

**GO E BANCO DE DADOS**

GANHANDO DESEMPENHO EM PROCESSOS DE CARGA DE DADOS

---

**MARCELO  
KRÜGER**

# BIG DATA

---

Grande Volume de Dados, estruturados e não estruturados,  
oriundos de diversas fontes

=  
Gerar conhecimento para auxiliar na tomada de decisão

# DESAFIOS DA CARGA DE DADOS

---

- Grande volume de dados

Dados Transacionais, Informações publicas, Dados de Sensores, ...

- Dados em diversos tipos de arquivo

CSV / Texto / Json / XML

- Quanto maior o volume, maior o tempo para carregar dados em uma estrutura

Escalabilidade / Velocidade / Facil manutenção

## **O BANCO DE DADOS**

---

- Recebe a maior carga dentro de uma estrutura de dados
- Recurso compartilhado entre outros processos e usuarios

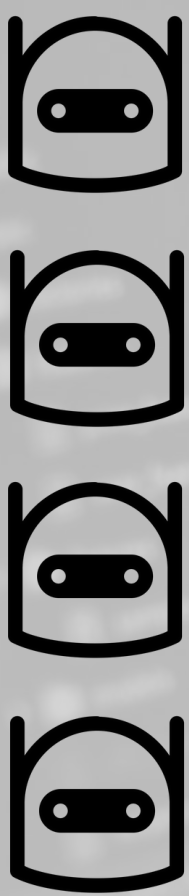
Uso constante de recursos (CPU / Memoria / Disco)

- Maior criticidade frente a outros processos

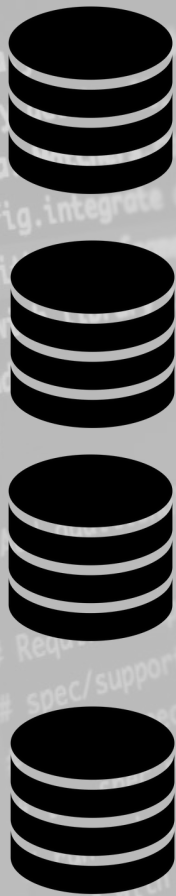
Como carregar dados de forma eficiente sem prejudicar o banco de dados?

# ESTRUTURA TOTALMENTE EM POSTGRESQL

---



Bots



I/O do Bot



Tratamento



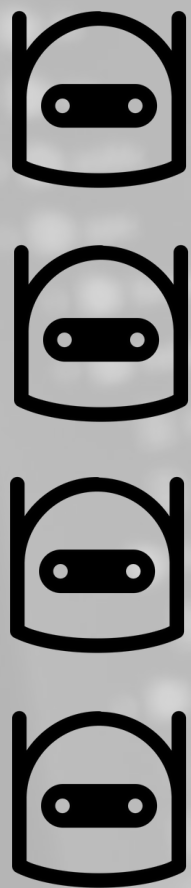
Repositorio

# ESTRUTURA TOTALMENTE EM POSTGRESQL

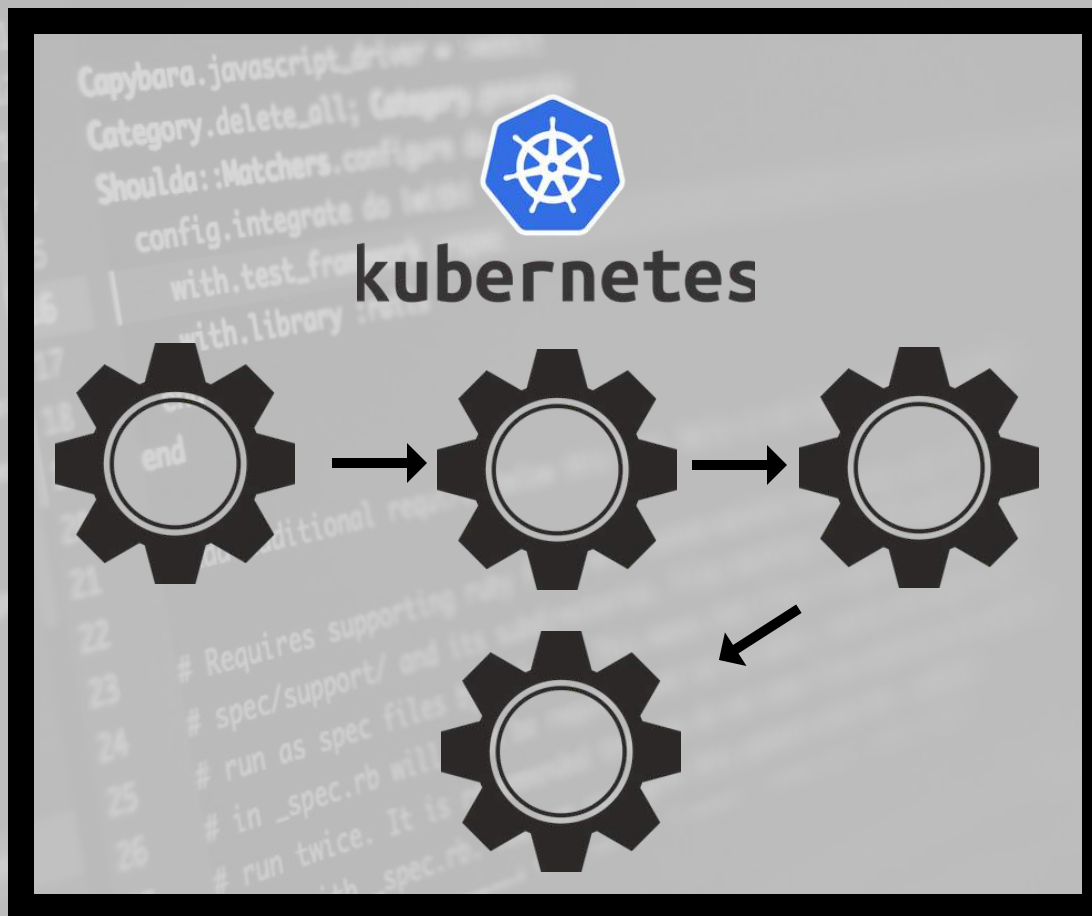
---

- Estrutura completamente em banco. Difícil de escalar
- Replicação com bucardo. Mesma estrutura de tabelas em todas as máquinas
- Custo elevado para manter todas as máquinas de banco basicamente como storage.

# ESTRUTURA COM MICROSSERVIÇOS



Bots



mongoDB



Repositorio

# ESTRUTURA COM MICROSERVIÇOS

---

- Facilidade para escalar serviços.
- Arquitetura com menor custo (R\$)
- Operações de tratamento podem ser realizadas com os dados que estão no streaming, não necessário um banco exclusivo para realizar o tratamento. Menos duplicação de dados, menos custo!

Como replicar dados para o PG de forma simples e eficiente?

Dados vem do streaming em JSON!



# FORMAS DE REALIZAR CARGA DE DADOS

---

- COPY
- PL/SQL
- Scripts (BASH)
- Softwares para ETL (PDI, )
- Via Linguagens  
JAVA / C# / Python / C++ / Go, ....

# QUAL UTILIZAR?

---

- Realizado teste de inserção em massa no PostgreSQL, utilizando as linguagens Java, C#, Python e GO
- Estrutura do teste

CPU: i7

Memoria: 8GB

Banco de Dados: PostgreSQL 9.6 em container Docker

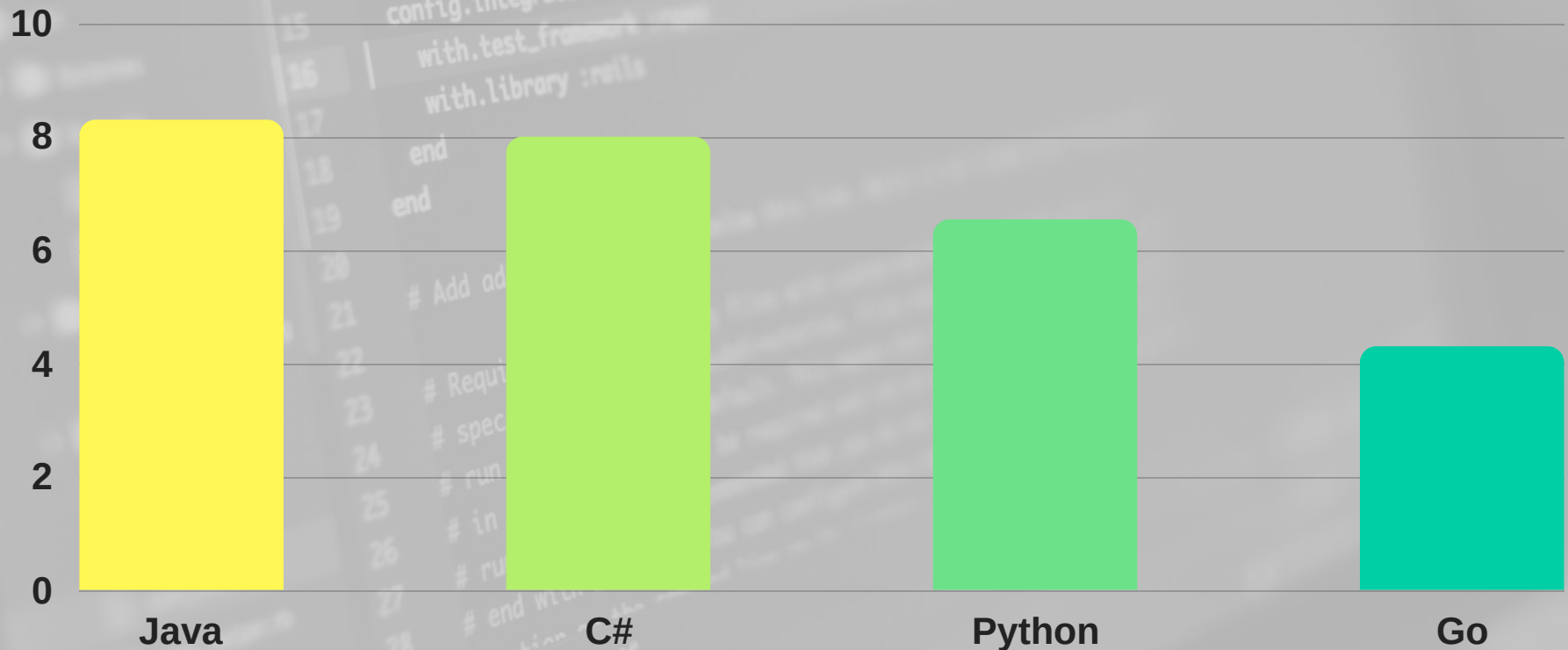
Carga de 5 milhões de empresas (CNPJ e Razão Social)

Processo e Banco de Dados estão executando na mesma estrutura.

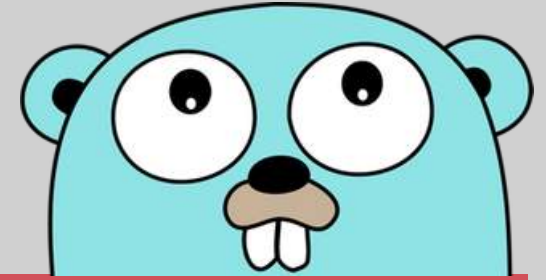
# QUAL UTILIZAR?

---

Tempo de Execução em cada linguagem (Minutos)



# QUAL UTILIZAR?



## Golang

- Além do desempenho para carga de dados

Programação concorrente/paralela nativa

Desempenho (Baseado em C \o/)

Multiplataforma

Open Source

Facilmente escalável

Simples!

GOLANG



# DEPENDÊNCIAS PARA CARGA DOS DADOS

---

- `database/sql`
- `github.com/lib/pq` (Driver PostgreSQL)

KEEP  
CALM  
AND  
USE  
GOLANG

## EXTRA

---

- Para melhorar o tempo de carga é possível ainda recorrer a estruturas de Pool no PostgreSQL.



Pgpool-II



PGBouncer

PostgreSQL

# PROJETOS EM GO PARA BANCO

---

- PGfutter - Carga de dados

<https://github.com/lukasmartinelli/pgfutter>

- SQL Exporter - Geração de métricas do Postgres para Prometheus

[https://github.com/justwatchcom/sql\\_exporter](https://github.com/justwatchcom/sql_exporter)

PostgreSQL

**OBRIGADO!**

---

**MARCELO.KRUGER@NEOWAY.COM.BR**  
**MARCELOKRUGER.COM.BR**

**VAGAS**  
**[HTTPS://JOBS.KENOBY.COM/NEOWAY/](https://jobs.kenoby.com/neoway/)**

Neoway 