



Painless: desenvolva sua
aplicação serverless sem
sofrimento

Evandro Pires da Silva

- Evangelista de Serverless
- Programador desde os 12 anos de idade (com Clipper)
- Filho de programador
- Intraempreendedor
- Coordenador do Senior Labs, laboratório de pesquisa da Senior Sistemas
- Founder e host do Sem Servidor, podcast dedicado ao tema Serverless
- Marido da Madi, e pai do Teodoro e da Olivia





VS



Quais as **principais dores** ao desenvolver uma aplicação **serverless**?



Principais dores

Quais?

- Linguagem de programação
- Design da função
- Arquitetura da aplicação
- Testes e Depuração
- Repositório
- CI/CD e IaC

Por onde começar?



Por onde começar?

- Mão na massa
- Escolha o cloud provider
- Estude os serviços do cloud provider que vão atender cada parte do projeto
- Defina a arquitetura do projeto
- Escolha a linguagem de programação
- Escolha um framework para desenvolvimento

Linguagem de programação

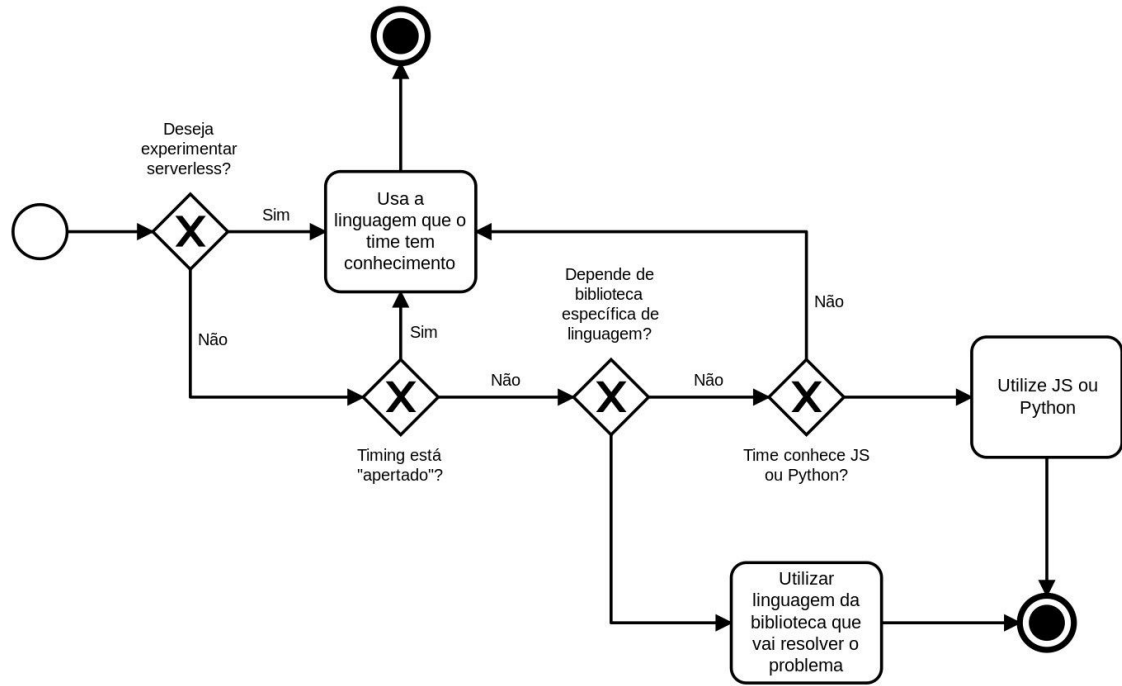


Linguagem de programação

Qual escolher?

- Aquele que resolve melhor o problema
- Cold start
- Know-how do time
- ...

Modelo de decisão



Linguagem de programação

Design da função



Design da função

- Separe a camada do handler da lógica do seu código
 - Camada handler
 - Camada business
 - Camada integration
- Utilize o ambiente de execução para tirar proveito de performance
 - Inicializar variáveis fora do handler
 - Arquivos no /tmp
- Faça conexões HTTP persistentes (*keep-alive*)
- Reduza o tamanho do seu pacote de deployment
- Utilize bibliotecas que são mais *lightweight*
- Não faça chamadas para a própria função



Design da função

- Não leve ao pé da letra a granularidade da função
 - Se não é reutilizado e é um mesmo processo, coloca na mesma função
 - Desenvolva de forma desacoplada, caso precise separar depois
- Se for usar um banco de dados relacional, não esqueça de usar com RDS Proxy (ou usar uma arquitetura que faça esse *throttling*)

Arquitetura da aplicação

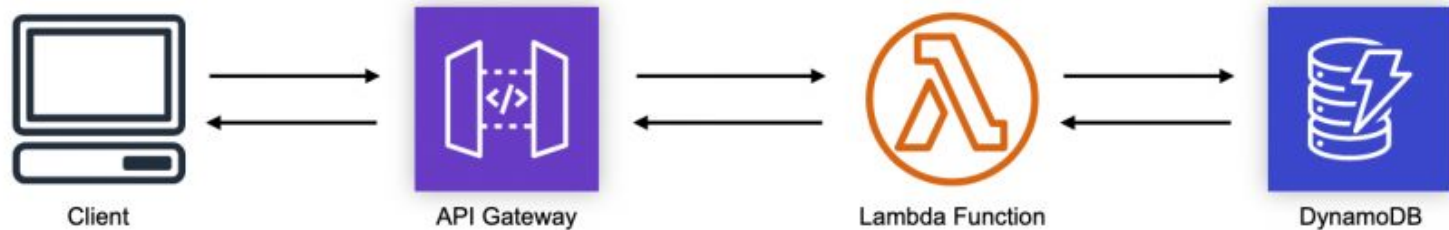
Arquitetura

Serverless Patterns for AWS

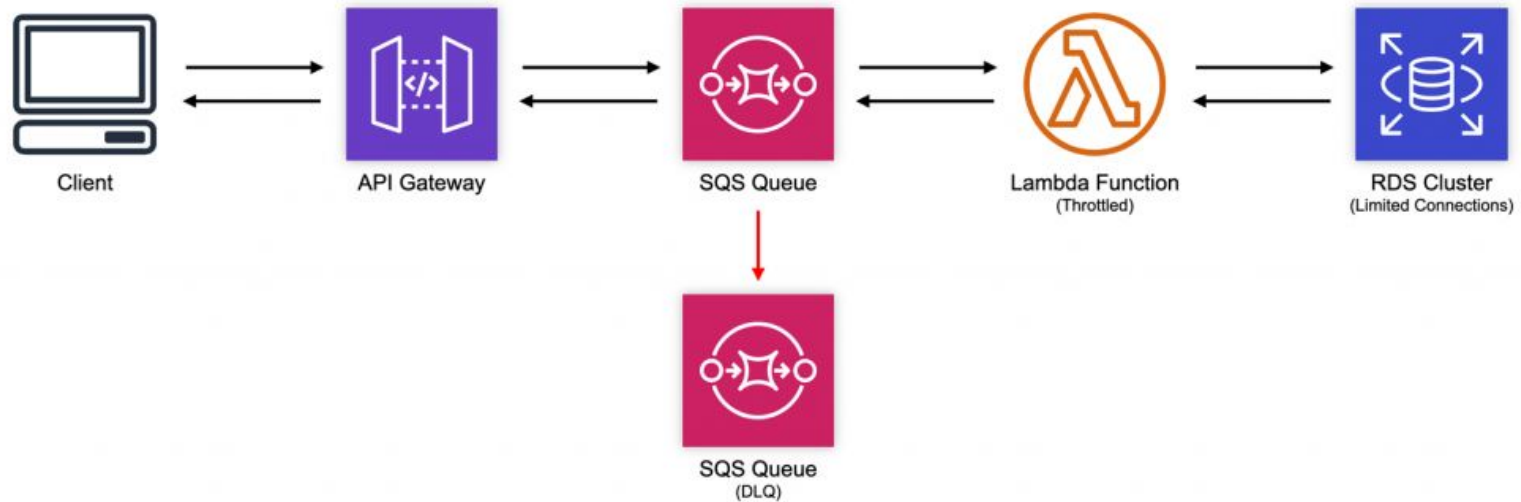
- Conhecimento dos serviços que o cloud provider oferece
- Padrões arquiteturais
 - The Simple Web Service
 - The Scalable Webhook
 - The Eventually Consistent
 -



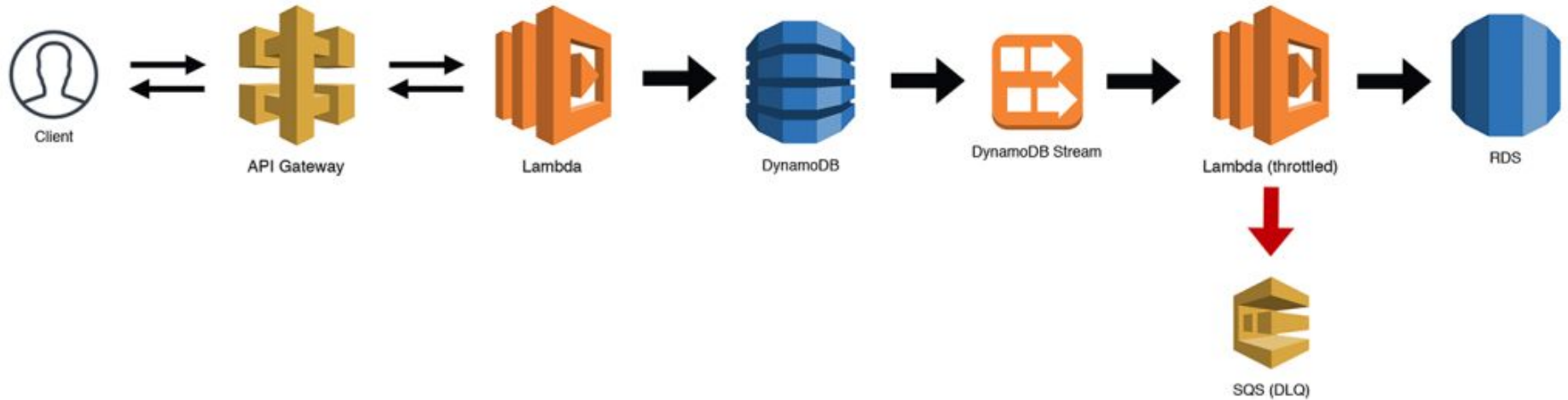
The Simple Web Service



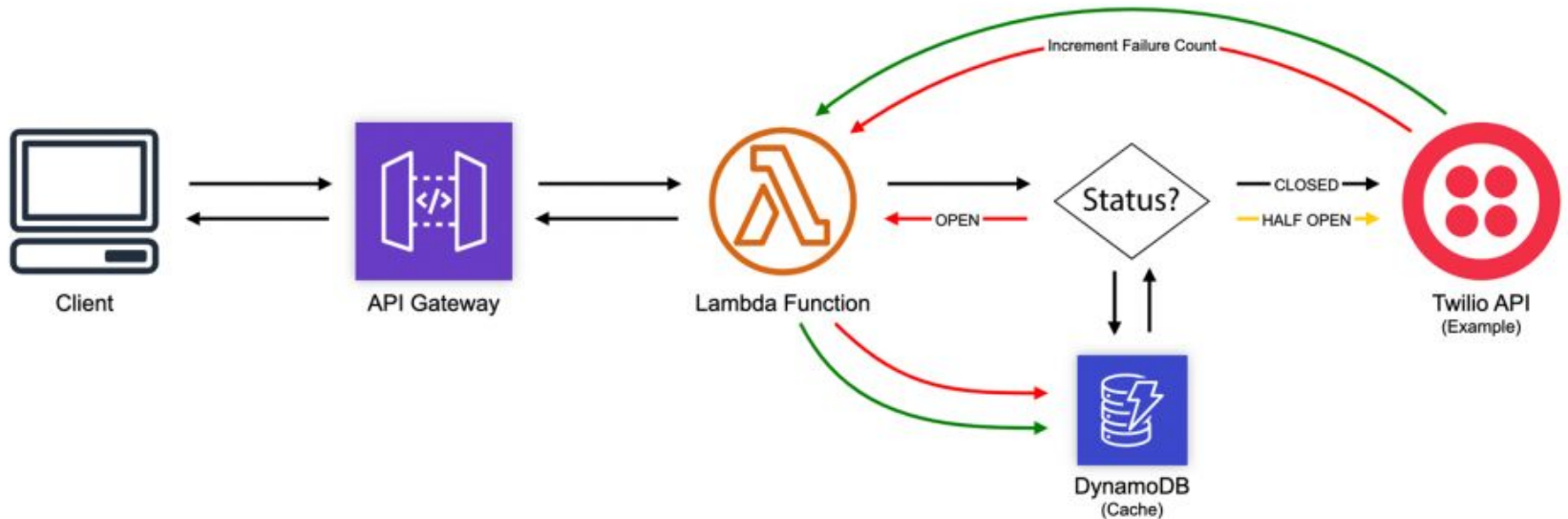
The Storage First Pattern



The Eventually Consistent



The Circuit Breaker





Microfrontend com Lambda Edge

Orquestração vs Coreografia



Orquestração

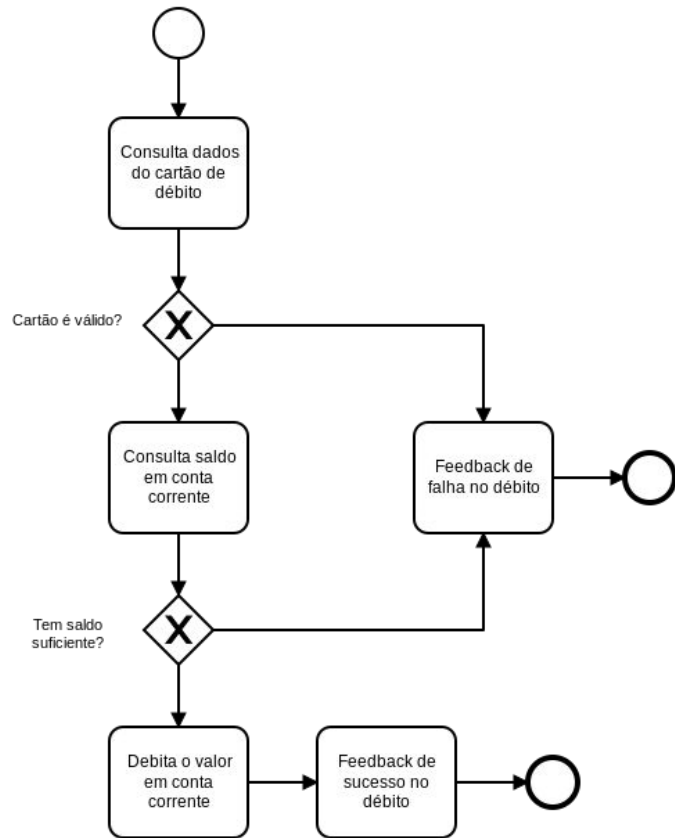
Características

- É como uma Orquestra Sinfônica
- Quando precisa controlar estados
- Utilizar StepFunctions
 - Máquina de estado
 - Precificação por cada mudança de estado



Orquestração

Fluxo de processamento de débito no cartão



Orquestração

Prós e contras

- **Prós**
 - Monitoramento de ponta a ponta é mais fácil com Step Functions
 - Fácil de implementar timeout
 - Lógica de negócio toda num local, mais fácil de gerenciar
 - O fluxo da orquestração é modelado e controlado por código
- **Contras**
 - Custa \$25 por milhão de transições de estados
 - Ponto único de falha: se o Step Functions cair, o processo não será executado



Coreografia

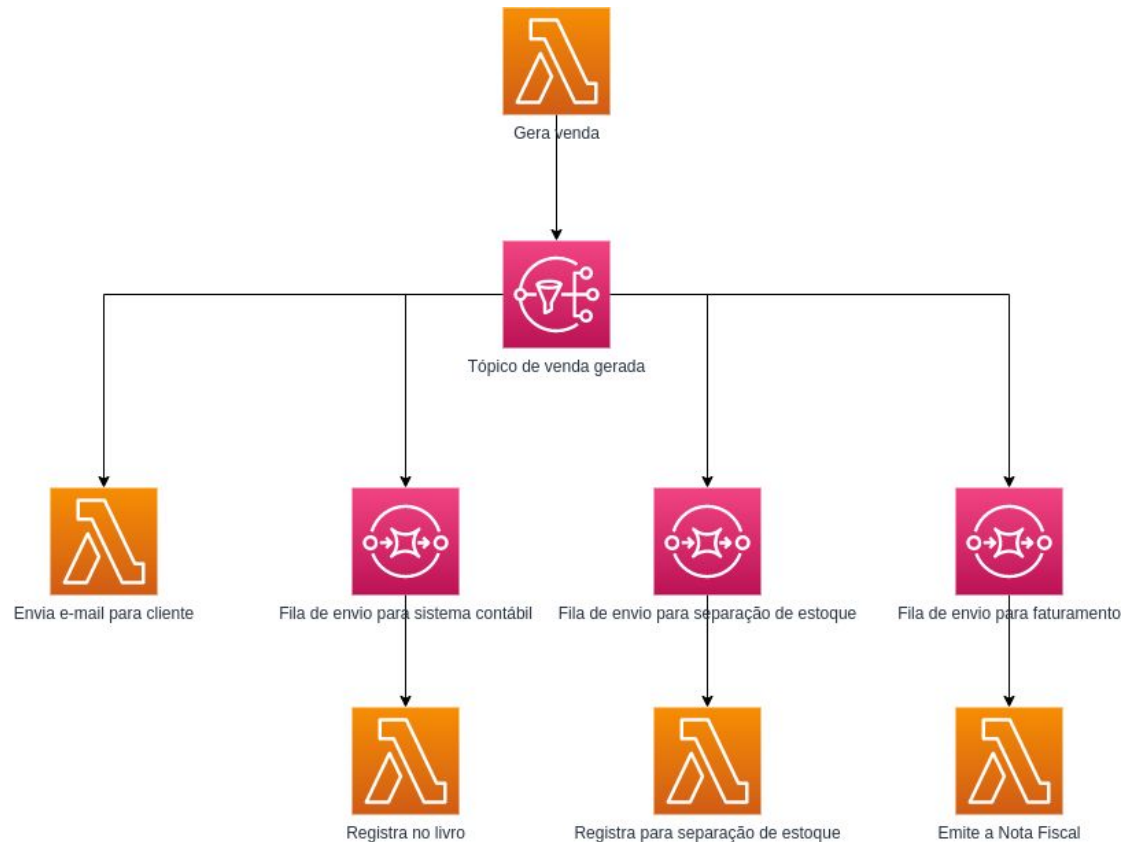
Características

- É como um grupo de dança
- Quando cada parte atua independentemente
- Utilizar EventBridge
 - Filtros
 - Múltiplos destinos
 - Recuperação de falha
 - Precificação: Grátis para eventos de serviços da AWS



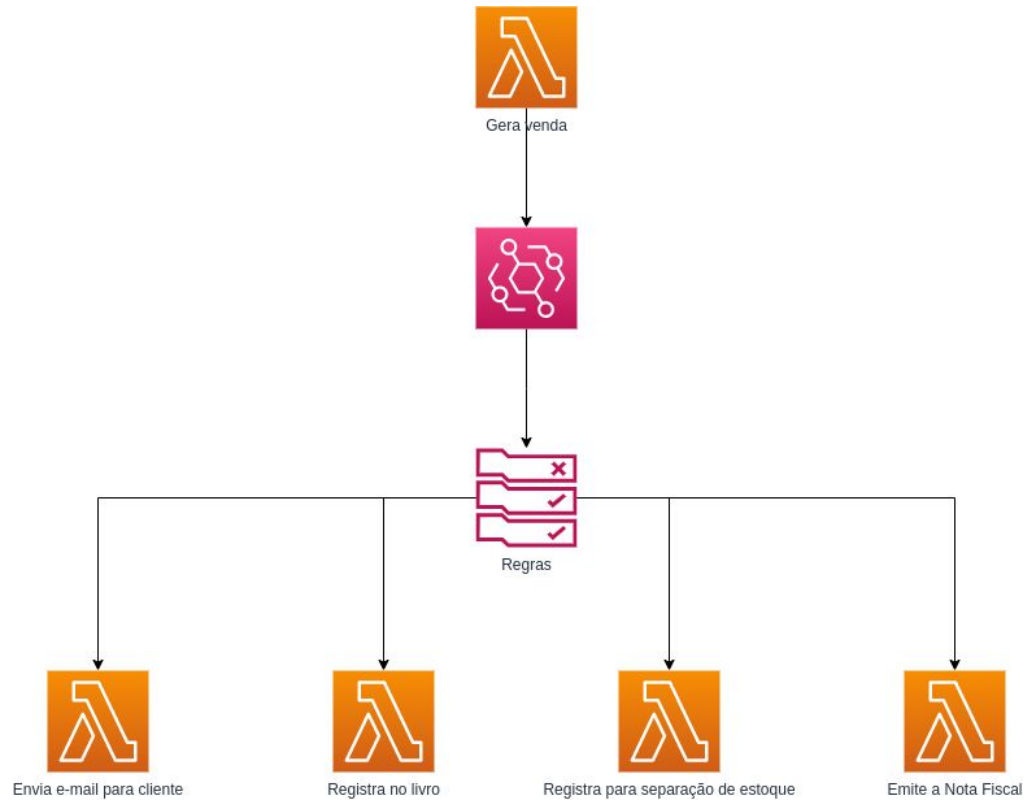
Coreografia

Usando SNS



Coreografia

Usando EventBridge



Coreografia

Prós e contras

- **Prós**
 - Cada etapa do processo pode ser mudado independentemente
 - Cada etapa do processo pode ser escalado independentemente
 - Baixo acoplamento, fácil de adicionar novas features
- **Contras**
 - Monitoramento de ponta a ponta é mais difícil
 - Mais difícil de implementar timeouts
 - Processo não é modelado explicitamente



Testes e Depuração



Testes e Depuração

- AWS Toolkits
 - AWS Toolkit for JetBrains
 - AWS Toolkit for PyCharm
 - AWS Toolkit for IntelliJ
 - AWS Toolkit for Visual Studio Code
- Ferramentas
 - Serverless Framework
 - SAM
 - Localstack
 - Moto



Depuração

- Logs e Alertas no CloudWatch
- X Ray
 - Cold start
 - Depuração de blocos



Repositório



Repositório

- Repositório por contexto
 - Divisão semelhante ao criar um microsserviço
- Template do SAM, CloudFormation ou Serverless Framework no mesmo repositório
 - Utilizar composição de templates
 - Recursos genéricos no arquivo raiz
 - Recursos específicos no arquivo dentro do pacote

`functions:`

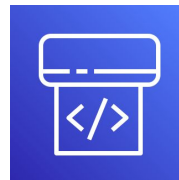
- `${file(number-in-words-java/serverless.yml)}`
- `${file(number-in-words-nodejs/serverless.yml)}`
- `${file(number-in-words-python/serverless.yml)}`

CI/CD e IaC

CI/CD e IaC

Build & Deploy

- Ferramentas
 - SAM
 - CDK
 - CloudFormation
 - Serverless Framework
- CodePipeline
 - CodeCommit
 - CodeBuild
 - Deploy diversos
 - S3
 - ECS
 - Plugins personalizados



Bônus



Bônus

Desafio com os gestores

- Medo do diferente/novo
 - Discurso tem que focar nos principais ganhos para o negócio
 - Time to market
 - Alta disponibilidade
 - Escalabilidade
 - Foco no negócio
- Cálculo dos custos
 - TCO (Total Cost of Ownership)
 - Gastos com recursos humanos
 - Licenças

Dicas práticas



Dicas práticas

- Comece logo a trabalhar com serviços serverless
- Configure o timeout das funções e certifique que está encerrando o processo
- Não leve ao pé da letra a granularidade
- Monitore os gastos, se não seu cartão precisa ser *limitless* (piada ruim)
- Pense nos serviços secundários, eles podem não aguentar a escala das funções
- Consumo de memória e tempo de execução importam
- Divulgue seu case por mais simples que seja. #shameless

Referências

- Artigo sobre definição de serverless:
<https://www.jeremydaly.com/stop-calling-everything-serverless>
- Artigo sobre Orquestração e Coreografia:
<https://theburningmonk.com/2020/08/choreography-vs-orchestration-in-the-land-of-serverless>
- Sem Servidor podcast: <https://semservidor.com.br>
- Serverless Chats: <https://www.serverlesschats.com>
- Talk Serverless Framework Plugins: <https://youtu.be/DtuvHRpMYSY>
- Meu GitHub: <https://github.com/epiresdasilva>

Obrigado!

Evandro Pires da Silva

<https://www.linkedin.com/in/epiresdasilva/>

