

CENTRO UNIVERSITÁRIO FARIAS BRITO
CIÊNCIA DA COMPUTAÇÃO

DISCIPLINA: INTELIGÊNCIA ARTIFICIAL
PROFESSOR: JOSÉ BELO ARAGÃO JÚNIOR

RELATÓRIO: LÓGICA FUZZY

ALUNOS: RENÊ VICTOR LUCAS 1611181
PEDRO VICTOR DA SILVA ÁVILA 1620237
DOUGLAS MENDES MARTINS 1620241

FORTALEZA – CE
MARÇO DE 2018

1. INTRODUÇÃO

Lógica fuzzy é uma forma de lógica multivalorada onde os valores verdadeiros de uma variável podem ser qualquer número entre 1 e 0, incluindo os próprios. É utilizada para explorar o conceito de verdade parcial, onde o valor verdadeiro pode existir em um intervalo entre completamente verdadeiro e completamente falso. É diferente da lógica booleana, onde os valores podem ser apenas inteiros e absolutos 1 ou 0.

O termo “lógica fuzzy”, também chamado alternativamente de lógica difusa, foi proposto por Lotfi Zadeh nos anos 60, embora já tenha sido estudada desde os anos 20 como parte da lógica de Lukasiewicz. A lógica fuzzy é baseada na premissa de que as pessoas fazem decisões embasadas em informações imprecisas e não-numéricas; assim, utiliza modelos matemáticos para representar dados e informações vagas. Estes modelos são capazes de reconhecer e manipular dados que têm um certo grau de incerteza. Dessa forma, a lógica fuzzy tem sido aplicada em inúmeros ramos do conhecimento e da tecnologia, dentre eles a inteligência artificial.

2. DESCRIÇÃO DO PROBLEMA

É sabido que, no Brasil, jogos eletrônicos têm preços desproporcionalmente altos em relação a outros países do mundo. Através de tributos e taxas em cima dos produtos em questão, os consumidores brasileiros encontram-se cada vez mais limitados em seu poder de compra, tendo muitas vezes que confiar em sites de análises e similares para saber se um jogo novo que está prestes a sair seria um jogo que o consumidor iria se interessar a ponto de investir seu dinheiro.

Entretanto, sites de análises podem muitas vezes ser imprecisos e até mesmo tendenciosos, levando em conta que, por mais imparcial que possam tentar ser, os jornalistas especializados em jogos eletrônicos por vezes não evitam de deixar um pouco de suas opiniões e gostos pessoais influenciarem na crítica de um ou outro jogo em questão. Desta forma, faz-se necessária a proposta de um novo método para auxiliar o consumidor, em conjunto com os demais, a decidir se um produto vale ou não o investimento de seu capital.

O problema da satisfação em comprar determinado jogo consiste em avaliar os meios e regras adequadas para conceder determinado valor ao jogo em questão. Este seria o “grau de satisfação” que se poderá ter, segundo alguns de nossos critérios, ao comprar determinado jogo. O grau de satisfação poderá ser usado para que o usuário tenha uma noção se vale a pena ou não comprar determinado jogo.

3. ESPECIFICAÇÕES DO ALGORITMO FUZZY

1. Entrada do sistema:

- Preço: Seria o preço do jogo na loja;
- Crítica: Seria uma pontuação, ou nota que vai determinar o quão bom é aquele jogo segundo olhar da crítica. Neste trabalho foi usado o sistema de pontuação, ou scores que pode ser encontrado na IGN, que é um sistema que vai 1 a 100;
- Tempo: Seria o gameplay, ou tempo médio que leva para o jogo ser zerado;

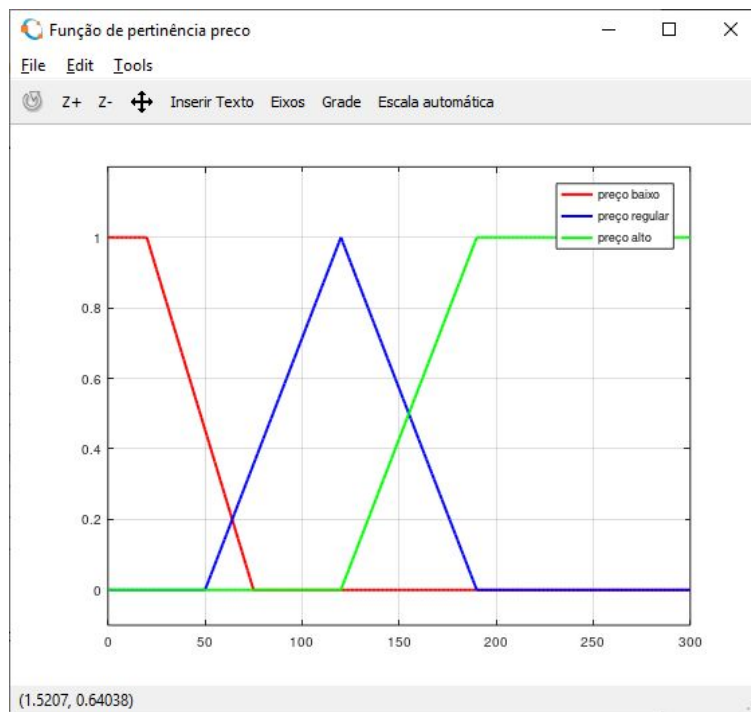
2. Saída do sistema:

- Satisfação (grau de satisfação): determina numa escala de 1 a 100 o grau de satisfação que se terá ao comprar determinado jogo segundo alguns critérios.

3.1 Função de pertinência

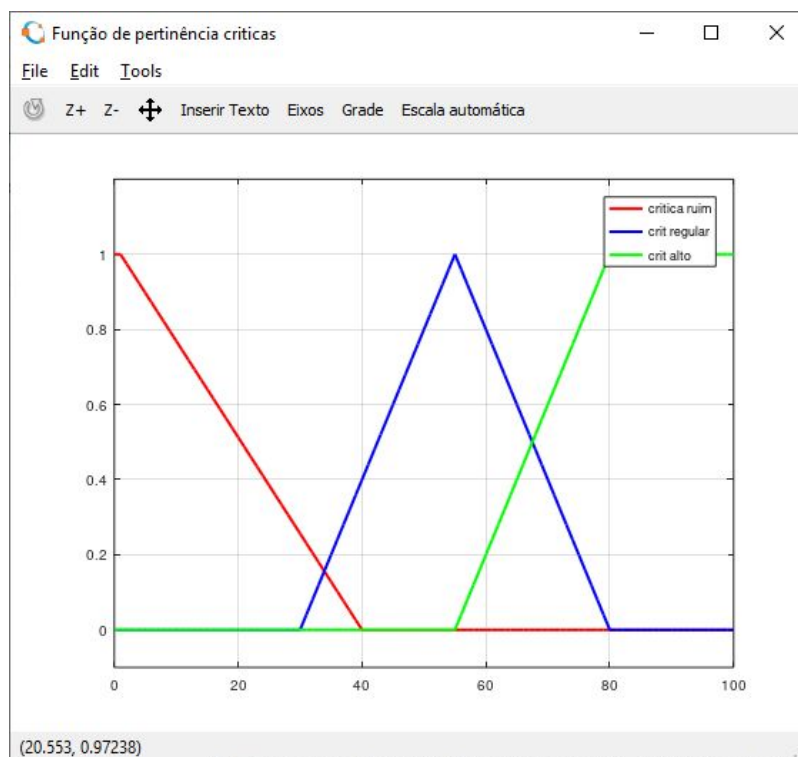
3.1.1 Função de pertinência para variável linguística preço

A variável linguística preço apresenta três conjuntos:baixo(varia de 20 a 75 reais), regular(varia de 50 a 190 reais), alto(varia de 120 a 300 reais)



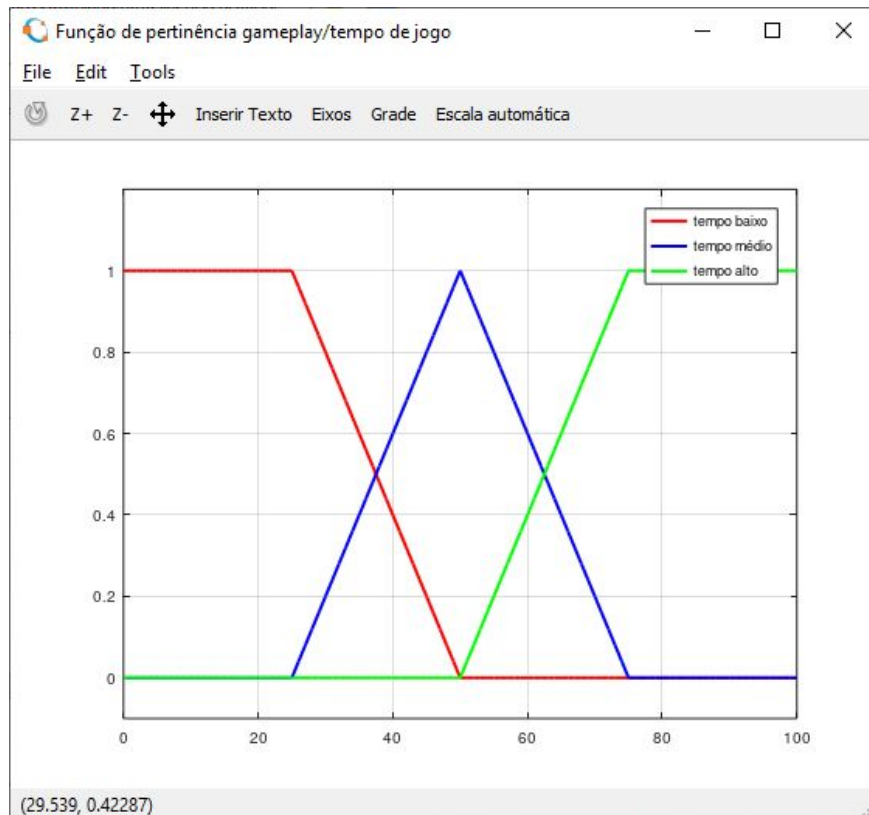
3.1.2 Função de pertinência para variável linguística crítica

A variável linguística crítica apresenta três conjuntos: ruim (varia de 1 a 40 pontos), regular (varia de 30 a 80 pontos), alto (varia de 55 a 100 pontos)



3.1.3 Função de pertinência para variável linguística tempo

A variável linguística tempo apresenta três conjuntos: baixo(varia de 25 a 50 horas), médio(varia de 25 a 75 horas), alto(varia de 50 a 100horas)



3.2 Conjunto de regras:

O conjunto de regras utilizado na base de regras que compõe o sistema fuzzy é apresentado abaixo para este caso particular da satisfação da compra do jogo.

1. Se tempo de jogo baixo e crítica do jogo ruim então satisfação igual a baixo
2. Se tempo de jogo médio e crítica do jogo regular então satisfação igual a médio
3. Se tempo de jogo alto e crítica de jogo alto então satisfação igual a alto
4. Se crítica do jogo alto e preço do jogo baixo então satisfação igual a alto
5. Se crítica do jogo regular e preço do jogo regular então satisfação igual médio
6. Se crítica do jogo ruim e preço do jogo alto então satisfação igual baixa
7. Se tempo do jogo alto e preço do jogo baixo então satisfação igual a alto
8. Se tempo de jogo médio e preço do jogo regular então satisfação igual a médio

9. Se tempo do jogo baixo e preço do jogo alto então satisfação igual a baixo

3.3 Métodos aplicados para desfuzificação

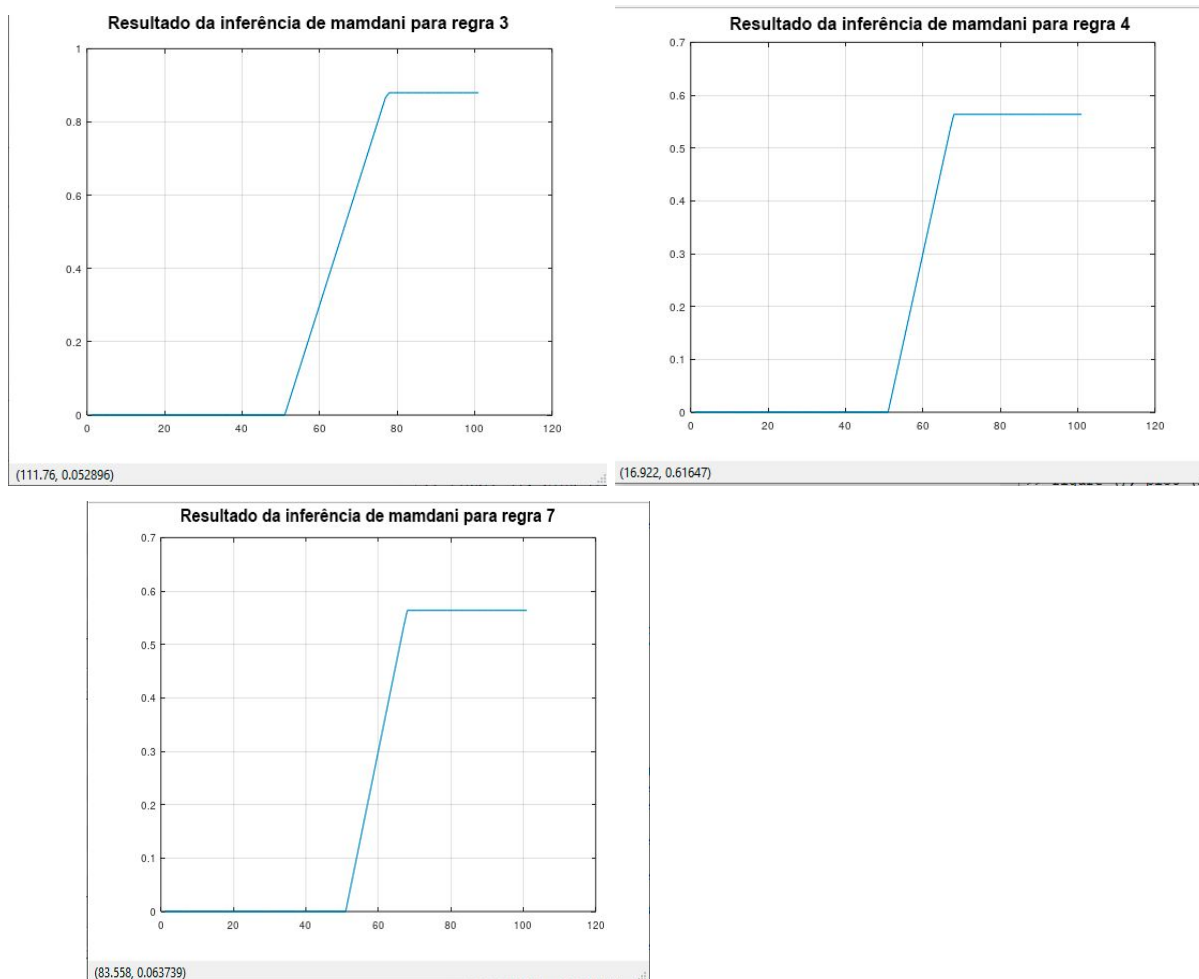
Neste foi aplicado o método de inferência de Mamdani, o resultado apresentado através do cálculo do centro de massa da figura obtida.

3.4 Testes realizados com o algoritmo fuzzy

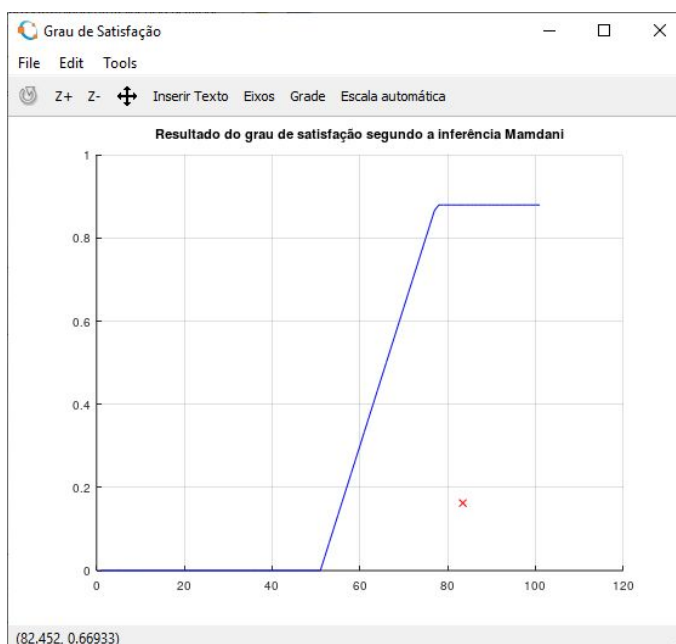
A Tabela abaixo mostra os resultados obtidos em diferentes situações onde é necessário verificar a satisfação em se comprar certos jogos:

testes	valor preço	valor crítica	valor tempo de jogo	Grau de satisfação(inferência de mamdani) centro de massa
1	45	78	80	83.46
2	90	35	67	50
3	190	75	60	69.59
4	165	35	1	15.57
5	201	3	35	7.23

3.4.1 Gráficos



Teste 1 (Inferência de Mamdani para as regras 3, 4 e 7).



Teste 1 (Resultado da Inferência de Mamdani para as regras 3, 4 e 7).

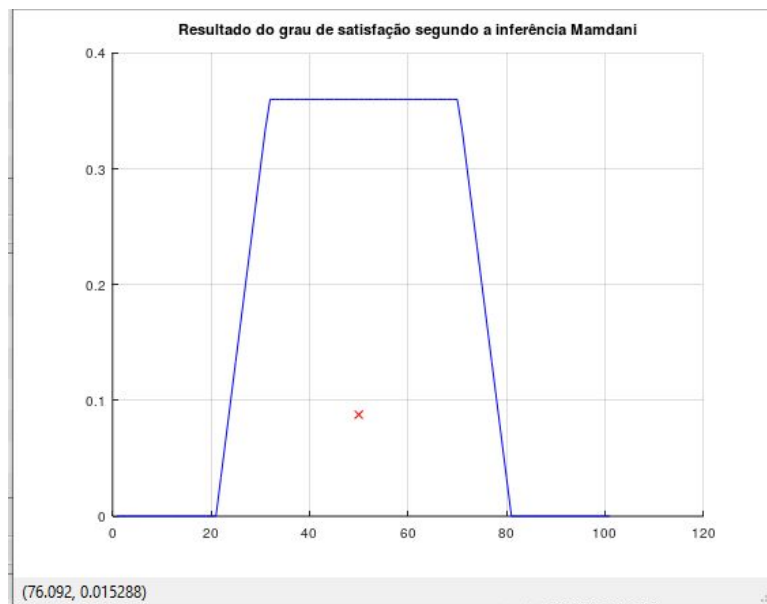


(6.4147, 0.33466)



(38.488, 0.18494)

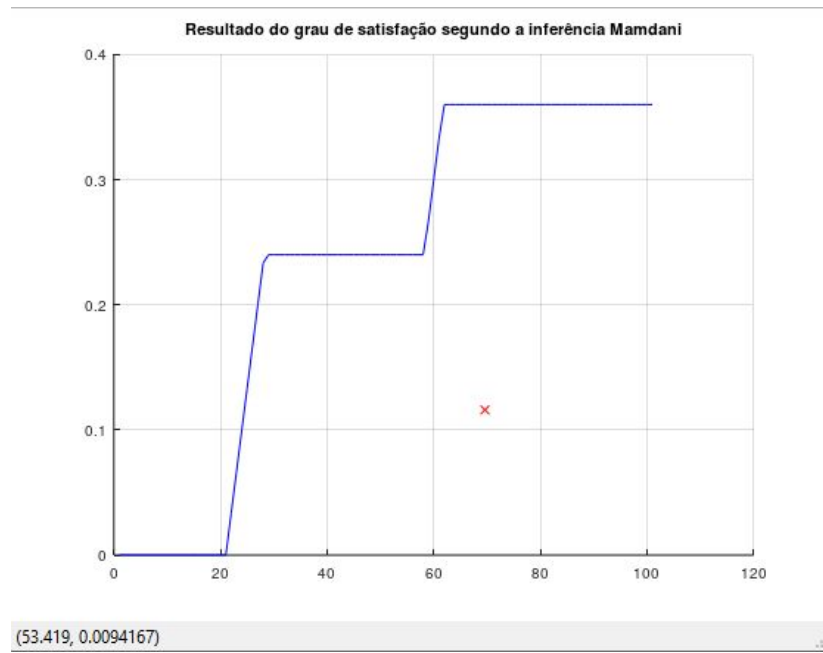
Teste 2 (Inferência de Mamdani para as regras 2, 5 e 8).



Teste 2 (Resultado da Inferência de Mamdani para as regras 2, 5 e 8).



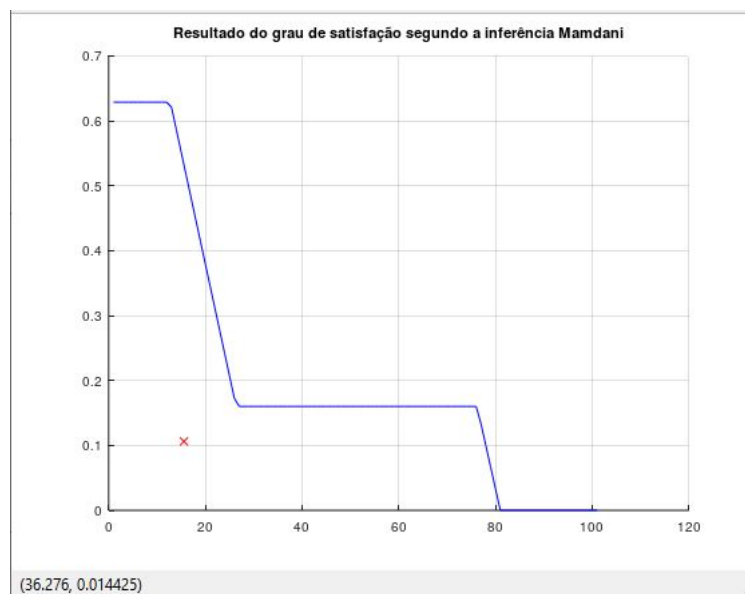
Teste 3 (Inferência de Mamdani para as regras 2, 3, 5 e 8).



Teste 3 (Resultado da Inferência de Mamdani para as regras 2, 3, 5 e 8).



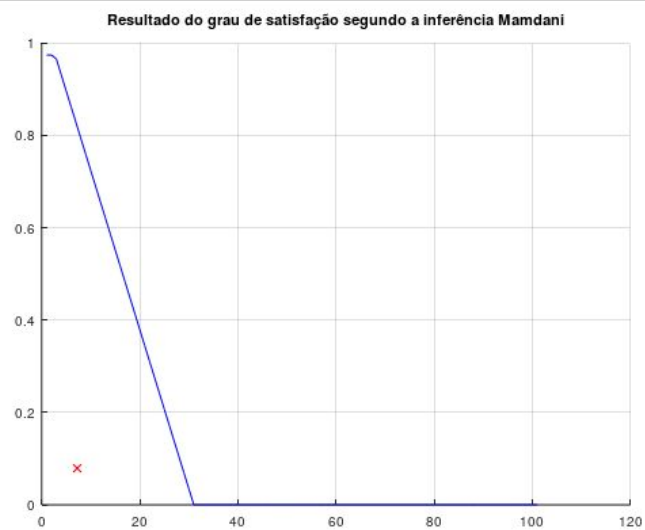
Teste 4 (Inferência de Mamdani para as regras 1, 5 , 6 e 9).



Teste 4 (Resultado da Inferência de Mamdani para as regras 1, 5 , 6 e 9).



Teste 5 (Inferência de Mamdani para as regras 1, 6 e 9).



Teste 5 (Resultado da Inferência de Mamdani para as regras 1, 6 e 9).

4. IMPLEMENTAÇÃO DO ALGORITMO FUZZY

%Programa de avaliação de satisfação ao comprar determinado jogo%

%mostrar na tela a mensagem que pede preço e logo abaixo a variavel que guarda o preço do jogo%

disp("Entre com o preço do jogo");

preco=input("Preço do jogo:");

%O preço do jogo vai ser de 1 a 300 reais%

%mostrar na tela a mensagem que pede a nota crítica ou pontuação do jogo segundo críticas e logo abaixo a variavel que vai guardar esse valor%

disp("Entre com a nota da crítica/score do jogo");

crit=input("Nota crítica do jogo:");

%a variavel de crítica vai de 1 a 100%

%%mostrar na tela a mensagem que pede o tempo de jogo/gameplay/tempo para terminar o jogo e logo a baixo a variavel que guarda esse valor%

disp("Entre com o gameplay/ tempo para terminar o jogo");

tempo=input("tempo do jogo:");

%tempo vai de 1 a 100 horas%

% função de pertinência preco%

x = 0:1:300;

preco_baixo=trapmf(x,[-1,0,20,75]);

preco_regular=trimf(x,[50,120,190]);

preco_alto=trapmf(x,[120,190,300,301]);

%função pertinencia pontuação/notas/crit%

x1=0:1:100;

crit_ruim=trapmf(x1,[-1,0,1,40]);

```
crit_regular=trimf(x1,[30,55,80]);  
crit_alto=trapmf(x1,[55,80,100,101]);
```

```
%função de pertinencia horas de jogo/gameplay/tempo para concluir o jogo%  
x2=0:1:100;  
tempo_baixo=trapmf(x2,[-1,0,25,50]);  
tempo_medio=trimf(x2,[25,50,75]);  
tempo_alto=trapmf(x2,[50,75,100,101]);
```

```
%grau de pertinencia da satisfação ao comprar o jogo/ou se vale a pena comprar%
```

```
x3=0:1:100;
```

```
sat_baixo=trapmf(x3,[-1,0,1,30]);  
sat_medio=trimf(x3,[20,50,80]);  
sat_alto=trapmf(x3,[50,80,100,101]);
```

```
%Inferência de Mamdani segundo o conjunto de Regras%
```

```
%R1: se tempo de jogo baixo e crítica do jogo ruim então satisfação igual a baixo%  
r1=min(tempo_baixo(tempo),crit_ruim(crit));  
regra1=min(r1,sat_baixo);
```

```
%R2: se tempo de jogo medio e crítica do jogo regular então satisfação igual a medio%  
r2=min(tempo_medio(tempo),crit_regular(crit));  
regra2=min(r2,sat_medio);
```

```
%R3: se tempo de jogo alto e crítica de jogo alto então satisfação igual a alto%  
r3=min(tempo_alto(tempo),crit_alto(crit));  
regra3=min(r3,sat_alto);
```

```
%R4: se crítica do jogo alto e preço do jogo baixo então satisfação igual a alto%
```

```
r4=min(crit_alto(crit),preco_baixo(preco));  
regra4=min(r4,sat_alto);
```

%R5: se crítica do jogo regular e preço do jogo regular então satisfação igual medio%

```
r5=min(crit_regular(crit),preco_regular(preco));  
regra5=min(r5,sat_medio);
```

%R6: se crítica do jogo ruim e preço do jogo alto então satisfação igual baixa%

```
r6=min(crit_ruim(crit),preco_alto(preco));  
regra6=min(r6,sat_baixo);
```

%R7: se tempo do jogo alto e preço do jogo baixo então satisfação igual a alto%

```
r7=min(tempo_alto(tempo),preco_baixo(preco));  
regra7=min(r7,sat_alto);
```

%R8: se tempo de jogo medio e preço do jogo regular então satisfação igual a medio%

```
r8=min(tempo_medio(tempo),preco_regular(preco));  
regra8=min(r8, sat_medio);
```

%R9: se tempo do jogo baixo e preço do jogo alto então satisfação igual a baixo%

```
r9=min(tempo_baixo(tempo),preco_alto(preco));  
regra9=min(r9,sat_baixo);
```

```
satisf_alt=max(regra3, regra4);  
satisf_alt1=max(satisf_alt,regra7);
```

```
satisf_med=max(regra2,regra5);  
satisf_med1=max(satisf_med,regra8);
```

```
satisf_baix=max(regra1,regra6);  
satisf_baix1=max(satisf_baix,regra9);
```

```
Y1=max(satisf_alt1,satisf_med1);  
ytotat=max(Y1, satisf_baix1);
```

```
figure('NumberTitle', 'off', 'Name', 'Grau de Satisfação');  
grid on;
```



```

hold on;
plot(ytotal,'b');
title('Resultado do grau de satisfação segundo a inferência Mamdani');

%desfuzzificação segundo o método do centro de massa%

%valor que será usado para o x%
resultadoMamdani = (ytotal.*x3)/ytotal;

%valor que será usado para encontrar o y%
Mamd = ytotal/2;
hold on;
plot(resultadoMamdani,mean(Mamd),'rx:');

%fim da desfuzzificação segundo o método do centro de massa%

%Resultado da inferencia de mamdani que será usado para determinar o grau de
satisfação%
fprintf('\n\n RESULTADOS \n\n');
disp('Grau de satisfação segundo a Inferencia de Mamdani');
fprintf('%g',resultadoMamdani);

%Condições que geram sugestões para o usuário%

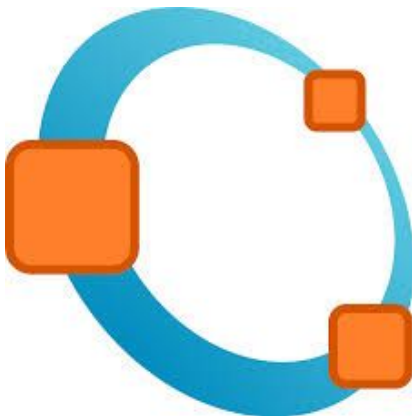
disp("\n Sugestão do sistema para o usuario");

if (resultadoMamdani<60)
    disp("\n\n Não vale muito a pena comprar o jogo");
elseif(resultadoMamdani<75)
    disp("\n\n Talvez valha a pena jogar o jogo");
else
    disp("\n\n Vale muito a pena jogar o jogo");
end

```

5. Instalação e execução da aplicação

Para essa etapa de instalação e execução será apresentado o programa utilizado para o desenvolvimento do trabalho, nesse caso, o Octave.



GNU Octave é uma linguagem computacional, desenvolvida para computação matemática.

Possui uma interface em linha de comando para a solução de problemas numéricos, lineares e não-lineares, também é usada em experimentos numéricos.

Antes de irmos para a etapa de execução do programa e testar a aplicação desenvolvida nele, precisamos primeiro instalar o Octave. Para isso, abra seu navegador e digite Octave na barra de busca dele.

Crunchyroll - Assista a Naruto Sh... x | www.etufor.ce.gov.br/consultas... x | ETUFOR - Ficha do Aluno x | octave - Pesquisa Google x +

https://www.google.com/search?q=octave&rlz=1C1EJFC_enBR827BR827&oq=octave&aqs=chrome..69i57j69i60j30l2.7247j0j7&sourceid=chrome&si...

Facebook | P! Punch! Fansubs | Crunchyroll - Assist... | MANGA Plus by SH...

Google octave

Todas Vídeos Imagens Notícias Shopping Mais Configurações Ferramentas

Aproximadamente 36.700.000 resultados (0,35 segundos)

GNU Octave
<https://www.gnu.org/software/octave/> Traduzir esta página
 The Octave syntax is largely compatible with Matlab. The Octave interpreter can be run in GUI mode, as a console, or invoked as part of a shell script.

Download
 GNU Octave is a programming language for scientific computing.

GNU Octave 4.4.1 Released
 Octave Version 4.4.1 has been released and is now available ...

Mais resultados de gnu.org »

Index of /gnu/octave/windows
 Index of /gnu/octave/windows. [ICO].
 Name · Last modified · Size ...

About
 Octave was originally conceived (in about 1988) to be ...

Octave Download - Baixaki
<https://www.baixaki.com.br/Programação/Ferramentas/Desenvolvimento>
 Octave é um programa desenvolvido por John W. Eaton. Acesse e veja mais informações, além de fazer o download e instalar o Octave.

Introdução ao Octave - UFT
www.uft.edu.br/enqambiental/prof/catalunha/arquivos/octave/octave_Souza.pdf

GNU Octave
 Linguagem de programação

GNU Octave é uma linguagem computacional, desenvolvida para computação matemática. Possui uma interface em linha de comando para a solução de problemas numéricos, lineares e não-lineares, também é usada em experimentos numéricos. Faz parte do projeto GNU, é um software livre sob os termos da licença GPL. Wikipédia

Licença: GNU General Public License
Escrito em: C, C++, Fortran

Itens também pesquisados Ver mais 15

Depois de feito a busca, entre no site de nome “Gnuu Octave” que é mostrado na imagem acima.

Crunchyroll - Assista a Naruto Sh... x | www.etufor.ce.gov.br/consultas... x | ETUFOR - Ficha do Aluno x | GNU Octave x +

https://www.gnu.org/software/octave/#install

Facebook | P! Punch! Fansubs | Crunchyroll - Assist... | MANGA Plus by SH...

GNU Octave About Donate Download Get Involved News Support/Help Docs

GNU Octave
 Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Windows
- Drop-in compatible with many Matlab scripts

Download **Docs**

Syntax Examples
 The Octave syntax is largely compatible with Matlab. The Octave interpreter can be run in GUI mode, as a console, or invoked as part of a shell script. More Octave examples can be found in the wiki.

`b = [4; 9; 2] # Column vector`

Logo dentro, no site, procure a opção de download que é mostrada na imagem acima e em seguida, clique em download. Depois disso você será direcionado para a parte do site onde você fará download.

Install

Source

GNU/Linux

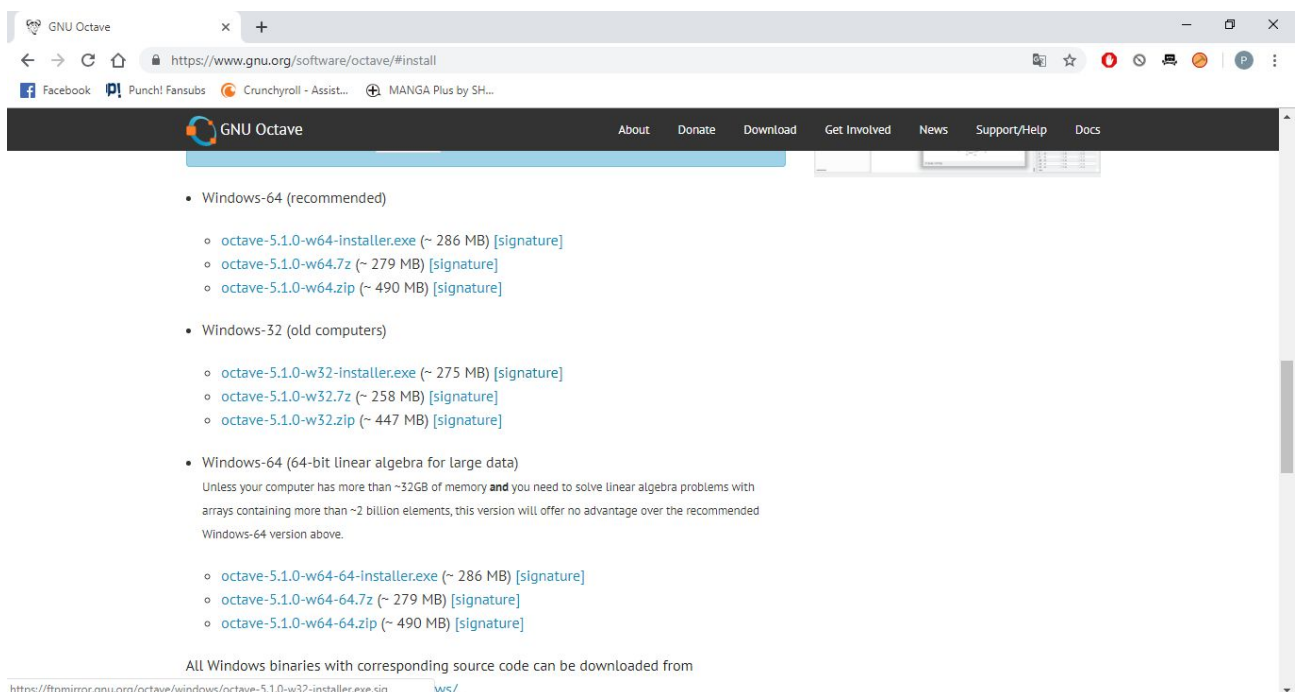
macOS

BSD

Windows

Note: All installers below bundle several **Octave Forge packages** so they don't have to be installed separately. After installation type `pkg list` to list them. [Read more.](#)

Escolha seu sistema operacional. No caso do desenvolvimento trabalho, ele foi feito no windows.

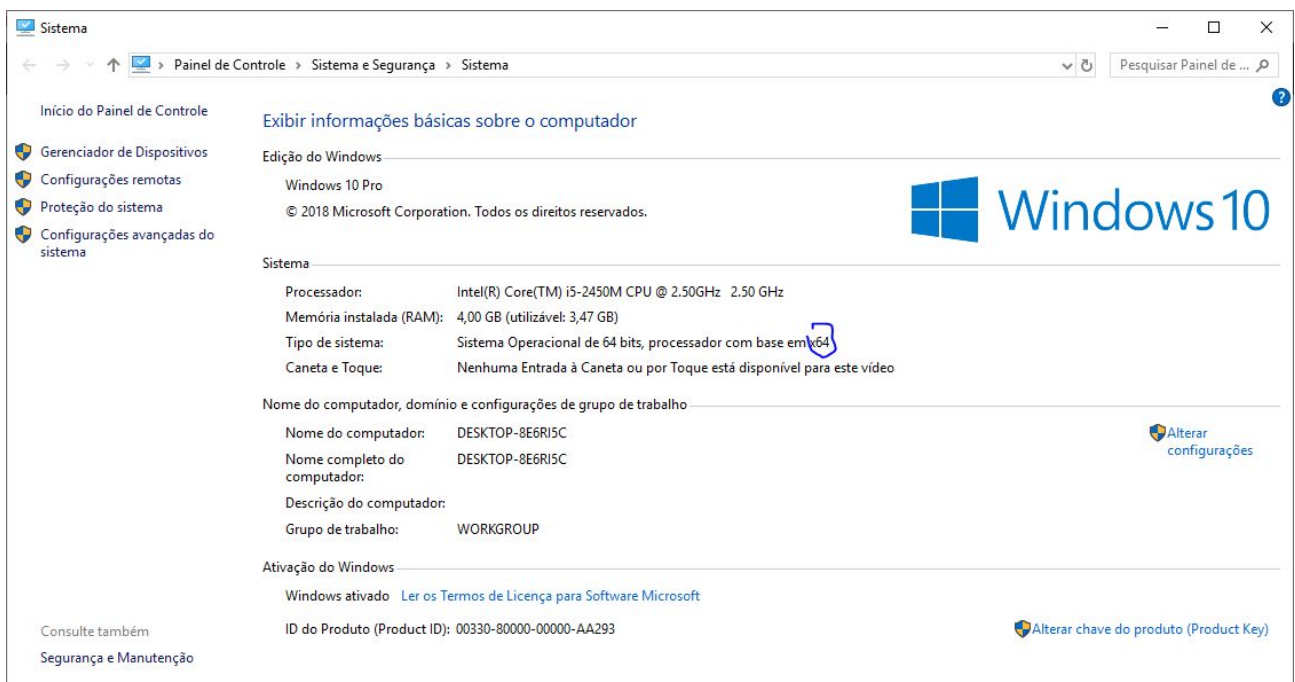


*Obs: caso você utilize outro Sistema Operacional, procure na internet algum tipo de tutorial de instalação do programa.

Depois de escolher o sistema operacional, no caso aqui, windows, você verá as seguintes informações mostradas acima. Você precisará escolher a versão do programa de acordo com a arquitetura do sistema operacional.

Para saber a arquitetura do seu sistema operacional siga um dos métodos abaixo:

método 1: Aperte as teclas Windows + Pause break para exibir as informações básicas do sistema.



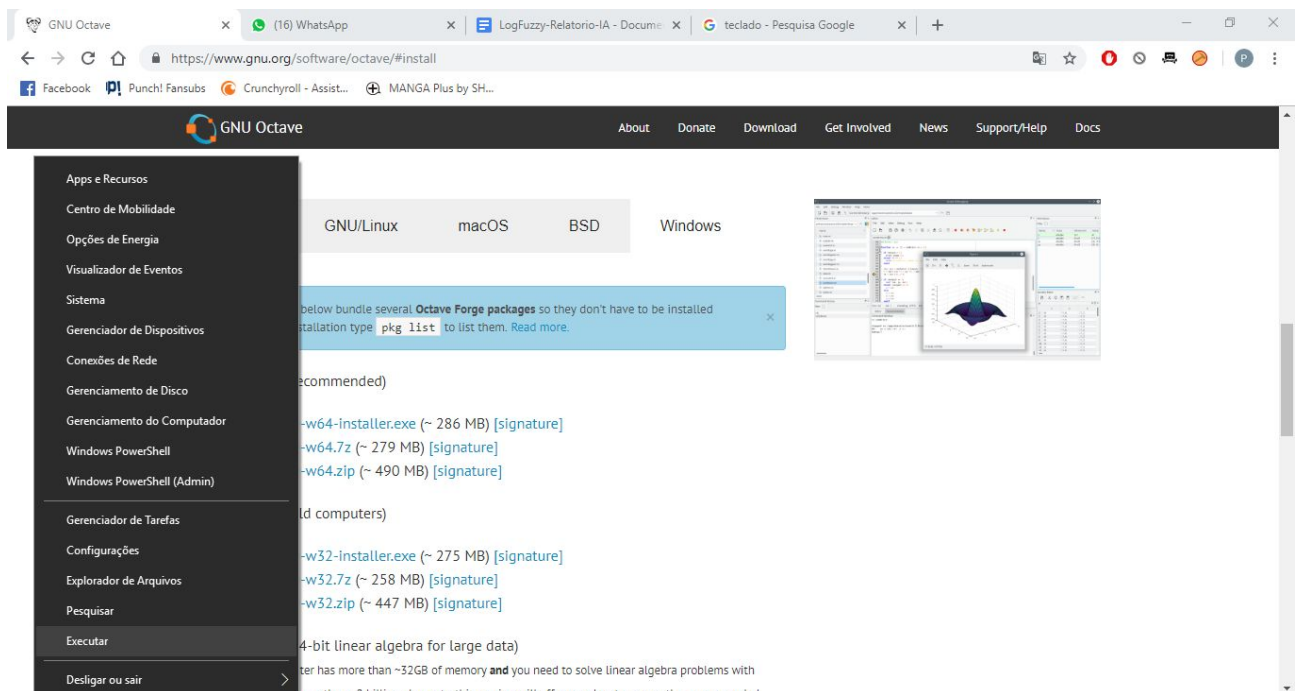
*obs: O elemento marcado é o tipo de arquitetura do sistema;

método 2: no caso de você utilizar windows 10, clique com o botão direito do mouse no símbolo do windows que fica perto da barra de pesquisa. Logo após fazer isso um painel irá surgir acima da região onde fica o símbolo do windows, e nele haverá uma opção com o nome “sistema”, então clique nela para abrir o painel de configurações que mostrará as informações do seu sistema.

*obs: na imagem abaixo, os elementos circulados são os referentes a arquitetura do sistema do seu computador



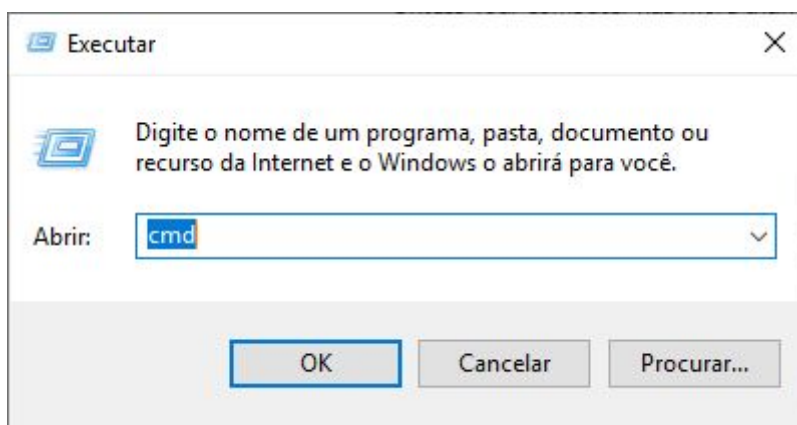
Método 3: No caso do windows 10 aperte o botão direito do mouse no símbolo do windows que fica ao lado da barra de pesquisa e logo em seguida clique em executar.



Caso você utilize outra versão do windows, aperte as teclas Window+R para ativar o comando “executar”.



Fazendo isso aparecerá um painel de comandos que você deverá digitar o seguinte comando “cmd”, sem aspas e em seguida aperte em “ok”.



Fazendo isso o prompt de comandos do windows será aberto e a partir de lá será possível descobrir a arquitetura do seu sistema através do seguinte comando que deverá ser digitado: `wmic os get osarchitecture`.

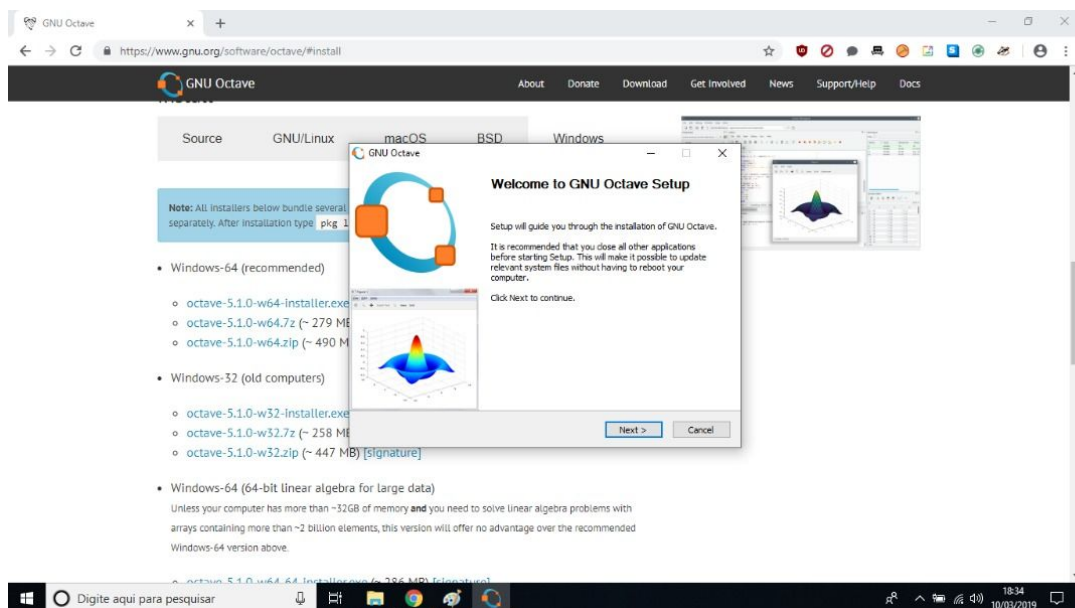
```
C:\Windows\system32\cmd.exe
Microsoft Windows [versão 10.0.17763.348]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

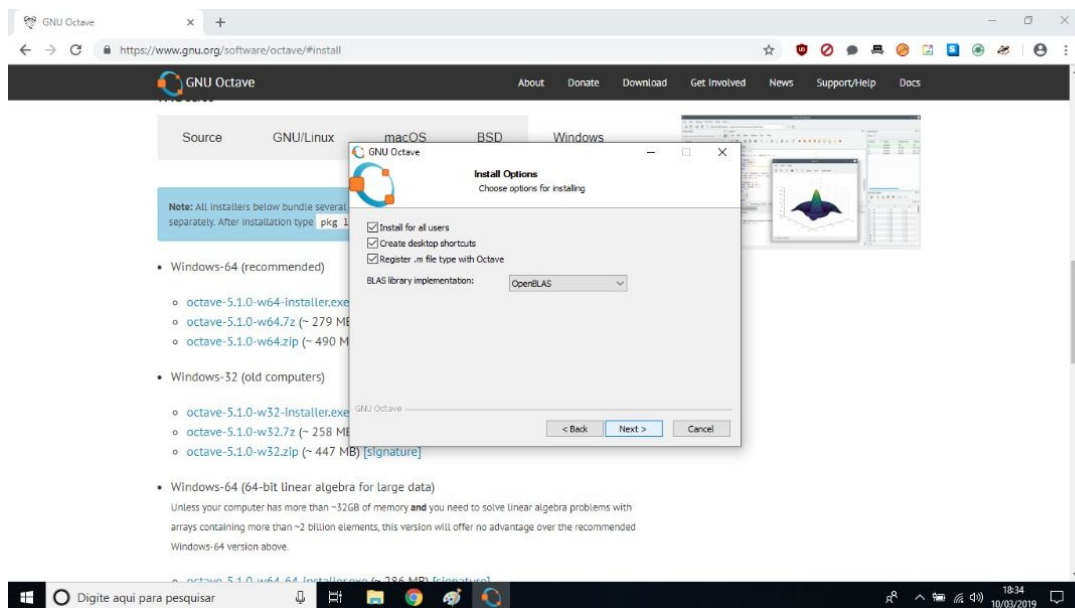
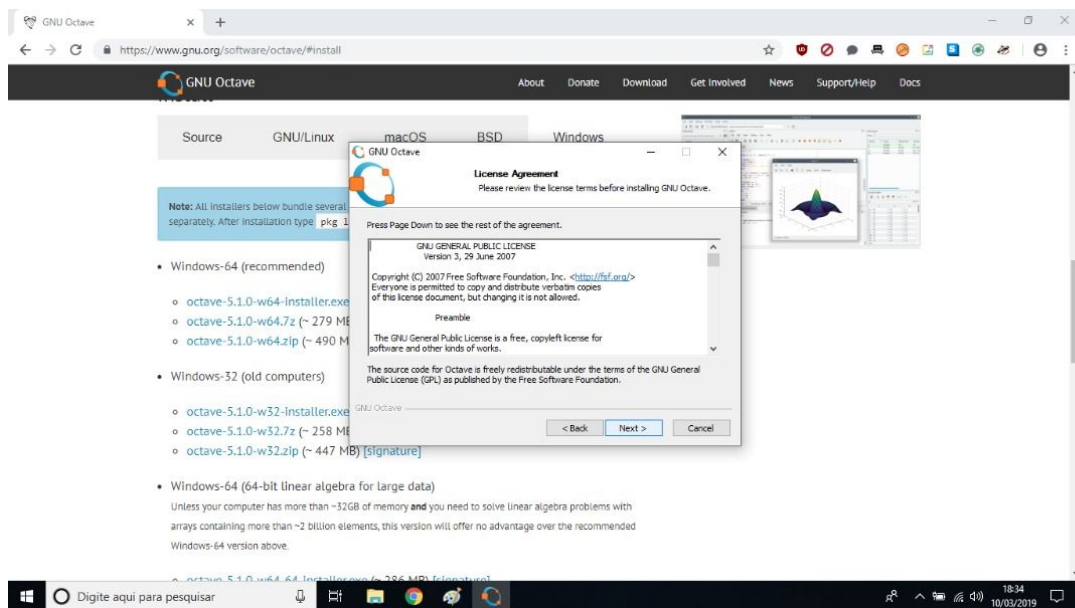
C:\Users\pedru>wmic os get OSArchitecture
OSArchitecture
64 bits

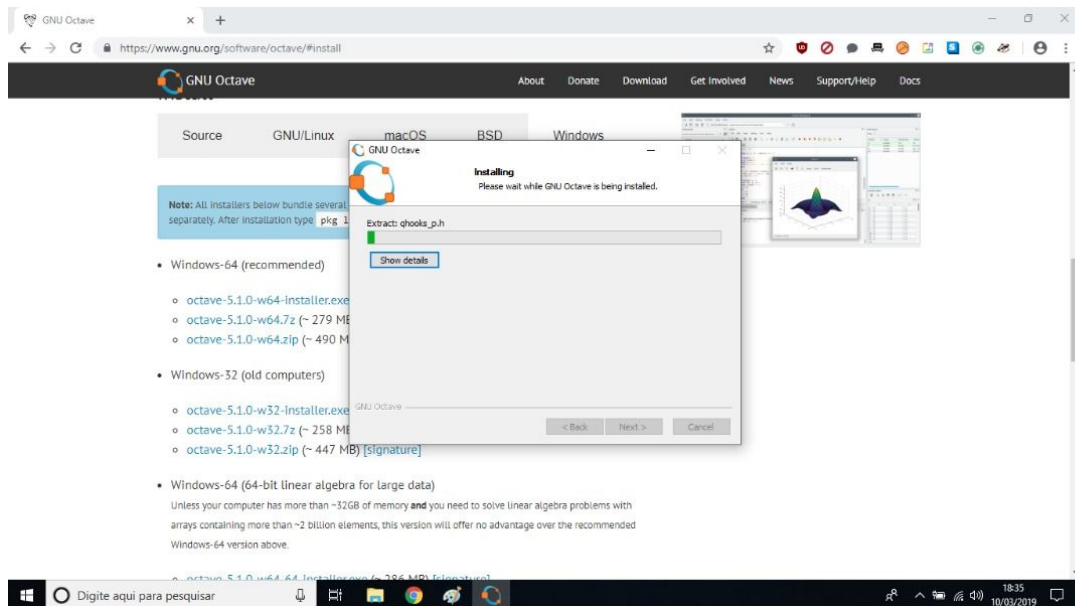
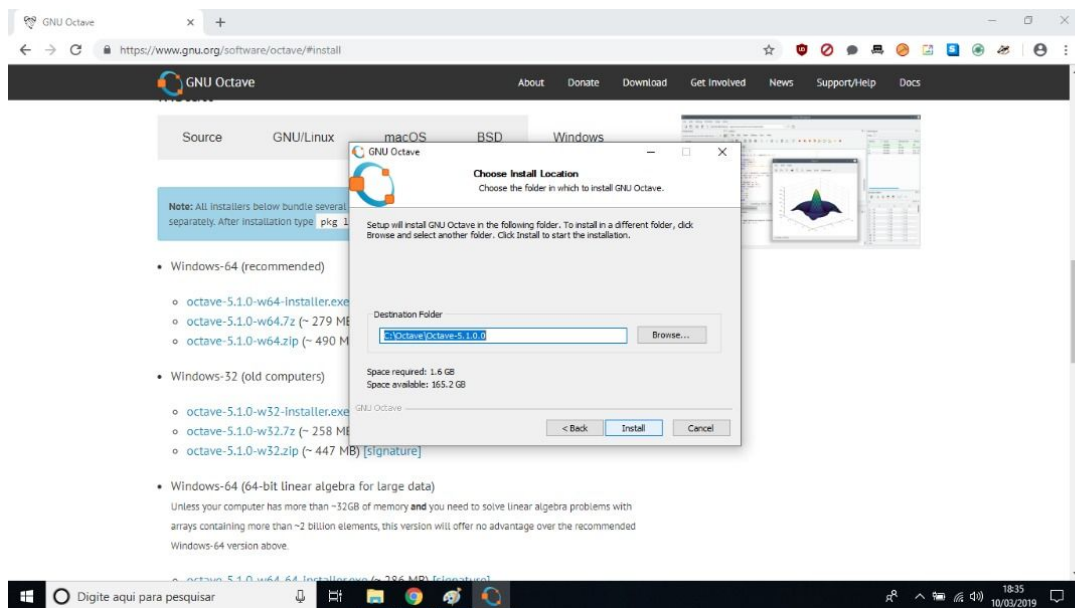
C:\Users\pedru>
```

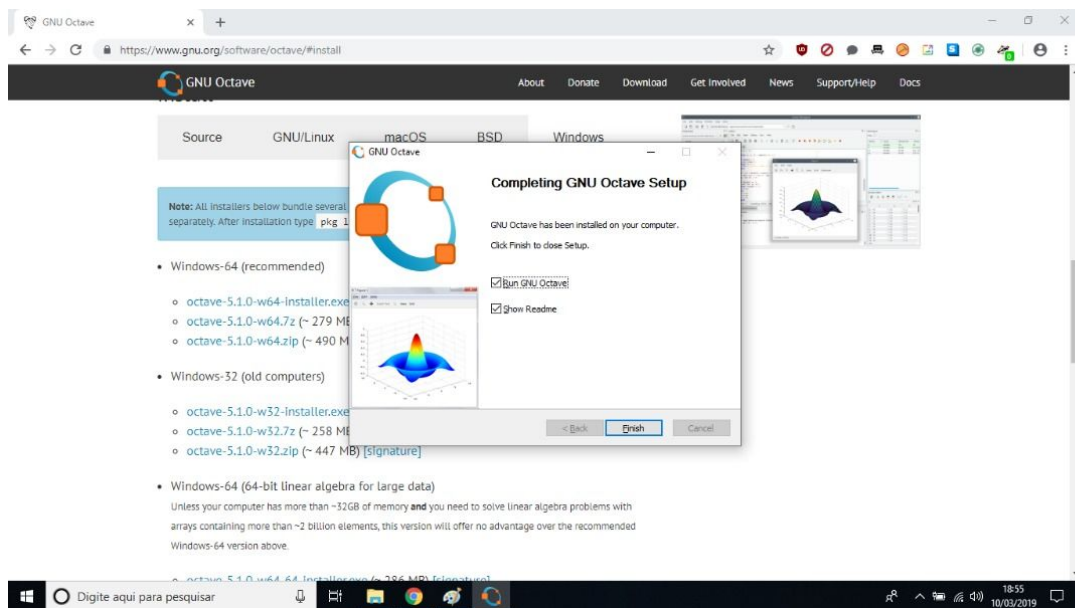
Depois descobrir a sua arquitetura de sistema, volte ao site, baixe e instale a última versão do octave baseada na sua arquitetura.

5.1 Processo de instalação









Depois de instalar o octave vá na área de trabalho, e procure o atalho. Em seguida abra o programa. caso utilize o linux ou outro sistema operacional, procure na internet como abrir o programa.



Antes de começar a executar o arquivo do trabalho, faz-se necessário ativar as ferramentas de fuzzy(fuzzy-logic-toolkit) para isso siga os procedimentos abaixo:

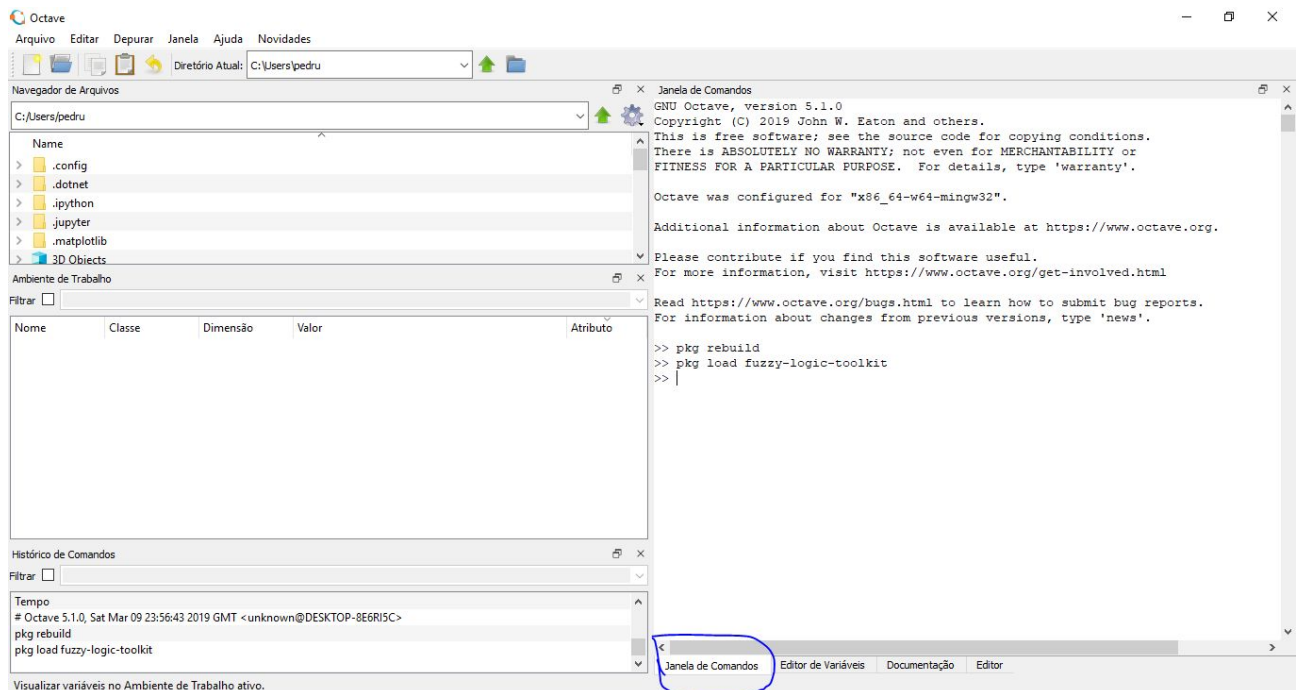
Com o programa aberto, vá na janela de comandos e digite os comandos:

comando 1: “pkg rebuild” sem aspas(comando para reconstruir pacotes do octave caso estejam corrompidos) - *obs: durante o desenvolvimento da aplicação foi necessário usar este comando por causa que os pacotes estavam corrompidos e nem apareciam na

lista de pacotes instalados e por conta disso não era possível ativar o pacote de ferramentas fuzzy. Por via das dúvidas use esse comando.

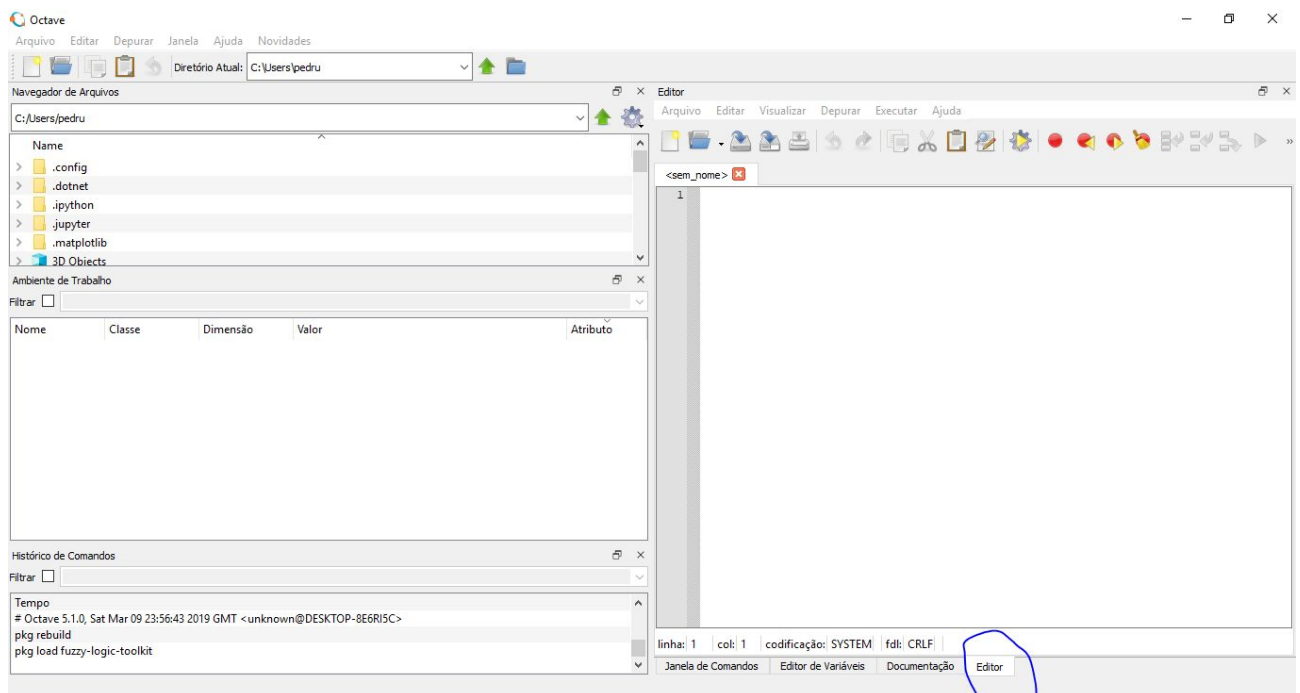
comando 2: “pkg load fuzzy-logic-toolkit” sem aspas(para ativar o pacote de ferramentas fuzzy)

*obs: Aperte a tecla enter depois de digitar cada comando.

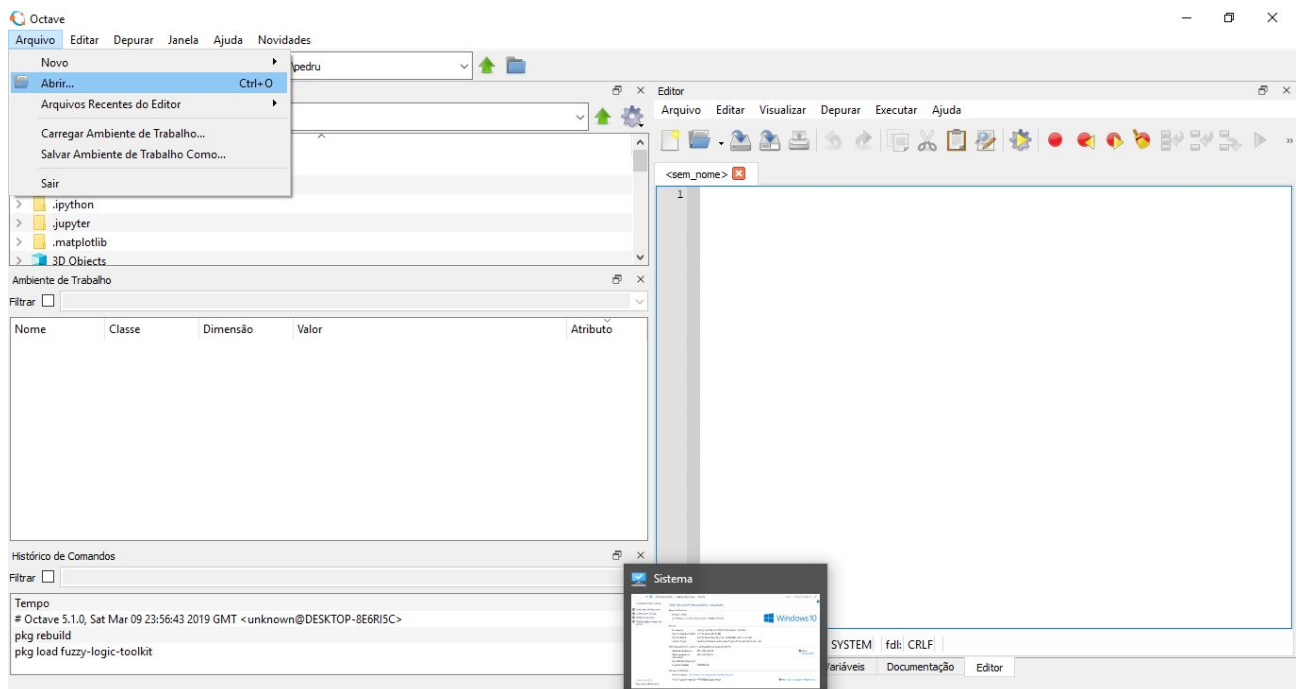


Depois de ativar o pacote de ferramentas fuzzy o programa está apto para executar a aplicação desenvolvida neste trabalho. Para isso, primeiramente devemos abrir o arquivo da aplicação no octave.

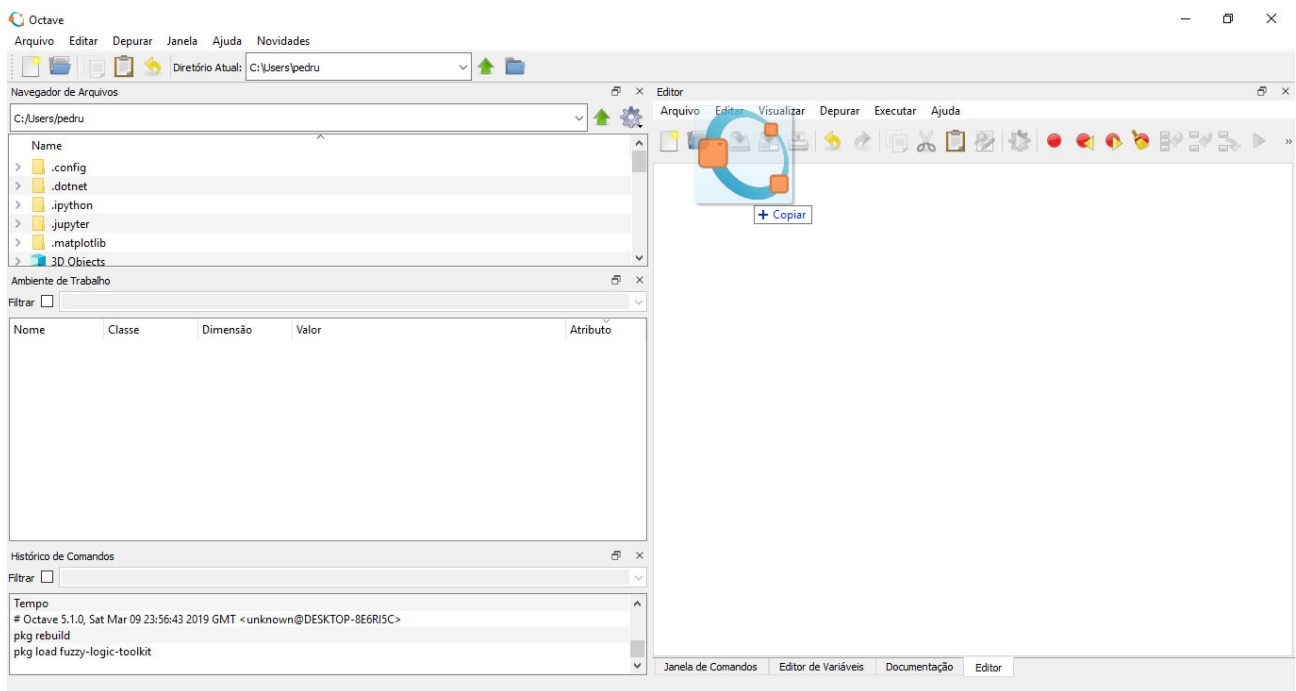
Método 1 para abrir o arquivo do octave do trabalho: vá na janela do editor



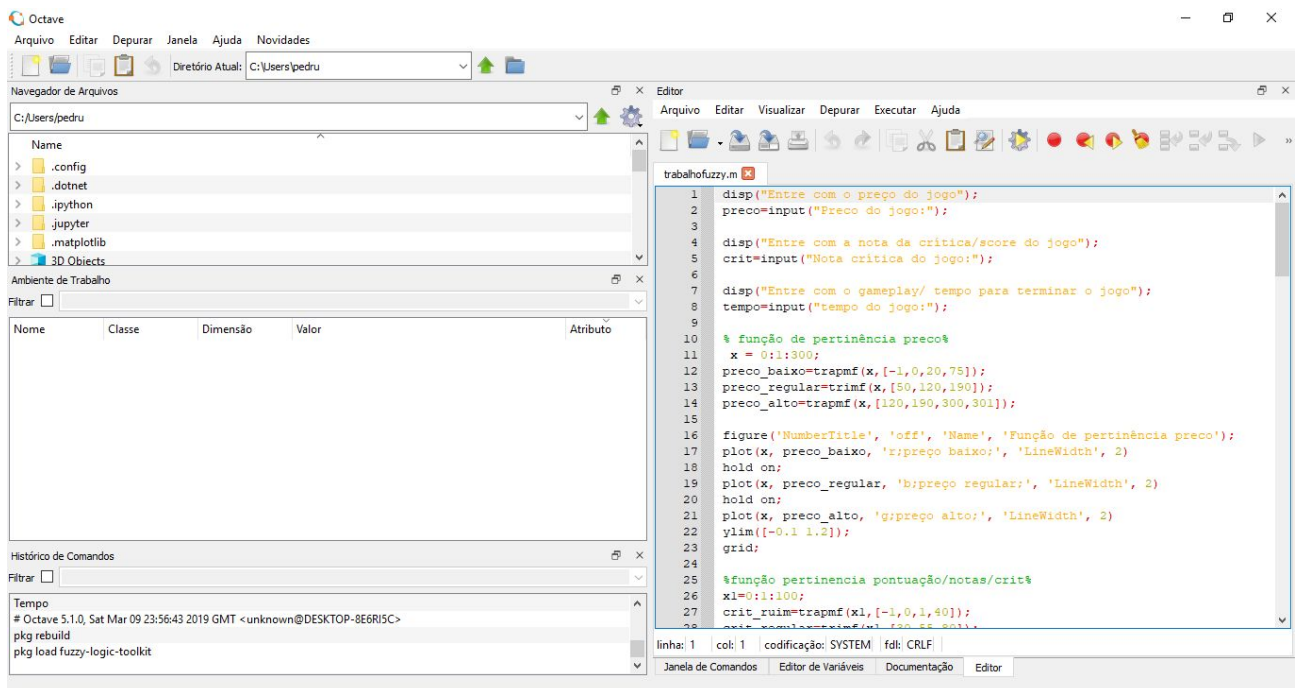
em seguida clique em “arquivo” e em seguida clique em abrir ou então simplesmente aperte ctrl +o.



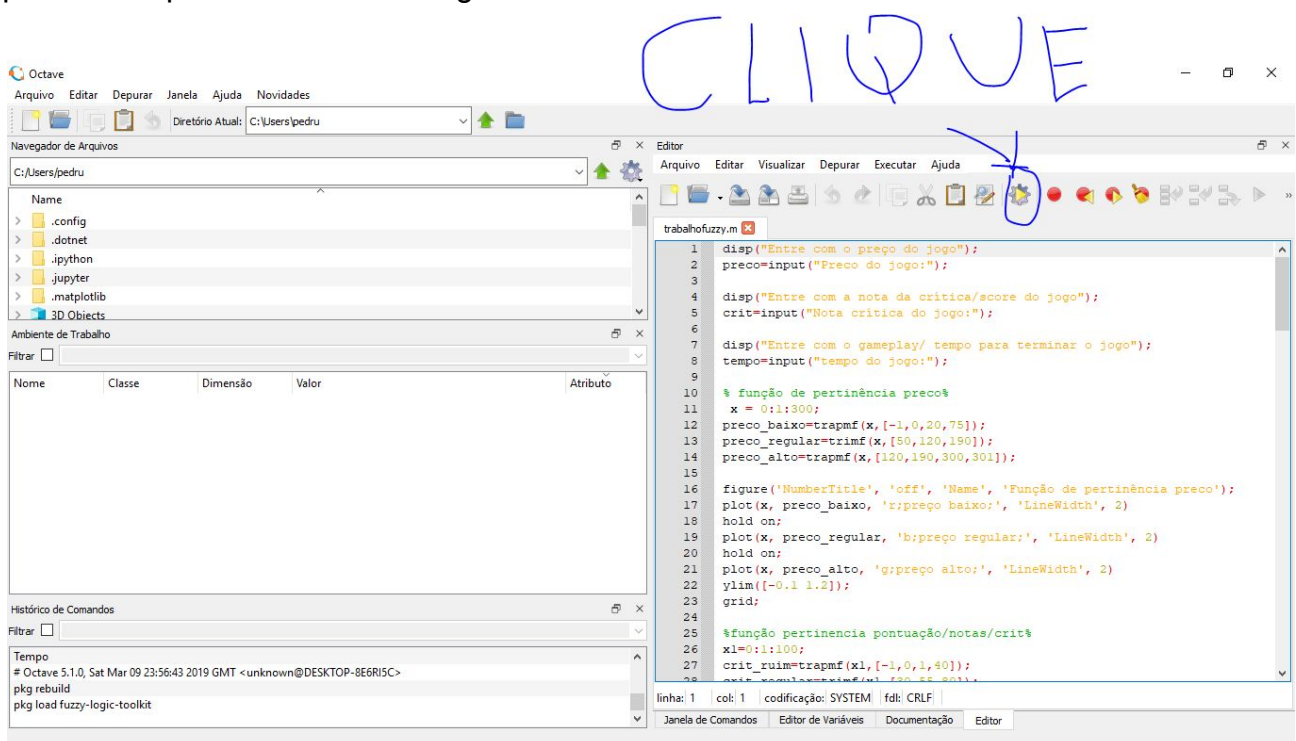
Depois de ter feito isso selecione o arquivo de nome trabalhofuzzy e clique em “open” ou abrir.



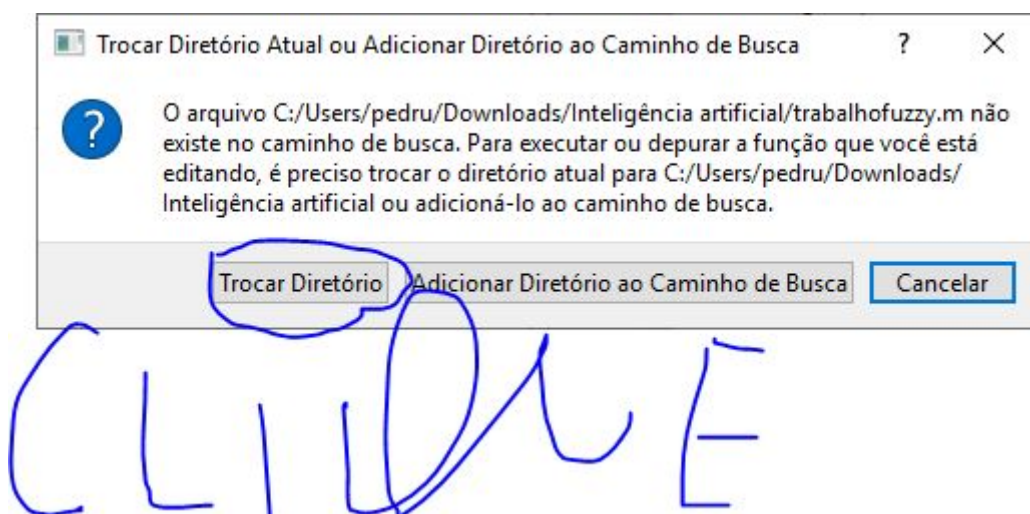
Depois de feito isso, o arquivo será aberto, e o código presente no trabalho será mostrado.



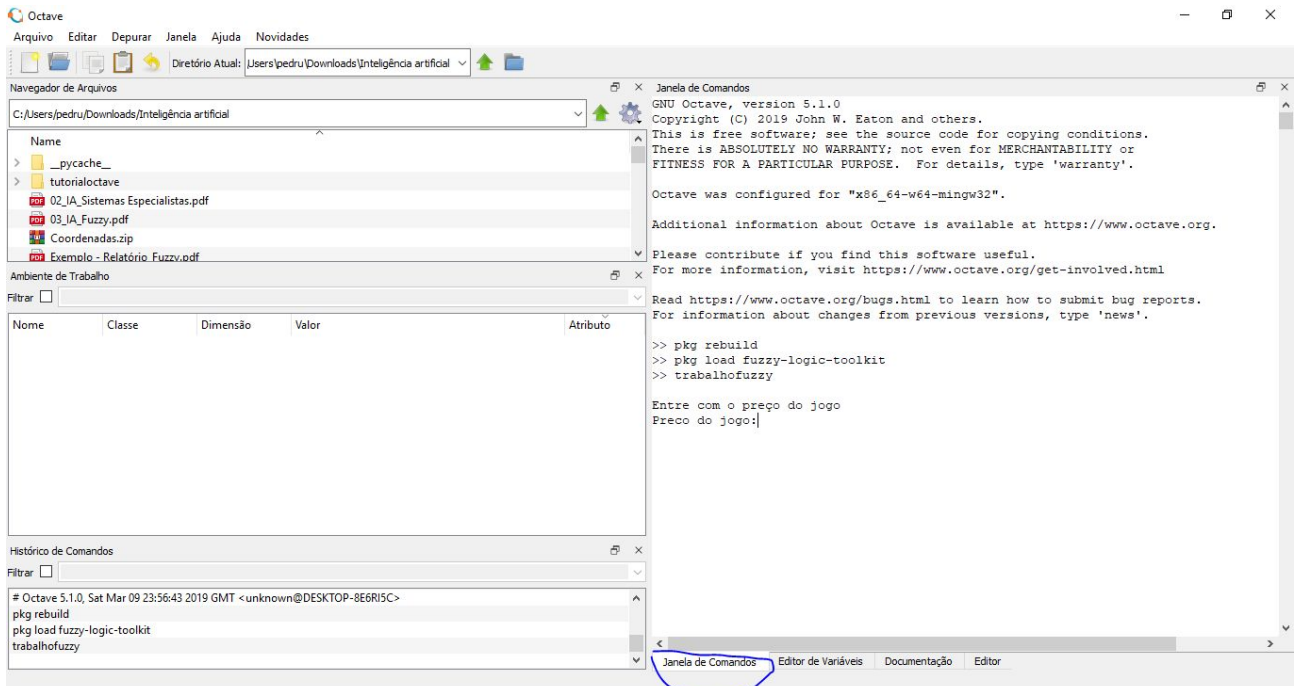
Depois de abrir o arquivo clique no símbolo de uma engrenagem com uma seta para frente para executar o código.



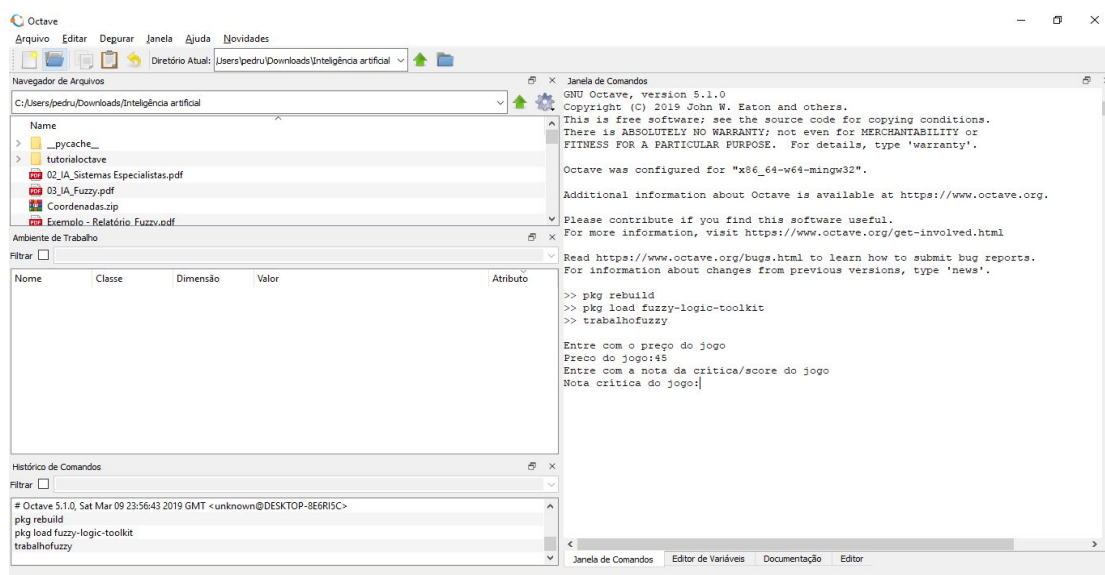
Tendo feito isso, talvez apareça uma mensagem caso o diretório do arquivo não se encontre no caminho de busca do programa. Caso apareça uma mensagem, clique em trocar diretórios.



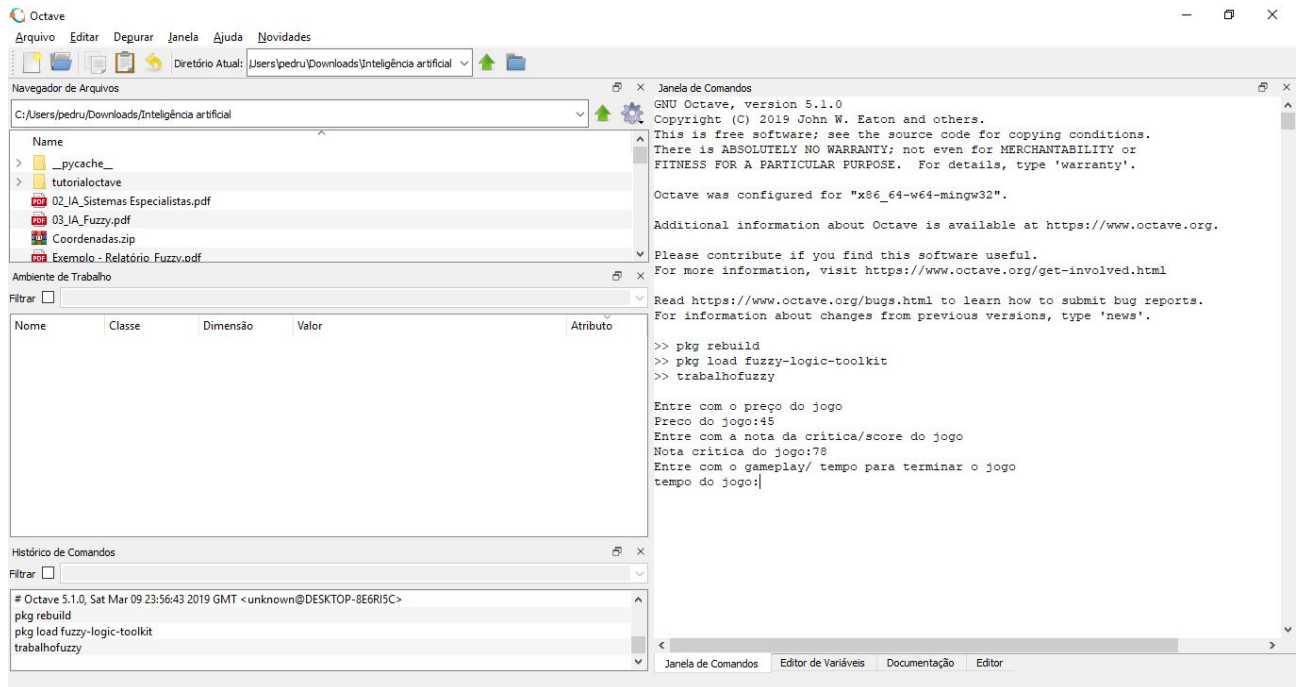
Depois de executar o código, vá no painel de comandos. É lá onde serão inseridos os valores que serão usados para gerar os resultados que mostrarão a satisfação da pessoa ao comprar certo jogo.



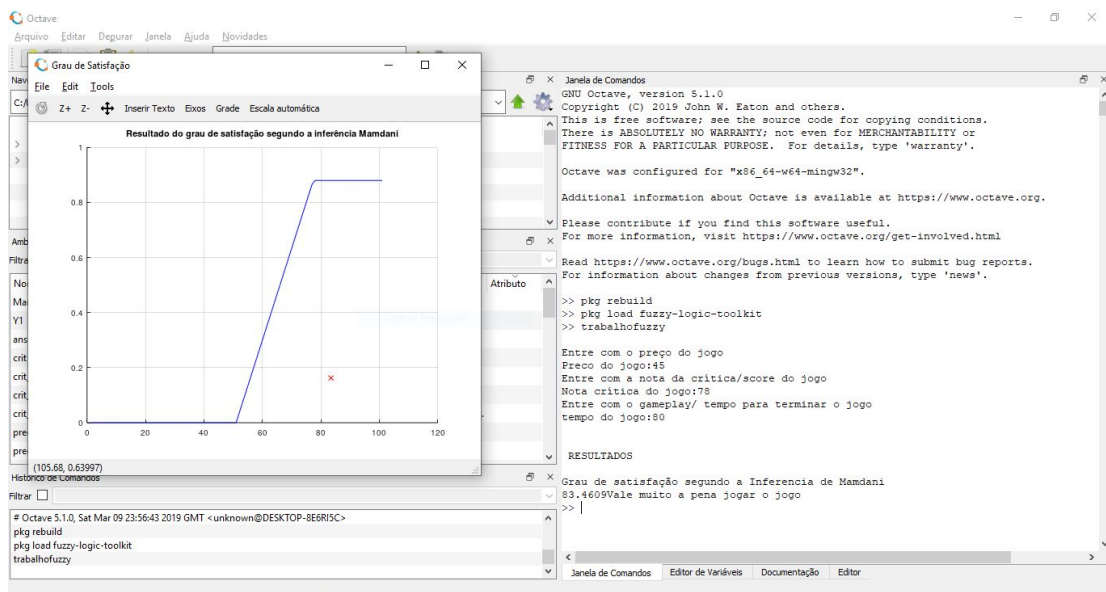
O primeiro valor pedido será o do preço do jogo ao qual será analisado. Digite um valor inteiro ou seja nada de "alguma coisa" e que seja um valor entre 1 e 300, ou melhor $\text{preço} \geq 1$ e $\text{preço} \leq 300$. Depois de digitar o valor aperte enter para que em seguida apareça um texto pedindo que insira um novo valor.



O segundo valor a ser inserido seria o da crit ou nota crítica. Esse valor seria uma pontuação dada a jogos por revistas de jogos, ou outros que mede a qualidade de um jogo de acordo com qualidade técnica entre outros fatores. Aqui deve ser inserido um valor entre 1 e 100, ou $\text{crit} \geq 1$ e $\text{crit} \leq 100$. Depois de digitar o valor aperte enter para que em seguida apareça um texto pedindo que insira um novo valor.



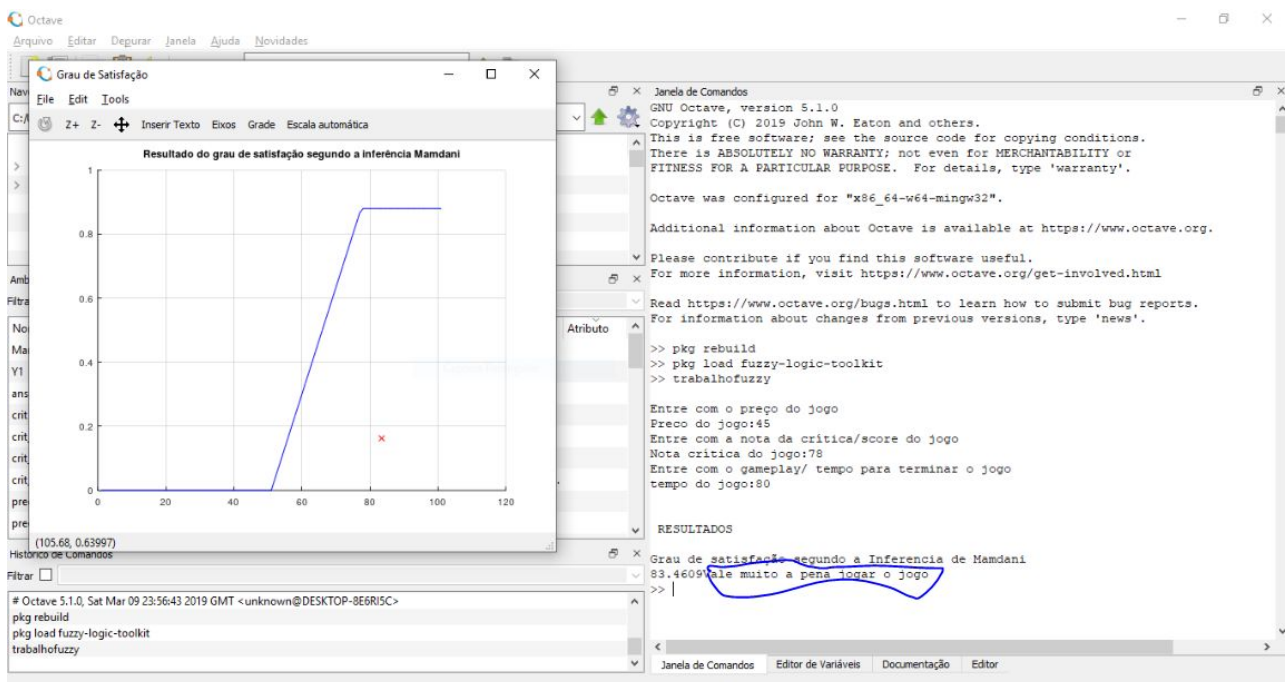
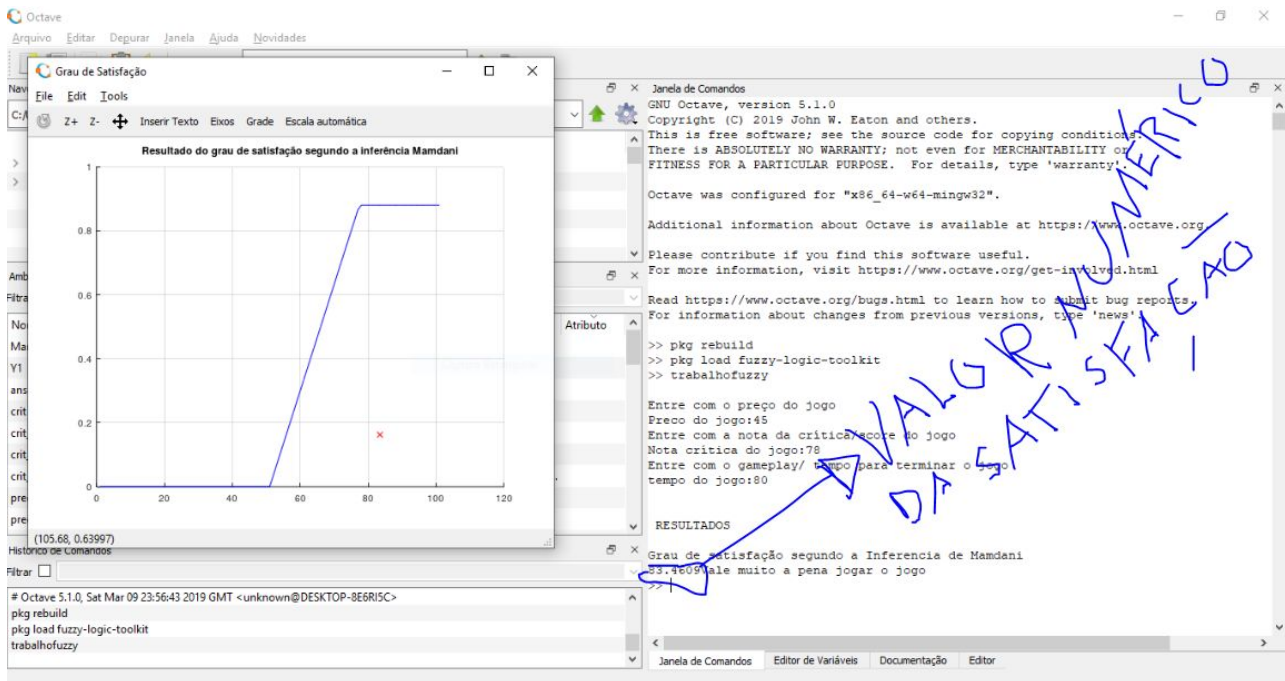
O terceiro valor seria o do tempo do jogo(tempo necessário para um jogo ser concluído). É necessário digitar um valor entre 1 e 100, ou melhor $\text{tempo} \geq 1$ e $\text{tempo} \leq 100$. Depois de digitar o valor aperte enter para que em seguida apareça uma nova janela com um gráfico e também os valores do resultado em forma de texto no painel de comandos.



No gráfico haverá o resultado da inferência de mamdani e que mostrará o grau de satisfação segundo nossos critérios.

O grau de satisfação poderá também ser visto na janela de comandos abaixo de “Resultados”. O grau de satisfação tenta mostrar, de forma numérica e segundo nossos critérios, o quão satisfeito alguém irá se sentir caso compre determinado jogo. Quanto mais próximo do 100 tiver, mais satisfeita a pessoa estará com determinado jogo

Além do valor numérico que representa satisfação, há também na janela de comandos, uma mensagem em texto, que indica se vale a pena ou não comprar um jogo segundo aquele valor de satisfação. Ela pode ajudar o usuário a saber o que significa aquele valor de satisfação.



6. CONCLUSÃO

É inegável que a lógica fuzzy se tornou uma ferramenta de destaque na solução de problemas que envolvem graus de incerteza. Este relatório se dispõe a concluir que a lógica fuzzy pode ser usada em prol da melhor orientação para o consumidor conseguir fazer a melhor decisão possível na hora de adquirir um jogo eletrônico com o melhor grau de confiabilidade em relação a seu investimento.