**Relatório sobre o teste de carga da leitura do BD de músicas com filtragem aplicada (por título e dos gêneros rock e MPB)**

**Tipo de operações(s)**

Leitura.

**Arquivos envolvidos**

* api/config/config.js
* api/controller/musicas.js
* api/model/connection.js
* api/index.js
* sql/criacao\_tabelas.sql
* sql/carga\_inicial.sql

**Download do k6**

https://github.com/grafana/k6-studio/releases

**Data da medição**

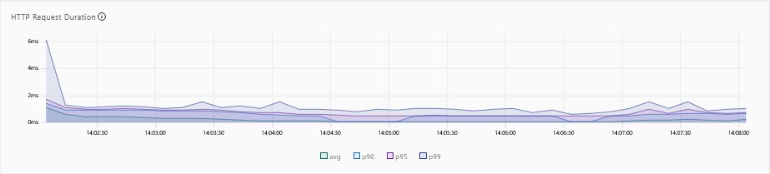
30/06/2025

**Descrição das configurações**

Todos os testes abaixo foram feitos em um computador com processador AMD Ryzen 5 5600G que contém Radeon Graphics 3.90 GHz, memória RAM de 16,0 GB (e 15,4 GB utilizáveis) e no sistema Windows 10. O disco usado foi da marca Xraydisk 1TB SSD, que possui velocidade de leitura sequencial de até 550 MB/s e escrita de 490 MB/s.

**Testes de carga (SLA):**

**Latência (Duração de Requisição HTTP):**



Legenda: avg, p90, p95, p99

**Concorrência (VUs) à esquerda e requisições feitas por segundo (http\_reqs por segundo) à direita:**



Legenda: vus, http\_reqs

**Vazão (Taxa de Transferência):**



Legenda: data\_received, data\_sent

**Levantamento de hipóteses**

* A não utilização de índices no sql do BD pode tornar o ato de filtragem bem mais lento do que necessário.
* A quantidade de dados transmitidos não aumenta proporcionalmente quando há muitos VUs simultâneos, possivelmente indicando que o socket único que leva as requisições ao BD fica sobrecarregado quando há muitos usuários).
* Recuperar todos os dados de uma música requer buscar dados de muitas tabelas diferentes (artista, musica, gênero, álbum, etc), o que pode deixar o ato de filtragem desnecessariamente lento.

**Relatório sobre o teste de carga da solicitação de nova música no BD de solicitações de músicas**

**Tipo de operações(s)**

Inserção.

**Arquivos envolvidos**

* api/config/config.js
* api/controller/musicas.js
* api/model/connection.js
* api/index.js
* sql/criacao\_tabelas.sql
* sql/carga\_inicial.sql

**Download do k6**

https://github.com/grafana/k6-studio/releases

**Data da medição**

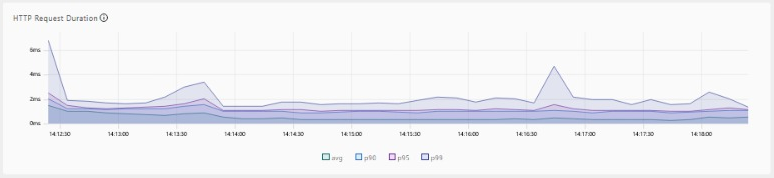
30/06/2025

**Descrição das configurações**

Todos os testes abaixo foram feitos em um computador com processador AMD Ryzen 5 5600G que contém Radeon Graphics 3.90 GHz, memória RAM de 16,0 GB (e 15,4 GB utilizáveis) e no sistema Windows 10. O disco usado foi da marca Xraydisk 1TB SSD, que possui velocidade de leitura sequencial de até 550 MB/s e escrita de 490 MB/s.

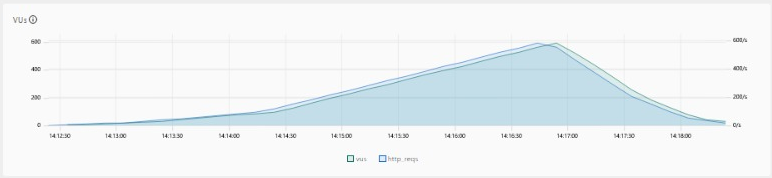
**Testes de carga (SLA):**

**Latência (Duração de Requisição HTTP):**



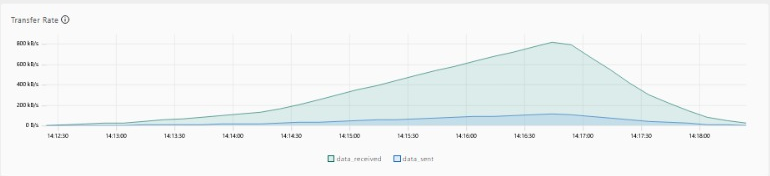
Legenda: avg, p90, p95, p99

**Concorrência (VUs) à esquerda e requisições por segundo (http\_reqs por segundo) à direita:**



Legenda: vus, http\_reqs

**Vazão (Taxa de Transferência):**



Legenda: data\_received, data\_sent

**Levantamento de hipóteses**

* Picos de latência ocorrem tanto em períodos com muitos VUs como também com poucos VUs, uma hipótese possível é a ocorrência ocasional de múltiplas requisições tentarem acessar ou escrever na mesma linha simultaneamente, causando maior lentidão.
* A quantidade de dados transmitidos é extremamente desproporcional quando há muitos VUs simultâneos, possivelmente sugerindo que o socket único que leva as requisições ao BD fica consideravelmente sobrecarregado quando há muitos usuários.
* Recuperar todos os dados de uma música requer buscar dados de muitas tabelas diferentes (artista, musica, gênero, álbum, etc), o que pode deixar o ato de filtragem desnecessariamente lento.