

# INTRODUÇÃO A APLICAÇÕES **SERVERLESS** NA AWS



Gabriel Prando;

Engenheiro de plataforma sênior no iFood;

Engenheiro de computação pela UTFPR e  
mestrando no PPGEEC;

Ex presidente da OCCAM Engenharia;

Em qualquer rede com @prandogabriel.



# Sumário

- Histórico modelos de computação;
- Conceitos sobre serverless;
- Comparação com modelos de PaaS, Containers e VMs;
- Casos de uso;
- Prós e contras;
- AWS e serviços;
- Serverless framework.

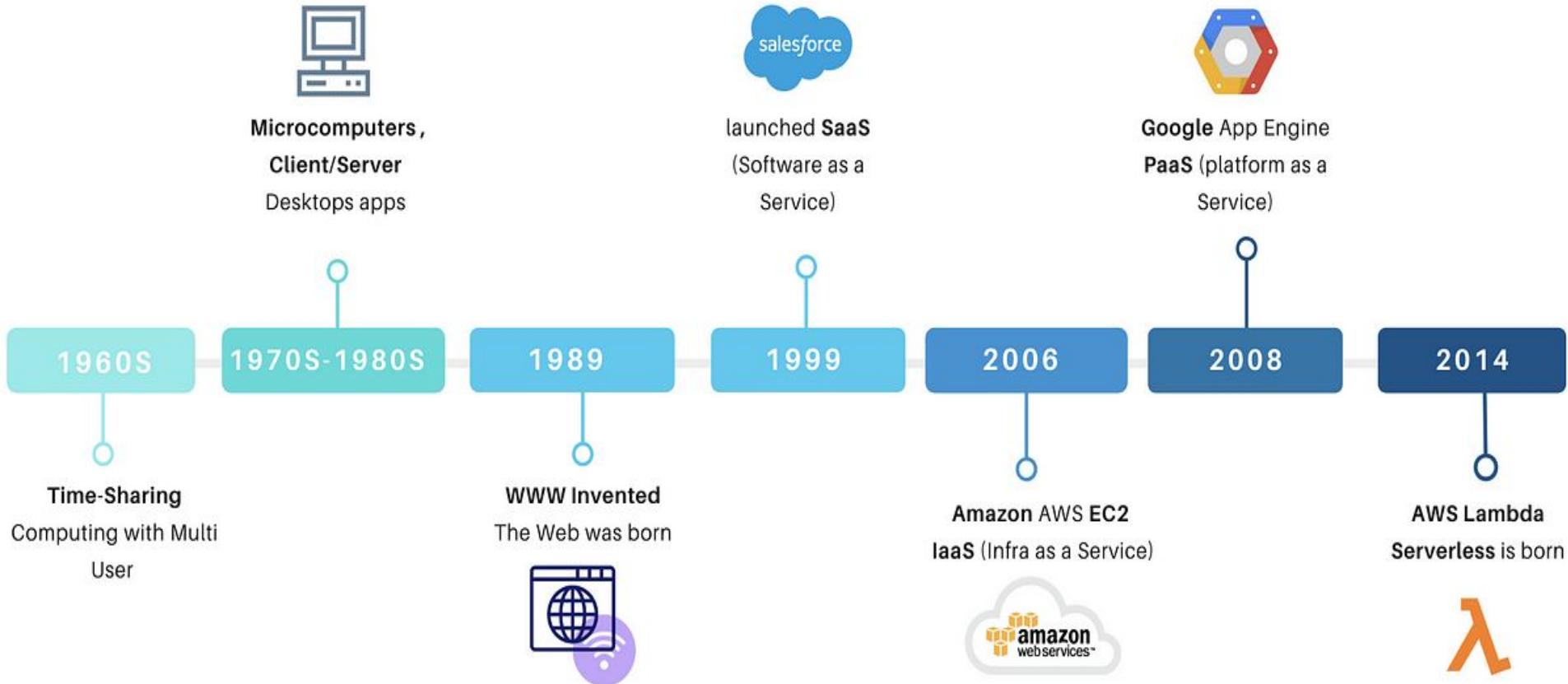
# Antes de falar sobre serverless...

Que tal vermos um pouco sobre a  
evolução dos modelos de computação



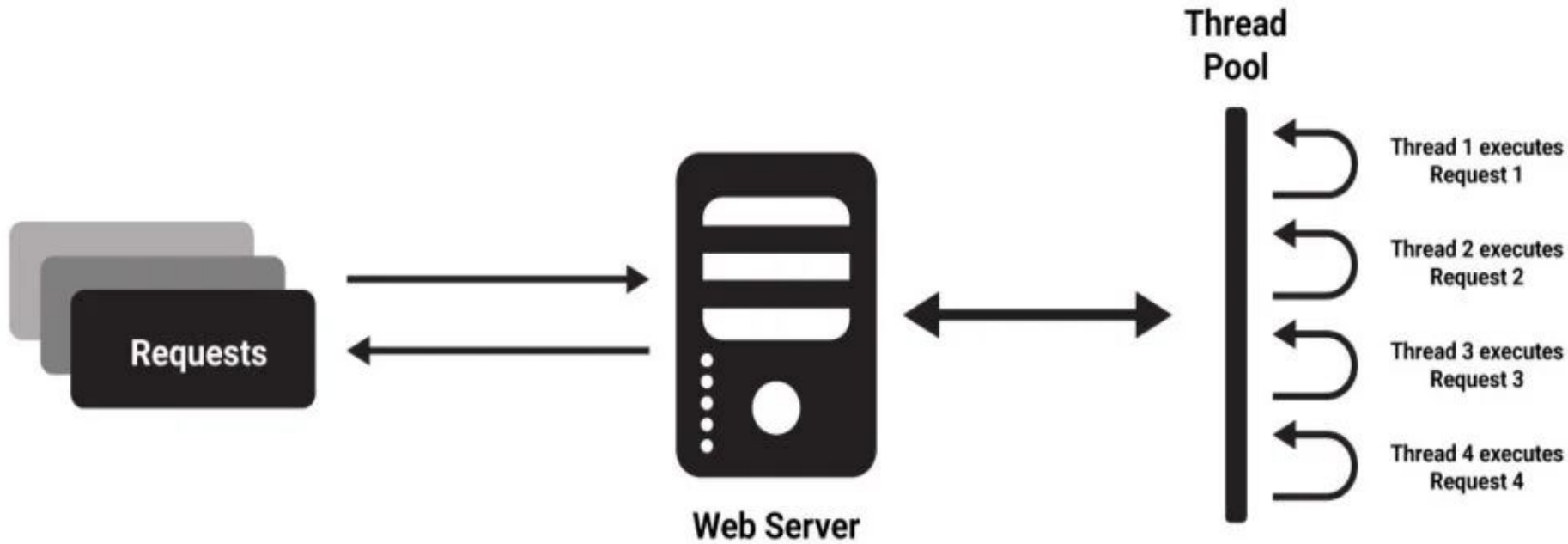
# HISTORY OF COMPUTING

journey towards serverless ...



— **WEB server tradicional**





# **Serverless**

Sem servidor, porém depende





**O que é serverless  
computing? (computação  
sem servidor)**



# **SERVERLESS**

**IS "NOT" NO SERVER. ITS NO SERVER  
FOR "YOU" TO MANAGE**

**'Sem servidor' descreve a experiência.**

**Os servidores são invisíveis para o desenvolvedor, que não os vê, gerencia ou interage, mas eles existem sim.**



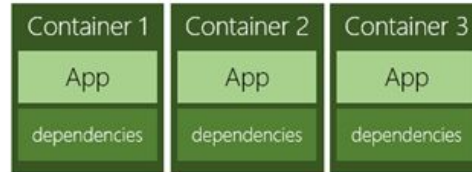
**SERVERLESS  
IS NOT REALLY SERVERLESS!**



## — VMs vs. Containers vs. Serverless



Virtual machines



Containers



Serverless



**FUNCTIONS, FUNCTIONS**

**EVERYWHERE**

## — Premissas

- Orientado a eventos / triggers
- Execução do código não gerenciado pela gente
- Agnóstico a tecnologia
- Pague somente pelo que usar
- Mais fácil de desenvolver = mais tempo para pensar
- Fácil elasticidade e escalabilidade
- Maior agilidade

## — Serverless vs PaaS vs. Containers vs. VMs

### **Tempo de provisionamento**

Milissegundos para serverless, versus segundos ou minutos para os outros modelos

### **Carga administrativa**

Zero para serverless, médio a pesado para PaaS, contêineres e VMs.





## — Serverless vs. PaaS vs. Containers vs. VMs

### Manutenção

Serverless e PaaS são gerenciadas 100% pelo provedor. Já

contêineres e VMs exigem manutenções significativas no SO, imagens de contêineres, conexões, etc.

### Escala

Simple para Serverless, um pouco mais manual e complexo para os demais



# Prós e contras

Olhar por outra  
perspectiva



## — Prós

Experiência de desenvolvimento

Pague pelo uso

Agnóstico a tech

DevOps simplificado

Desempenho

Visibilidade uso

## — Contrás

Latência inaceitável  
para determinados  
aplicativos

Custos mais altos  
para cargas de  
trabalho estáveis ou  
previsíveis

Problemas de  
monitoramento e  
depuração

Aprisionamento do  
fornecedor

## — Casos de uso

- Triggers baseadas em tarefas
- Rest APIs
- Processamento assíncrono
- Checkers de segurança
- CI/CD
- Processamento de dados (ETL)
- Processamento e manipulação de imagens/vídeos
- Startups (economia, foco dev, fácil desenvolvimento e manutenção)

## **E como a gente começa a usar serverless?**

A resposta é simples



## — Encontre um provedor de nuvem



## — E um serviço específico de FaaS



Back4App



Parse



Kinvey



AWS Lambda



Azure Functions

### Serverless Hosting Providers



Cloudflare Workers



Google Cloud  
Functions



IBM Cloud  
Functions



Oracle Functions



Knative



## — Uma ótima opção de mercado



serverless



Lambda

## — AWS

- Oferece serviços de computação em nuvem
- Entre as principais clouds do mercado
- Pioneira em FaaS
- Regiões/servers por todo o mundo +30 locais nos 5 continentes



# Serviços da AWS que rodam no modelo — serverless

## Compute



AWS Lambda

## Messaging



Amazon SQS

## Integration



Amazon  
API Gateway

## Security



Amazon  
Cognito

## Monitoring



Amazon  
Cloudwatch

## Database



Amazon  
DynamoDB

## Analytics



Amazon  
Kinesis



AWS Fargate



Amazon SNS



AWS Step  
Functions



AWS Key  
Management  
Service



Amazon  
Quickstart

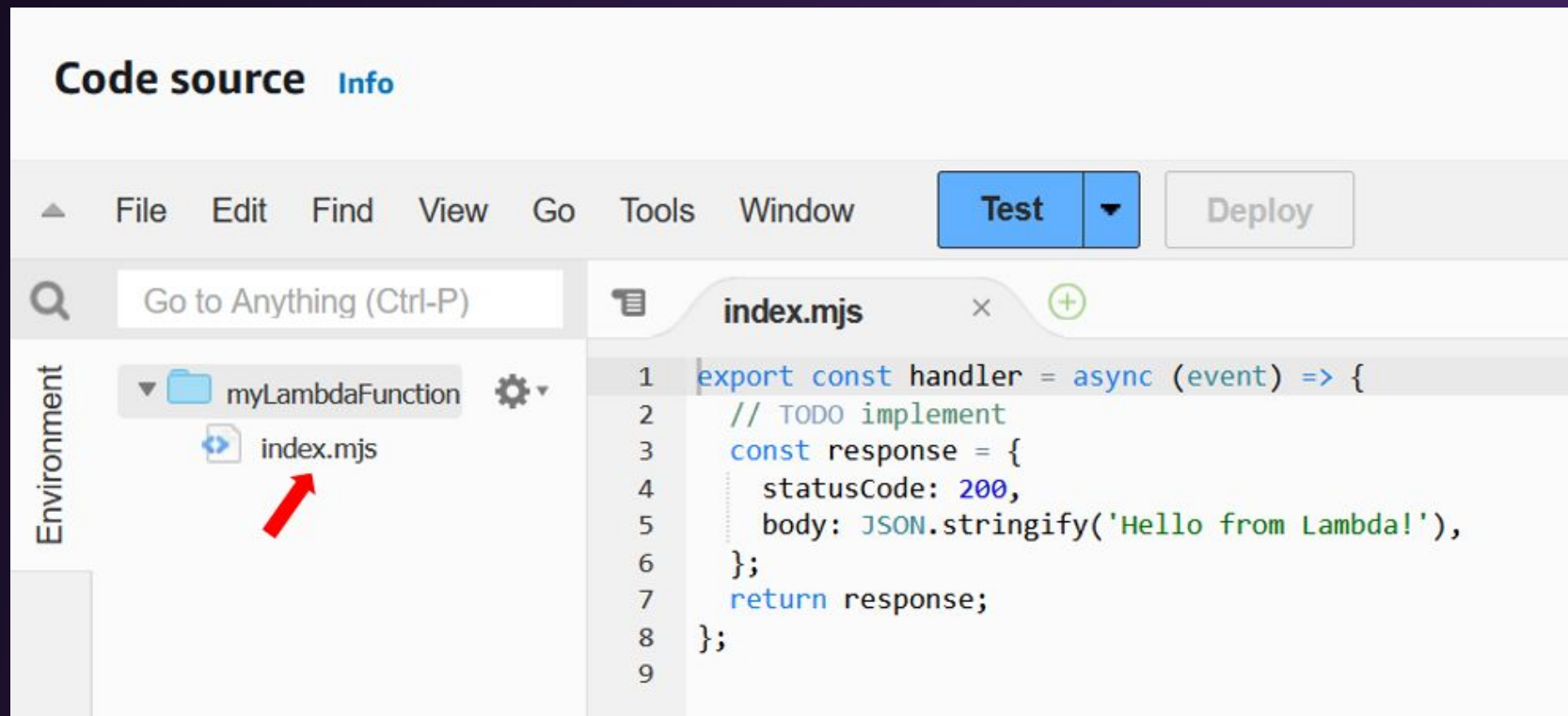


Amazon  
Aurora



Amazon  
Athena

## — Hello world AWS Lambda



## — Exemplo função de cálculo de área

```
export const handler = async (event, context) => {  
  
  const length = event.length;  
  const width = event.width;  
  let area = calculateArea(length, width);  
  console.log(`The area is ${area}`);  
  
  console.log('CloudWatch log group: ', context.logGroupName);  
  
  let data = {  
    "area": area,  
  };  
  return JSON.stringify(data);  
  
  function calculateArea(length, width) {  
    return length * width;  
  }  
};
```



## Exemplo função de cálculo de área na prática....

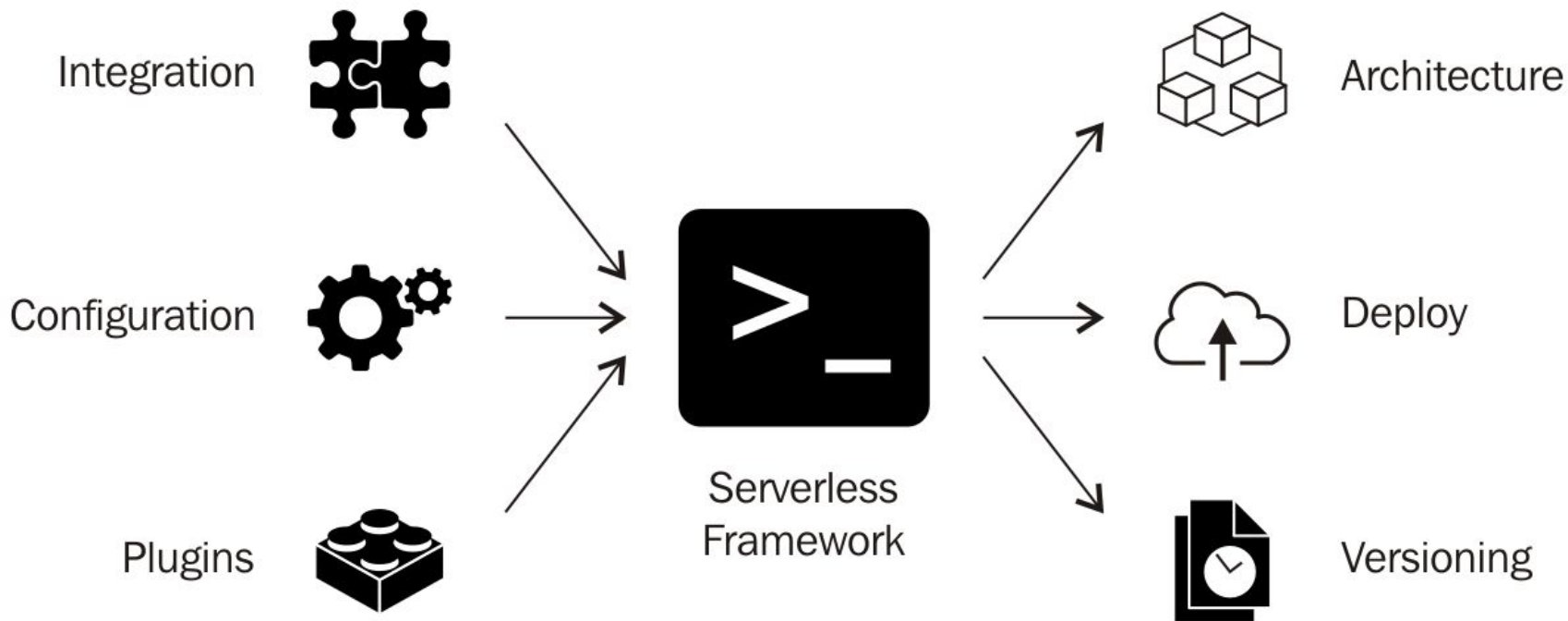
## — Até agora tudo legal, mas..

Qual o problema de desenvolver dessa forma?

E como podemos melhorar isso?



## — Serverless framework





**Bora ver na prática  
tudo isso!**