

Trabalho Prático

Banco de Dados II

Douglas Raimundo de Oliveira Silva - 2019018540

Vítor Ramos de Carvalho - 2019004231

Vitor Siqueira Lobao - 2018004809

Webster Aurelio Carvalho Ramos - 2018014716

Yerro Cândido Ferreira - 2019004142

ROTEIRO DA APRESENTAÇÃO

02

1. API de dados utilizada

2. Modelagem

3. Carga no banco







6. JMeter

7. Conclusão

The Rick and Morty API

rickandmortyapi.com

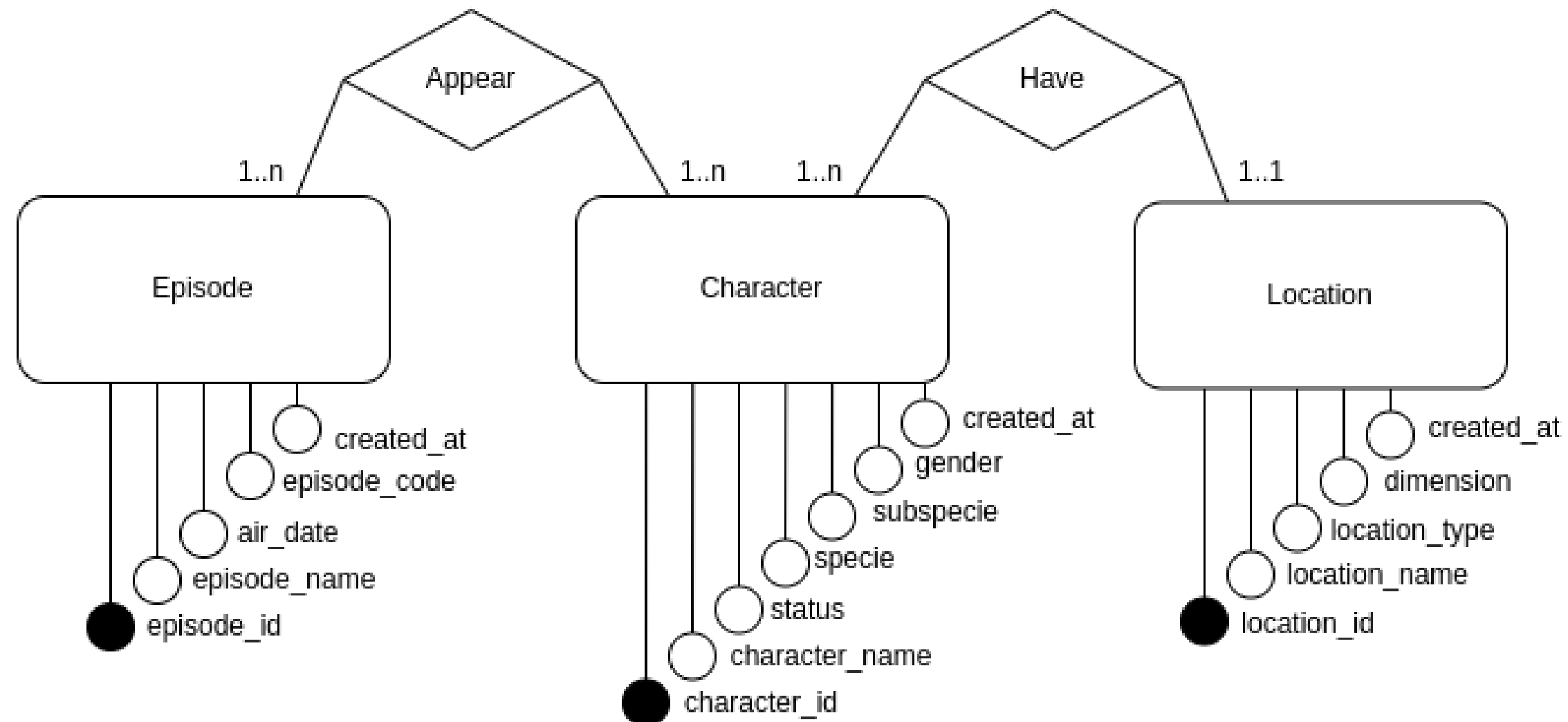
The Rick and Morty API

	<p>Morty K-22</p> <p>• Alive - Human</p> <p>Last known location: Earth (Replacement Dimension)</p> <p>First seen in: The Ricklantis Mixup</p>		<p>Noob-Noob</p> <p>• Alive - Poopybutthole</p> <p>Last known location: Vindicator's Base</p> <p>First seen in: Vindicators 3: The Return of Worldender</p>		<p>SEAL Team Rick</p> <p>• Dead - Human</p> <p>Last known location: Citadel of Ricks</p> <p>First seen in: The Rickshank Rickdemption</p>
	<p>Wasp Summer</p> <p>• Alive - Animal</p> <p>Last known location: Earth (Wasp Dimension)</p> <p>First seen in: Edge of Tomorty: Rick, Die, Rickpeat</p>		<p>Birdperson</p> <p>• Alive - Alien</p> <p>Last known location: Story Train</p> <p>First seen in: Never Ricking Morty</p>		<p>Hairspray Fan</p> <p>• Alive - Alien</p> <p>Last known location: Story Train</p> <p>First seen in: Never Ricking Morty</p>

Modelagem

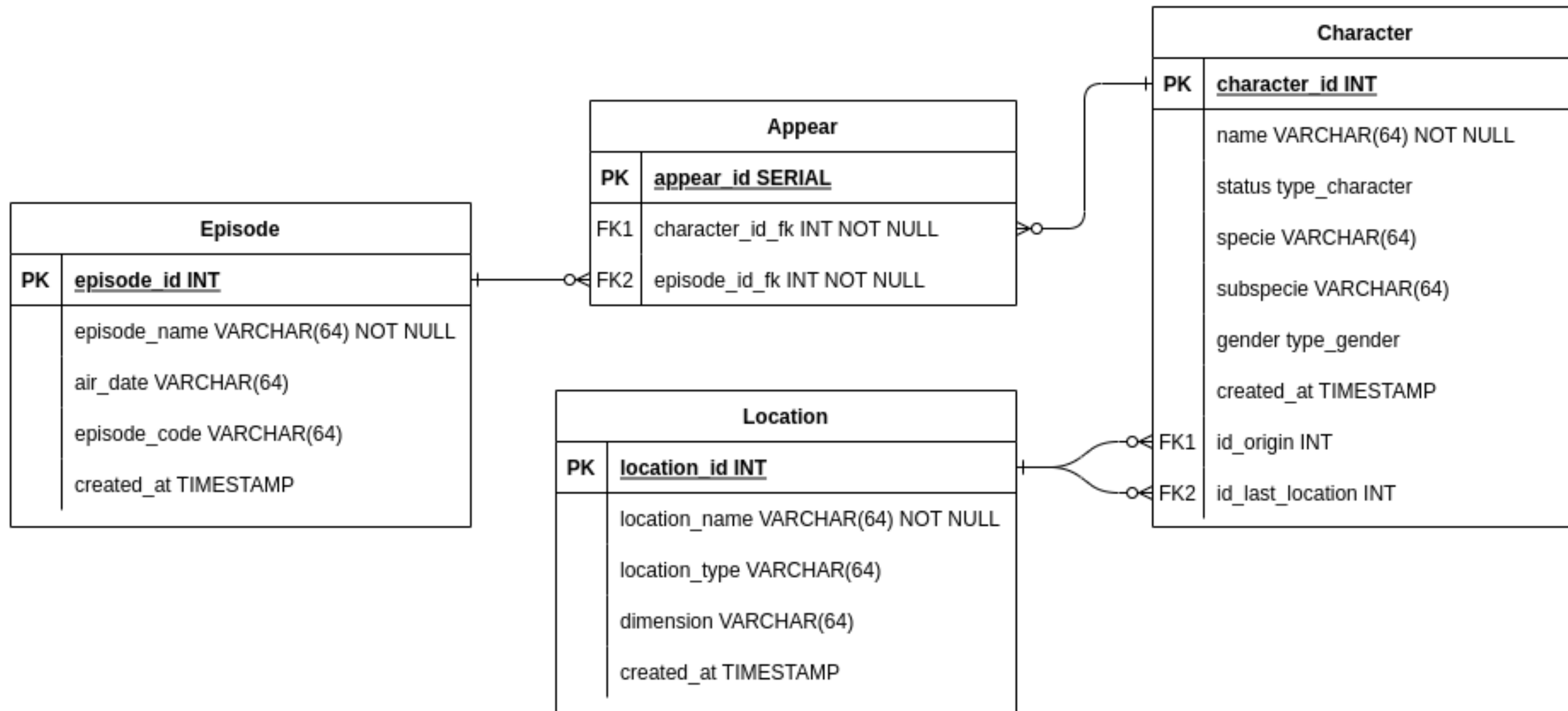
04

Modelo Entidade Relacionamento



Modelagem

Modelo Lógico



Modelagem

Índices - sem índice

1

EXPLAIN ANALYSE SELECT * FROM character WHERE specie = 'Humanoid'

Data Output

Explain

Messages

Notifications

QUERY PLAN

text

1

Seq Scan on "character" (cost=0.00..20.33 rows=68 width=59) (actual time=0.025..0.294 rows=68 loops=1)

2

[...] Filter: ((specie)::text = 'Humanoid'::text)

3

[...] Rows Removed by Filter: 758

4

Planning Time: 0.090 ms

5

Execution Time: 0.333 ms

Modelagem

Índices - com índice

```
1 CREATE INDEX character_specie_index ON character(specie)
2 EXPLAIN ANALYSE SELECT * FROM character WHERE specie = 'Humanoid'
3
```

Data Output Explain Messages Notifications

	QUERY PLAN	
▲	text	🔒
1	Bitmap Heap Scan on "character" (cost=4.68..15.53 rows=68 width=59) (actual time=0.038..0.083 rows=68 lo...	
2	[...] Recheck Cond: ((specie)::text = 'Humanoid'::text)	
3	[...] Heap Blocks: exact=10	
4	[...] -> Bitmap Index Scan on character_specie_index (cost=0.00..4.66 rows=68 width=0) (actual time=0.025..0...	
5	[...] Index Cond: ((specie)::text = 'Humanoid'::text)	
6	Planning Time: 0.121 ms	
7	Execution Time: 0.123 ms	

```
1 SELECT
2     schemaname,
3     relname,
4     n_live_tup
5 FROM pg_stat_user_tables
6 ORDER BY n_live_tup;
```

	Data Output	Explain	Messages	Notificat
	<div>schemaname</div> <div>name</div>	<div>relname</div> <div>name</div>	<div>n_live_tup</div> <div>bigint</div>	
1	public	episode	51	
2	public	location	126	
3	public	character	826	
4	public	appear	1267	

CARGA NO BANCO

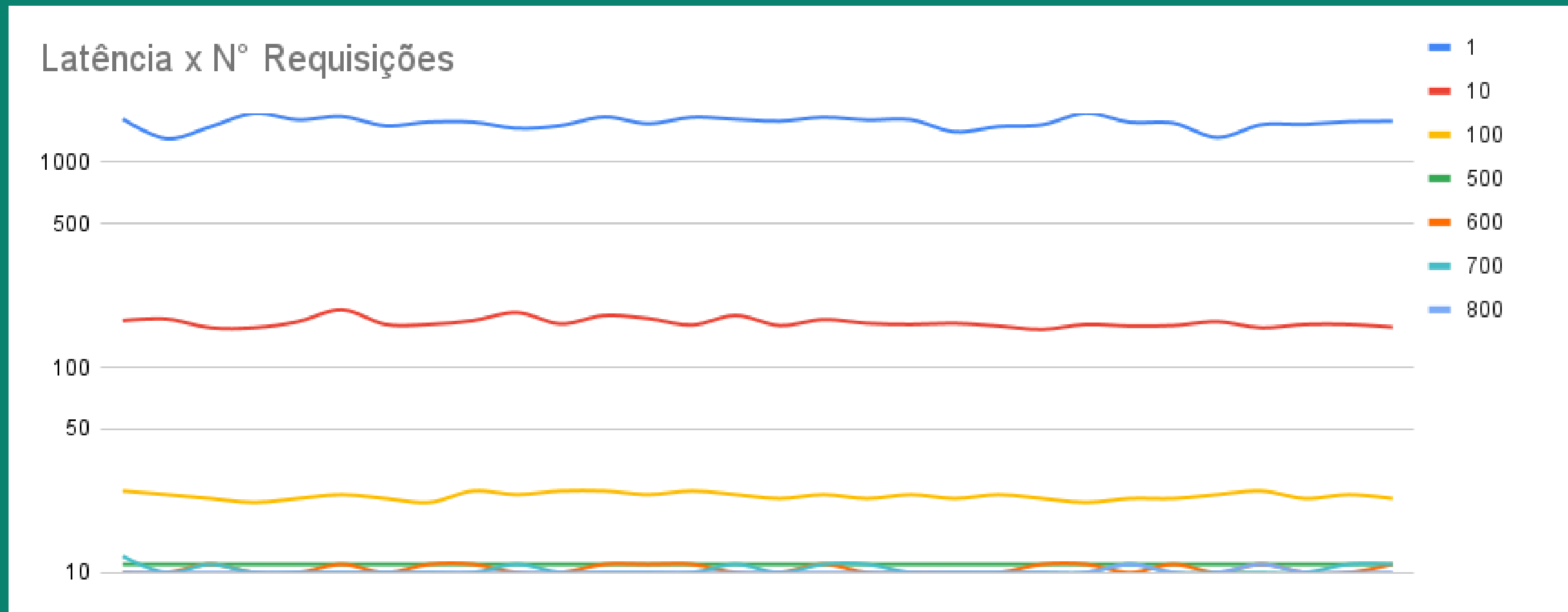
Uma aplicação, em Python, para consumir os dados da API.

Acesso e download dos dados utilizando um filtro e retorno dos dados.

Extração dos dados do JSON e carga nas tabelas do banco de dados.

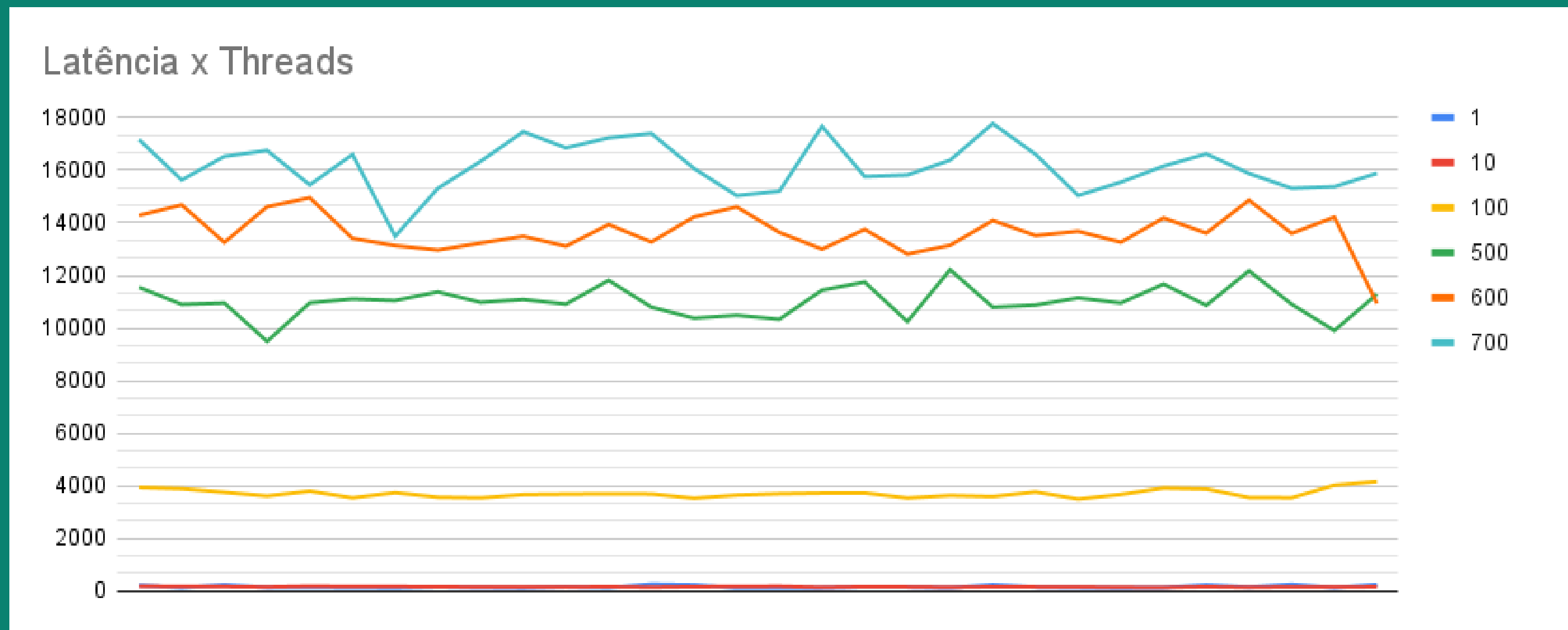
JMeter - Teste de Latência

Um usuário e diversas requisições



JMeter - Teste de Latência

Diversos usuários e uma requisição





Conclusão

- Prática de conceitos da disciplina
- Aprendizado de consumo e processamento de dados de uma API
- Realização de testes e estudo dos resultados

Obrigado!