

Pós Tech - IA para Devs

Alunos

Rafael Alves Cardoso - RM 360124

Silvio Cezer Saczuck - RM 360204

Luciano Giles Soares - RM 359834

William Judice Yizima - RM 360214

Luiz Ricardo Zinsly Calmon - RM359894

GitHub do TCC2: <https://github.com/WilliamYizima/POS-4IADT-2>

Vídeo da apresentação: https://youtu.be/b_O_AEDGMRc

Projeto: Otimização de Alocação de Tarefas com Base em Custo-Benefício

1. Contexto

A empresa possui uma fila de tarefas (tickets), cada uma com um nível de dificuldade (peso), que precisa ser resolvida por um time de profissionais com diferentes níveis de capacidade técnica (capacidade de resolução) e custo-hora. Atualmente, não há um critério claro para distribuição dessas tarefas, o que pode levar a alocações ineficientes, com alto custo ou desperdício de capacidade técnica.

2. Problema

A ausência de um algoritmo que distribua as tarefas de forma otimizada entre os profissionais tem causado:

- Alocações ineficientes de recursos humanos.
 - Aumento de custos operacionais (utilização de profissionais mais caros sem necessidade).
 - Tempo de resolução dos chamados maior do que o necessário.
 - Desbalanceamento da carga de trabalho entre os membros da equipe.
-

3. Objetivo

O objetivo principal é **otimizar a alocação das tarefas** com base em:

- Menor tempo total de resolução dos chamados.
- Menor custo possível por tarefa alocada.
- Utilização inteligente da capacidade técnica de cada profissional.

- Equilíbrio na distribuição de tarefas.
-

4. Regras de Negócio

1. **Tempo mínimo cobrado:** Cada tarefa alocada consome **pelo menos uma hora de trabalho**, mesmo que a capacidade do profissional não seja totalmente utilizada.
 2. **Múltiplas tarefas por profissional:** Um mesmo profissional pode receber mais de uma tarefa.
 3. **Custo-benefício:** Dê preferência ao profissional com **menor custo-hora possível**, desde que sua **capacidade de resolução seja compatível** com o peso da tarefa.
 4. **Otimização de tempo e custo:** O algoritmo deve considerar a melhor combinação de alocação que **minimize o tempo total de execução e o custo total do projeto**.
-

5. Proposta de Solução

Desenvolver um **algoritmo Genético de alocação de tarefas**, que analise todas as combinações possíveis e encontre a melhor distribuição das tarefas entre os profissionais, obedecendo às regras de negócio estabelecidas.

O algoritmo deve:

- Para cada tarefa, verificar todos os profissionais cuja capacidade seja **maior ou igual** ao peso da tarefa.
 - Calcular o custo de resolução da tarefa com base no valor hora do profissional e nas horas necessárias (divisão peso / capacidade, arredondando para cima).
 - Escolher o profissional com o **menor custo total de execução da tarefa**.
 - Repetir esse processo até alocar todas as tarefas.
-

6. Critérios de Aceite

- Todas as tarefas devem ser alocadas a pelo menos um profissional.
 - O custo final total deve ser o **menor possível** dentro das regras.
 - O tempo estimado total de resolução das tarefas deve ser o **menor possível** dentro das regras.
 - Nenhum profissional deve receber uma tarefa cujo peso exceda sua capacidade de resolução.
-

7. Exemplo Ilustrativo

Dados de Entrada:

Tarefas

Nome	Peso
ticket_9	30

Profissionais

Nome	Capacidade	Valor Hora
Lucio	6	R\$ 98
Ana	11	R\$ 131
Gabriel	11	R\$ 131
Helena	17	R\$ 137
Igor	22	R\$ 182

Análise da tarefa ticket_9:

- **Lucio:** $30 / 6 = 5h \rightarrow \text{Custo} = 5 \times R\$98 = R\$490$
- **Ana:** $30 / 11 = 2.73 \rightarrow 3h \rightarrow \text{Custo} = 3 \times R\$131 = R\$393$
- **Gabriel:** Mesmo custo da Ana = R\$393
- **Helena:** $30 / 17 = 1.76 \rightarrow 2h \rightarrow \text{Custo} = 2 \times R\$137 = \mathbf{R\$274}$
- **Igor:** $30 / 22 = 1.36 \rightarrow 2h \rightarrow \text{Custo} = 2 \times R\$182 = R\$364$

✓ Melhor opção: **Helena**, custo total R\$274.

8. Demonstração prática da solução

Executando o algoritmo Genético

```

print("\nCarga de trabalho por profissional:")
print(df_carga.to_string(index=False))

# Visualizações
print("\nCriando visualizações gráficas...")
visualizar_evolucao(historico_fitness)
visualizar_pareto(historico_populacoes[-1], historico_fitness[-1], hof)
visualizar_solucao(solucao_equilibrada, "Solução Equilibrada")

# Comparar soluções extremas
print("\nComparando soluções extremas:")
comparar_solucoes(solucao_menor_custo, solucao_menor_tempo, "Menor Custo", "Menor Tempo")

```

↳ Executando algoritmo genético para alocação de tarefas...

```

-- Geração 1 --
Min: [2650.    29.4]
Avg: [5190.20666667  61.81703333]
-- Geração 2 --
Min: [2490.    28.2]
Avg: [5206.9325   62.06195]
-- Geração 3 --
Min: [2440.    27.5]
Avg: [5225.69916667  62.36208333]
-- Geração 4 --
Min: [2430.    27.3]
Avg: [5236.75583333  62.58951667]
-- Geração 5 --
Min: [2430.    27.3]
Avg: [5269.20583333  63.10591667]
-- Geração 6 --
Min: [2420.    27.2]
Avg: [5288.02416667  63.43948333]
-- Geração 7 --
Min: [2420.    27.2]
Avg: [5303.6775   63.71175]
-- Geração 8 --
Min: [2390.    26.9]
Avg: [5311.98166667  63.85033333]

```

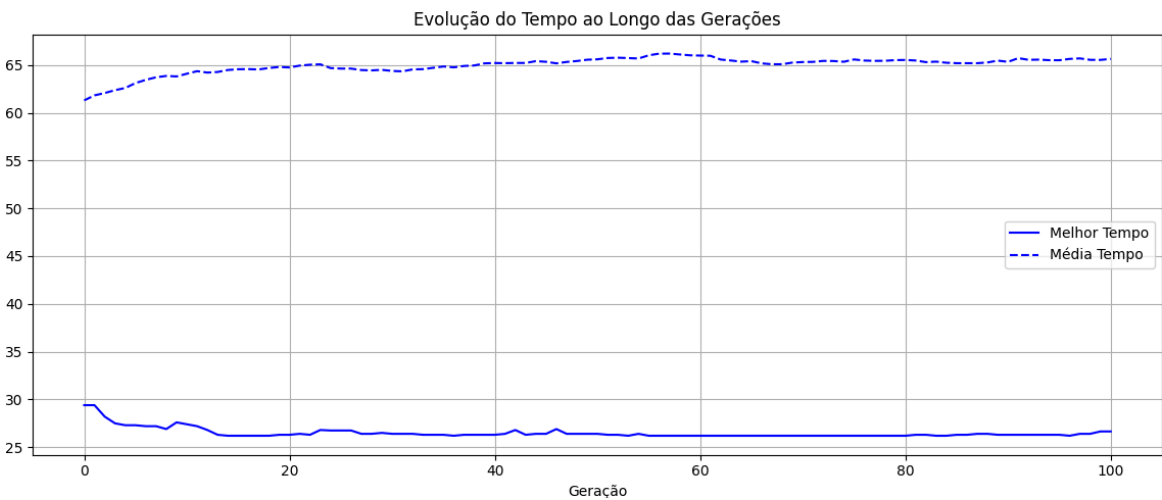
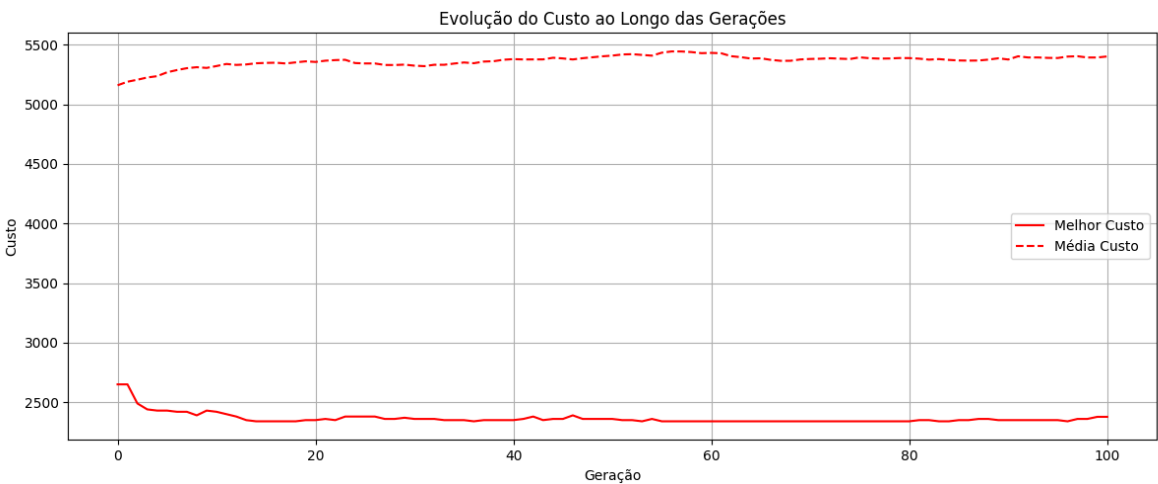
Solução com melhor equilíbrio entre custo e tempo:
Tempo total do processo: 4.90 horas
Custo total: R\$ 1960

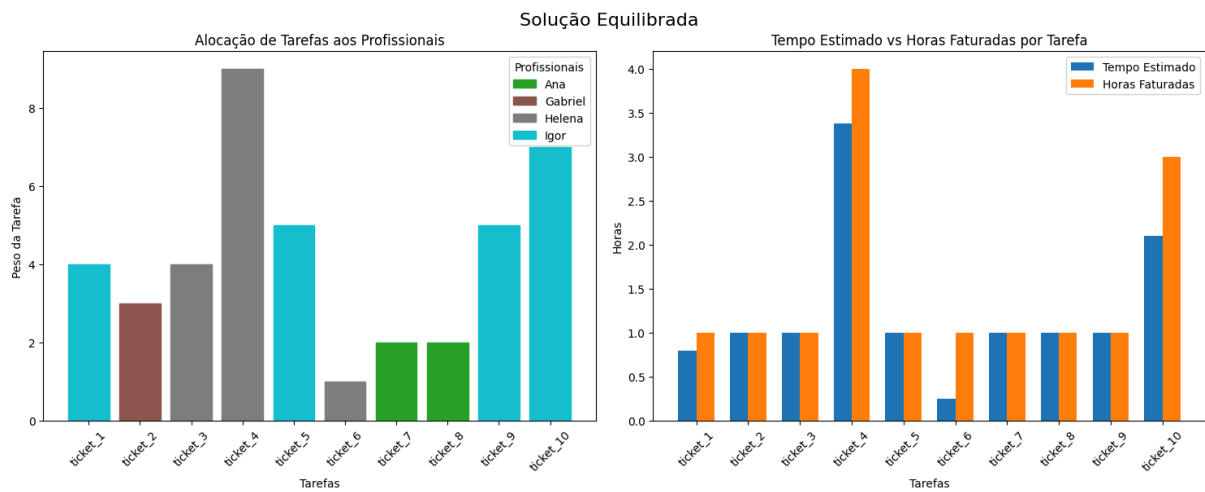
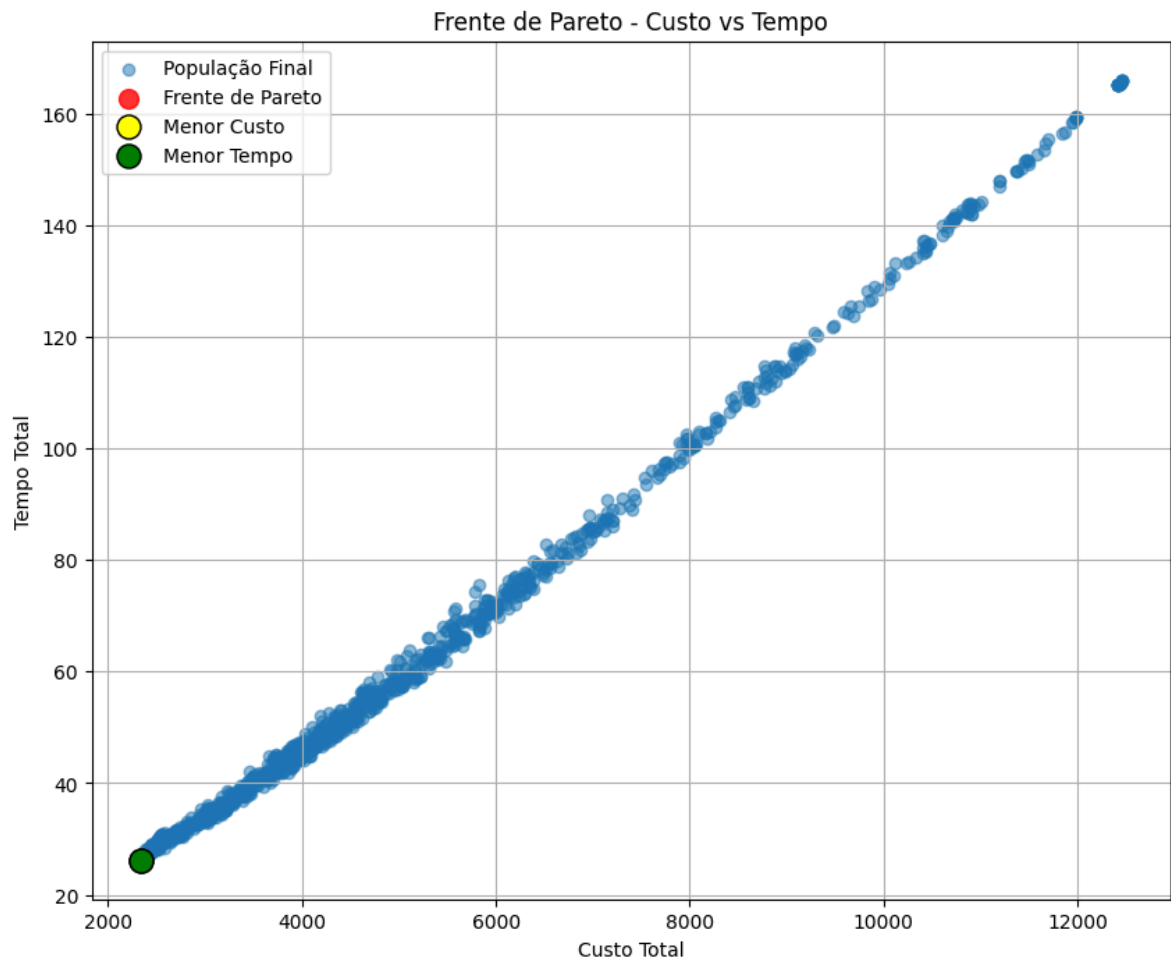
Alocação de tarefas:

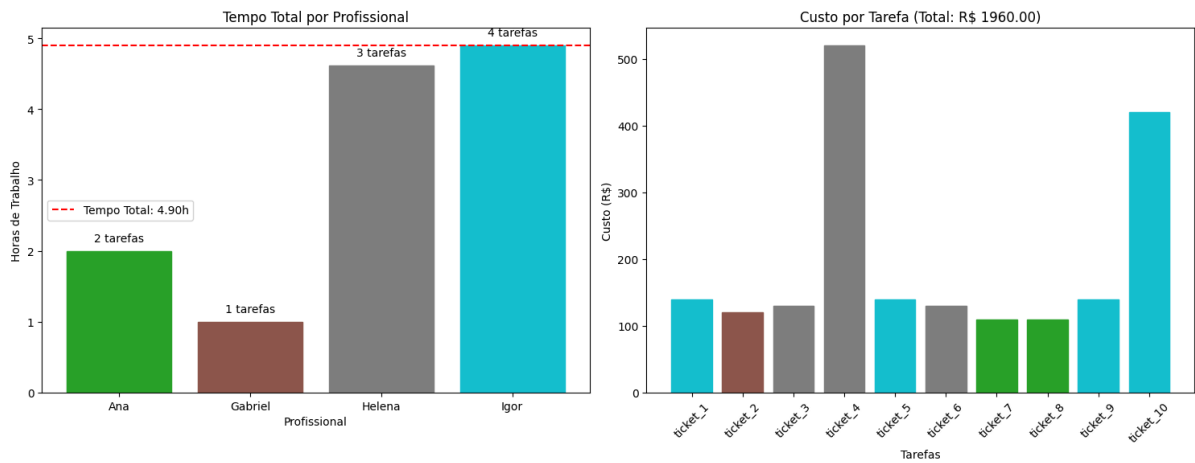
tarefa	peso_tarefa	profissional	capacidade	tempo_estimado	horas_faturadas	custo
ticket_1	4	Igor	5	0.80	1	140
ticket_2	3	Gabriel	3	1.00	1	120
ticket_3	4	Helena	4	1.00	1	130
ticket_4	9	Helena	4	3.38	4	520
ticket_5	5	Igor	5	1.00	1	140
ticket_6	1	Helena	4	0.25	1	130
ticket_7	2	Ana	2	1.00	1	110
ticket_8	2	Ana	2	1.00	1	110
ticket_9	5	Igor	5	1.00	1	140
ticket_10	7	Igor	5	2.10	3	420

Carga de trabalho por profissional:

profissional	tempo_total	tarefas
Ana	2.00	2
Gabriel	1.00	1
Helena	4.62	3
Igor	4.90	4







Comparando soluções extremas:

Comparação de soluções:

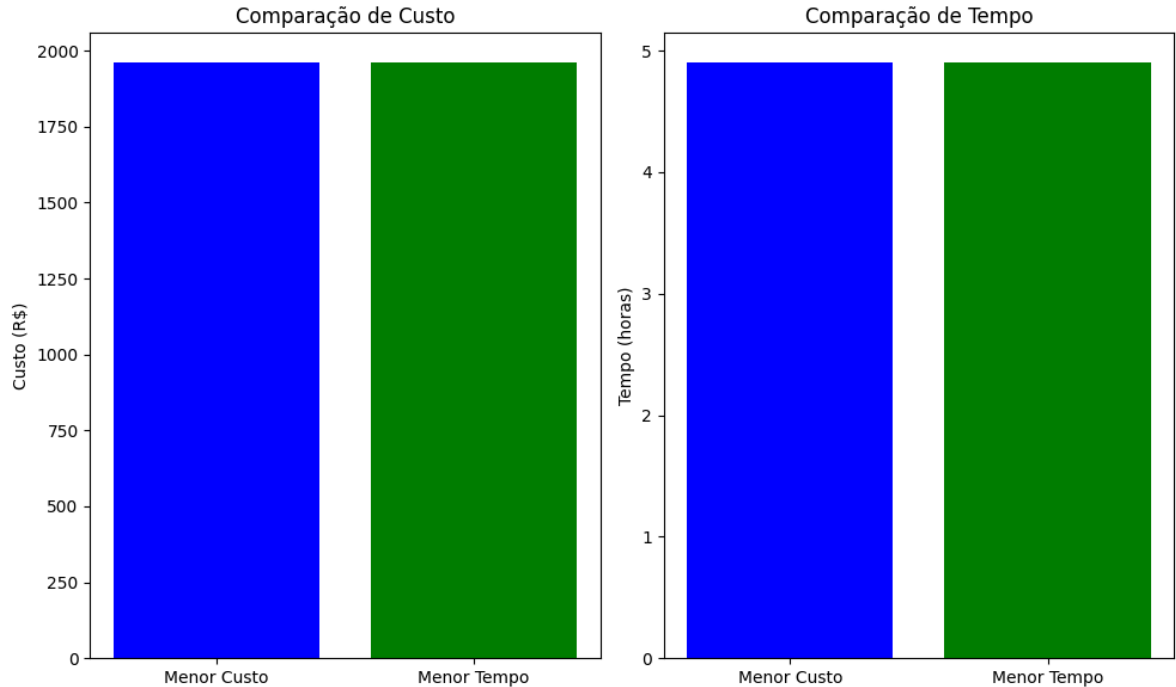
Menor Custo: Tempo = 4.9 horas, Custo = R\$ 1960

Menor Tempo: Tempo = 4.9 horas, Custo = R\$ 1960

Diferença:

Tempo: 0.0 horas

Custo: R\$ 0



	tarafa	peso_tarafa	profissional	capacidade	tempo_estimado	\
0	ticket_1	4	Igor	5	0.80	
1	ticket_2	3	Gabriel	3	1.00	
2	ticket_3	4	Helena	4	1.00	
3	ticket_4	9	Helena	4	3.38	
4	ticket_5	5	Igor	5	1.00	
5	ticket_6	1	Helena	4	0.25	
6	ticket_7	2	Ana	2	1.00	
7	ticket_8	2	Ana	2	1.00	
8	ticket_9	5	Igor	5	1.00	
9	ticket_10	7	Igor	5	2.10	

	horas_faturadas	custo
0	1	140
1	1	120
2	1	130
3	4	520
4	1	140
5	1	130
6	1	110
7	1	110
8	1	140
9	3	420

	tarafa	peso_tarafa	profissional	capacidade	tempo_estimado	\
0	ticket_1	4	Igor	5	0.80	
1	ticket_2	3	Gabriel	3	1.00	
2	ticket_3	4	Helena	4	1.00	
3	ticket_4	9	Helena	4	3.38	
4	ticket_5	5	Igor	5	1.00	
5	ticket_6	1	Helena	4	0.25	
6	ticket_7	2	Ana	2	1.00	
7	ticket_8	2	Ana	2	1.00	
8	ticket_9	5	Igor	5	1.00	
9	ticket_10	7	Igor	5	2.10	

	horas_faturadas	custo
0	1	140
1	1	120
2	1	130
3	4	520
4	1	140
5	1	130
6	1	110
7	1	110
8	1	140
9	3	420

4.8999999999999995,
4.8999999999999995,
np.int64(1960),
np.int64(1960))

9. Considerações Finais

Essa otimização permite tomadas de decisão mais estratégicas na distribuição das tarefas, promovendo economia de custos e melhor aproveitamento da equipe. Com base nos dados, será possível gerar uma **ordem de alocação inteligente**, permitindo ajustes conforme novos tickets e profissionais sejam adicionados.

