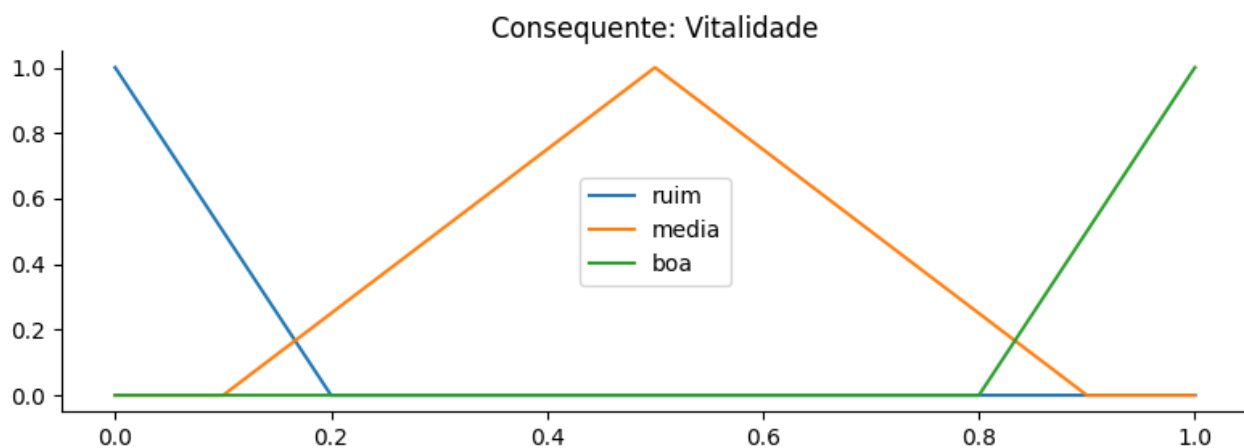
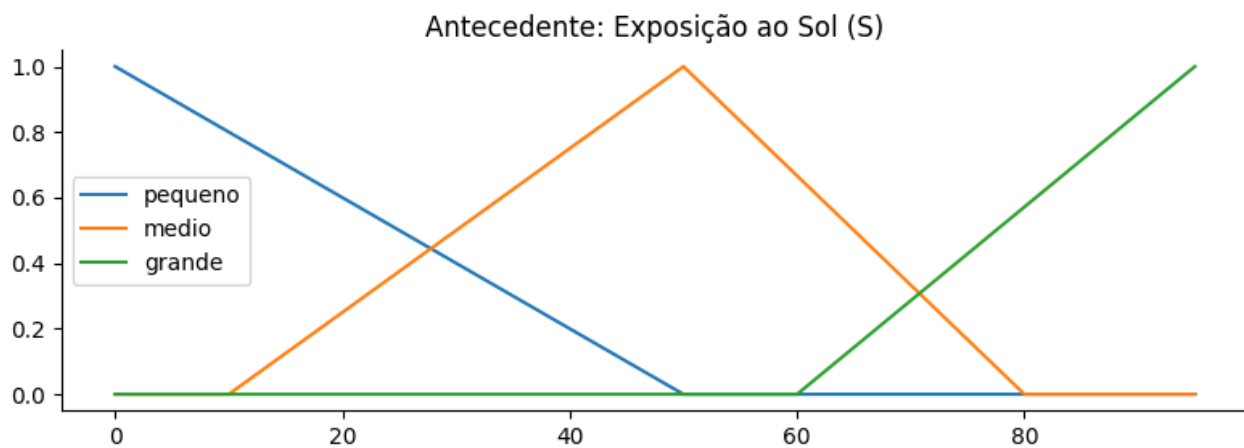
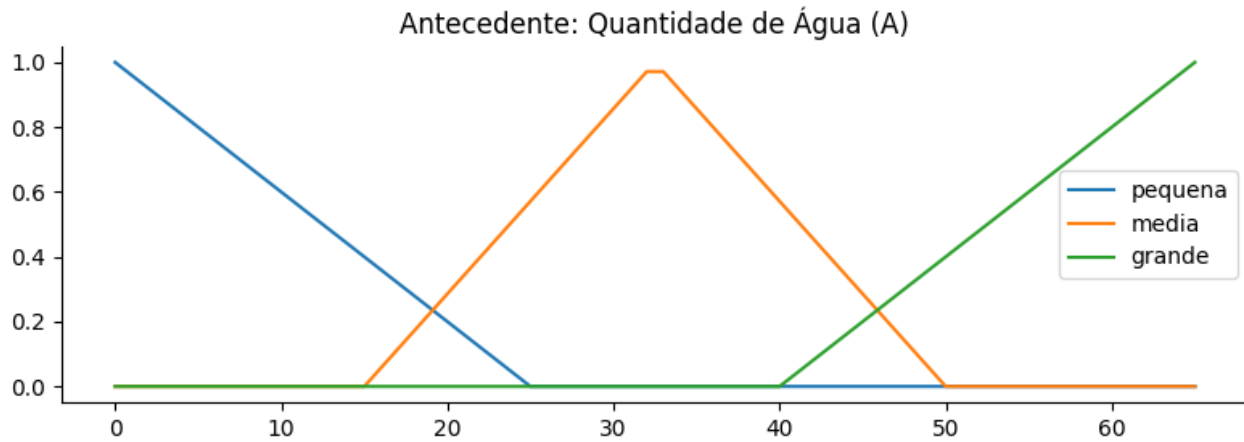


# 2º Trabalho Prático sobre Lógica Fuzzy

Aluno(a): Pedro Henrique Vilaça Valverde

## Problema A: Vitalidade das Violetas

### Funções de Pertinência:



### Base de Regras:

SE sol é pequeno E água é pequena ENTÃO vitalidade é media  
SE sol é pequeno E água é media ENTÃO vitalidade é boa  
SE sol é pequeno E água é grande ENTÃO vitalidade é ruim  
SE sol é medio E água é pequena ENTÃO vitalidade é media  
SE sol é medio E água é media ENTÃO vitalidade é boa

## 2º Trabalho Prático sobre Lógica Fuzzy

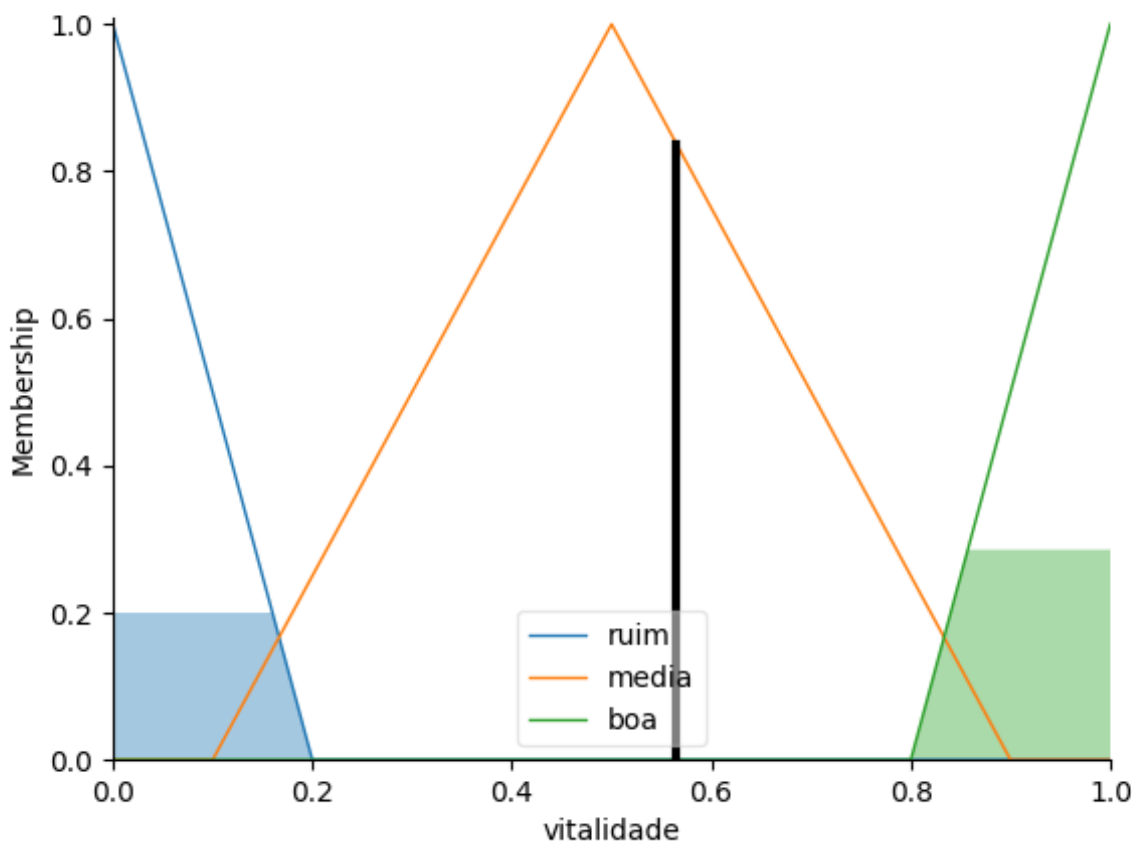
SE sol é medio E água é grande ENTÃO vitalidade é ruim  
SE sol é grande E água é pequena ENTÃO vitalidade é ruim  
SE sol é grande E água é media ENTÃO vitalidade é media  
SE sol é grande E água é grande ENTÃO vitalidade é ruim

### Resultado da Simulação:

Entradas: Quantidade de Água = 45 ml, Exposição ao Sol = 50 minutos

Valor da Vitalidade (Defuzzificado): 0.565

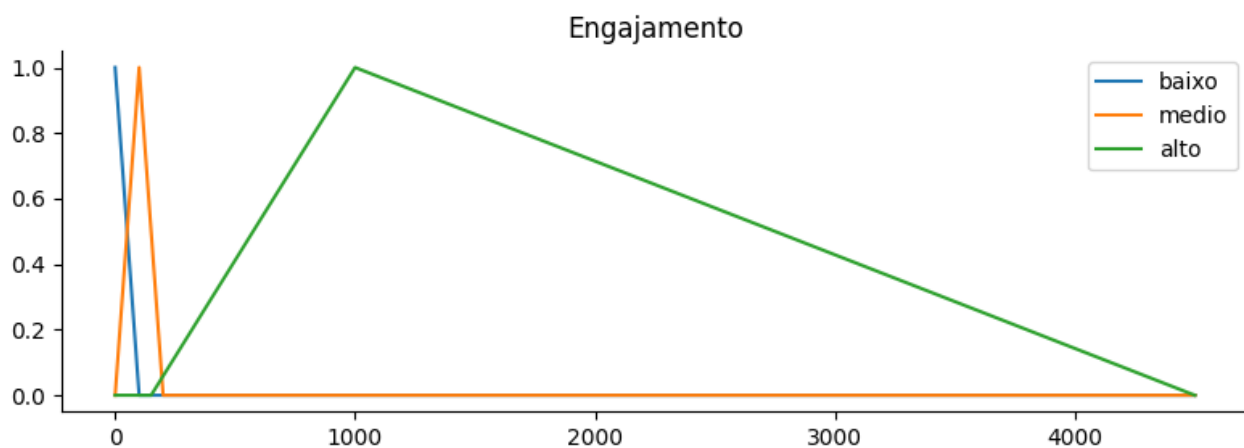
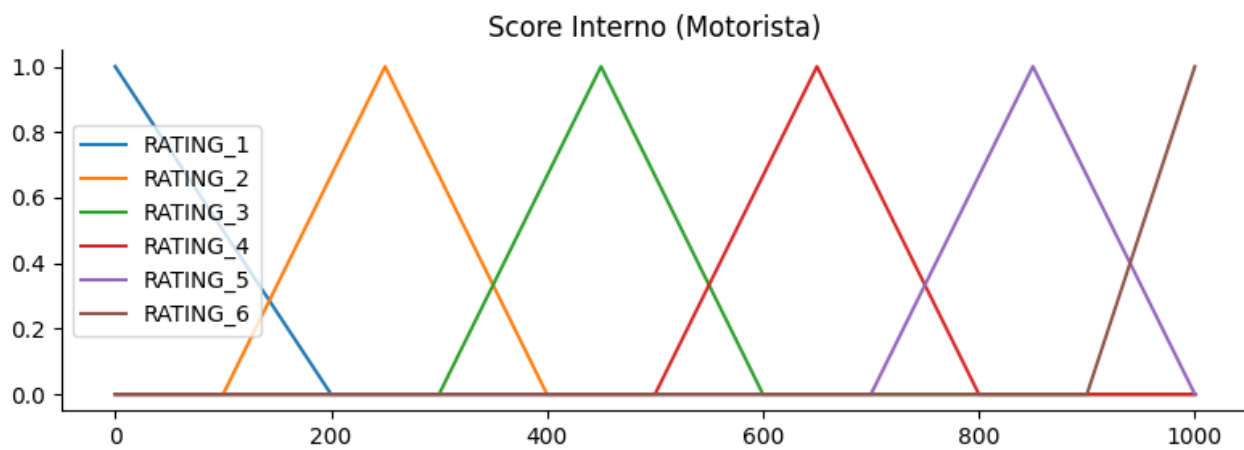
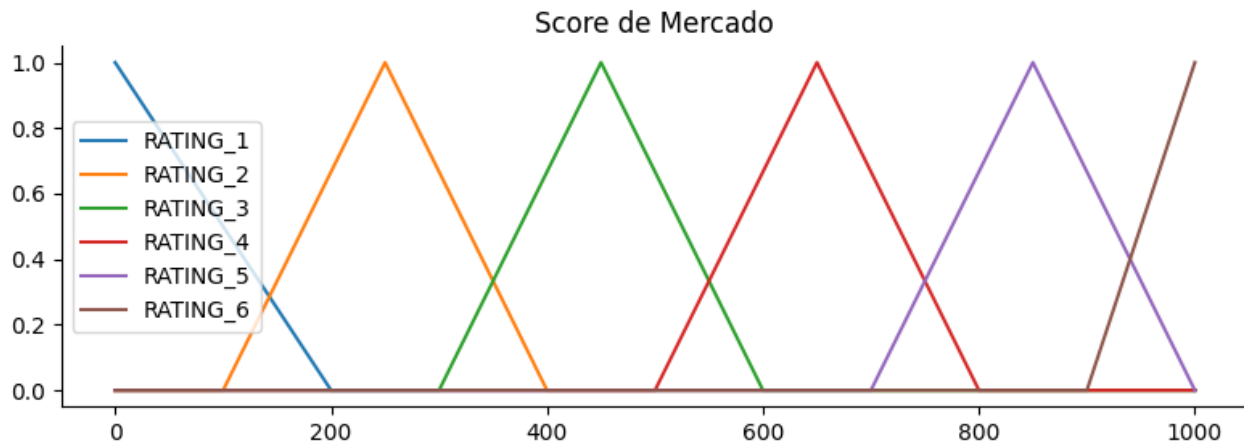
### Gráfico de Saída (Defuzzificação):



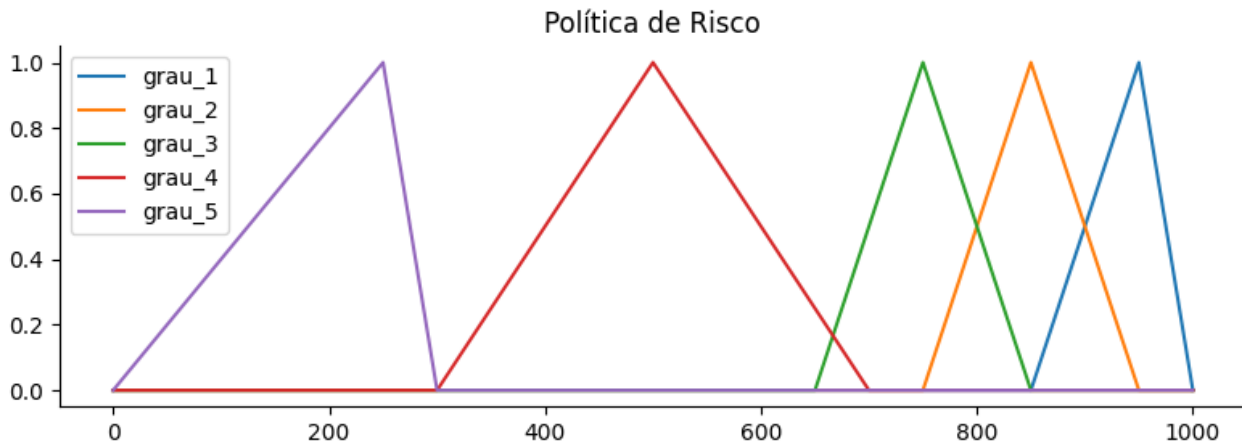
## 2º Trabalho Prático sobre Lógica Fuzzy

### Problema B: Modelagem da Política de Crédito

#### Funções de Pertinência:



## 2º Trabalho Prático sobre Lógica Fuzzy



### Base de Regras (Interpretadas):

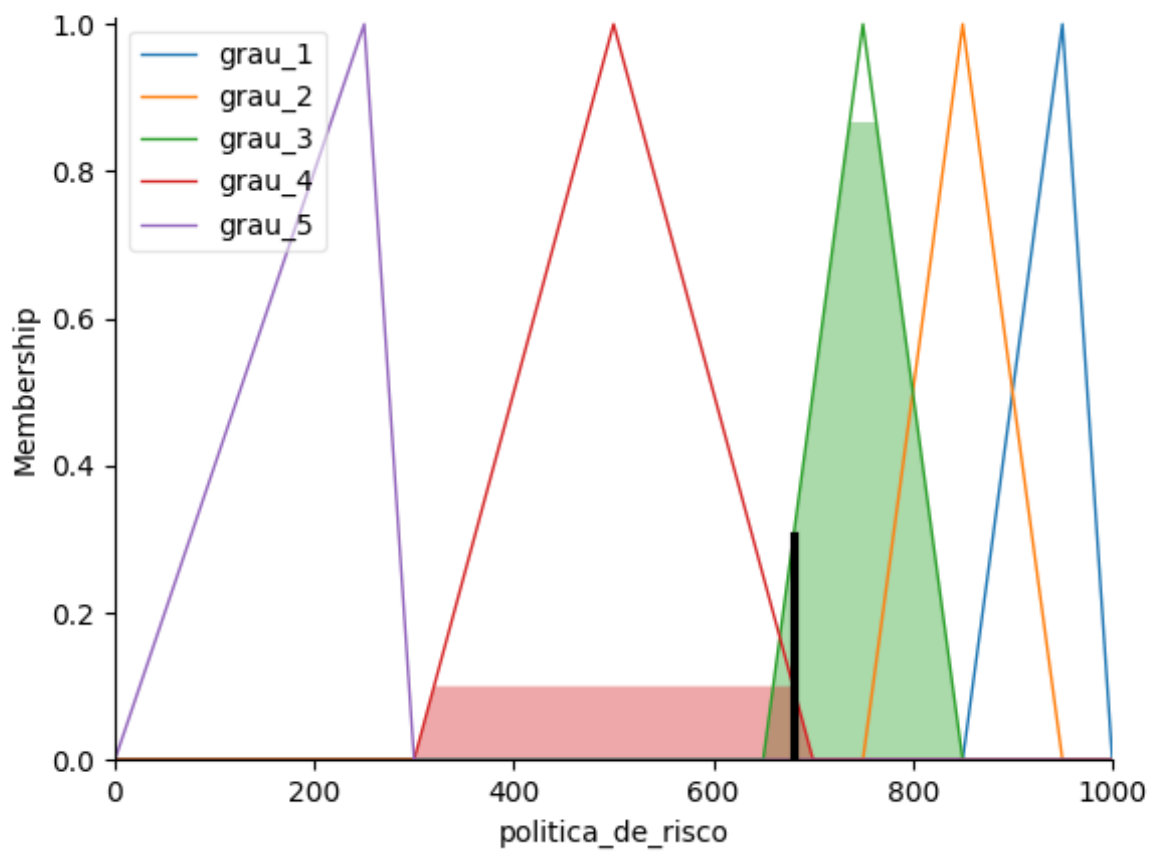
SE engajamento Baixo E Ratings em {3,4,5,6} ENTÃO Risco Grau 5  
SE engajamento Baixo E Ratings em {1,2} ENTÃO Risco Grau 4  
SE engajamento Médio E Ratings em {4,5,6} ENTÃO Risco Grau 5  
SE engajamento Médio E Ratings em {1,2,3} ENTÃO Risco Grau 3  
SE engajamento Alto E Ratings são 6 ENTÃO Risco Grau 4  
SE engajamento Alto E Ratings em {4,5} ENTÃO Risco Grau 3  
SE engajamento Alto E Ratings são 3 ENTÃO Risco Grau 2  
SE engajamento Alto E Ratings em {1,2} ENTÃO Risco Grau 1

### Resultado da Simulação:

Entradas: score\_mercado = 2, score\_interno = 230, engajamento = 90  
Política de Risco (Defuzzificado): 680.46

### Gráfico de Saída (Defuzzificação):

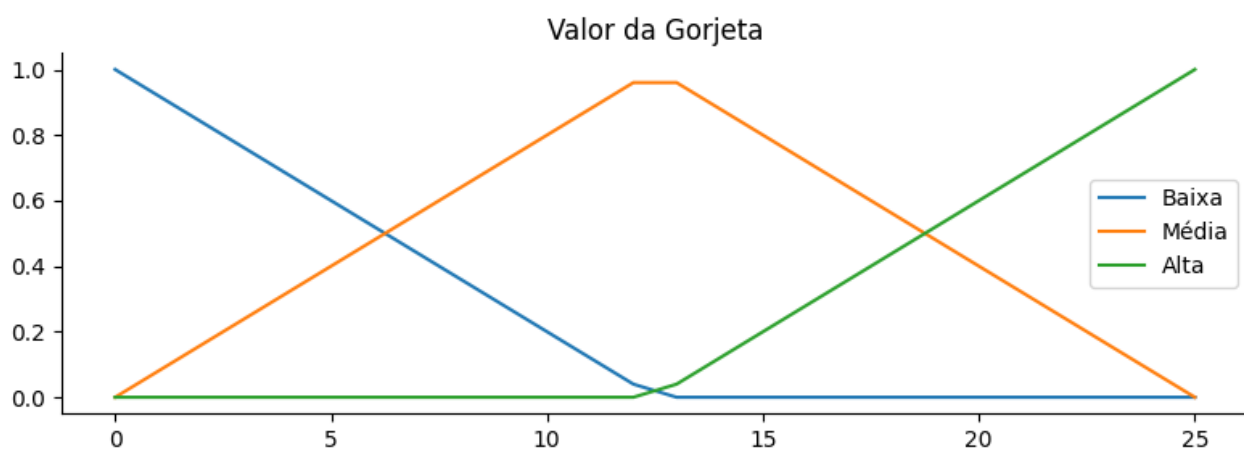
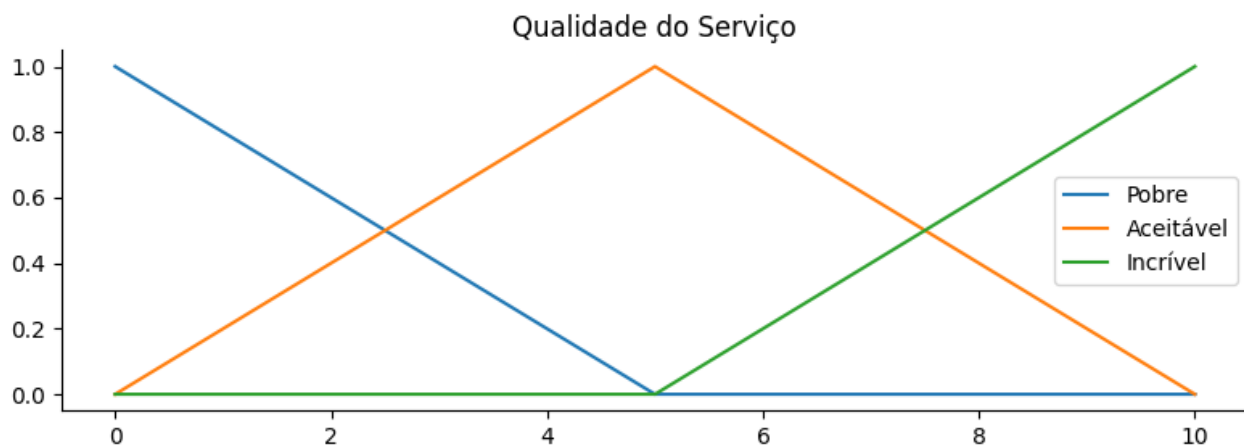
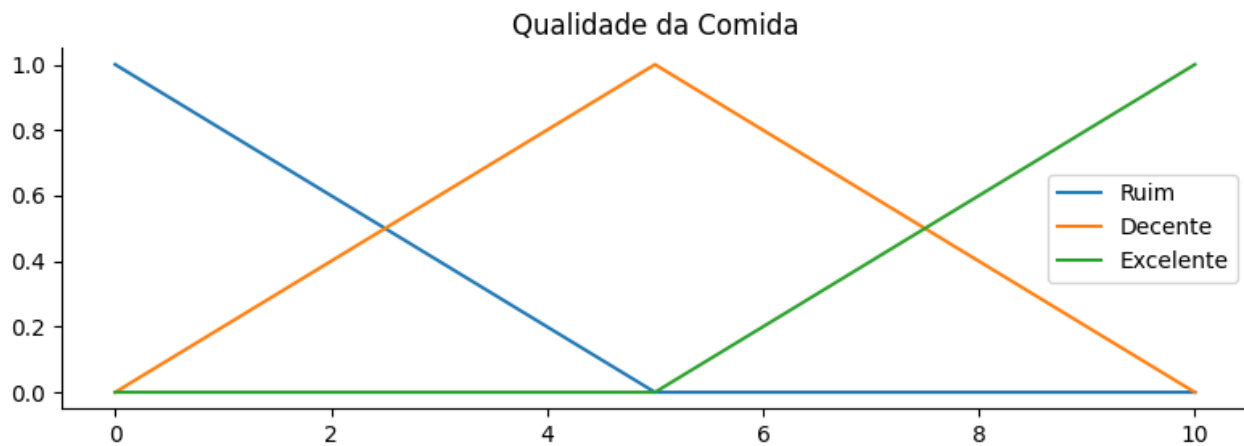
## 2º Trabalho Prático sobre Lógica Fuzzy



## 2º Trabalho Prático sobre Lógica Fuzzy

### Problema C: Problema da Gorjeta

#### Funções de Pertinência:



#### Base de Regras:

1. Se a comida é Ruim OU o serviço é Pobre, então a gorjeta será Baixa
2. Se o serviço for Aceitável, a gorjeta será Média
3. Se a comida é Excelente OU o serviço é Incrível, então a gorjeta será Alta

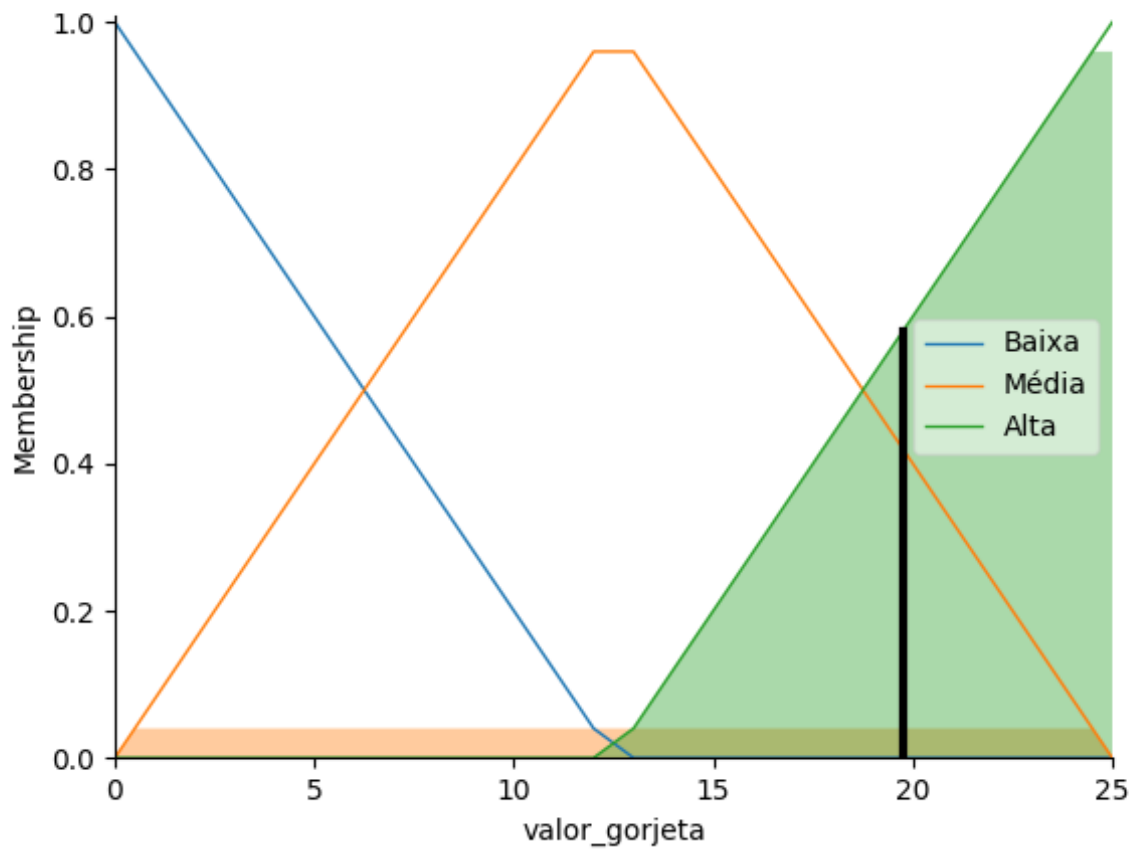
#### Resultado da Simulação:

Entradas: Qualidade da Comida = 6.5, Qualidade do Serviço = 9.8

Valor da Gorjeta (Defuzzificado): 19.76

## 2º Trabalho Prático sobre Lógica Fuzzy

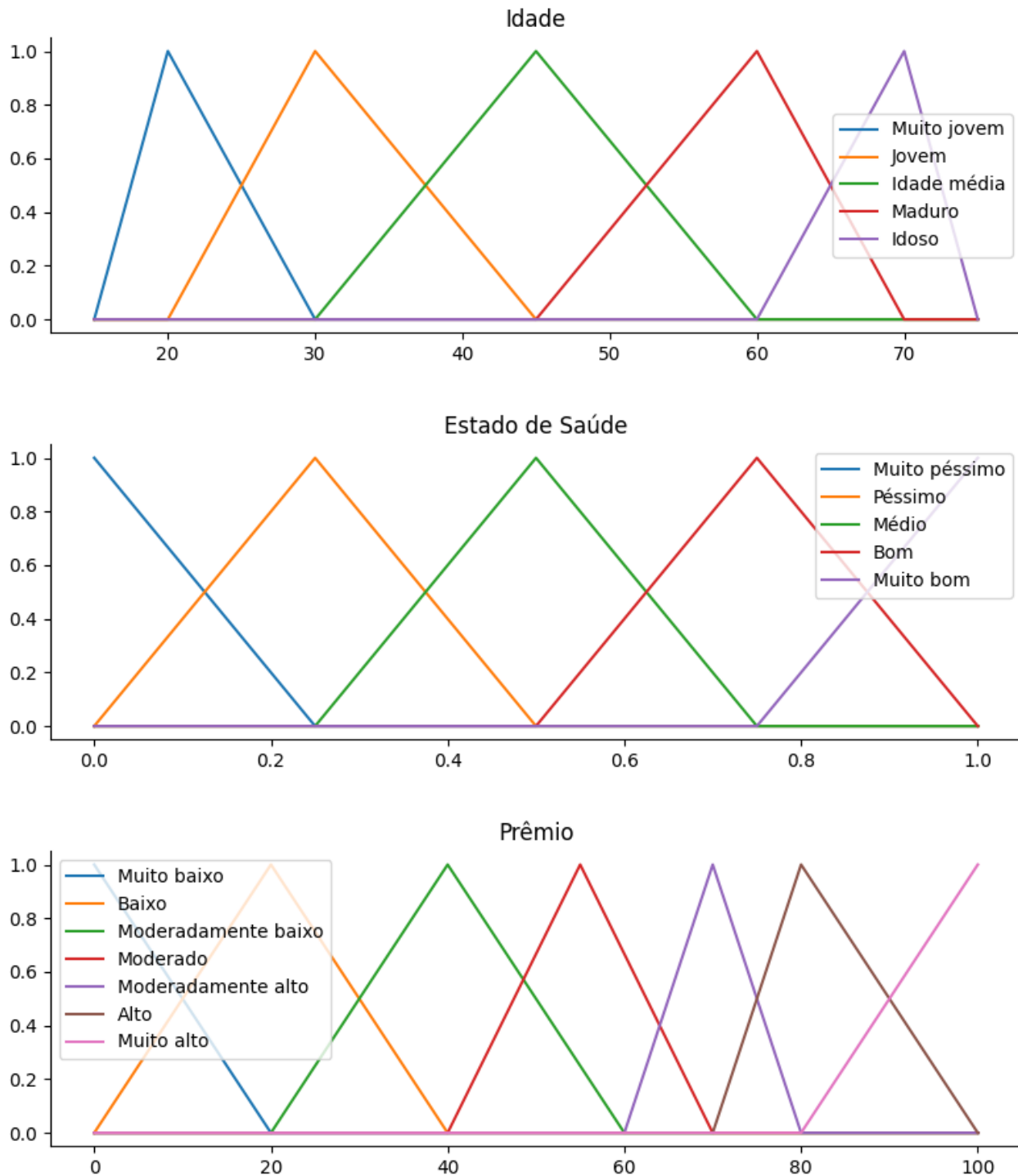
Gráfico de Saída (Defuzzificação):



## 2º Trabalho Prático sobre Lógica Fuzzy

### Problema D: Cálculo do Prêmio de Seguro

#### Funções de Pertinência:



#### Base de Regras:

SE idade é 'Muito jovem' E saúde é 'Muito péssimo' ENTÃO prêmio é 'Moderado'  
SE idade é 'Jovem' E saúde é 'Muito péssimo' ENTÃO prêmio é 'Moderadamente alto'  
SE idade é 'Idade média' E saúde é 'Muito péssimo' ENTÃO prêmio é 'Moderadamente alto'  
SE idade é 'Maduro' E saúde é 'Muito péssimo' ENTÃO prêmio é 'Alto'  
SE idade é 'Idoso' E saúde é 'Muito péssimo' ENTÃO prêmio é 'Muito alto'  
SE idade é 'Muito jovem' E saúde é 'Péssimo' ENTÃO prêmio é 'Moderadamente baixo'  
SE idade é 'Jovem' E saúde é 'Péssimo' ENTÃO prêmio é 'Moderado'



## 2º Trabalho Prático sobre Lógica Fuzzy

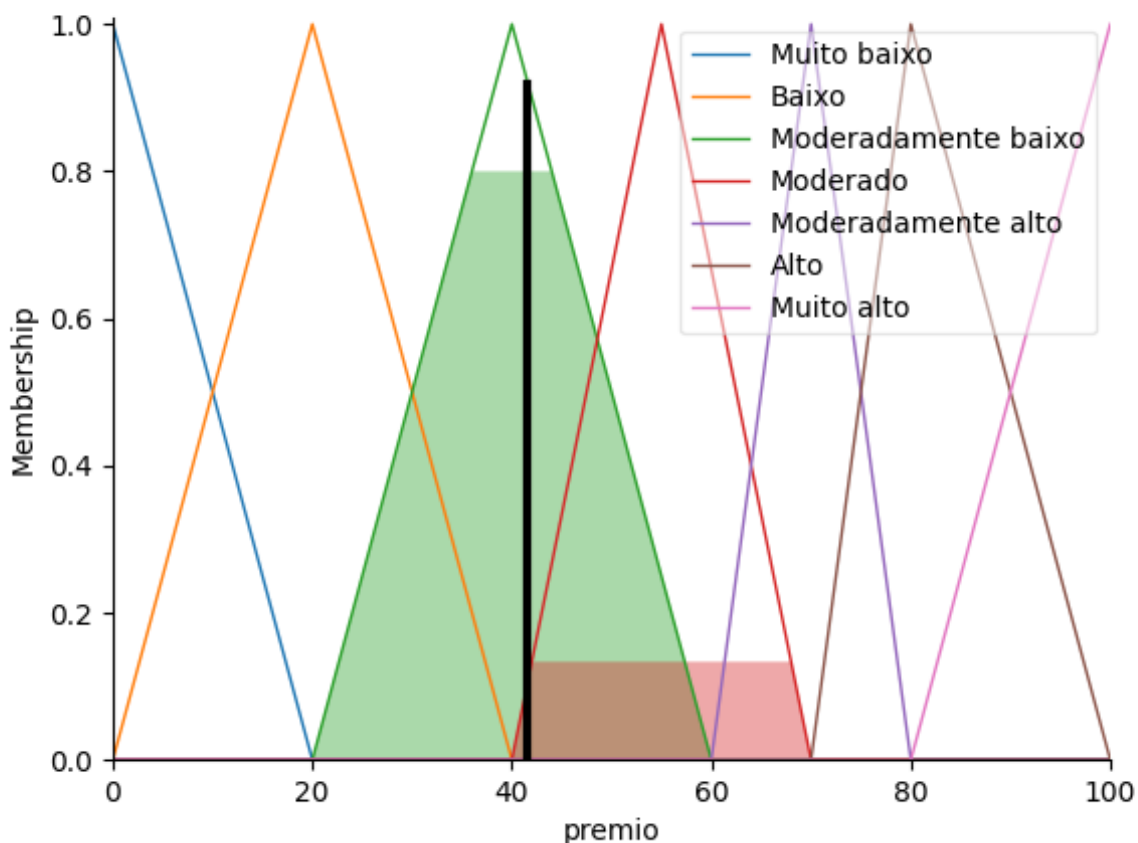
SE idade é 'Idade média' E saúde é 'Péssimo' ENTÃO prêmio é 'Moderadamente alto'  
SE idade é 'Maduro' E saúde é 'Péssimo' ENTÃO prêmio é 'Moderadamente alto'  
SE idade é 'Idoso' E saúde é 'Péssimo' ENTÃO prêmio é 'Moderadamente alto'  
SE idade é 'Muito jovem' E saúde é 'Médio' ENTÃO prêmio é 'Moderadamente baixo'  
SE idade é 'Jovem' E saúde é 'Médio' ENTÃO prêmio é 'Moderadamente baixo'  
SE idade é 'Idade média' E saúde é 'Médio' ENTÃO prêmio é 'Moderado'  
SE idade é 'Maduro' E saúde é 'Médio' ENTÃO prêmio é 'Moderadamente alto'  
SE idade é 'Idoso' E saúde é 'Médio' ENTÃO prêmio é 'Moderadamente alto'  
SE idade é 'Muito jovem' E saúde é 'Bom' ENTÃO prêmio é 'Baixo'  
SE idade é 'Jovem' E saúde é 'Bom' ENTÃO prêmio é 'Moderadamente baixo'  
SE idade é 'Idade média' E saúde é 'Bom' ENTÃO prêmio é 'Moderadamente baixo'  
SE idade é 'Maduro' E saúde é 'Bom' ENTÃO prêmio é 'Moderado'  
SE idade é 'Idoso' E saúde é 'Bom' ENTÃO prêmio é 'Alto'  
SE idade é 'Muito jovem' E saúde é 'Muito bom' ENTÃO prêmio é 'Muito baixo'  
SE idade é 'Jovem' E saúde é 'Muito bom' ENTÃO prêmio é 'Baixo'  
SE idade é 'Idade média' E saúde é 'Muito bom' ENTÃO prêmio é 'Moderadamente baixo'  
SE idade é 'Maduro' E saúde é 'Muito bom' ENTÃO prêmio é 'Moderadamente baixo'  
SE idade é 'Idoso' E saúde é 'Muito bom' ENTÃO prêmio é 'Moderado'

### Resultado da Simulação:

Entradas: Idade = 32 anos, Estado de Saúde = 0.7

Valor do Prêmio (Defuzzificado): 41.60

### Gráfico de Saída (Defuzzificação):



# 2º Trabalho Prático sobre Lógica Fuzzy

## Código Fonte: resolver\_fuzzy.py

---

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt
import os

# --- FUNÇÃO AUXILIAR PARA SALVAR GRÁFICOS ---
def save_variable_plot(variable, file_name, title):
    """
    Plota e salva o gráfico de uma variável fuzzy em um arquivo.
    """
    fig, ax = plt.subplots(figsize=(8, 3))
    for term in variable.terms:
        ax.plot(variable.universe, variable[term].mf, label=term)
    ax.set_title(title)
    ax.legend()
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.get_xaxis().tick_bottom()
    ax.get_yaxis().tick_left()
    plt.tight_layout()
    plt.savefig(file_name)
    plt.close(fig)

# --- CRIAÇÃO DO DIRETÓRIO DE RESULTADOS ---
RESULT_DIR = './result'
if not os.path.exists(RESULT_DIR):
    os.makedirs(RESULT_DIR)
print(f"Diretório '{RESULT_DIR}' pronto para receber os resultados.")

# --- PROBLEMA A: VITALIDADE DAS VIOLETAS ---
def solve_problem_A():
    print("\n--- Iniciando Problema A: Vitalidade das Violetas ---")

    agua = ctrl.Antecedent(np.arange(0, 66, 1), 'agua')
    sol = ctrl.Antecedent(np.arange(0, 96, 1), 'sol')
    vitalidade = ctrl.Consequent(np.arange(0, 1.01, 0.01), 'vitalidade')

    agua['pequena'] = fuzz.trimf(agua.universe, [-25, 0, 25])
    agua['media'] = fuzz.trimf(agua.universe, [15, 32.5, 50])
    agua['grande'] = fuzz.trimf(agua.universe, [40, 65, 90])

    sol['pequeno'] = fuzz.trimf(sol.universe, [-50, 0, 50])
    sol['medio'] = fuzz.trimf(sol.universe, [10, 50, 80])
    sol['grande'] = fuzz.trimf(sol.universe, [60, 95, 130])

    vitalidade['ruim'] = fuzz.trimf(vitalidade.universe, [-0.2, 0, 0.2])
    vitalidade['media'] = fuzz.trimf(vitalidade.universe, [0.1, 0.5, 0.9])
    vitalidade['boa'] = fuzz.trimf(vitalidade.universe, [0.8, 1, 1.2])

    save_variable_plot(agua, os.path.join(RESULT_DIR, 'exA_agua_mf.png'), 'Antecedente: Quantidade de Água (A)')
    save_variable_plot(sol, os.path.join(RESULT_DIR, 'exA_sol_mf.png'), 'Antecedente: Exposição ao Sol (S)')
    save_variable_plot(vitalidade, os.path.join(RESULT_DIR, 'exA_vitalidade_mf.png'), 'Consequente:
```

## 2º Trabalho Prático sobre Lógica Fuzzy

```
Vitalidade')
print("Gráficos das funções de pertinência do Problema A salvos.")

regras_texto = []
regras = [
    ctrl.Rule(sol['pequeno'] & agua['pequena'], vitalidade['media']),
    ctrl.Rule(sol['pequeno'] & agua['media'], vitalidade['boa']),
    ctrl.Rule(sol['pequeno'] & agua['grande'], vitalidade['ruim']),
    ctrl.Rule(sol['medio'] & agua['pequena'], vitalidade['media']),
    ctrl.Rule(sol['medio'] & agua['media'], vitalidade['boa']),
    ctrl.Rule(sol['medio'] & agua['grande'], vitalidade['ruim']),
    ctrl.Rule(sol['grande'] & agua['pequena'], vitalidade['ruim']),
    ctrl.Rule(sol['grande'] & agua['media'], vitalidade['media']),
    ctrl.Rule(sol['grande'] & agua['grande'], vitalidade['ruim']),
]

regras_texto = [
    "SE sol é pequeno E água é pequena ENTÃO vitalidade é media",
    "SE sol é pequeno E água é media ENTÃO vitalidade é boa",
    "SE sol é pequeno E água é grande ENTÃO vitalidade é ruim",
    "SE sol é medio E água é pequena ENTÃO vitalidade é media",
    "SE sol é medio E água é media ENTÃO vitalidade é boa",
    "SE sol é medio E água é grande ENTÃO vitalidade é ruim",
    "SE sol é grande E água é pequena ENTÃO vitalidade é ruim",
    "SE sol é grande E água é media ENTÃO vitalidade é media",
    "SE sol é grande E água é grande ENTÃO vitalidade é ruim",
]

with open(os.path.join(RESULT_DIR, 'exA_regras.txt'), 'w', encoding='utf-8') as f:
    f.write('\n'.join(regras_texto))

sistema_ctrl = ctrl.ControlSystem(regras)
simulador = ctrl.ControlSystemSimulation(sistema_ctrl)

simulador.input['agua'] = 45
simulador.input['sol'] = 50
simulador.compute()
resultado = simulador.output['vitalidade']

print(f"Resultado (Vitalidade) para Água=45ml e Sol=50min: {resultado:.3f}")

with open(os.path.join(RESULT_DIR, 'exA_resultado.txt'), 'w', encoding='utf-8') as f:
    f.write(f"Entradas: Quantidade de Água = 45 ml, Exposição ao Sol = 50 minutos\n")
    f.write(f"Valor da Vitalidade (Defuzzificado): {resultado:.3f}\n")

vitalidade.view(sim=simulador)
plt.savefig(os.path.join(RESULT_DIR, 'exA_vitalidade_resultado_grafico.png'))
plt.close()
print("--- Problema A Finalizado ---")

# --- PROBLEMA B: POLÍTICA DE CRÉDITO (VERSÃO CORRIGIDA) ---
def solve_problem_B():
    print("\n--- Iniciando Problema B: Política de Crédito ---")

    score_mercado = ctrl.Antecedent(np.arange(0, 1001, 1), 'score_mercado')
    score_interno = ctrl.Antecedent(np.arange(0, 1001, 1), 'score_interno')
    engajamento = ctrl.Antecedent(np.arange(0, 4501, 1), 'engajamento')
    risco = ctrl.Consequent(np.arange(0, 1001, 1), 'politica_de_risco')
```

## 2º Trabalho Prático sobre Lógica Fuzzy

```
def define_ratings(variable):
    variable['RATING_1'] = fuzz.trimf(variable.universe, [-150, 0, 200])
    variable['RATING_2'] = fuzz.trimf(variable.universe, [100, 250, 400])
    variable['RATING_3'] = fuzz.trimf(variable.universe, [300, 450, 600])
    variable['RATING_4'] = fuzz.trimf(variable.universe, [500, 650, 800])
    variable['RATING_5'] = fuzz.trimf(variable.universe, [700, 850, 1000])
    variable['RATING_6'] = fuzz.trimf(variable.universe, [900, 1000, 1100])

define_ratings(score_mercado)
define_ratings(score_interno)

engajamento['baixo'] = fuzz.trimf(engajamento.universe, [0, 0, 100])
engajamento['medio'] = fuzz.trimf(engajamento.universe, [0, 100, 200])
engajamento['alto'] = fuzz.trimf(engajamento.universe, [150, 1000, 4500])

risco['grau_1'] = fuzz.trimf(risco.universe, [850, 950, 1000])
risco['grau_2'] = fuzz.trimf(risco.universe, [750, 850, 950])
risco['grau_3'] = fuzz.trimf(risco.universe, [650, 750, 850])
risco['grau_4'] = fuzz.trimf(risco.universe, [300, 500, 700])
risco['grau_5'] = fuzz.trimf(risco.universe, [0, 250, 300])

save_variable_plot(score_mercado, os.path.join(RESULT_DIR, 'exB_score_mercado_mf.png'), 'Score de Mercado')
save_variable_plot(score_interno, os.path.join(RESULT_DIR, 'exB_score_interno_mf.png'), 'Score Interno
(Motorista)')
save_variable_plot(engajamento, os.path.join(RESULT_DIR, 'exB_engajamento_mf.png'), 'Engajamento')
save_variable_plot(risco, os.path.join(RESULT_DIR, 'exB_risco_mf.png'), 'Política de Risco')
print("Gráficos das funções de pertinência do Problema B salvos.")

# Criando as condições de rating para CADA variável de score
cond_motorista_6_3 = score_interno['RATING_6'] | score_interno['RATING_5'] | score_interno['RATING_4'] |
score_interno['RATING_3']
cond_mercado_6_3 = score_mercado['RATING_6'] | score_mercado['RATING_5'] | score_mercado['RATING_4'] |
score_mercado['RATING_3']

cond_motorista_1_2 = score_interno['RATING_1'] | score_interno['RATING_2']
cond_mercado_1_2 = score_mercado['RATING_1'] | score_mercado['RATING_2']

cond_motorista_6_4 = score_interno['RATING_6'] | score_interno['RATING_5'] | score_interno['RATING_4']
cond_mercado_6_4 = score_mercado['RATING_6'] | score_mercado['RATING_5'] | score_mercado['RATING_4']

cond_motorista_1_3 = score_interno['RATING_1'] | score_interno['RATING_2'] | score_interno['RATING_3']
cond_mercado_1_3 = score_mercado['RATING_1'] | score_mercado['RATING_2'] | score_mercado['RATING_3']

cond_motorista_4_5 = score_interno['RATING_4'] | score_interno['RATING_5']
cond_mercado_4_5 = score_mercado['RATING_4'] | score_mercado['RATING_5']

# Aplicando as regras com as condições para AMBAS as variáveis de score
regras = [
    ctrl.Rule(engajamento['baixo'] & cond_motorista_6_3 & cond_mercado_6_3, risco['grau_5']),
    ctrl.Rule(engajamento['baixo'] & cond_motorista_1_2 & cond_mercado_1_2, risco['grau_4']),
    ctrl.Rule(engajamento['medio'] & cond_motorista_6_4 & cond_mercado_6_4, risco['grau_5']),
    ctrl.Rule(engajamento['medio'] & cond_motorista_1_3 & cond_mercado_1_3, risco['grau_3']),
    ctrl.Rule(engajamento['alto'] & score_interno['RATING_6'] & score_mercado['RATING_6'],
risco['grau_4']),
    ctrl.Rule(engajamento['alto'] & cond_motorista_4_5 & cond_mercado_4_5, risco['grau_3']),
    ctrl.Rule(engajamento['alto'] & score_interno['RATING_3'] & score_mercado['RATING_3'],
risco['grau_2']),
    ctrl.Rule(engajamento['alto'] & cond_motorista_1_2 & cond_mercado_1_2, risco['grau_1'])
]
```

## 2º Trabalho Prático sobre Lógica Fuzzy

```
]

regras_texto = [
    "SE engajamento Baixo E Ratings em {3,4,5,6} ENTÃO Risco Grau 5",
    "SE engajamento Baixo E Ratings em {1,2} ENTÃO Risco Grau 4",
    "SE engajamento Médio E Ratings em {4,5,6} ENTÃO Risco Grau 5",
    "SE engajamento Médio E Ratings em {1,2,3} ENTÃO Risco Grau 3",
    "SE engajamento Alto E Ratings são 6 ENTÃO Risco Grau 4",
    "SE engajamento Alto E Ratings em {4,5} ENTÃO Risco Grau 3",
    "SE engajamento Alto E Ratings são 3 ENTÃO Risco Grau 2",
    "SE engajamento Alto E Ratings em {1,2} ENTÃO Risco Grau 1"
]

with open(os.path.join(RESULT_DIR, 'exB_regras.txt'), 'w', encoding='utf-8') as f:
    f.write('\n'.join(regras_texto))

sistema_ctrl = ctrl.ControlSystem(regras)
simulador = ctrl.ControlSystemSimulation(sistema_ctrl)

simulador.input['score_mercado'] = 2
simulador.input['score_interno'] = 230
simulador.input['engajamento'] = 90
simulador.compute()
resultado = simulador.output['politica_de_risco']

print(f"Resultado (Risco) para Score Mercado=2, Score Interno=230, Engajamento=90: {resultado:.2f}")

with open(os.path.join(RESULT_DIR, 'exB_resultado.txt'), 'w', encoding='utf-8') as f:
    f.write("Entradas: score_mercado = 2, score_interno = 230, engajamento = 90\n")
    f.write(f"Política de Risco (Defuzzificado): {resultado:.2f}\n")

risco.view(sim=simulador)
plt.savefig(os.path.join(RESULT_DIR, 'exB_risco_resultado_grafico.png'))
plt.close()
print("--- Problema B Finalizado ---")

# --- PROBLEMA C: PROBLEMA DA GORJETA ---
def solve_problem_C():
    print("\n--- Iniciando Problema C: Problema da Gorjeta ---")

    comida = ctrl.Antecedent(np.arange(0, 11, 1), 'qualidade_comida')
    servico = ctrl.Antecedent(np.arange(0, 11, 1), 'qualidade_servico')
    gorjeta = ctrl.Consequent(np.arange(0, 26, 1), 'valor_gorjeta')

    comida['Ruim'] = fuzz.trimf(comida.universe, [0, 0, 5])
    comida['Decente'] = fuzz.trimf(comida.universe, [0, 5, 10])
    comida['Excelente'] = fuzz.trimf(comida.universe, [5, 10, 10])

    servico['Pobre'] = fuzz.trimf(servico.universe, [0, 0, 5])
    servico['Aceitável'] = fuzz.trimf(servico.universe, [0, 5, 10])
    servico['Incrível'] = fuzz.trimf(servico.universe, [5, 10, 10])

    gorjeta['Baixa'] = fuzz.trimf(gorjeta.universe, [0, 0, 12.5])
    gorjeta['Média'] = fuzz.trimf(gorjeta.universe, [0, 12.5, 25])
    gorjeta['Alta'] = fuzz.trimf(gorjeta.universe, [12.5, 25, 25])

    save_variable_plot(comida, os.path.join(RESULT_DIR, 'exC_comida_mf.png'), 'Qualidade da Comida')
    save_variable_plot(servico, os.path.join(RESULT_DIR, 'exC_servico_mf.png'), 'Qualidade do Serviço')
    save_variable_plot(gorjeta, os.path.join(RESULT_DIR, 'exC_gorjeta_mf.png'), 'Valor da Gorjeta')
```

## 2º Trabalho Prático sobre Lógica Fuzzy

```
print("Gráficos das funções de pertinência do Problema C salvos.")

regras_texto = [
    "1. Se a comida é Ruim OU o serviço é Pobre, então a gorjeta será Baixa",
    "2. Se o serviço for Aceitável, a gorjeta será Média",
    "3. Se a comida é Excelente OU o serviço é Incrível, então a gorjeta será Alta"
]

regra1 = ctrl.Rule(comida['Ruim'] | servico['Pobre'], gorjeta['Baixa'])
regra2 = ctrl.Rule(servico['Aceitável'], gorjeta['Média'])
regra3 = ctrl.Rule(comida['Excelente'] | servico['Incrível'], gorjeta['Alta'])

with open(os.path.join(RESULT_DIR, 'exC_regras.txt'), 'w', encoding='utf-8') as f:
    f.write('\n'.join(regras_texto))

sistema_ctrl = ctrl.ControlSystem([regra1, regra2, regra3])
simulador = ctrl.ControlSystemSimulation(sistema_ctrl)

simulador.input['qualidade_comida'] = 6.5
simulador.input['qualidade_servico'] = 9.8
simulador.compute()
resultado = simulador.output['valor_gorjeta']

print(f"Resultado (Gorjeta) para Comida=6.5 e Serviço=9.8: {resultado:.2f}")

with open(os.path.join(RESULT_DIR, 'exC_resultado.txt'), 'w', encoding='utf-8') as f:
    f.write(f"Entradas: Qualidade da Comida = 6.5, Qualidade do Serviço = 9.8\n")
    f.write(f"Valor da Gorjeta (Defuzzificado): {resultado:.2f}\n")

gorjeta.view(sim=simulador)
plt.savefig(os.path.join(RESULT_DIR, 'exC_gorjeta_resultado_grafico.png'))
plt.close()
print("--- Problema C Finalizado ---")

# --- PROBLEMA D: CÁLCULO DE PRÊMIO DE SEGURO ---
def solve_problem_D():
    print("\n--- Iniciando Problema D: Prêmio de Seguro ---")

    idade = ctrl.Antecedent(np.arange(15, 76, 1), 'idade')
    saude = ctrl.Antecedent(np.arange(0, 1.01, 0.01), 'estado_saude')
    premio = ctrl.Consequent(np.arange(0, 101, 1), 'premio')

    idade['Muito jovem'] = fuzz.trimf(idade.universe, [15, 20, 30])
    idade['Jovem'] = fuzz.trimf(idade.universe, [20, 30, 45])
    idade['Idade média'] = fuzz.trimf(idade.universe, [30, 45, 60])
    idade['Maduro'] = fuzz.trimf(idade.universe, [45, 60, 70])
    idade['Idoso'] = fuzz.trimf(idade.universe, [60, 70, 75])

    saude['Muito péssimo'] = fuzz.trimf(saude.universe, [0, 0, 0.25])
    saude['Péssimo'] = fuzz.trimf(saude.universe, [0, 0.25, 0.5])
    saude['Médio'] = fuzz.trimf(saude.universe, [0.25, 0.5, 0.75])
    saude['Bom'] = fuzz.trimf(saude.universe, [0.5, 0.75, 1.0])
    saude['Muito bom'] = fuzz.trimf(saude.universe, [0.75, 1.0, 1.0])

    premio['Muito baixo'] = fuzz.trimf(premio.universe, [0, 0, 20])
    premio['Baixo'] = fuzz.trimf(premio.universe, [0, 20, 40])
    premio['Moderadamente baixo'] = fuzz.trimf(premio.universe, [20, 40, 60])
    premio['Moderado'] = fuzz.trimf(premio.universe, [40, 55, 70])
    premio['Moderadamente alto'] = fuzz.trimf(premio.universe, [60, 70, 80])
```

## 2º Trabalho Prático sobre Lógica Fuzzy

```
premio['Alto'] = fuzz.trimf(premio.universe, [70, 80, 100])
premio['Muito alto'] = fuzz.trimf(premio.universe, [80, 100, 100])

save_variable_plot(idade, os.path.join(RESULT_DIR, 'exD_idade_mf.png'), 'Idade')
save_variable_plot(saude, os.path.join(RESULT_DIR, 'exD_saude_mf.png'), 'Estado de Saúde')
save_variable_plot(premio, os.path.join(RESULT_DIR, 'exD_premio_mf.png'), 'Prêmio')
print("Gráficos das funções de pertinência do Problema D salvos.")

mapa_regras = {
    'Muito péssimo': ['Moderado', 'Moderadamente alto', 'Moderadamente alto', 'Alto', 'Muito alto'],
    'Péssimo': ['Moderadamente baixo', 'Moderado', 'Moderadamente alto', 'Moderadamente alto',
'Moderadamente alto'],
    'Médio': ['Moderadamente baixo', 'Moderadamente baixo', 'Moderado', 'Moderadamente alto',
'Moderadamente alto'],
    'Bom': ['Baixo', 'Moderadamente baixo', 'Moderadamente baixo', 'Moderado', 'Alto'],
    'Muito bom': ['Muito baixo', 'Baixo', 'Moderadamente baixo', 'Moderadamente baixo', 'Moderado']
}

regras = []
regras_texto = []
idade_terms = ['Muito jovem', 'Jovem', 'Idade média', 'Maduro', 'Idoso']
saude_terms = ['Muito péssimo', 'Péssimo', 'Médio', 'Bom', 'Muito bom']

for s_term in saude_terms:
    for i, i_term in enumerate(idade_terms):
        p_term = mapa_regras[s_term][i]
        regra = ctrl.Rule(idade[i_term] & saude[s_term], premio[p_term])
        regras.append(regra)
        regras_texto.append(f"SE idade é '{i_term}' E saúde é '{s_term}' ENTÃO prêmio é '{p_term}'")

with open(os.path.join(RESULT_DIR, 'exD_regras.txt'), 'w', encoding='utf-8') as f:
    f.write('\n'.join(regras_texto))

sistema_ctrl = ctrl.ControlSystem(regras)
simulador = ctrl.ControlSystemSimulation(sistema_ctrl)

simulador.input['idade'] = 32
simulador.input['estado_saude'] = 0.7
simulador.compute()
resultado = simulador.output['premio']

print(f"Resultado (Prêmio) para Idade=32 e Saúde=0.7: {resultado:.2f}")

with open(os.path.join(RESULT_DIR, 'exD_resultado.txt'), 'w', encoding='utf-8') as f:
    f.write(f"Entradas: Idade = 32 anos, Estado de Saúde = 0.7\n")
    f.write(f"Valor do Prêmio (Defuzzificado): {resultado:.2f}\n")

premio.view(sim=simulador)
plt.savefig(os.path.join(RESULT_DIR, 'exD_premio_resultado_grafico.png'))
plt.close()
print("--- Problema D Finalizado ---")

# --- EXECUÇÃO PRINCIPAL ---
if __name__ == "__main__":
    solve_problem_A()
    solve_problem_B()
    solve_problem_C()
```

## 2º Trabalho Prático sobre Lógica Fuzzy

```
solve_problem_D()  
print("\nTodos os exercícios foram processados e os resultados foram salvos em './result'.")
```