

# Uso de Metaheurística em uma Ferramenta de Cotação para Compras de Cartas de *Magic: The Gathering*

Luiz Eduardo Pereira



Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus Formiga – IFMG

29 de novembro de 2019

# Conteúdo

- 1 Introdução
- 2 Fundamentação Teórica
- 3 Materiais e Métodos
- 4 Projeto e Desenvolvimento
- 5 Resultados
- 6 Considerações Finais

- 1 Introdução
- 2 Fundamentação Teórica
- 3 Materiais e Métodos
- 4 Projeto e Desenvolvimento
- 5 Resultados
- 6 Considerações Finais

# Introdução

*Magic: The Gathering* é um *Trading Card Game* criado pela Wizards of the Coast em 1993. O objetivo do jogador é reduzir a vida do oponente a zero. Os jogadores podem utilizar qualquer carta desde o lançamento do jogo para construir um *deck*.

Figura 1 - Carta de *Magic*.



Fonte - Wizards of the Coast (2019)

*Magic* conta com mais de **20.000** cartas diferentes. Todo ano são lançadas novas coleções. Para que os jogadores tenham acesso a tantas cartas, existem **lojas especializadas no jogo**.

Essas lojas **não conseguem manter estoque** de todas as cartas, devido a isso, os jogadores geralmente precisam **comprar em lojas virtuais** fora de suas cidades.

A LigaMagic é um portal que reúne os **preços** de cartas de várias **lojas brasileiras**.

- Lojas **não conseguem** manter estoque de todas as cartas;
- Comprar online exige pagamento de **frete**;
- Encontrar a combinação de lojas que possua valor **ótimo manualmente** é muito **difícil e demorado**.

# Objetivos

## Objetivo Geral

Desenvolver um protótipo de um sistema *Web* que realiza **cotações de preço** para cartas de *Magic: The Gathering*, buscando através de metaheurística, **minimizar o valor final da compra**, sugerindo **combinações de lojas** para o usuário.

## Objetivos Específicos

- Implementar uma *Single Page Application*;
- Utilizar *web crawler* para **extrair** dados a partir do site LigaMagic;
- Gerar opções de compra utilizando a metaheurística **Simulated Annealing** e técnicas de **pós-otimização**;
- Notificar o usuário via **e-mail** com os resultados da cotação;

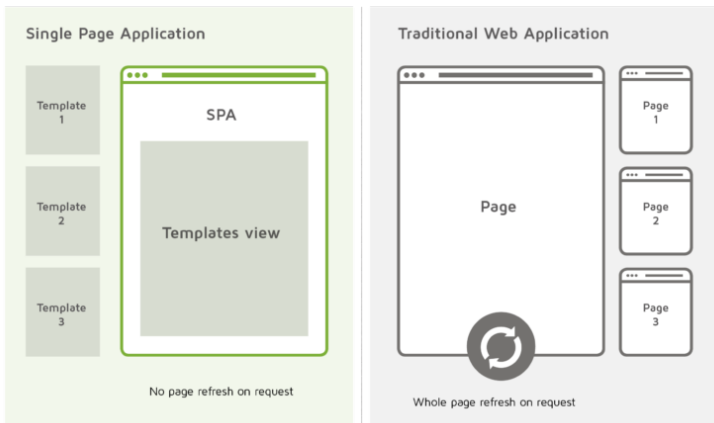
# Conteúdo

- 1 Introdução
- 2 Fundamentação Teórica**
- 3 Materiais e Métodos
- 4 Projeto e Desenvolvimento
- 5 Resultados
- 6 Considerações Finais



# Single Page Application

Figura 2 - Single Page Application.



Fonte - GearHeart (2019)

Figura 3 - Usos de *web crawler*.



Fonte - APIFY (2019)

## Algoritmo 1 - *Simulated Annealing* Multiobjetivo.

---

```
Gere uma solução inicial x
M := Ø
Atualize o conjunto M das potenciais soluções eficientes com x
C := Número de cadeias de Markov
repeat
  T := Temperatura inicial
  repeat
    Gere uma solução y a partir da vizinhança de x
    if y não é dominado por x then
      Atualize o conjunto M das potenciais soluções eficientes com y
      x := y com probabilidade  $P(x, y, T, \Lambda)$ 
    end if
    Diminui T utilizando algum esquema de resfriamento
  until T = Temperatura final
  C := C - 1
until C = 0
```

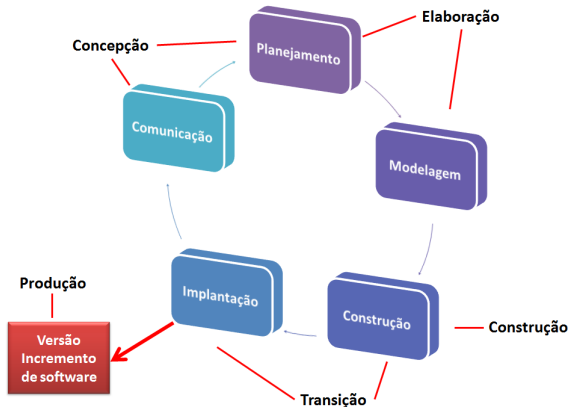
---

Fonte - Czyzak e Jaszkievicz (1998)

# Conteúdo

- 1 Introdução
- 2 Fundamentação Teórica
- 3 Materiais e Métodos**
- 4 Projeto e Desenvolvimento
- 5 Resultados
- 6 Considerações Finais

Figura 4 - Processo Unificado



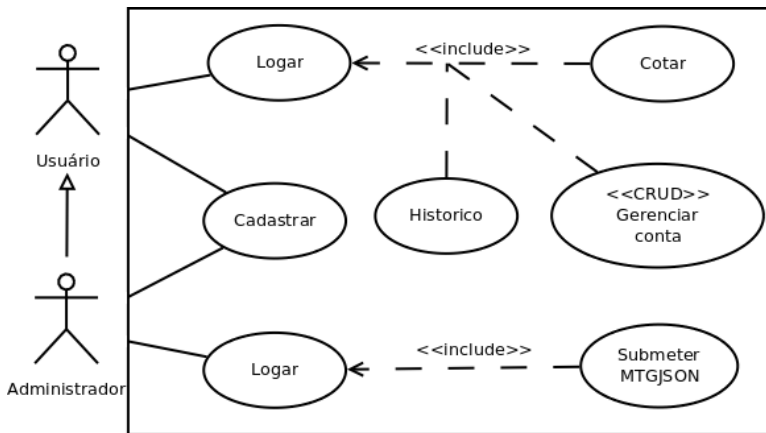
Fonte - Pressman e Maxim (2016)



# Conteúdo

- 1 Introdução
- 2 Fundamentação Teórica
- 3 Materiais e Métodos
- 4 Projeto e Desenvolvimento**
- 5 Resultados
- 6 Considerações Finais

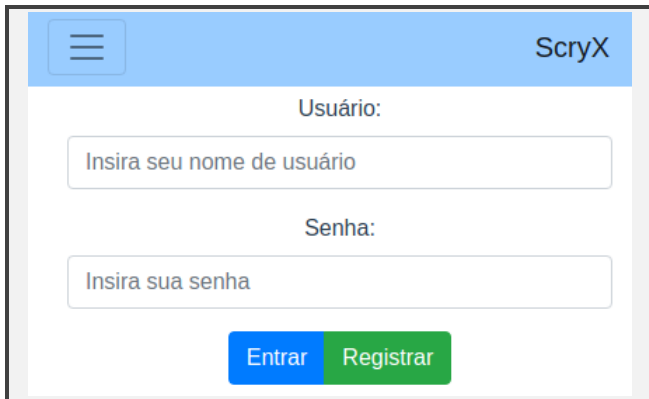
Figura 5 - Diagrama de casos de uso.



Fonte - Elaborado pelo autor.



Figura 6 - Interface de login.



The image shows a login interface for a system named "ScryX". At the top, there is a light blue header bar. On the left side of this bar is a hamburger menu icon (three horizontal lines). On the right side is the text "ScryX". Below the header, the interface is white. It starts with the label "Usuário:" followed by a text input field containing the placeholder text "Insira seu nome de usuário". Below this is the label "Senha:" followed by another text input field containing the placeholder text "Insira sua senha". At the bottom, there are two buttons: a blue button labeled "Entrar" and a green button labeled "Registrar".

Fonte - Elaborado pelo autor.

Figura 7 - Interface do perfil - Administrador.

The screenshot shows a web interface for an administrator profile. At the top, there is a light blue header with a hamburger menu icon on the left and the text 'ScryX' on the right. Below the header, there is a section titled 'Submeter lista de nomes de cartas - MTGJson'. This section contains a large text input field with the placeholder 'Arraste o arquivo aqui...', a 'Browse' button, and an 'Enviar' button. Below this section, the title 'Perfil:' is centered. The profile information is displayed in two columns. The left column contains 'Nome: admin', 'Usuário: admin', and 'E-mail: admin@admin'. The right column contains 'Data de Cadastro: 2019-09-17' and 'Data de Nascimento: 2000-01-01'. At the bottom of the profile section, there are three buttons: 'Alterar E-mail' (blue), 'Alterar Senha' (blue), and 'Apagar Conta' (red).

Submeter lista de nomes de cartas - MTGJson

Arraste o arquivo aqui... Browse Enviar

**Perfil:**

Nome: admin	Data de Cadastro: 2019-09-17
Usuário: admin	Data de Nascimento: 2000-01-01
E-mail: admin@admin	Alterar Senha
Alterar E-mail	Apagar Conta

Fonte - Elaborado pelo autor.

Figura 8 - Interface do perfil - Usuário.

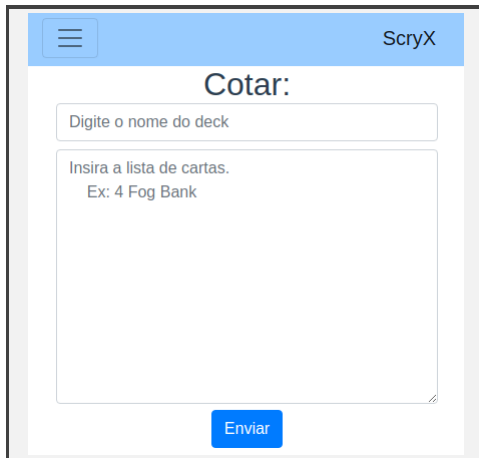


The image shows a user profile interface. At the top, there is a light blue header bar. On the left side of the header is a hamburger menu icon (three horizontal lines inside a square). On the right side of the header is the text 'ScryX'. Below the header, the word 'Perfil:' is centered in a large, bold font. Underneath, the profile information is displayed in two columns. The left column contains 'Nome: José Andrades', 'Usuário: jandrades', and 'E-mail: joseandrades1@gmail.com'. The right column contains 'Data de Cadastro: 2019-10-04' and 'Data de Nascimento: 1990-01-01'. At the bottom of the profile section, there are three buttons: a blue button labeled 'Alterar E-mail' on the left, a blue button labeled 'Alterar Senha' on the right, and a red button labeled 'Apagar Conta' on the right.

Perfil:	
Nome: José Andrades	Data de Cadastro: 2019-10-04
Usuário: jandrades	Data de Nascimento: 1990-01-01
E-mail: joseandrades1@gmail.com	<a href="#">Alterar Senha</a>
<a href="#">Alterar E-mail</a>	<a href="#">Apagar Conta</a>

Fonte - Elaborado pelo autor.

Figura 9 - Interface de cotação.



The image shows a web interface titled 'Cotar' within a light blue header bar. The header bar contains a hamburger menu icon on the left and the text 'ScryX' on the right. Below the header, the title 'Cotar:' is centered. There are two main input areas: a text box with the placeholder 'Digite o nome do deck' and a larger text area with the placeholder 'Insira a lista de cartas.' and an example 'Ex: 4 Fog Bank'. At the bottom center, there is a blue button labeled 'Enviar'.

Fonte - Elaborado pelo autor.

Figura 10 - Interface de histórico - Opções.



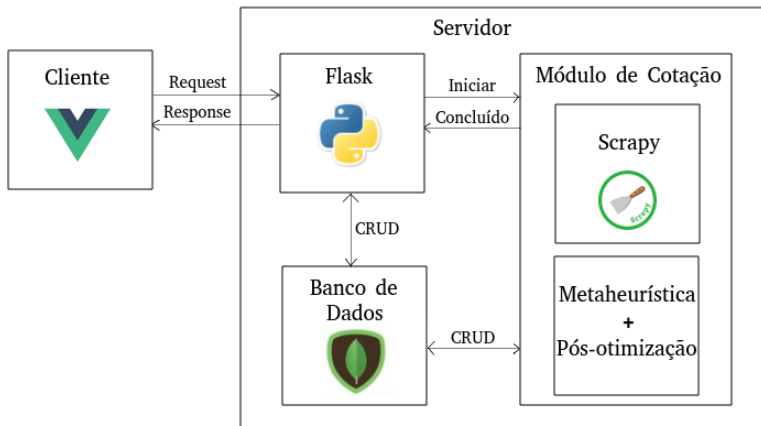
Fonte - Elaborado pelo autor.

Figura 11 - Interface de histórico - Opção expandida.

Opção 1			
Valor Total:	1435.49	Quantidade de lojas:	3
Loja:	Magicbembarato	Valor:	503.78
Nome	Quantidade	Preço Unitário	Preço
absorb	2	5.26	10.52
cast down	3	2.69	8.07
lyra dawnbringer	2	21.56	43.12
negate	2	0.41	0.82
teferi, hero of dominaria	4	102.09	408.36
thief of sanity	1	12.03	12.03
Loja:	Pi Games	Valor:	512.06
Nome	Quantidade	Preço Unitário	Preço


Fonte - Elaborado pelo autor.

Figura 12 - Módulos da aplicação.



Fonte - Elaborado pelo autor.

Figura 13 - Página da LigaMagic.



**Niv-Mizzet, Parun**

Cor: Multi Colorida    Custo de Mana: (CMC: 6)

Tipo: Legendary Creature - Dragon Wizard (5/5)

Formatos Válidos: Standard, Modern, Legacy, Brawl

Edição: Guildas de Ravnicar    Raridade: Rara

Artista: Svetlin Velinov

**R\$ 4,00**    **R\$ 7,74 ▼**    **R\$ 15,00**

Alerta de Preço ?

Menor Preço **R\$ 4,00**    Maior Preço **R\$ 49,90**

Me avise quando preço chegar em

R\$  [Criar Alerta](#)

Utilizar filtros auxiliares

	Guildas de Ravnicar	<del>R\$ 5,00</del> <b>R\$ 4,00</b>	M	4 unids	<input type="text" value="1"/>	<a href="#">Comprar</a>	<a href="#">Ir à loja</a>
	Guildas de Ravnicar	<del>R\$ 5,00</del> <b>R\$ 4,00</b>	M	4 unids	<input type="text" value="1"/>	<a href="#">Comprar</a>	<a href="#">Ir à loja</a>
	Guildas de Ravnicar	<b>R\$ 4,24</b>	NM	1 unid	<input type="text" value="1"/>	<a href="#">Comprar</a>	<a href="#">Ir à loja</a>
	Guildas de Ravnicar	<b>R\$ 4,40</b>	NM	1 unid	<input type="text" value="1"/>	<a href="#">Comprar</a>	<a href="#">Ir à loja</a>

Fonte - LigaMagic (2019)



Figura 14 - Página de uma loja filiada a LigaMagic.

**Brontodonte Destruidor / Thrashing Brontodon**

Edição Idioma Q Extras Estoque Preço

M20	PT	NM	-	4 unid.	<div> <div>RS 1,50</div> <div>RS 1,43</div> </div>	1	+	Comprar
	PTEN	NM	-	4 unid.	<div> <div>RS 1,50</div> <div>RS 1,43</div> </div>	1	+	Comprar
	PTEN	NM	Foil	2 unid.	RS 6,98	1	+	Comprar
M20	PT	NM	Foil	1 unid.	RS 6,98	1	+	Comprar
	PTEN	NM	Foil	0 unid.	RS 6,98	Me avise quando chegar.		
	-	-	-	0 unid.		Me avise quando chegar.		
	-	-	-	0 unid.		Me avise quando chegar.		

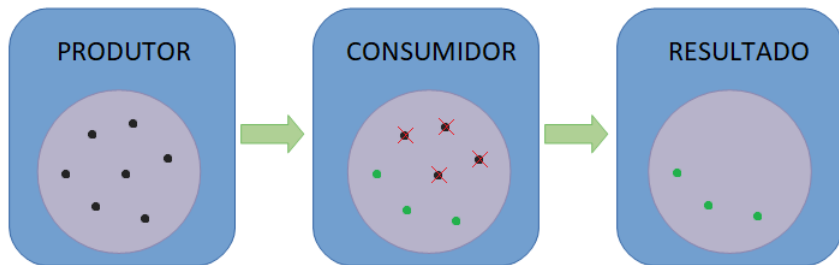
Fonte - LigaMagic (2019)

```
1 /html/body/table/tbody/tr[2]/td/table[3]/tbody/tr/td[2]/table/tbody/tr/td[2]/div[2]/table/tbody/tr/td[2]/div[2]/table/tbody/tr[2]/td[6]/font
```

XPath

# Simulated Annealing

Figura 15 - Funcionamento do Simulated Annealing.



Fonte - Elaborado pelo autor.

- Objetivo 1: Reduzir o preço;
- Objetivo 2: Reduzir o número de lojas;
- Objetivo 3: Reduzir o número de cartas faltantes.

Figura 16 - Estrutura de Dados.

(a) "content\_table"

	L1	L2	L3
C1	(10, 0.10)	(5, 0.2)	(4, 1), (2, 2)
C2	(5, 1), (2, 3)	(4, 0.5)	(4, 0.5)
C3	(4, 0.5)	(5, 1.5)	(5, 0.2)

(b) "cards".

C1	4
C2	2
C1	1

(c) "result\_table".

	L1	L2	L3
C1	3	0	1
C2	0	2	0
C3	1	0	0

Fonte - Elaborado pelo autor.

## Algoritmo 2 - Produtor.

```
1 result_table = gera_solucao_inicial()
2 for i in range(CADEIAS_MARKOV):
3     temperatura_atual = TEMPERATURA_INICIAL
4     while ((temperatura_atual > TEMPERATURA_FINAL)):
5         for card in card_dict.items():
6             new_result_table = swap_change_all(result_table, card)
7             if (random.uniform(0, 1) <= criterio_aceitacao(
8                 result_table, new_result_table, temperatura_atual)):
9                 result_table = new_result_table
10                solutions.append(result_table)
11            if (len(solutions) >= MAXIMO_SOLUCOES):
12                solution_deliver.put(solutions)
13                solutions = []
14        temperatura_atual = resfria_temperatura(temperatura_atual)
```

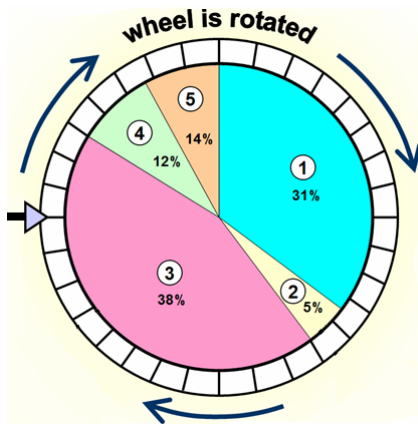
Fonte - Elaborado pelo autor.

Figura 17 - Swap - Troca de elementos na “result\_table”.

Solução atual				
0	0	0	4	...
SWAP – Passo 1				
0	0	0	0	...
SWAP – Passo 2				
0	2	0	0	...
SWAP – Passo 2				
0	2	2	0	...

Fonte - Elaborado pelo autor.

Figura 18 - *Roulette Whell.*



Fonte - Newcastle (2019)

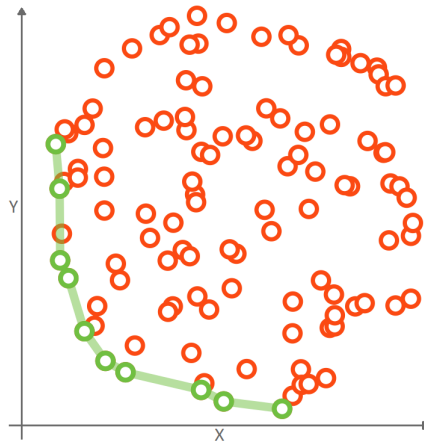
- *Fitness*;
- Critério de aceitação:

$$P(x, y, T, \Lambda) = \min \left\{ 1, \exp \left( \sum_{j=1}^j \lambda_j (f_j(x) - f_j(y) / T) \right) \right\} \quad (1)$$

Onde  $\sum_{i=1}^n \lambda_i = 1$ ,  $n$  é o número de objetivos. A equação retorna a probabilidade de uma **solução gerada ser aceita**.

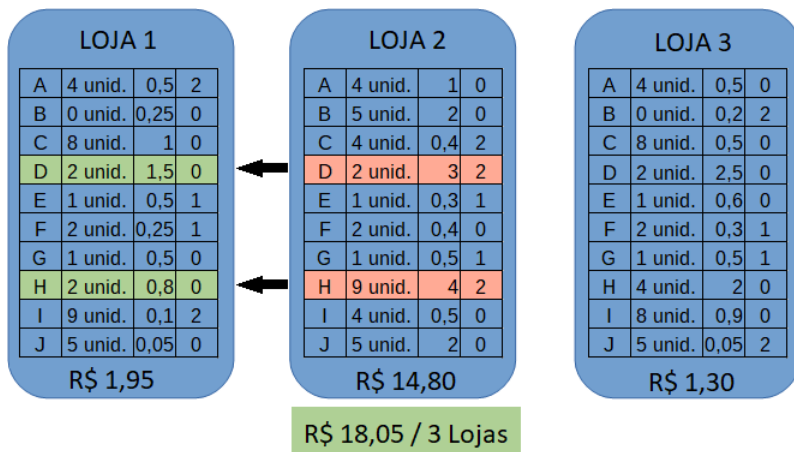


Figura 19 - Fronteira de Pareto.



Fonte - Ormont (2016)

Figura 20 - Fase 1 - Realocação.



Fonte - Elaborado pelo autor.

Figura 21 - Resultado realocação.

LOJA 1			
A	4 unid.	0,5	2
B	0 unid.	0,25	0
C	8 unid.	1	0
D	2 unid.	1,5	2
E	1 unid.	0,5	1
F	2 unid.	0,25	1
G	1 unid.	0,5	0
H	2 unid.	0,8	2
I	9 unid.	0,1	2
J	5 unid.	0,05	0
R\$ 6,55			

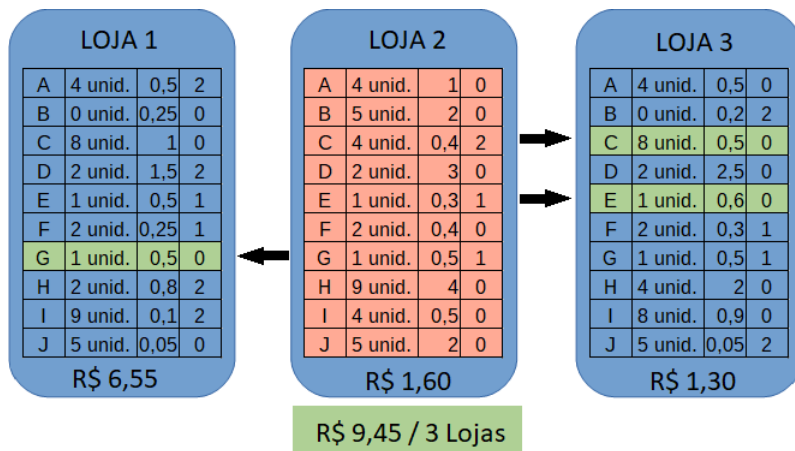
LOJA 2			
A	4 unid.	1	0
B	5 unid.	2	0
C	4 unid.	0,4	2
D	2 unid.	3	0
E	1 unid.	0,3	1
F	2 unid.	0,4	0
G	1 unid.	0,5	1
H	9 unid.	4	0
I	4 unid.	0,5	0
J	5 unid.	2	0
R\$ 1,60			

LOJA 3			
A	4 unid.	0,5	0
B	0 unid.	0,2	2
C	8 unid.	0,5	0
D	2 unid.	2,5	0
E	1 unid.	0,6	0
F	2 unid.	0,3	1
G	1 unid.	0,5	1
H	4 unid.	2	0
I	8 unid.	0,9	0
J	5 unid.	0,05	2
R\$ 1,30			

R\$ 9,45 / 3 Lojas

Fonte - Elaborado pelo autor.

Figura 22 - Fase 2 - Eliminação.



Fonte - Elaborado pelo autor.

Figura 23 - Resultado eliminação.

LOJA 1			
A	4 unid.	0,5	2
B	0 unid.	0,25	0
C	8 unid.	1	0
D	2 unid.	1,5	2
E	1 unid.	0,5	1
F	2 unid.	0,25	1
G	1 unid.	0,5	1
H	2 unid.	0,8	2
I	9 unid.	0,1	2
J	5 unid.	0,05	0
R\$ 7,05			

LOJA 3			
A	4 unid.	0,5	0
B	0 unid.	0,2	2
C	8 unid.	0,5	2
D	2 unid.	2,5	0
E	1 unid.	0,6	1
F	2 unid.	0,3	1
G	1 unid.	0,5	1
H	4 unid.	2	0
I	8 unid.	0,9	0
J	5 unid.	0,05	2
R\$ 2,90			

R\$ 9,95 / 2 Lojas

Fonte - Elaborado pelo autor.

# Conteúdo

- 1 Introdução
- 2 Fundamentação Teórica
- 3 Materiais e Métodos
- 4 Projeto e Desenvolvimento
- 5 Resultados**
- 6 Considerações Finais

Tabela 1 - Instâncias.

	Formato	Faixa Preço	Qtd. Cartas	Qtd. Únicas
Atraxa	<i>Commander</i>	Alto	100	100
Niv-Mizzet	<i>Commander</i>	Baixo	100	83
Storm	<i>Modern</i>	Alto	75	28
UB-Mill	<i>Modern</i>	Baixo	75	23
Esper	<i>Standard</i>	Alto	75	27
Mono-Blue	<i>Standard</i>	Baixo	75	22

Fonte - Elaborado pelo autor.

- *Commander*: Desde 1993;
- *Modern*: Desde 2003;
- *Standard*: Somente as coleções mais recentes;

# Validação Simulated Annealing

Figura 24 - Atraxa *Commander*.

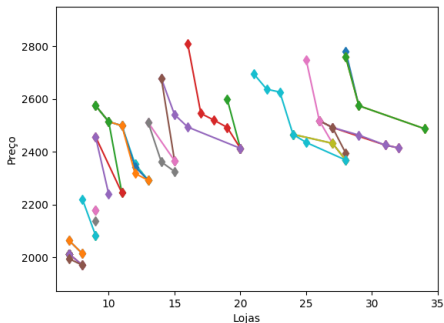
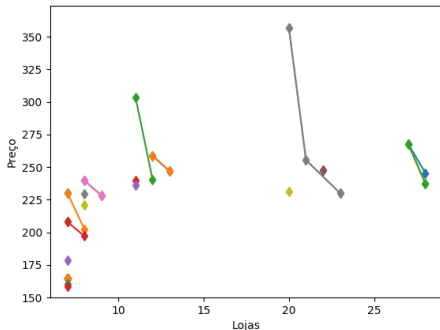


Figura 25 - Niv-Mizzet *Commander*.



Fonte - Elaborado pelo autor.



Figura 26 - Atraxa Commander.

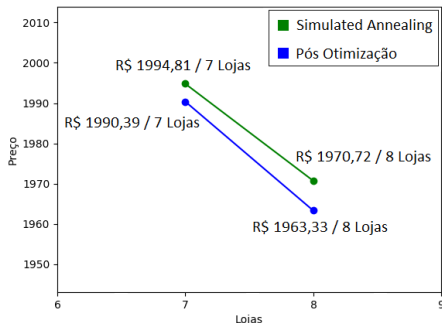
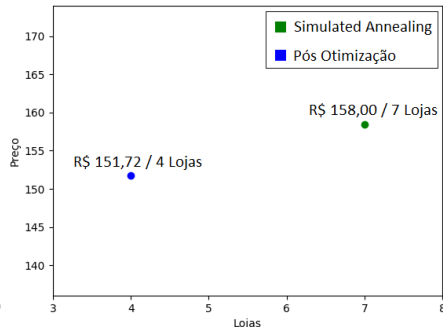


Figura 27 - Niv-Mizzet Commander.



Fonte - Elaborado pelo autor.

- ScryX - LigaMagic;
- Frete carta registrada  $\approx$  R\$ 10,00.

Figura 28 - Esper *Standard*.

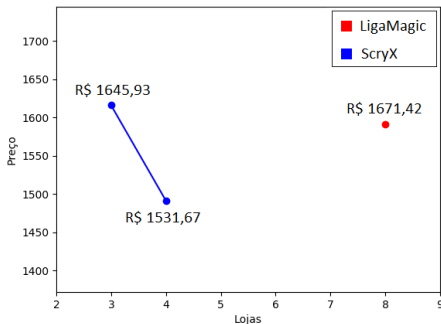
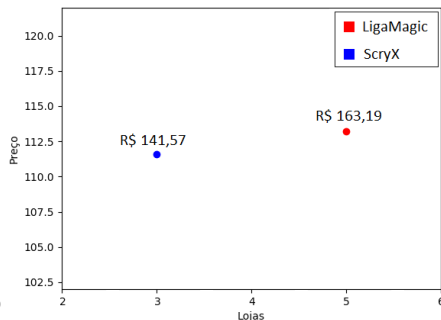


Figura 29 - Mono-Blue *Standard*.



Fonte - Elaborado pelo autor.

+1,53% / +8,37%

+13,25%

Figura 30 - Storm *Modern*.

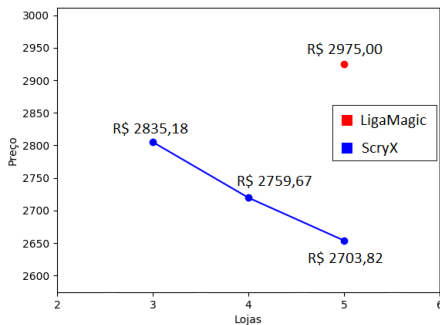
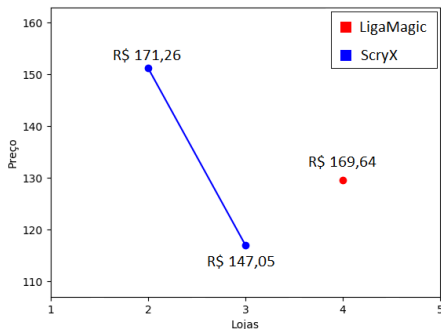


Figura 31 - UB-Mill *Modern*.



Fonte - Elaborado pelo autor.

+4,7% / +7,24% / +9,12%

-0,95% / +13,32%

Figura 32 - Atraxa Commander.

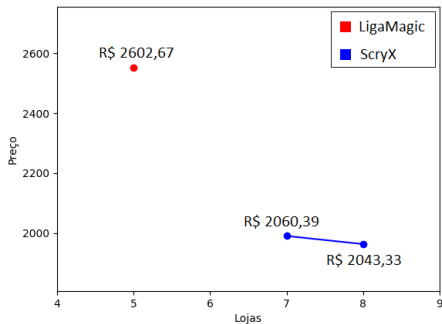
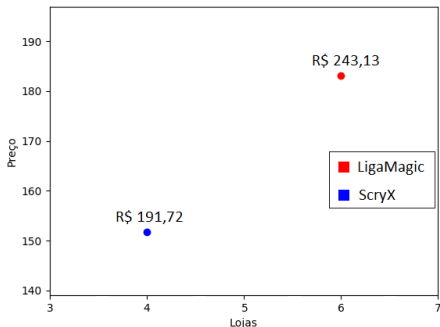


Figura 33 - Niv-Mizzet Commander.



Fonte - Elaborado pelo autor.

+20,84% / 21,5%

+21,15%

# Conteúdo

- 1 Introdução
- 2 Fundamentação Teórica
- 3 Materiais e Métodos
- 4 Projeto e Desenvolvimento
- 5 Resultados
- 6 Considerações Finais**

# Considerações Finais

- Elimina a necessidade de cotação de forma manual;
- Não existia nenhuma ferramenta igual no início do projeto;
- Encontrou bons resultados em relação a ferramenta da LigaMagic.

## Trabalhos Futuros:

- Vincular o cálculo de frete aos Correios, para valores exatos;
- Desenvolver diferentes metaheurísticas com o objetivo de tentar encontrar melhores resultados;
- *Yu-Gi-Oh!* e *Pokémon*;
- Desenvolver ferramentas de cotações para outras áreas, como exemplo, para aplicativos como Mercado Livre e Taki APP.





## APIFY.

What is the difference between web scraping and crawling?, 2019.  
Acessado em: 19 de Nov. 2019.



## Piotr Czyzak and Andrzej Jaskiewicz.

Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization.  
*Journal of Multi-criteria Decision Analysis*, 7(1):34–47, 1998.



## Gearhear.

Pros and cons of building single page applications in 2019, 2019.  
Acessado em: 25 de Nov. 2019.



## Ligamagic.

Ligamagic, 2019.  
Acessado em: 21 de Nov. 2019.

# Referências II



University Newcastle.

Roulette wheel selection, 2019.

Acessado em: 25 de Nov. 2019.



Justin Ormont.

pareto-frontier, 2016.

Acessado em: 21 de Nov. 2019.



Roger S. Pressman and Bruce R. Maxim.

*Engenharia de Software: Uma Abordagem Profissional.*

McGraw Hill Brasil, 8 edition, 2016.



Wizards of the Coast LLC.

The history of magic, 2019.

Acessado em: 27 de Nov. 2019.

# Uso de Metaheurística em uma Ferramenta de Cotação para Compras de Cartas de *Magic: The Gathering*

Luiz Eduardo Pereira



Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus Formiga – IFMG

29 de novembro de 2019

```
1 # Criacao da requisicao de cotacao de preco
2 @app.route("/request_list", methods=['POST'])
3 @jwt_required
4 def request_list():
5     post_data = request.get_json()
6     if (len(post_data.get("card_list")) == 0):
7         return jsonify({'status': '3'})
8     try:
9         error_list = register_request(db, post_data.get("deck_name"),
10                                     post_data.get("card_list"), post_data.get("username"))
11         if (error_list == None):
12             return jsonify({'status': '3'})
13         elif (error_list == []):
14             return jsonify({'status': '0'})
15         else:
16             return jsonify({'status': '2', 'error_list': error_list})
17     except:
```

```
1 submitList() {  
2     this.errorList = [];  
3     let payload = {  
4         'card_list': this.card_list,  
5         'deck_name': this.name,  
6         'username': localStorage.getItem('username')  
7     }  
8     axiosAuth.post('http://localhost:5000/request_list', payload)  
9         .then((res) => {  
10         if (res.data.status == 0) {  
11             this.message = 'Sucesso'  
12         } else if (res.data.status == 1) {  
13             this.message = 'Ocorreu um erro. Verifique as seguintes cartas  
14                 : '  
15             this.errorList = res.data.error_list;  
16         }  
17     })  
18 }
```

```
1 # Extrai os links das lojas
2 def parse_page(self, response):
3
4     lines = response.xpath('//div[@class="e-col1"]/a')
5     links = lines.xpath('./@href').extract()
6     names = lines.xpath('./img/@title').extract()
7
8     for link, name in zip(links, names):
9         yield scrapy.Request(url=store_link_mount(link), callback=self.
            parse_item, meta={'card_store': (response.meta['card'], name)
            })
```

# SA - Parâmetros

```
#### TEMPERATURE  
INITIAL_TEMPERATURE 50  
FINAL_TEMPERATURE 0.1  
COOLING_SCHEDULE GEOMETRIC  
ALPHA 0.995  
REHEAT 2
```

```
#### ACCEPTANCE CRITERIA  
#### LAMBDA1: Preço total  
#### LAMBDA2: Quantidade de cartas  
#### LAMBDA3: Quantidade de lojas  
ACCEPTANCE_OPTION W  
LAMBDA1 0.4  
LAMBDA2 0  
LAMBDA3 0.6
```

```
#### FITNESS  
PRICE_INCREASE 10  
QUANTITY_INCREASE 1000  
STORE_INCREASE 100  
PERCENT_MISSING 10
```

```
#### THREADS  
N_THREAD 3  
N_SOLUTION_POOL 500
```

```
#### ROULETTE WHEEL  
ROULETTE_OPTION QUANTITY
```

```
1 result_table = gera_solucao_inicial()
2 for i in range(CADEIAS_MARKOV):
3     temperatura_atual = TEMPERATURA_INICIAL
4     while ((temperatura_atual > TEMPERATURA_FINAL)):
5         for card in card_dict.items():
6             new_result_table = swap_change_all(result_table, card)
7             if (random.uniform(0, 1) <= criterio_aceitacao(
8                 result_table, new_result_table, temperatura_atual)):
9                 result_table = new_result_table
10                solutions.append(result_table)
11                if (len(solutions) >= MAXIMO_SOLUCOES):
12                    solution_deliver.put(solutions)
13                    solutions = []
14            temperatura_atual = resfria_temperatura(temperatura_atual)
```



```
1 # Encontra a fronteira de pareto
2 def pareto_thread(solution_deliver, result_deliver):
3     while (n_thread_cont > 0):
4         message = solution_deliver.get()
5         if (message == 'thread_finished'):
6             n_thread_cont -= 1
7         else:
8             solutions += message
9             solutions = prepara_pareto(solutions)
10
11     result_deliver.put(solutions)
```