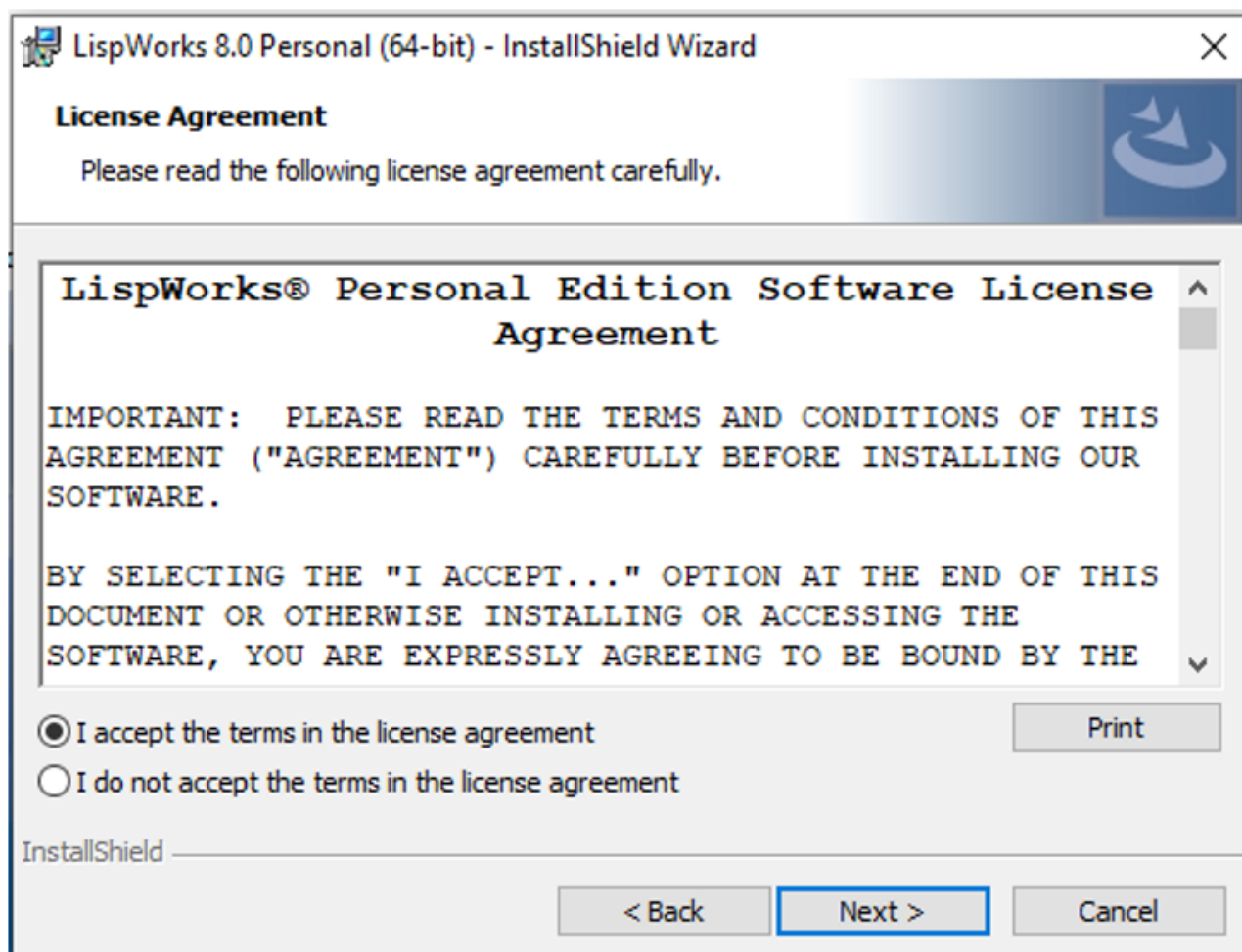
Items marked * are required.

[Proceed to Download](#)

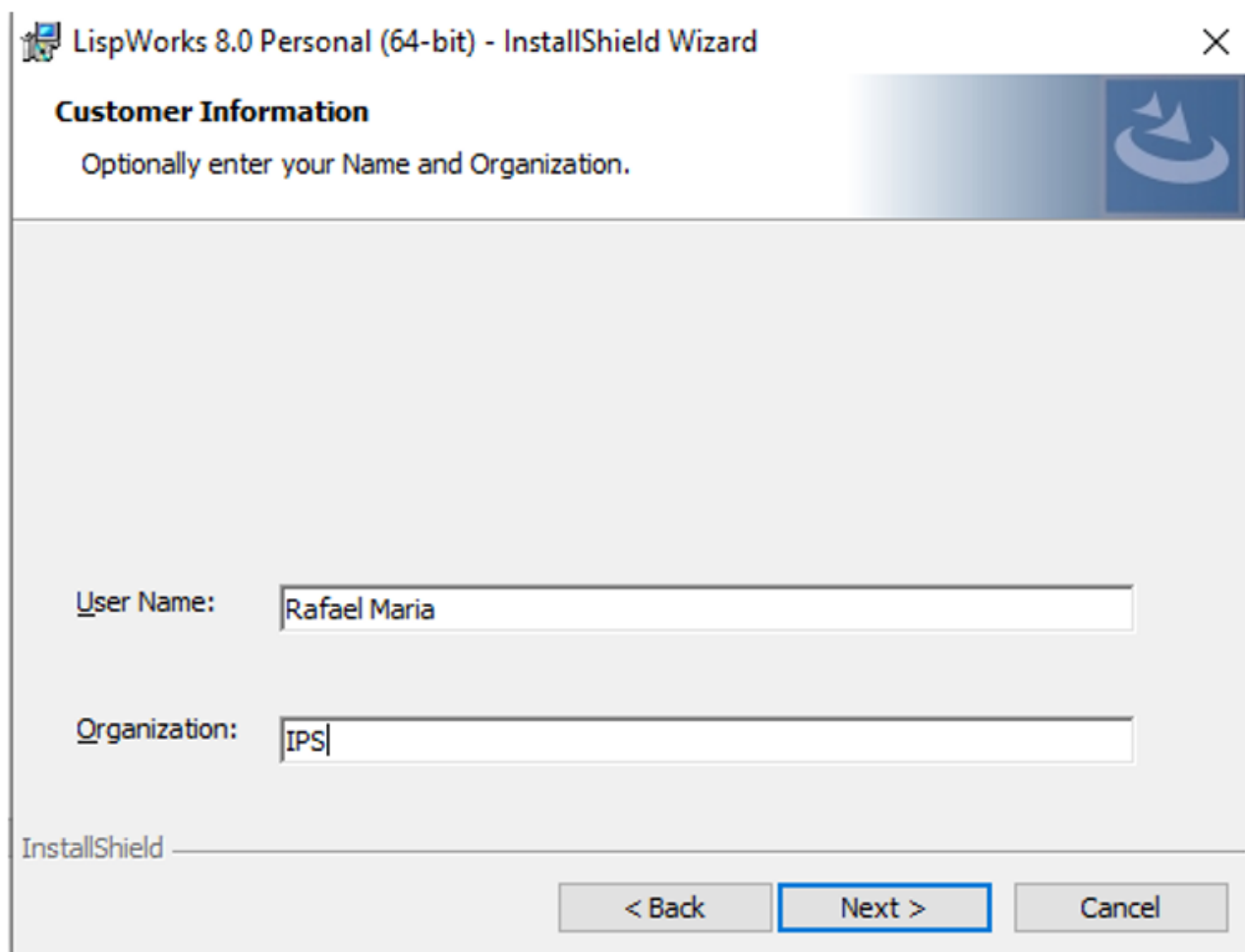
2. Após a finalização do Download, devem proceder á execução do mesmo e clicar em Next.



3. Concordar com os termos de Licença e utilização e clicar em Next.



4. Clicar em Next (Opcional) Escolher o nome de utilizador e a organização que pertence.



LispWorks 8.0 Personal (64-bit) - InstallShield Wizard

Customer Information
Optionally enter your Name and Organization.

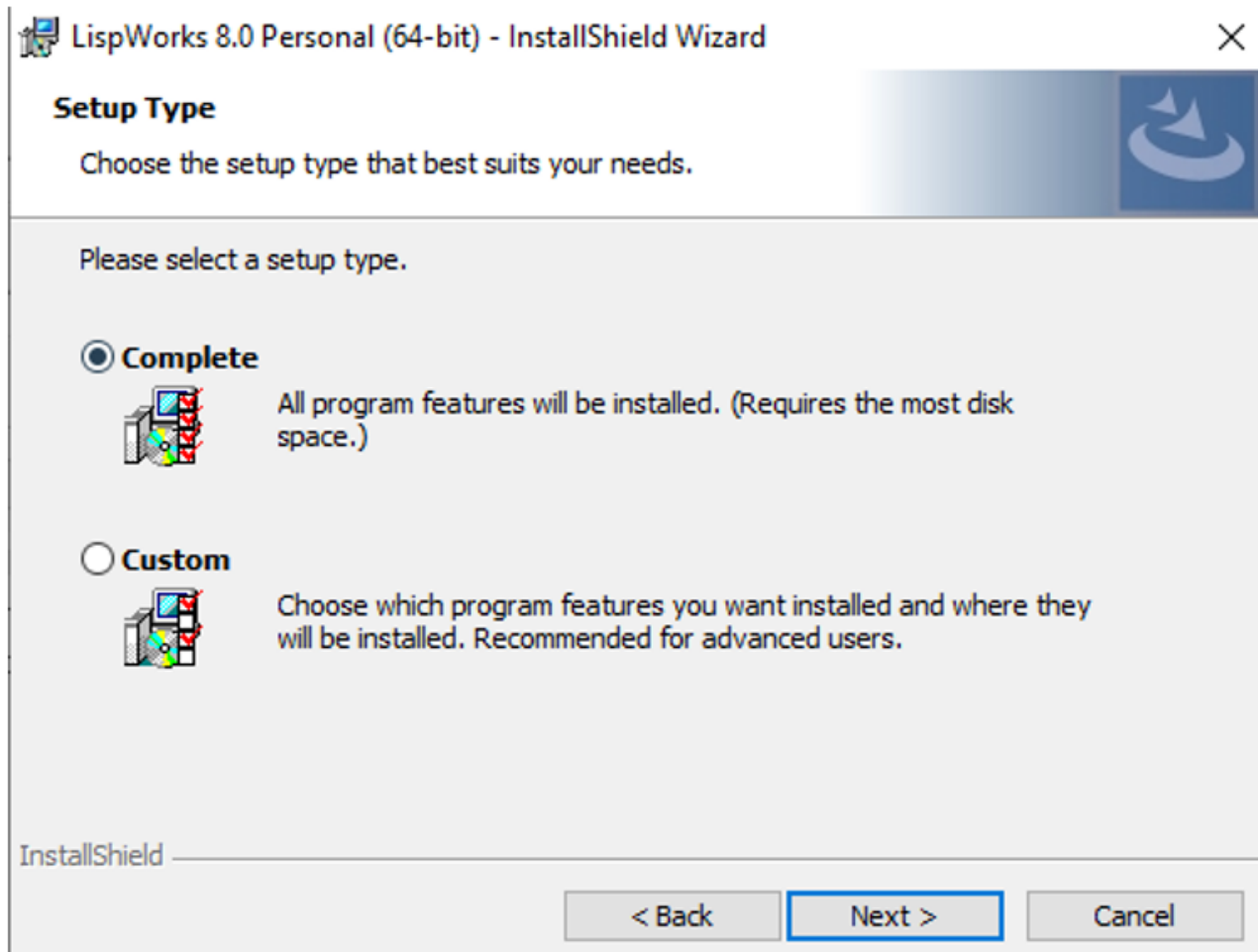
User Name:

Organization:

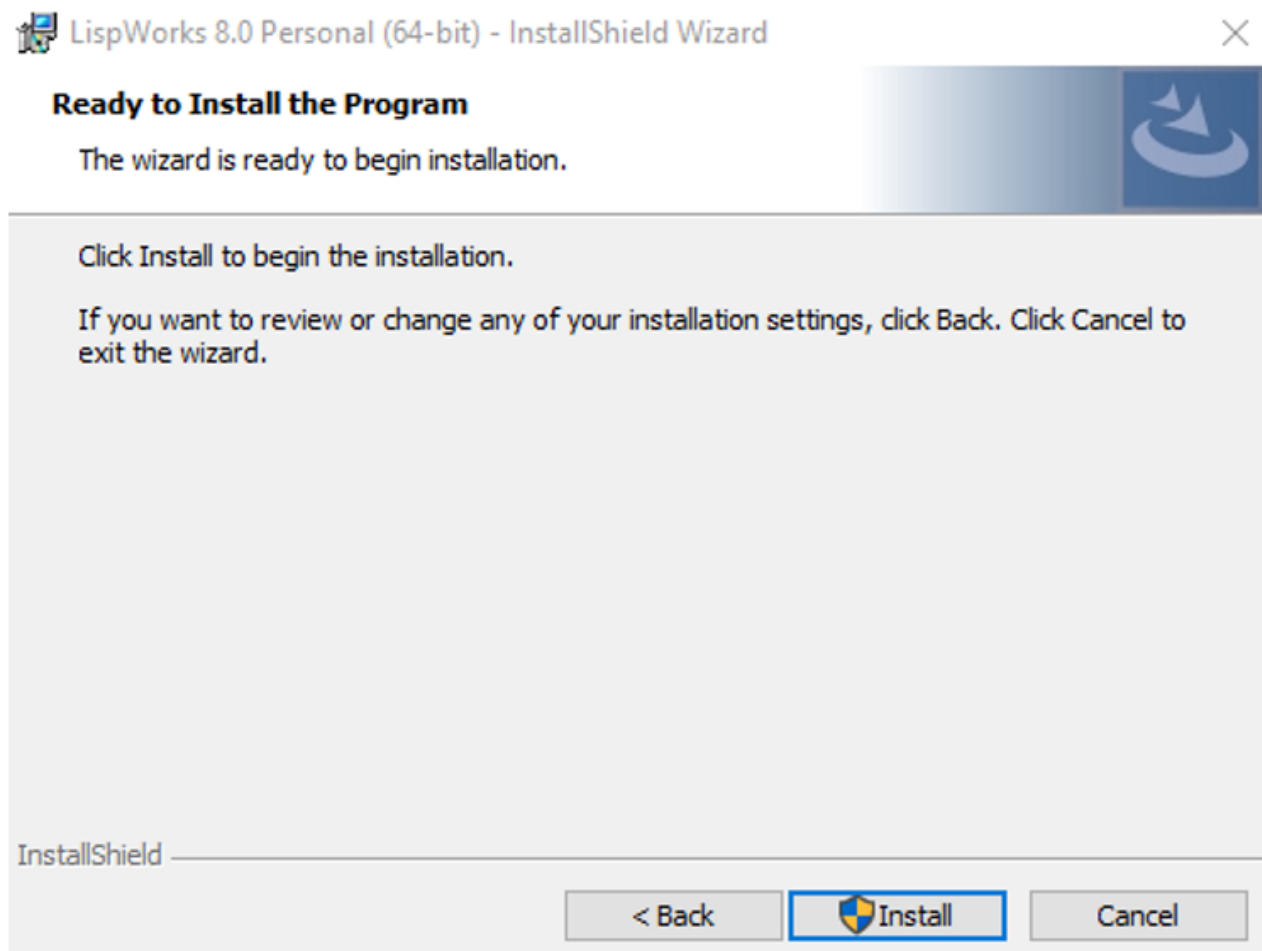
InstallShield

< Back **Next >** Cancel

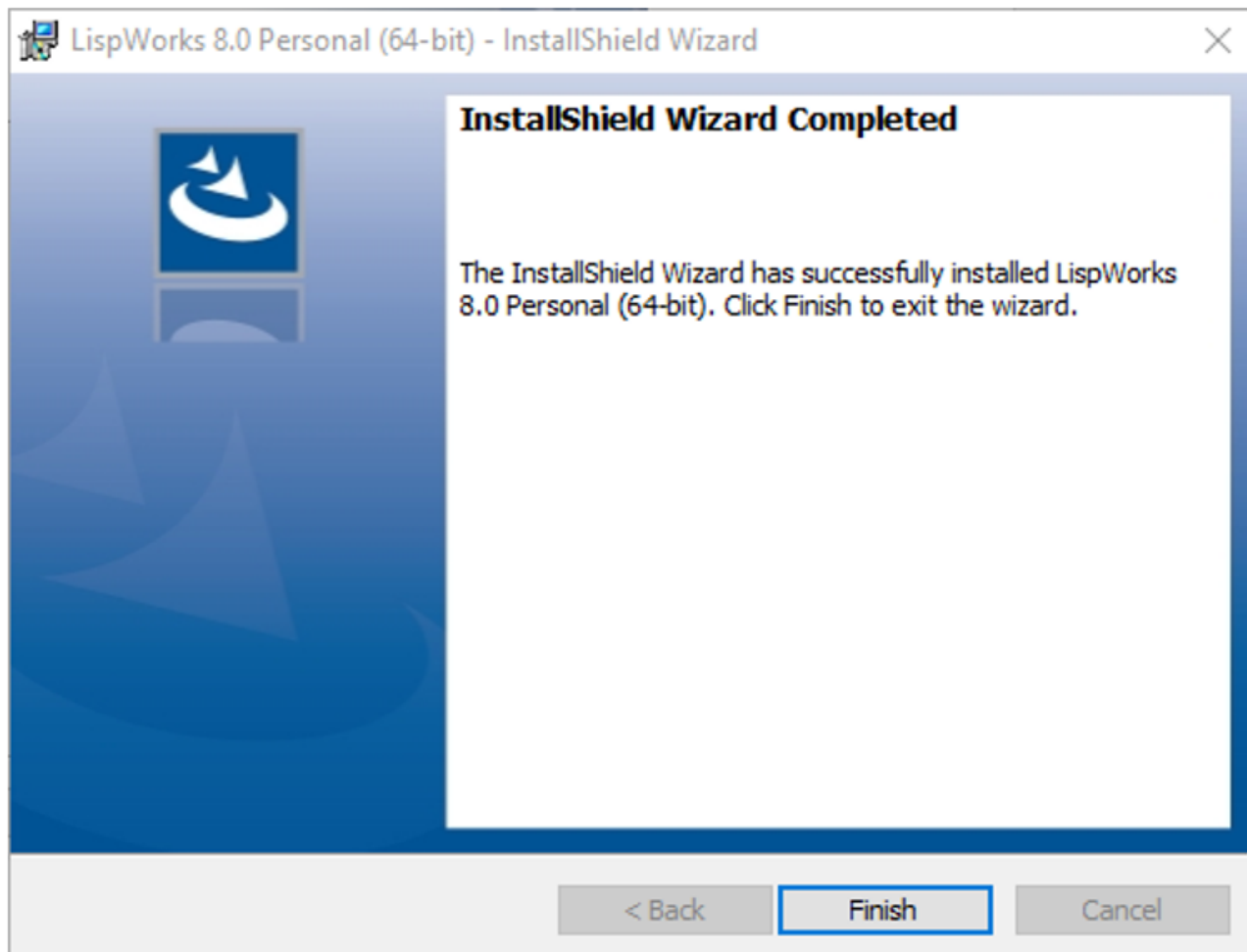
5. Clicar em Next (Opcional) Escolher o método de instalação customizável, porém não nos responsabilizamos em caso de não ser possível executar o software a partir deste método.



6. Clicar em Install. (Nota: pode ser que seja necessário dar permissões de Administrador).



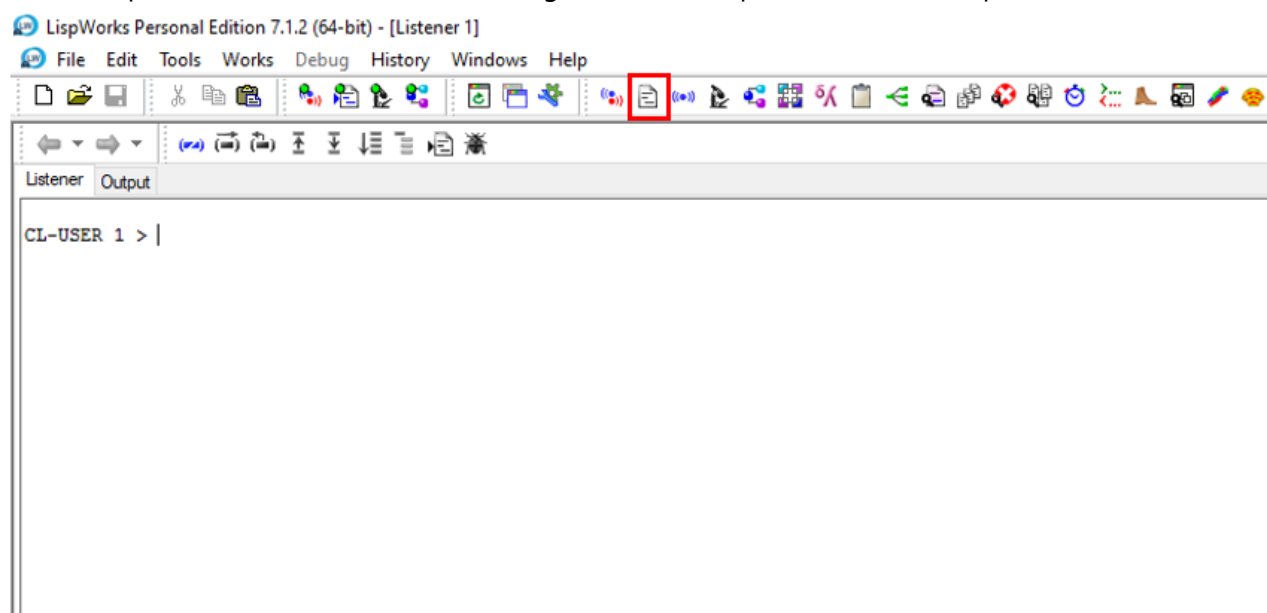
7. Clicar em Finish para terminar a instalação. (Nota: após a instalação é necessário reiniciar o dispositivo para utilizar LispWorks).



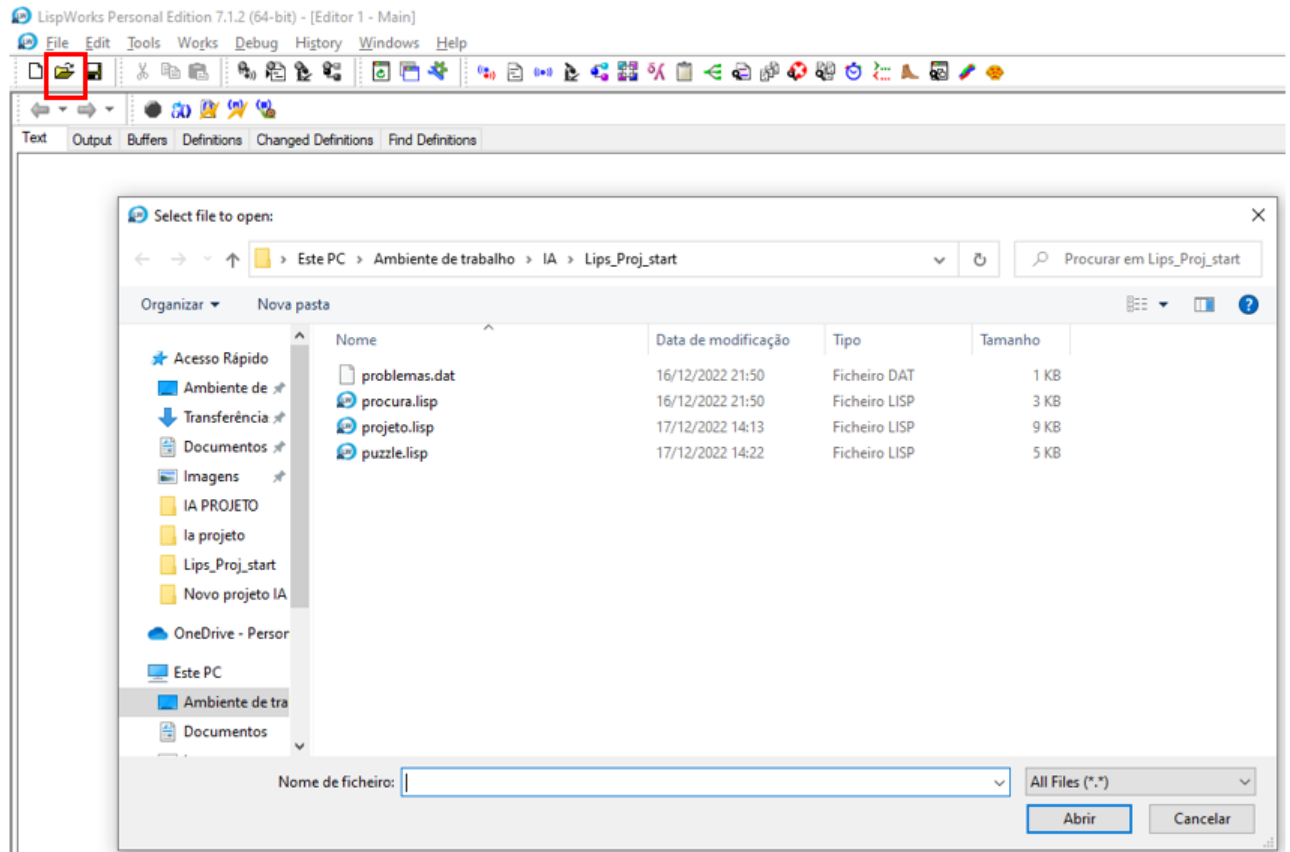
Utilização do Software

Nota: Apesar de ser permitida do LispWorks, em vários Sistemas operativos, entre Windows, Linux, Android e ios. Só foi testado em Windows, pois era a única forma de teste dos Elementos do Grupo.

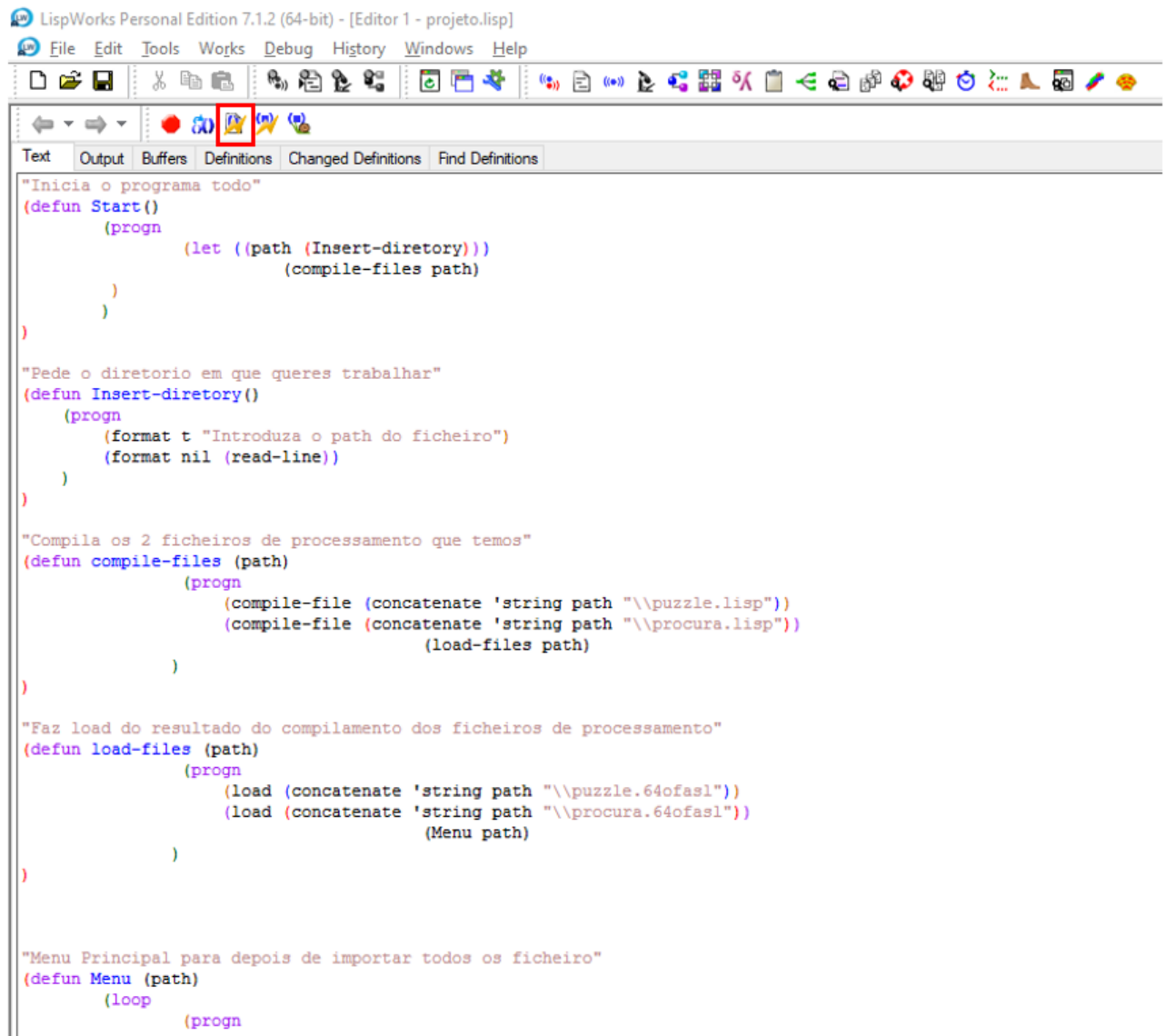
1. Abrir o LispWorks em modo de editor, carregando no icon que está dentro do quadrado.



2. Carregamos em abrir um ficheiro, carregando no icon dentro do quadrado e seleccionamos o ficheiro, "projeto.lisp".



3. Compilar o buffer de modo a executar o programa que acabamos de abrir no editor, carregar no botão dentro do quadrado para executar o compiler.



```

LispWorks Personal Edition 7.1.2 (64-bit) - [Editor 1 - projeto.lisp]
File Edit Tools Works Debug History Windows Help

"Inicia o programa todo"
(defun Start()
  (progn
    (let ((path (Insert-directory)))
      (compile-files path)
    )
  )
)

"Pede o directorio em que queres trabalhar"
(defun Insert-directory()
  (progn
    (format t "Introduza o path do ficheiro")
    (format nil (read-line))
  )
)

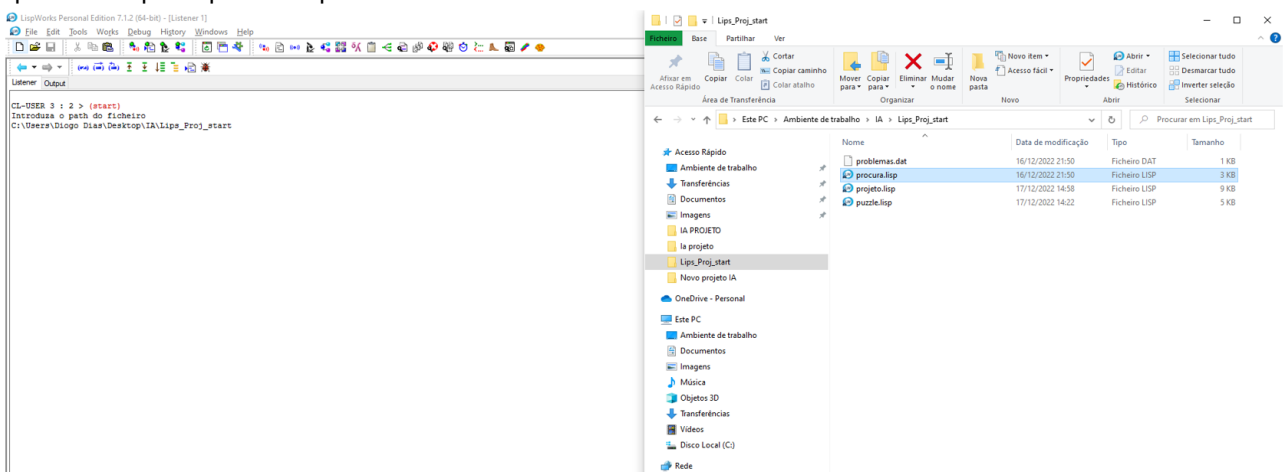
"Compila os 2 ficheiros de processamento que temos"
(defun compile-files (path)
  (progn
    (compile-file (concatenate 'string path "\\puzzle.lisp"))
    (compile-file (concatenate 'string path "\\procura.lisp"))
    (load-files path)
  )
)

"Faz load do resultado do compilamento dos ficheiros de processamento"
(defun load-files (path)
  (progn
    (load (concatenate 'string path "\\puzzle.64ofasl"))
    (load (concatenate 'string path "\\procura.64ofasl"))
    (Menu path)
  )
)

"Menu Principal para depois de importar todos os ficheiro"
(defun Menu (path)
  (loop
    (progn

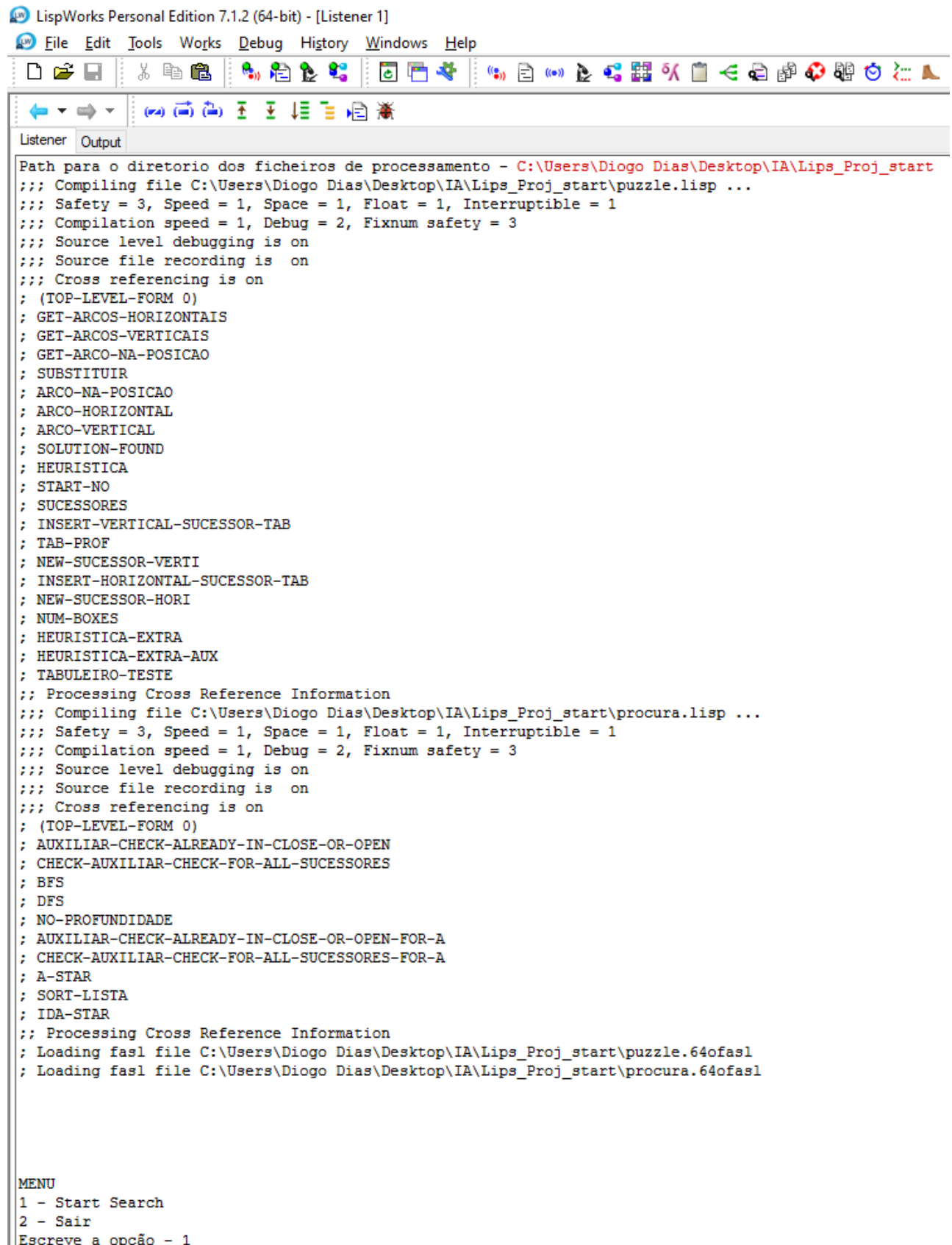
```

4. Entrar no Listener para iniciar o programa, chamando inicialmente a função "(start)", para dar início ao programa, passando depois a diretoria de onde se encontram os ficheiros de processamento, "procura.lisp" e "puzzle.lisp".



5. Após passar o path do directorio dos ficheiros de processamento, o programa vai fazer o LOAD dos ficheiros e das funções dentro delas, pedindo em seguida início do programa, basta escrever 1 ou 2. 1

inicia o programa, 2 finaliza o programa.



```

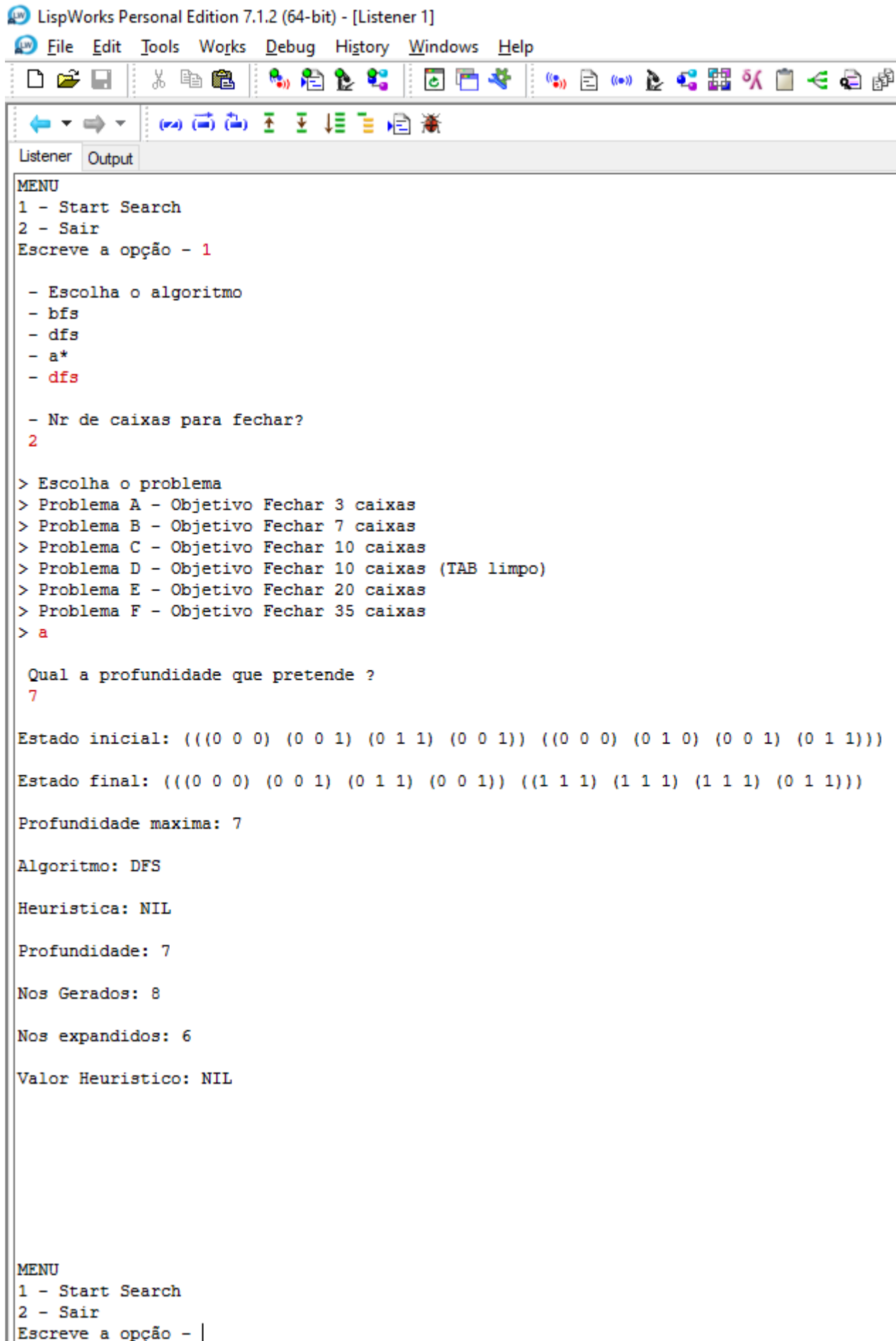
LispWorks Personal Edition 7.1.2 (64-bit) - [Listener 1]
File Edit Tools Works Debug History Windows Help

Path para o directorio dos ficheiros de processamento - C:\Users\Diogo Dias\Desktop\IA\Lips_Proj_start
;;; Compiling file C:\Users\Diogo Dias\Desktop\IA\Lips_Proj_start\puzzle.lisp ...
;;; Safety = 3, Speed = 1, Space = 1, Float = 1, Interruptible = 1
;;; Compilation speed = 1, Debug = 2, Fixnum safety = 3
;;; Source level debugging is on
;;; Source file recording is on
;;; Cross referencing is on
; (TOP-LEVEL-FORM 0)
; GET-ARCOS-HORIZONTAIS
; GET-ARCOS-VERTICAIS
; GET-ARCO-NA-POSICAO
; SUBSTITUIR
; ARCO-NA-POSICAO
; ARCO-HORIZONTAL
; ARCO-VERTICAL
; SOLUTION-FOUND
; HEURISTICA
; START-NO
; SUCESSORES
; INSERT-VERTICAL-SUCCESSOR-TAB
; TAB-PROF
; NEW-SUCCESSOR-VERTI
; INSERT-HORIZONTAL-SUCCESSOR-TAB
; NEW-SUCCESSOR-HORI
; NUM-BOXES
; HEURISTICA-EXTRA
; HEURISTICA-EXTRA-AUX
; TABULEIRO-TESTE
;; Processing Cross Reference Information
;;; Compiling file C:\Users\Diogo Dias\Desktop\IA\Lips_Proj_start\procura.lisp ...
;;; Safety = 3, Speed = 1, Space = 1, Float = 1, Interruptible = 1
;;; Compilation speed = 1, Debug = 2, Fixnum safety = 3
;;; Source level debugging is on
;;; Source file recording is on
;;; Cross referencing is on
; (TOP-LEVEL-FORM 0)
; AUXILIAR-CHECK-ALREADY-IN-CLOSE-OR-OPEN
; CHECK-AUXILIAR-CHECK-FOR-ALL-SUCCESSORES
; BFS
; DFS
; NO-PROFUNDIDADE
; AUXILIAR-CHECK-ALREADY-IN-CLOSE-OR-OPEN-FOR-A
; CHECK-AUXILIAR-CHECK-FOR-ALL-SUCCESSORES-FOR-A
; A-STAR
; SORT-LISTA
; IDA-STAR
;; Processing Cross Reference Information
; Loading fasl file C:\Users\Diogo Dias\Desktop\IA\Lips_Proj_start\puzzle.64ofasl
; Loading fasl file C:\Users\Diogo Dias\Desktop\IA\Lips_Proj_start\procura.64ofasl

MENU
1 - Start Search
2 - Sair
Escreve a opção - 1

```

6. De seguida basta inserir o algoritmo que queremos utilizar, o número de caixas que queremos fechar, o problema ou tabuleiro inicial que queremos utilizar e consoante o algoritmo, qual a profundidade, para o DFS e qual a heurística para os algoritmos de espaço de estados.



```

LispWorks Personal Edition 7.1.2 (64-bit) - [Listener 1]
File Edit Tools Works Debug History Windows Help

Listener Output
MENU
1 - Start Search
2 - Sair
Escreve a opção - 1

- Escolha o algoritmo
- bfs
- dfs
- a*
- dfs

- Nr de caixas para fechar?
2

> Escolha o problema
> Problema A - Objetivo Fechar 3 caixas
> Problema B - Objetivo Fechar 7 caixas
> Problema C - Objetivo Fechar 10 caixas
> Problema D - Objetivo Fechar 10 caixas (TAB limpo)
> Problema E - Objetivo Fechar 20 caixas
> Problema F - Objetivo Fechar 35 caixas
> a

Qual a profundidade que pretende ?
7

Estado inicial: (((0 0 0) (0 0 1) (0 1 1) (0 0 1)) ((0 0 0) (0 1 0) (0 0 1) (0 1 1)))
Estado final: (((0 0 0) (0 0 1) (0 1 1) (0 0 1)) ((1 1 1) (1 1 1) (1 1 1) (0 1 1)))

Profundidade maxima: 7

Algoritmo: DFS

Heuristica: NIL

Profundidade: 7

Nos Gerados: 8

Nos expandidos: 6

Valor Heuristico: NIL

MENU
1 - Start Search
2 - Sair
Escreve a opção - |

```

Por fim

o programa chama a sua função start recursivamente podendo então experimentar os restantes algoritmos e tabuleiros.

Input/Output

O programa recebe informação por parte de diversos ficheiros, tais como:

- Puzzle.lisp - Onde vai buscar as funções necessárias para processamento das operações.
- Procura.lisp - Onde vai buscar os diversos algoritmos implementados.
- Problemas.dat - Onde vai buscar os tabuleiros dos problemas que queremos resolver.

Por sua vez, o programa tem o output representado na figura 6, onde diz:

- Estado inicial do tabuleiro escolhido;
- Estado final do tabuleiro escolhido;
- Profundidade maxima atingida;
- A heurística utilizada;
- Todos os nós gerados;
- Todos os nós expandidos;
- O valor heurístico atingido;

Exemplo de aplicação

Os comandos necessários para utilização da aplicação são básicos, contudo sempre que existe a necessidade de preenchimento de um campo para continuar com a utilização do programa, temos um exemplo do que escrever para que o programa corra corretamente.

Questões (Pede)	Commandos	Insucesso	Sucesso
Diretoria	Diretoria dos ficheiros	Termina o programa	Continua o programa
Começar	1 ou 2	Repete Pergunta	N/A
Algoritmo	DFS/BFS/A*/IDA*	Repete Pergunta	N/A
Nr caixas a fechar	0 a 100	Repete Pergunta	N/A
Escolha do problema	A a F	Repete Pergunta	N/A
Profundidade	0 a 100	Termina o programa	Continua o programa
Heurística	1 ou 2	Repete Pergunta	N/A