

DANIEL NORIO TAKASU REBELO  
LUCAS ARTHUR FELGUEIRAS  
LUIZ GUSTAVO DOS SANTOS  
VICTOR FRANÇA FERREIRA

# COMPUTAÇÃO EM NUVEM: INTRODUÇÃO

São Paulo  
2018

**DANIEL NORIO TAKASU REBELO  
LUCAS ARTHUR FELGUEIRAS  
LUIZ GUSTAVO DOS SANTOS  
VICTOR FRANÇA FERREIRA**

## **COMPUTAÇÃO EM NUVEM: INTRODUÇÃO**

Trabalho apresentado à Escola Politécnica  
da Universidade de São Paulo para a  
disciplina de Engenharia de Sistemas.

São Paulo  
2018

**DANIEL NORIO TAKASU REBELO**  
**LUCAS ARTHUR FELGUEIRAS**  
**LUIZ GUSTAVO DOS SANTOS**  
**VICTOR FRANÇA FERREIRA**

## **COMPUTAÇÃO EM NUVEM: INTRODUÇÃO**

Trabalho apresentado à Escola Politécnica  
da Universidade de São Paulo para a  
disciplina de Engenharia de Sistemas.

Área de Concentração:  
Engenharia de Sistemas

Orientador:  
Prof<sup>ª</sup>. Dr<sup>ª</sup>. Selma Shin Shimizu  
Melnikoff

# RESUMO

A tecnologia de computação em nuvem é uma tendência para as novas aplicações do mundo. Grandes empresas estão migrando suas estruturas para a nuvem, outras constroem ambientes para abrigar essas novas aplicações. O objetivo do trabalho consiste em compreender melhor como funciona essas tecnologias, sua motivação e as principais opções disponíveis no mercado para uso.

**Palavras-Chave** – Nuvem, Arquitetura, Computação, Engenharia, Programa.

# ABSTRACT

Cloud computing technology is a trend for new applications around the world. Large companies are migrating their structures to the cloud, others build environments to house these new applications. The objective of the work is to better understand how these technologies work, their motivation and the main options available in the market to use.

**Keywords** – Cloud, Architecture, Computing, Enginnering, Software.

## LISTA DE FIGURAS

1	Arquitetura Básica do Heroku(?) . . . . .	22
2	Ligação entre <i>router</i> e <i>dynos</i> do Heroku(?) . . . . .	23
3	Exemplo do <i>Dashboard</i> do Heroku com seus recursos . . . . .	24
4	Regiões disponibilizadas pela Azure . . . . .	26
5	Camadas de Segurança da Azure . . . . .	27

## LISTA DE TABELAS

# SUMÁRIO

<b>Parte I: INTRODUÇÃO</b>	<b>8</b>
<b>1 Objetivo</b>	<b>9</b>
1.1 Introdução . . . . .	9
<b>2 Motivação</b>	<b>10</b>
2.1 Contexto . . . . .	10
<b>3 Justificativa</b>	<b>11</b>
3.1 Introdução . . . . .	11
<b>Parte II: ASPECTOS CONCEITUAIS</b>	<b>12</b>
<b>4 Necessidades</b>	<b>13</b>
4.1 Introdução . . . . .	13
<b>5 Funcionamento Básico</b>	<b>14</b>
5.1 Introdução . . . . .	14
<b>6 Conceitos Usados</b>	<b>15</b>
6.1 Introdução . . . . .	15
<b>Parte III: ASPECTOS TÉCNICOS</b>	<b>16</b>
<b>7 Estrutura</b>	<b>17</b>
7.1 Introdução . . . . .	17
<b>8 Evolução Tecnológica</b>	<b>18</b>
8.1 Introdução . . . . .	18



<b>Parte IV: SOLUÇÕES DO MERCADO</b>	<b>19</b>
<b>9 Amazon Web Services</b>	<b>20</b>
9.1 Introdução . . . . .	20
<b>10 Google Cloud</b>	<b>21</b>
10.1 Introdução . . . . .	21
<b>11 Heroku</b>	<b>22</b>
11.1 Introdução . . . . .	22
11.2 Funcionamento básico . . . . .	22
11.3 Tarifação . . . . .	23
11.4 <i>Containers</i> . . . . .	24
11.5 Conclusão . . . . .	25
<b>12 Azure</b>	<b>26</b>
12.1 Introdução . . . . .	26
12.2 Vantagens e diferenciais . . . . .	27
12.3 Desvantagens . . . . .	28
12.4 Conclusão . . . . .	28

# PARTE I

## INTRODUÇÃO

# **1 OBJETIVO**

## **1.1 Introdução**

## 2 MOTIVAÇÃO

### 2.1 Contexto

## **3 JUSTIFICATIVA**

### **3.1 Introdução**

# PARTE II

## ASPECTOS CONCEITUAIS

## 4 NECESSIDADES

### 4.1 Introdução

## **5    FUNCIONAMENTO BÁSICO**

### **5.1   Introdução**



## **6 CONCEITOS USADOS**

### **6.1 Introdução**

# PARTE III

## ASPECTOS TÉCNICOS

## 7 ESTRUTURA

### 7.1 Introdução

## 8 EVOLUÇÃO TECNOLÓGICA

### 8.1 Introdução

# **PARTE IV**

## **SOLUÇÕES DO MERCADO**

## 9 AMAZON WEB SERVICES

### 9.1 Introdução

## 10 GOOGLE CLOUD

### 10.1 Introdução

## 11 HEROKU

### 11.1 Introdução

O Heroku é uma plataforma de computação em nuvem conhecida no mercado, com a possibilidade de subir aplicações nas linguagens Ruby, Node.js, Java, Python, Clojure, Scala, Go e PHP. O grande destaque do Heroku está na facilidade em subir uma aplicação com facilidade e de maneira gratuita, o que possibilita testar e validar ideias básicas antes de escalar de fato.

A estrutura básica do Heroku funciona com o uso de *dynos*, que servem tanto para hospedar sua aplicação principal quanto máquinas auxiliares para serviços externos e/ou paralelos. Porém, recentemente o Heroku disponibilizou também a estrutura de *containers* em Docker (tecnologia de gerenciamento de *containers*) (assim como seus principais concorrentes), gerando maior flexibilidade para o serviço.

### 11.2 Funcionamento básico

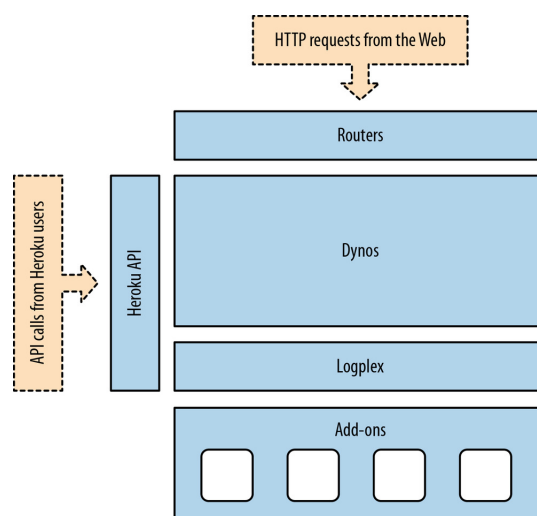


Figura 1: Arquitetura Básica do Heroku(?)



O funcionamento do Heroku consiste no uso de *dynos* para hospedar as aplicações e nos *routers* para tratar e encaminhar as requisições dos usuários. Além disso, o próprio Heroku disponibiliza extensões para gerenciar sua aplicação, como por exemplo o gerenciador de IP estático, ou o banco de dados necessário para a aplicação.

Um *dyno* é um *container* pronto, com 512MB de memória RAM, responsável por abrigar uma ou mais instâncias de sua aplicação, permitindo escalabilidade e tolerância de erros. Além disso, os *dynos* permitem rodar tarefas de sua aplicação à parte, como filas, requisições assíncronas, entre outros.

Já os *routers* são responsáveis por gerenciar os acessos dos usuários às aplicações correspondentes, dado que não é padrão do Heroku estruturar IP estático para cada *dyno*. Ou seja, a estrutura é responsável por fazer sua aplicação funcionar corretamente, sem acessar aplicações alheias.

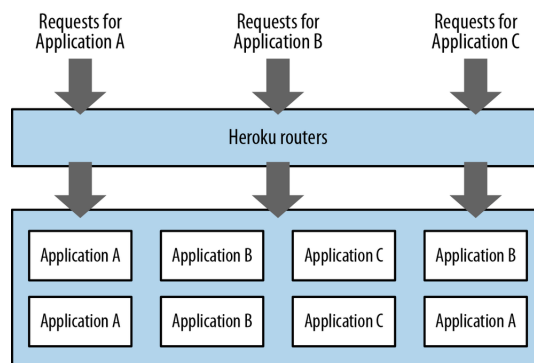


Figura 2: Ligação entre *router* e *dynos* do Heroku(?)

Graças a essa estrutura, é possível, com facilidade, gerenciar versionamento e escalar a aplicação de maneira horizontal, com novos *dynos* gerenciados pelo *router*, dando uma popularidade considerável ao Heroku, em especial nas aplicações onde velocidade de validar hipóteses é o principal foco.

### 11.3 Tarifação

O Heroku possui um diferencial em relação à tarifação: possui um plano básico gratuito que possibilita testar aplicações de maneira fácil e sem dificuldades de expansão. Essencialmente, a cobrança do Heroku ocorre via uso de seus *dynos*-hora. Além disso, há a cobrança pelas extensões usadas dentro da aplicação, onde, no geral, há um plano gratuito de testes ou até para projetos pequenos funcionarem com tranquilidade sem necessidades de escalabilidade.

Para qualquer usuário do Heroku, é disponibilizado um pacote de 750 *dynos*-hora durante um mês, ou seja, uma aplicação gratuita usando um simples *dyno* pode durar tranquilamente. Além disso, o Heroku de uso gratuito limita o tempo em que um *dyno* fica ligado de maneira ociosa: passando trinta minutos desde a última requisição feita pelo cliente (e repassada pelo *router*), o *dyno* é derrubado, sendo religado após uma nova requisição.

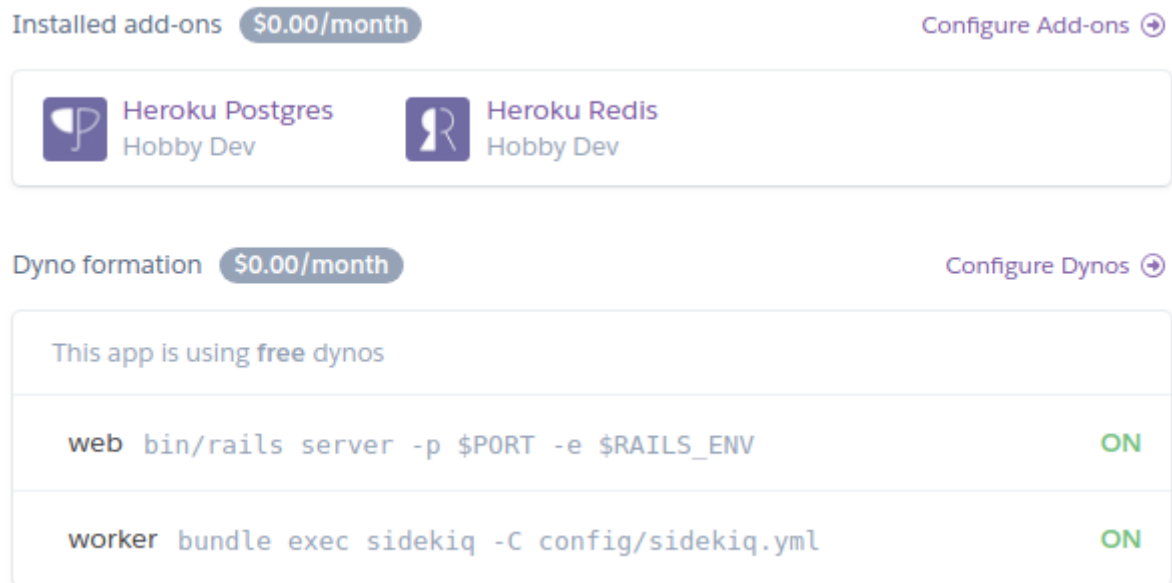


Figura 3: Exemplo do *Dashboard* do Heroku com seus recursos

Além da cobrança do uso dos *dynos*, há também a cobrança para cada extensão utilizada por sua aplicação, o que de fato encarece o custo final para escalar uma aplicação. Por exemplo, o banco de dados relacional padrão do Heroku é gratuito até certo limite de dados, passando desse limite, há a extensão do plano que permite, além de mais dados armazenados, aumenta o fluxo possível de acesso ao banco.

## 11.4 *Containers*

Recentemente, o Heroku possibilitou em seu catálogo de soluções o uso de estruturas isoladas customizadas: os *containers*. *Containers*, segundo o site do Docker(?) (em inglês):

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

Ou seja, são imagens autossuficientes e virtualizadas que permitem rodar aplicações prontas nos mais diversos dispositivos, nos moldes da virtualização.

O Heroku possibilitou, em 2017, uma maneira eficiente de receber *containers* em seu serviço, com o **Heroku Container Registry**(?). Esse serviço facilita colocar em produção as máquinas isoladas contruídas com o uso do *Docker*, aumentando as possibilidades de operação com a plataforma. A tarifação do serviço segue o mesmo padrão do Heroku tradicional, dado o fato de que *dynos* são uma espécie de *containers*.

## 11.5 Conclusão

O Heroku é uma solução que se destaca pela velocidade e facilidade de colocar uma ideia em produção, de maneira gratuita, com possibilidades eficientes de escalabilidade. Porém, financeiramente, não é a melhor solução, com uma tarifação elevada para escalar a aplicação. Além disso, sua estrutura atende apenas aplicações prontas, não permitindo outras estruturas avançadas como o uso de *containers*, sendo incorporado ao catálogo de produtos recentemente.

## 12 AZURE

### 12.1 Introdução

A Azure é uma plataforma de computação em nuvem, agindo como provedor *SaaS*, *PaaS* e *IaaS*. Tem como proprietária a empresa americana *Microsoft*, e disponibiliza serviços como *builds*, testes, *deploys* e gerenciamento de aplicações que estão hospedadas em *data centers* administrados pela própria empresa.

Atualmente possui mais de 600 ferramentas para os mais diversos usos na área da computação, como serviços de armazenamento, soluções para desenvolvimento *Mobile*, administração de dados, messageria, CDN e etc. Além disso, seus *data centers* que constituem 36 regiões ao redor do globo, provendo estabilidade e escalabilidade sem problemas de latência.



Figura 4: Regiões disponibilizadas pela Azure

## 12.2 Vantagens e diferenciais

O primeiro atributo positivo a considerar em relação aos serviços de nuvem da Azure é a confiabilidade da infraestrutura do provedor. Seu Contrato de Nível de Serviço (em inglês, *SLA*) estima uma porcentagem de disponibilidade de 99,95%, o que equivale a pouco mais de quatro horas anuais de tempo de *downtime*. Essa é uma conquista que até mesmo uma solução *on premises* simplesmente não consegue atingir de forma consistente. No entanto, hospedar um aplicativo ou armazenar dados na nuvem geralmente está associado a uma alta disponibilidade - certamente um dos maiores apelos de *cloud computing*. Porém a alta disponibilidade dos Serviços de Nuvem do Azure da *Microsoft* continua a impressionar, inspirando muitas vezes *CIOs* a migrar pelo menos uma parte de suas soluções mais pesadas de dados *on-line*.

Além disso, uma das maiores preocupações das empresas ao migrar para a nuvem é a segurança. A Azure segue o modelo de segurança padrão, que consiste em “detectar, avaliar, diagnosticar, estabilizar e fechar”. Esse modelo, alinhado com outras políticas rígidas de segurança, rendeu à plataforma uma série de certificações de *compliance*, a estabelecendo como líder em segurança de *IaaS*. Fornece proteção em vários níveis e serviços simples e fáceis de usar.

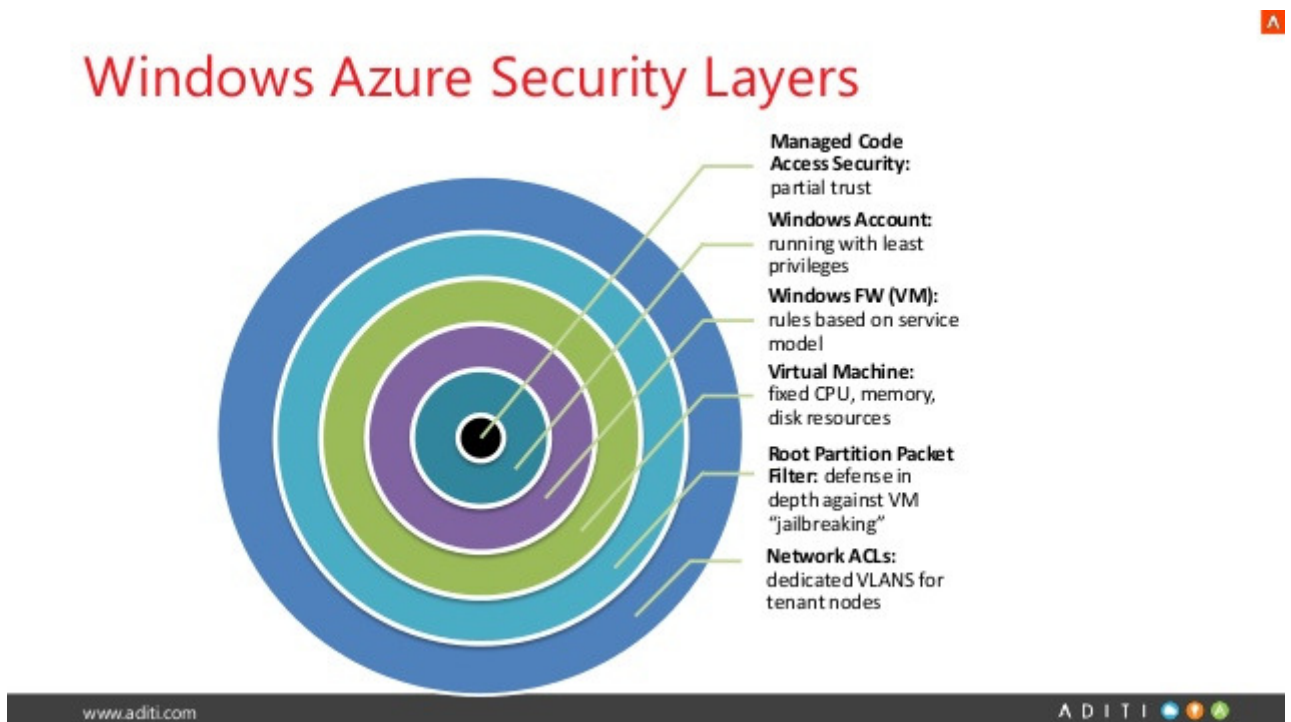


Figura 5: Camadas de Segurança da Azure

Finalmente, os serviços de nuvem da Azure também oferecem a escalabilidade necessária para o crescimento de qualquer solução, fornecendo maior poder computacional conforme necessário. Seja uma pequena empresa, ou uma empresa com a necessidade de um grande *data warehouse*, os serviços de nuvem do Azure têm armazenamento e capacidade computacional para lidar com praticamente qualquer tipo de demanda.

## 12.3 Desvantagens

Ao contrário das plataformas *SaaS*, nas quais o usuário final está consumindo informações (por exemplo, o *Office 365*), a Azure migra o poder computacional de dentro de sua empresa (*datacenter*) ou escritório para a nuvem. Como ocorre com a maioria dos provedores de serviços em nuvem, o Azure precisa ser gerenciado e mantido cuidadosamente, o que inclui aplicação de *patches* e o monitoramento dos servidores.

Ao contrário dos servidores locais, o Azure exige experiência para garantir que todas as partes móveis trabalhem juntas de forma eficiente. Um erro comum dos administradores de negócios que não estão totalmente envolvidos em quão bem (ou mal) seus servidores de nuvem estão operando é o de fazer o chamado *over-provisioning* dos serviços em nuvem. Embora seja um erro corriqueiro, essa falta de conhecimento dos administradores pode custar às empresas milhares de dólares por ano.

## 12.4 Conclusão

No geral, a Azure é uma excelente plataforma de *cloud computing*, ficando atrás de seus concorrentes apenas em alguns poucos pontos. Possui serviços robustos e bem consolidados, bastante segurança, e boa disponibilidade e escalabilidade. Além disso, torna-se quase que uma escolha obrigatória para empresas que já utilizam ou tem familiaridade com ambientes *Microsoft*, devido às inúmeras facilidades extras fornecidas (integrações e possíveis cortes de gastos com licenças).