GOLANG, GO BIG

GOLANG EM UM GRANDE PRODUTO

THE DEVELOPER'S CONFERENCE

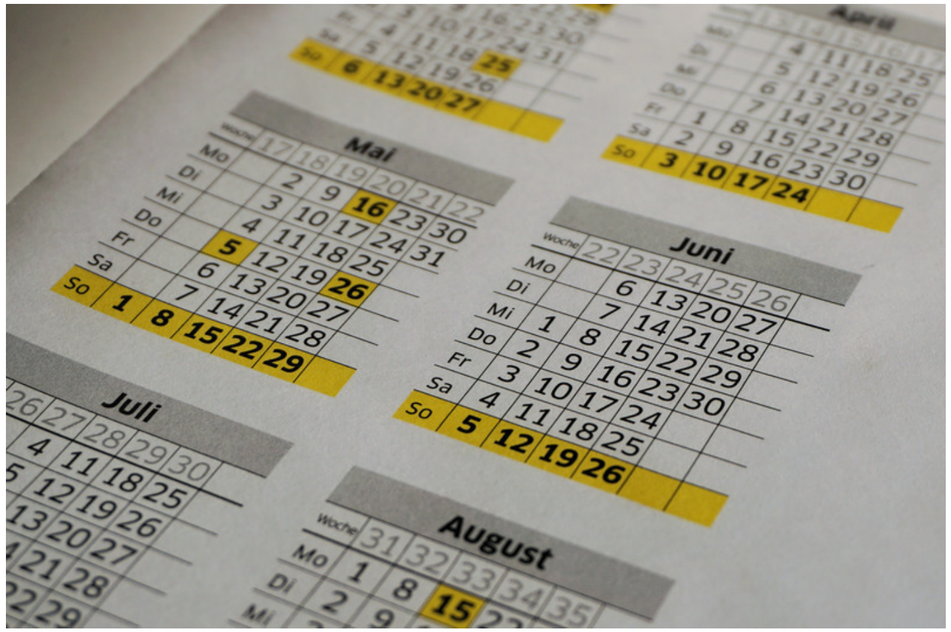
Diego Bernardes Gaulke

DIEGO GAULKE
ENGENHEIRO DE SOFTWARE

diegobgaulke@gmail.com
https://github.com/diegogaulke
https://www.linkedin.com/in/diegobernardesgaulke/

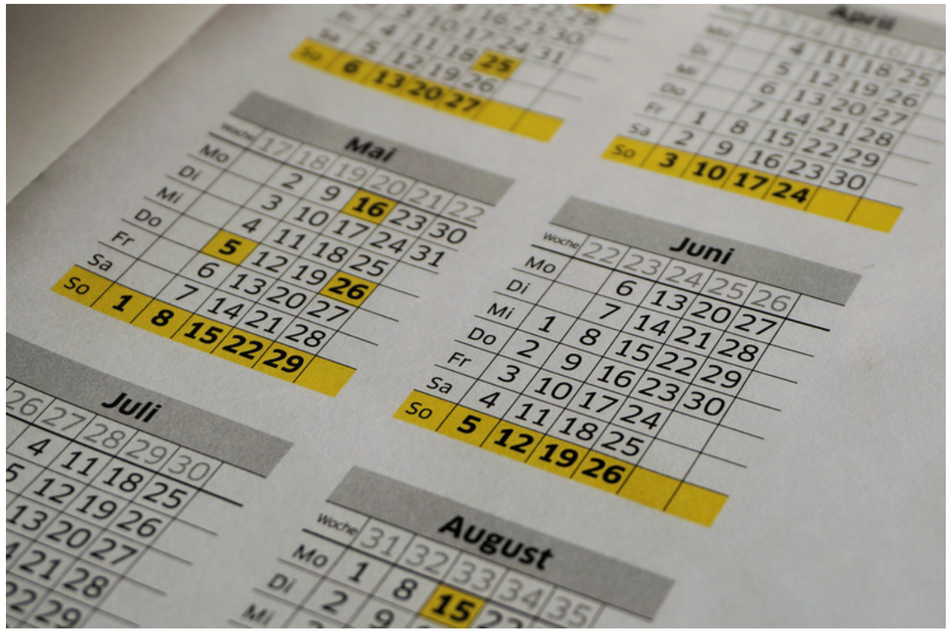
OS DADOS...

BRASILEIROS ENDIVIDADOS



BRASILEIROS ENDIVIDADOS

50%



POPULAÇÃO ECONOMICAMENTE ATIVA (PEA)

120 Milhões de brasileiros

INADIMPLENTES

60 milhões de Brasileiros

50% da PEA

QUANTAS DÍVIDAS EXISTEM NO BRASIL? NA BASE DA MAIOR EMPRESA DE RECUPERAÇÃO DE CRÉDITO?

QUANTAS DÍVIDAS EXISTEM NO BRASIL? NA BASE DA MAIOR EMPRESA DE RECUPERAÇÃO DE CRÉDITO?

350 milhões de dívidas

O PROBLEMA...

- **Aumento de acessos no site**
- **Disponibilizar informações em tempo real**
- **Concorrência computacional com grandes parceiros**
- **Quedas constantes (throttling mainframe)**

E TEM MAIS...

- **Custo de mainframe**
- **Um novo produto entrando no ar**
- **Campanhas de mídia online e offline já agendadas**
- **Pouco tempo para resolução**

COMO RESOLVER?

- **Buscar uma fonte de dados alternativa**
- **Não concorrer com outras áreas ou parceiros**
- **Manter uma base consistente**

DESAFIOS

- **6 tipos de dividas**
- **Atualização D-1**
- **Regras de negócio na carga dos dados**

MAIS DESAFIOS...

- **Grande quantidade de dados sensíveis**
- **Falta de documentação e normalização dos dados disponíveis**
- **Armazenamento de dados sensíveis na nuvem (compliance)**

PRIMEIRA "SOLUÇÃO" ...

- Golang para ETL
- DynamoDB
- Indice CPF criptografado
- Goroutines controladas (channels)

PROBLEMAS

- **Goroutines controladas (channels)**
- **DynamoDB**

DYNAMODB

DynamoDB

amazon web services™

- Não é feito para escalar de maneira rápida
- Deixar ele "escalado" tem um custo muito alto
- Preso a uma única cloud
- Demora na carga dos dados (mais de 10 horas para 50 milhões de registros)

MESMO ASSIM...

DynamoDB

amazon web services™

- A solução importou 3 tipos de dívidas
- Aparentemente estava rodando sem problemas
- Trazia rapidamente os registros de dívidas pelo CPF

MAS...

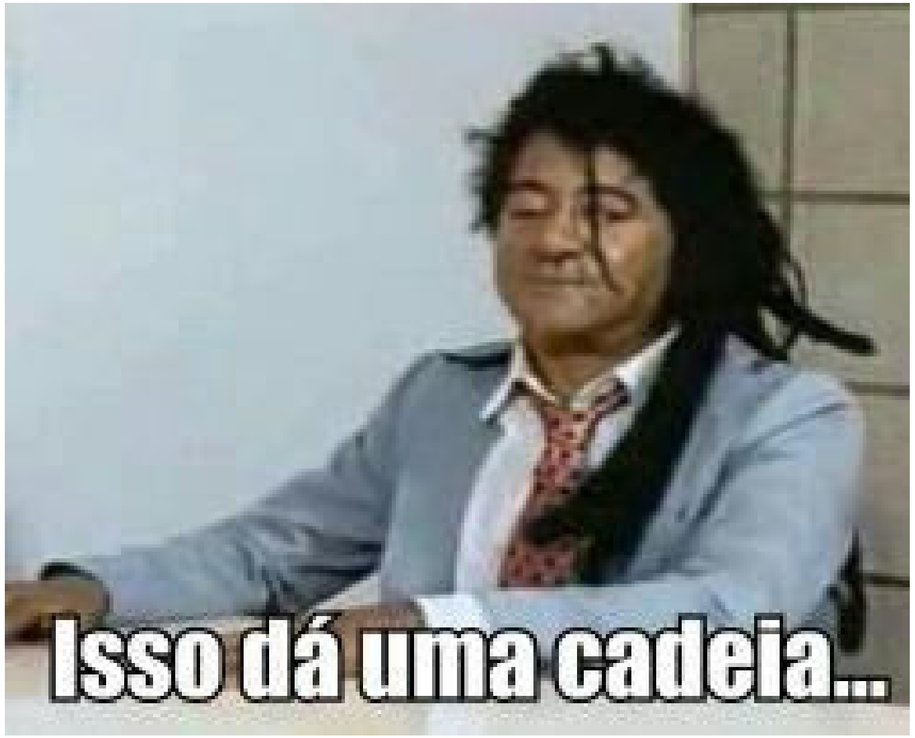
- Como conferir a consistência do que foi importado?
- Como contar a quantidade de dívidas importadas por cada tipo?
- Como aumentar a performance da carga?
- Como diminuir a complexidade das rotinas no Go?

OUTROS PROBLEMAS...

- **Para um CPF com 5 dívidas apareciam apenas 3**
- **DynamoDB fazendo throttling e API da aws não dava erro**

"VOCÊS DEVEM ESTAR FAZENDO ERRADO"

"CLANDESTINO"



- Diminuir a complexidade do código Go
- Reduzir o custo de cloud
- Aumentar a velocidade da carga dos dados
- Monitorar e aferir a consistência da base

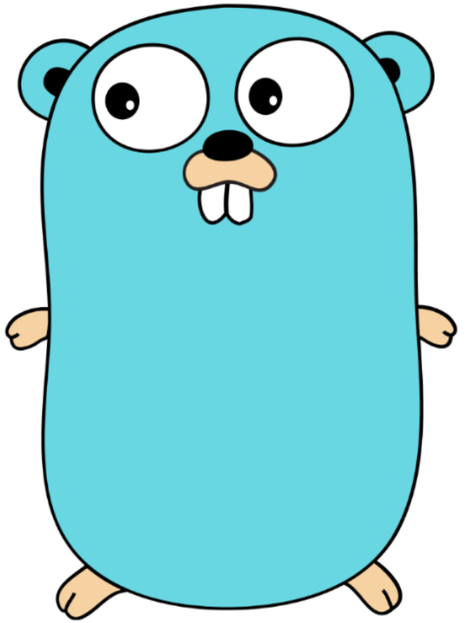
SOLUÇÃO "1"

- **Logstash**
- **Elastic Search**

DESCARTE SOLUÇÃO "1"

- **Go como "plugin" para o LogstashElastic Search**
- **Uma camada a mais de controle**

"SOLUÇÃO 2" (ESCOLHIDA)

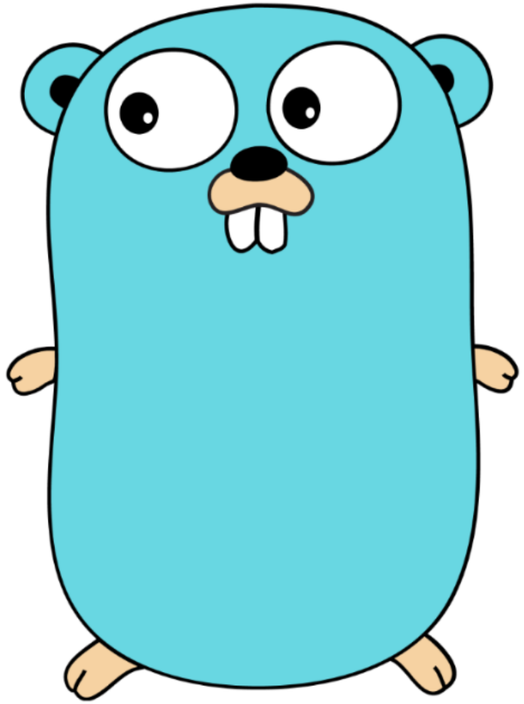


- **Elastic Search**
- **Golang**
- **Biblioteca ElasticSearch para Go, Olivere**

MODELAGEM
INDEXAÇÃO

- **Índice por tipo de dívida e pela data da carga**
- **Possibilidade em caso de falha de refazer a carga**

CONCLUSÃO



- **Redução de custo**
- **Baixa complexidade de código**
- **Possibilidade de aferição rápida dos dados**
- **Consistência dos dados importados**

AGRADECIMENTOS

FIM

- **Dúvidas?**

diegobgaulke@gmail.com
github.com/diegogaulke