



<Coders> in

</Rio>

Arquitetura Serveless com Azure Functions





Lucas Marques

Developer @ Braspag Tecnologia em Pagamentos
Founder @ MoreCoddling



medium.com/@lmarquesmoreira



linkedin.com/in/lmarquesmoreira



github.com/lmarquesmoreira



morecoddling.com.br

O que esperar?

- Evolução da Nuvem
 - Breve pontos sobre modelos IaaS, PaaS e FaaS
- Como chegamos as Funções?
 - Algumas arquiteturas conhecidas e suas diferenças

- Azure Functions
 - O que é o Azure Functions?
 - Utilizando o Azure functions

Evolução da Nuvem

<Coders>in </Rio>



Que comecem as VMs



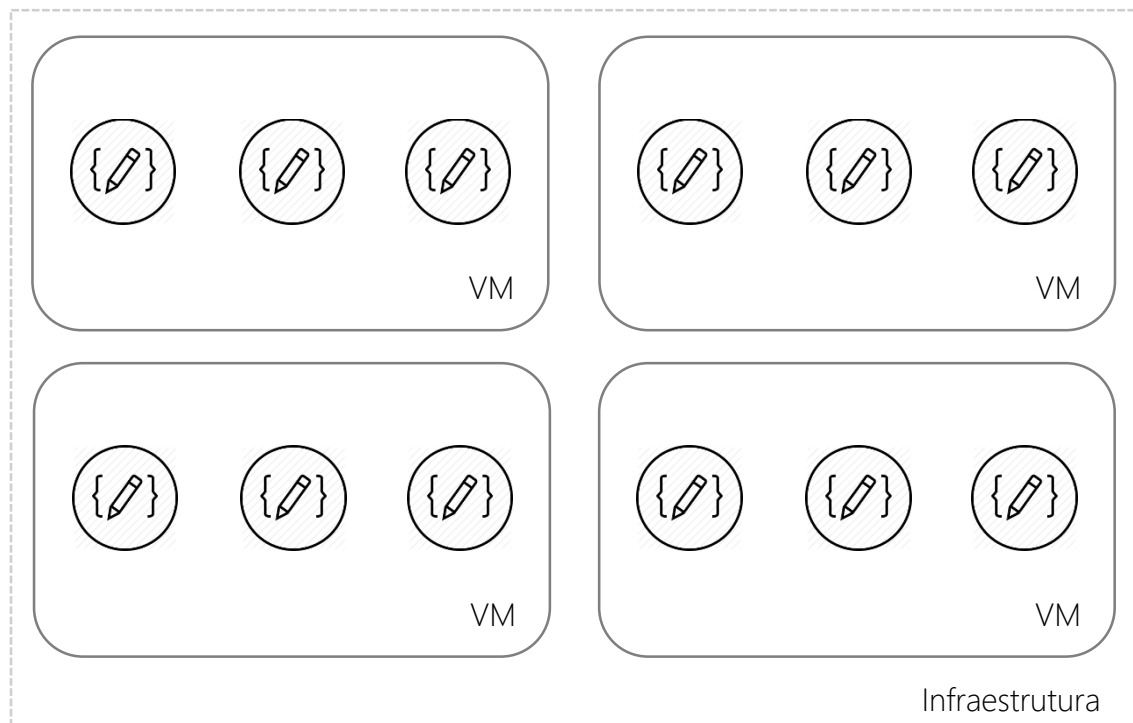
Vemos o modelo IaaS (Infraestrutura como serviço).

Onde temos basicamente, VMs altamente configuráveis e nelas, executamos nosso código.

Nosso modelo de escalabilidade é aumentando o número de instâncias dessas VMs.

Não se preocupem com as Vms

"You deploy your apps but try not to think about the Virtual Machines underneath. They exist, but we pretend they don't until forced." by Scott Hanselman - Microsoft

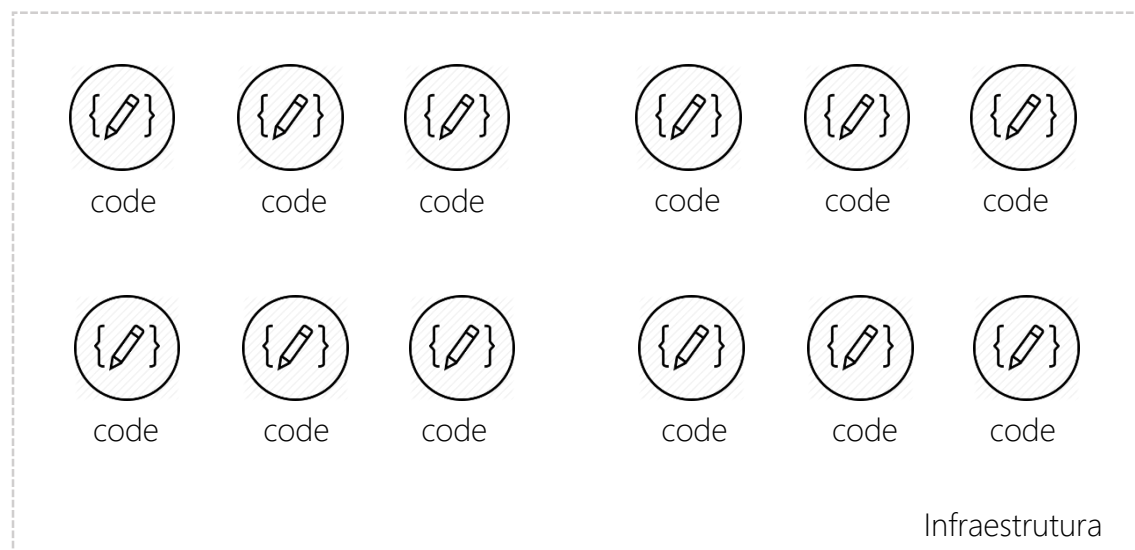


Vemos o modelo PaaS
(Plataforma como serviço).

Em síntese é:

*Gerencie os aplicativos e
serviços que a nuvem cuida do
resto.*

Esqueçam as VMs



*Basicamente é um PaaS, só que de alto nível,
zero preocupação com as VMs.*

Inicia-se a era do Serverless, também conhecida como FaaS, Function as a Service.

Que são "pequenas peças discretas de software executadas na nuvem" by Scott Hanselman – Microsoft

Arquiteturalmente

Como chegamos as Funções?

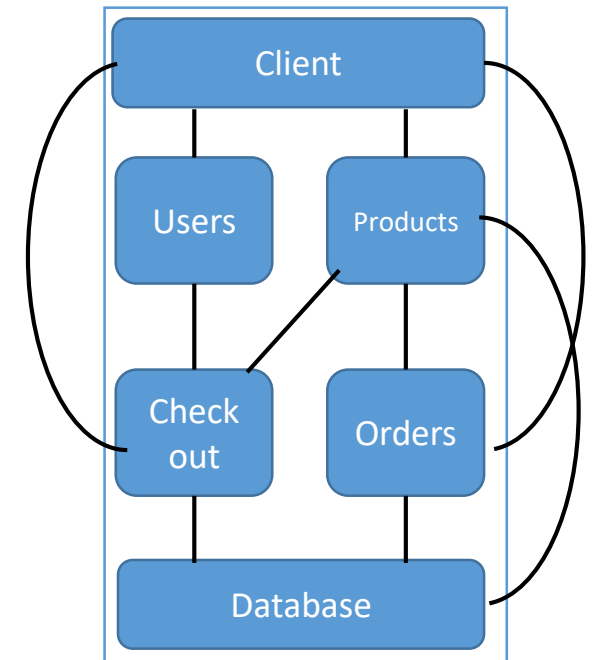
Um breve olhar sobre os tipos de arquiteturas mais conhecidas...

Arquitetura Monolítica

É o padrão mais utilizado para o desenvolvimento de aplicações corporativas e que possui como pontos fortes o desenvolvimento, testes e implantação mais simples.

Já os pontos fracos...

- Difícil de gerenciar seu desenvolvimento, pode se tornar uma bicho de sete cabeças.
- Uma pequena alteração em um módulo do projeto, impacta **todo o projeto** exigindo um deployment da aplicação inteira. **Deploys sempre críticos.**
- Dificulta a utilização de uma entrega contínua.
- Limita a adoção de novas tecnologias



Arquitetura baseada em microservices

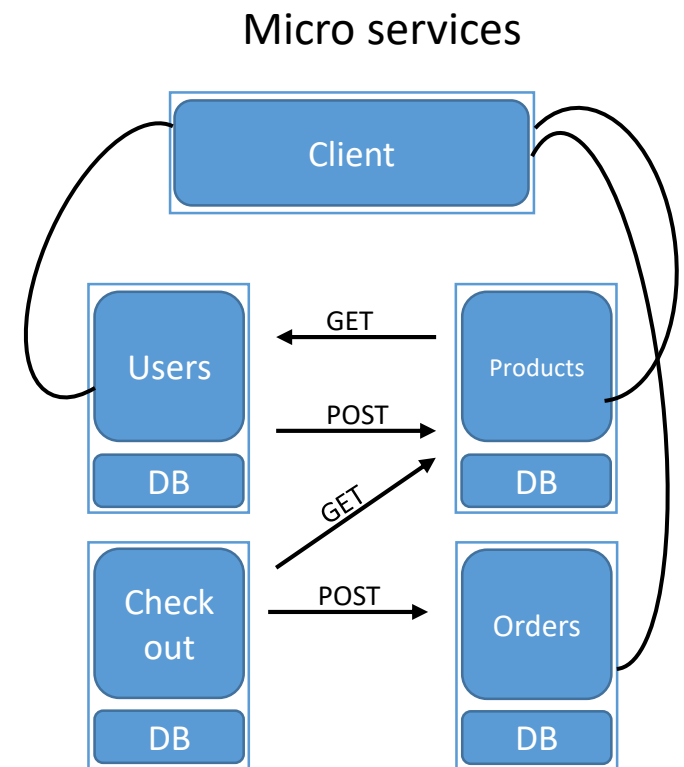
"A ideia da arquitetura de microservices é projetar aplicações grandes, complexas e duráveis em um conjunto coeso de serviços que evoluem ao longo do tempo." by
Chris Richardson – Founder at CloudFoundry

Pontos fortes

- Serviços individuais são mais fáceis de entender e manter.
- Deployment independente
- Adoção de novas tecnologias é bem mais simples
- Possibilitam a Entrega contínua

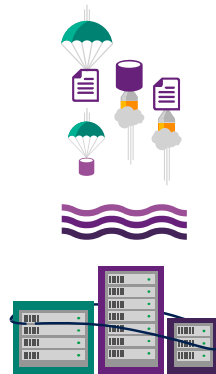
Pontos fracos

- Aplicações são bem mais complexas, logo problemas mais complexos.
- Exigem um alto nível de automação
- Os dados estão distribuídos.

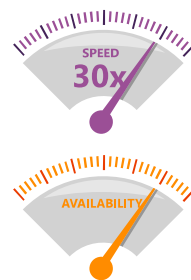


Arquiteturas sem servidor Serverless

"Uma aplicação que roda em ambientes, como containers, sem estado que são disparadas através de eventos, efêmeros (só podem durar uma vez a cada invocação) e são completamente gerenciadas por terceiros" by Mike Roberts – CoFounder at Symphonia



Abstração dos
servidores



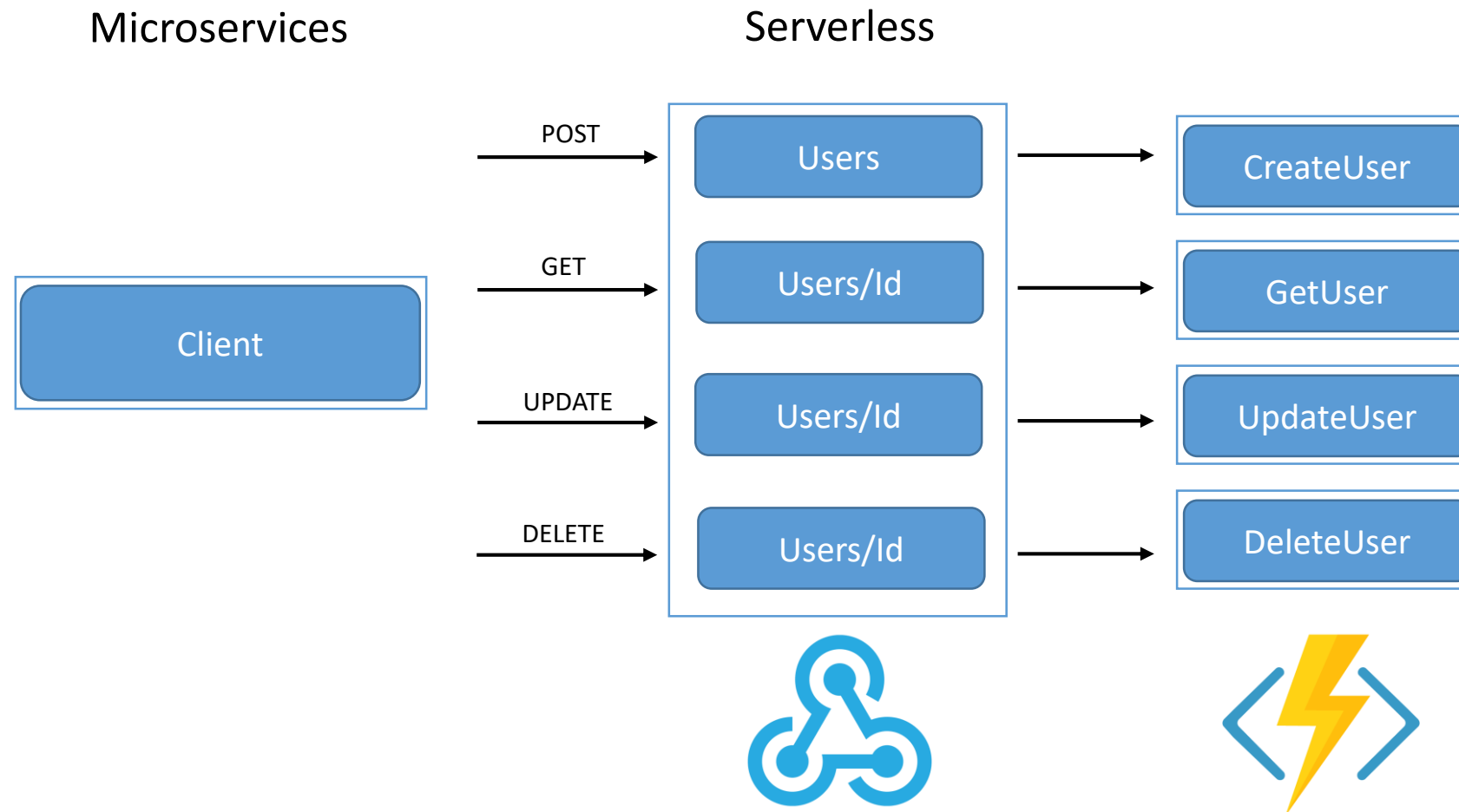
Baseada em eventos/
scale instantâneo



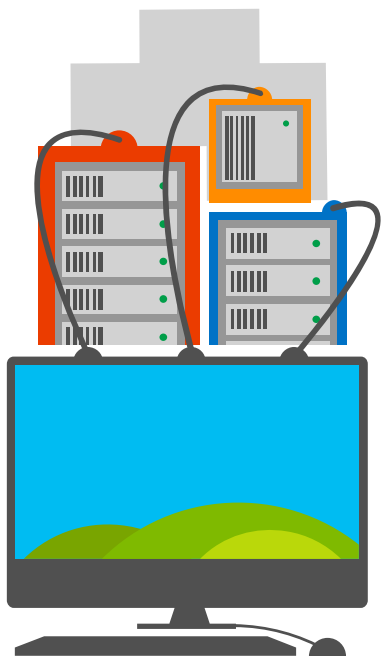
Faturamento por tempo
de execução da função

Arquitetura sem servidor

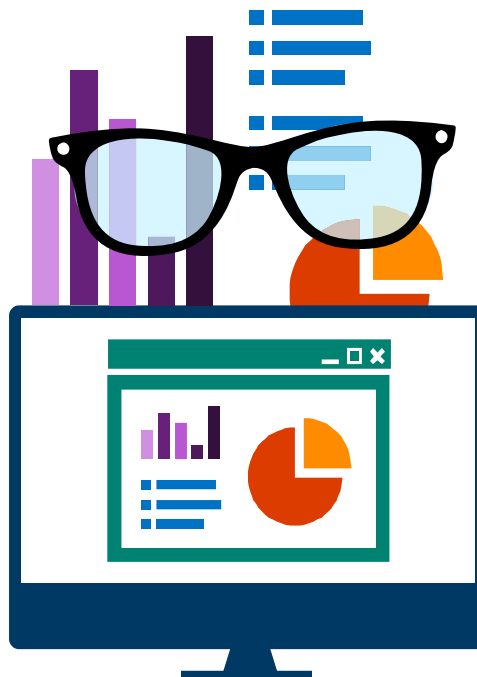
Serverless



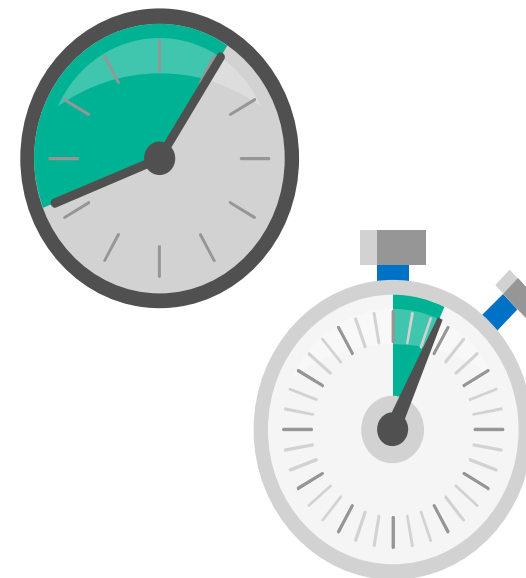
Benefícios do Serverless



DevOps
reduzido



Foco na lógica
do negócio



Reduz o Time To
Market

Créditos: Yochay Kiriathy - Microsoft

Azure Functions

<Coders>in </Rio>



O que é o Azure functions?

Arquitetura sem servidor, serviço de computação orientado a eventos



= microservice sem servidor

Suas features

- **Escolha uma linguagem**
 - Escreva as funções usando C#, F#, Node.js, Python, PHP, batch, bash, ou qualquer executável
- **Pague pelo uso**
 - Pague somente pelo tempo gasto pela sua função
- **Suporta suas dependências**
 - Suporta dependências via NPM ou Nuget
- **Segurança integrada**
 - Funções acionadas por HTTP podem ser protegidas via OAuth
- **Integração simplificada**
 - Funções podem ser integradas através de serviços do Azure ou de terceiros
- **Desenvolvimento flexível**
 - Codificação diretamente pelo portal ou através de integração continua
- **Open Source**
 - Runtime é openSource e está disponível no github



Integrações

Esses serviços podem acionar sua função e iniciar a execução, ou podem servir como entrada e saída.

- Azure DocumentDB
- Azure Event Hubs
- Azure Mobile Apps (tables)
- Azure Notification Hubs
- Azure Service Bus (queues and topics)
- Azure Storage (blob, queues, and tables)
- GitHub (webhooks)
- On-premises (using Service Bus)
- Twilio (SMS messages)



Triggers e Bindings possíveis

Um **trigger**, define como uma função será invocada.

Input e Output **bindings**, fornecem uma forma declarativa de conectar os dados dentro do função.

Type	Service	Trigger	Input	Output
Schedule	Azure Functions	✓		
HTTP (REST or webhook)	Azure Functions	✓		✓*
Blob Storage	Azure Storage	✓	✓	✓
Events	Azure Event Hubs	✓		✓
Queues	Azure Storage	✓		✓
Tables	Azure Storage		✓	✓
Tables	Azure Mobile Apps		✓	✓
No-SQL DB	Azure DocumentDB		✓	✓
Push Notifications	Azure Notification Hubs			✓

(* - The http out binding requires an http trigger)



Demo

<Coders>in </Rio>



Obrigado 😊