

Microsserviços: Teoria e prática com o projeto Spring Cloud e Netflix OSS.

Elena A. Araujo

DCC/UFLA

25 de maio de 2018



Elena Augusta Araujo

- Diretora de Qualidade (SetApp);
- Graduada em Ciência da Computação (2016) na **UFV-Campus Florestal.** ❤
- Mestranda em Ciência da Computação na Universidade Federal de Lavras (UFLA).

Contato



elena.araujo@posgrad.ufla.br



<https://www.facebook.com/elena.araujo.524>



<https://www.linkedin.com/in/elena-araujo-546001bb/>

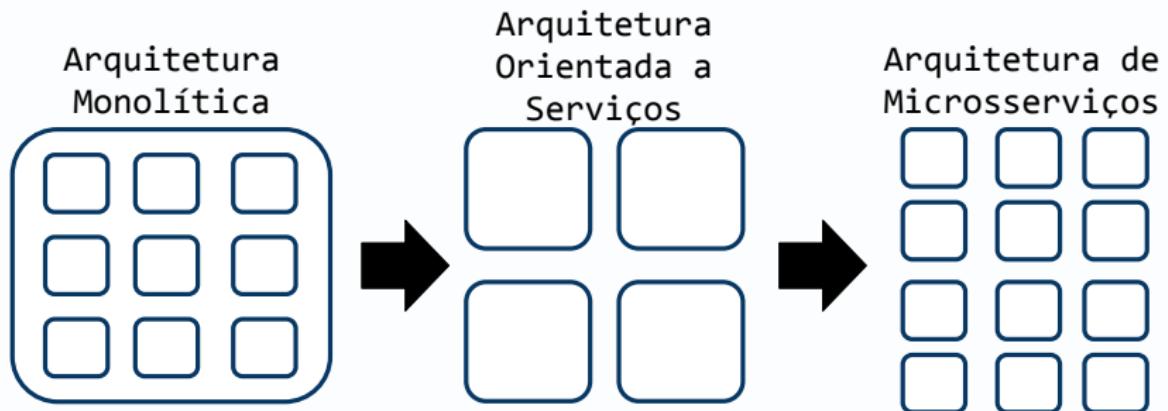




- Afinal, o que são microsserviços?
- Arquitetura Monolítica e Microsserviços;
- Projeto Spring Cloud Netflix OSS;
- Implementação de uma aplicação de controle de vendas.

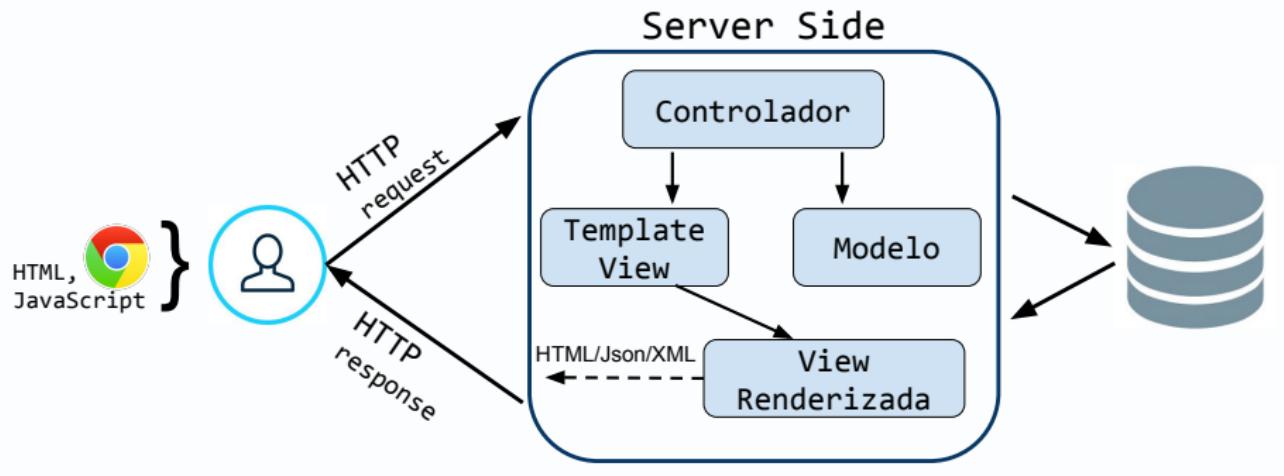
**Microsserviços:
De onde surgiram?
Quem utiliza?
Funciona mesmo??**

Hoje na SECOM!



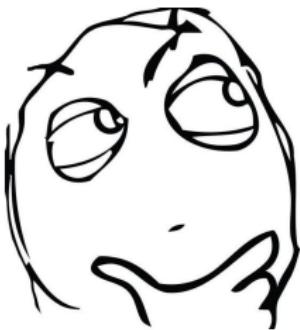
- Consiste de uma **única** unidade lógica executável.
- Aplicações grandes e robustas. **Verdadeiro faz tudo!**
- Em geral, são mantidas por um **único** time e distribuída como um todo pela organização.







Nessa arquitetura, como poderíamos modelar um sistema de controle de vendas??



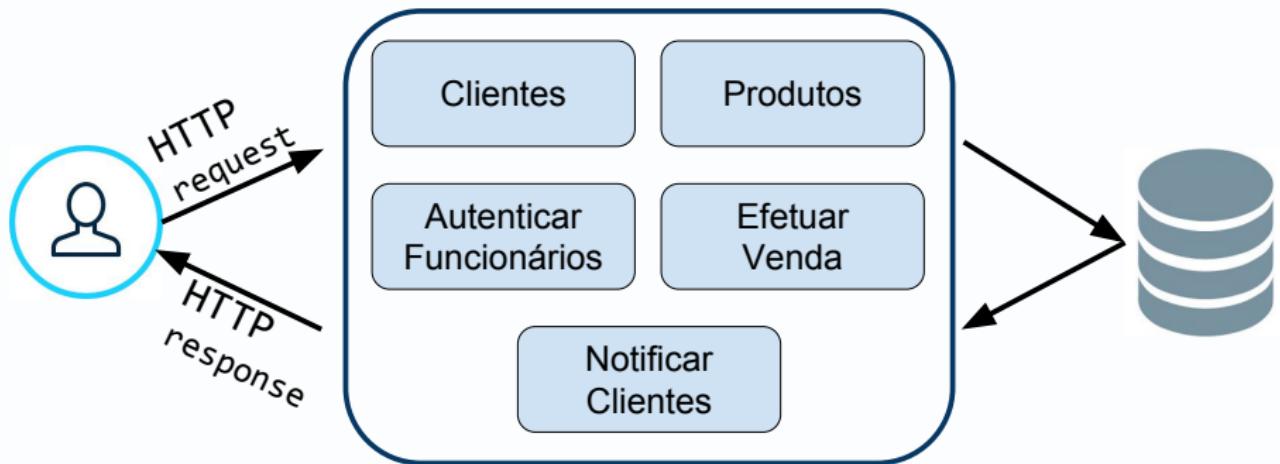


Descrição do problema

- Um funcionário autenticado deverá efetuar venda de produtos a determinados clientes.
- Um cliente poderá efetuar a compra de mais de um produto.
- O sistema deverá notificar os clientes interessados em determinados produtos quando o mesmo estiver disponível em estoque. Isso ocorre devido a uma tentativa de compra malsucedida (indisponibilidade do produto).

Sistema de Controle de Vendas

Server Side





- Toda a lógica para manipular uma requisição é efetuada em um único processo.
- É simples de **desenvolver, testar e implantar.**
- É natural, pois é assim que aprendemos!



Porém, com o crescimento do negócio...

- O número de linhas de código aumentam;
- Difícil manter, evoluir, entender;
- Baixa produtividade dos desenvolvedores;
- Preso às tecnologias;
- Baixa reutilização.





Mais além...

- Aumentar o número de produtos ofertados;
- Aumentar a quantidade de clientes da loja;
- Na **Black Friday**, como suportar a quantidade de requisições de vendas ao mesmo tempo??



Quais são as consequências?

The screenshot shows a browser window with the URL <https://vendas> in the address bar. The page content is as follows:

**Se um simples método falha,
toda a aplicação pode falhar!!**

Esta página não está funcionando

`periodicos.pucpr.br` não consegue atender a esta solicitação no momento.

HTTP ERROR 500

**Pior, toda a aplicação pode
ficar indisponível!!**

[RECARREGAR](#)

Qual seria uma alternativa?

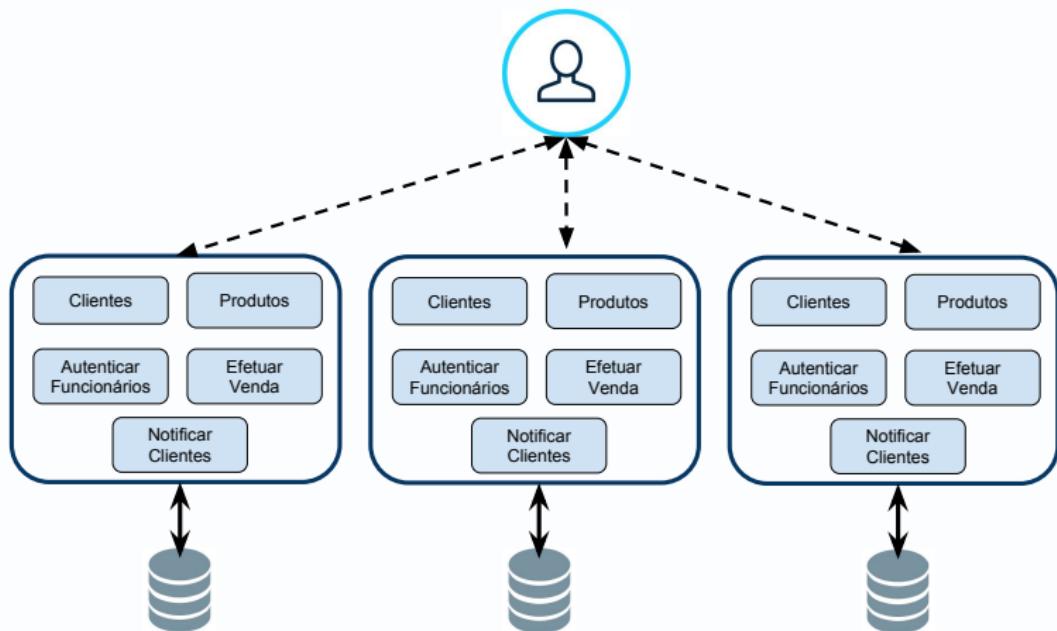


Qual seria uma alternativa?



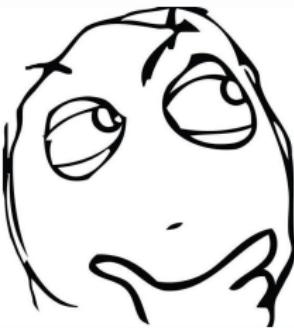


- Arquiteturas monolíticas são escaladas de forma **horizontal!**
- Mais servidores são adicionadas para aumentar o desempenho!





Vamos pensar mais um pouquinho??



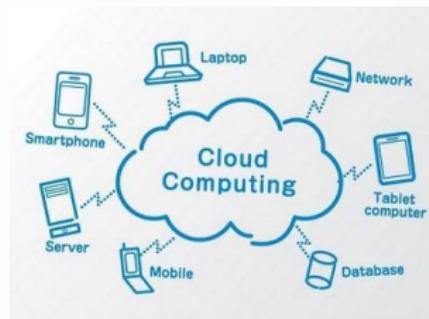


Impactos da escalabilidade horizontal

- Mudanças ficam amarradas -> **republicar toda a aplicação;**
- Dificuldade em manter uma estrutura modular;
- Impacto no custo (**escalar apenas o módulo mais requisitado**).

Cloud Computing

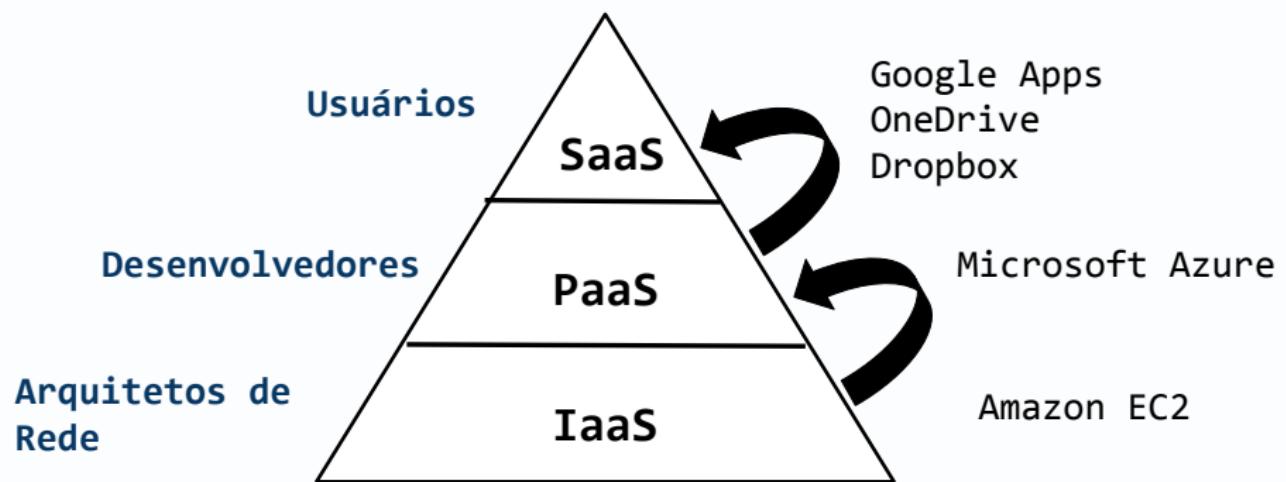
- Conjunto de computadores distribuídos que fornecem recursos ou serviços sobre demanda através de uma rede com escalabilidade e confiabilidade (Randles, 2010).
- O preço é calculado conforme o uso.
- Provisionar o tipo e tamanho dos recursos necessários.





Seis vantagens e benefícios da computação na nuvem (Amazon AWS, 2018).

- Substitua despesas de capital por despesas variáveis;
- Beneficie de economias massivas de escala;
- Pare de adivinhar a capacidade;
- Aumente a velocidade e agilidade;
- Pare de gastar dinheiro com execução e manutenção de datacenters;
- Torne-se global em minutos.

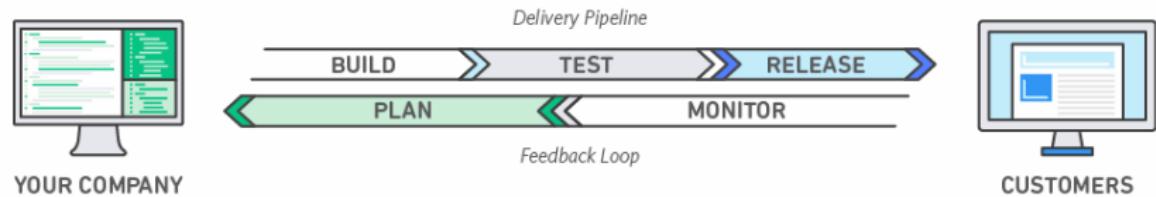




- Combinação de **filosofias** culturais, **práticas** e **ferramentas** que aumentam a capacidade de uma empresa de **distribuir** aplicativos e serviços em alta velocidade (Amazon AWS, 2018).



DevOps





Como funciona?

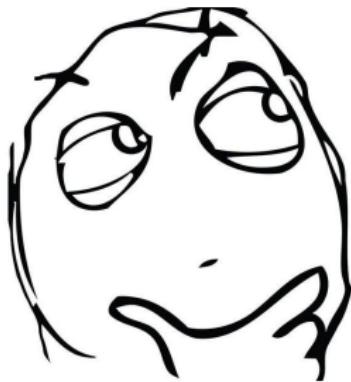
- Equipes de desenvolvimento e operações não são mais separadas em silos;
- Engenheiros trabalham durante todo o ciclo de vida da aplicação (desenvolvimento, testes, implantação, operações);
- A equipe desenvolve várias qualificações, não sendo limitadas apenas às suas funções.



Benefícios

- Velocidade;
- Entrega contínua;
- Confiabilidade.

Oow, afinal, o que são microsserviços??





Microsserviços x Big Data

Comparar



● microservices
Termo de pesquisa

● bigdata
Termo de pesquisa

+ Adicionar comparação

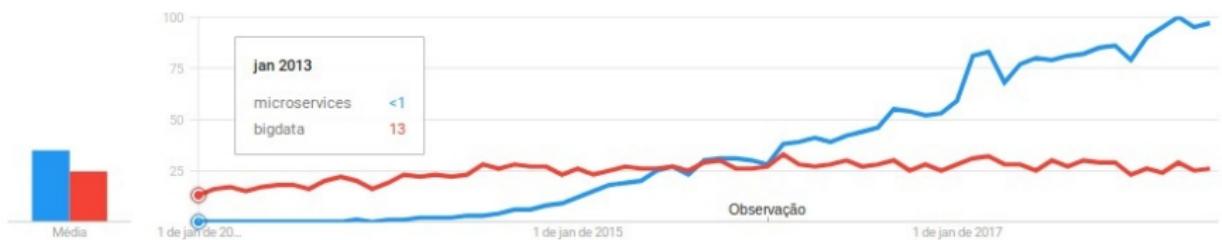
Todo o mundo ▾

01/01/2013 – 19/05/2018 ▾

Todas as categorias ▾

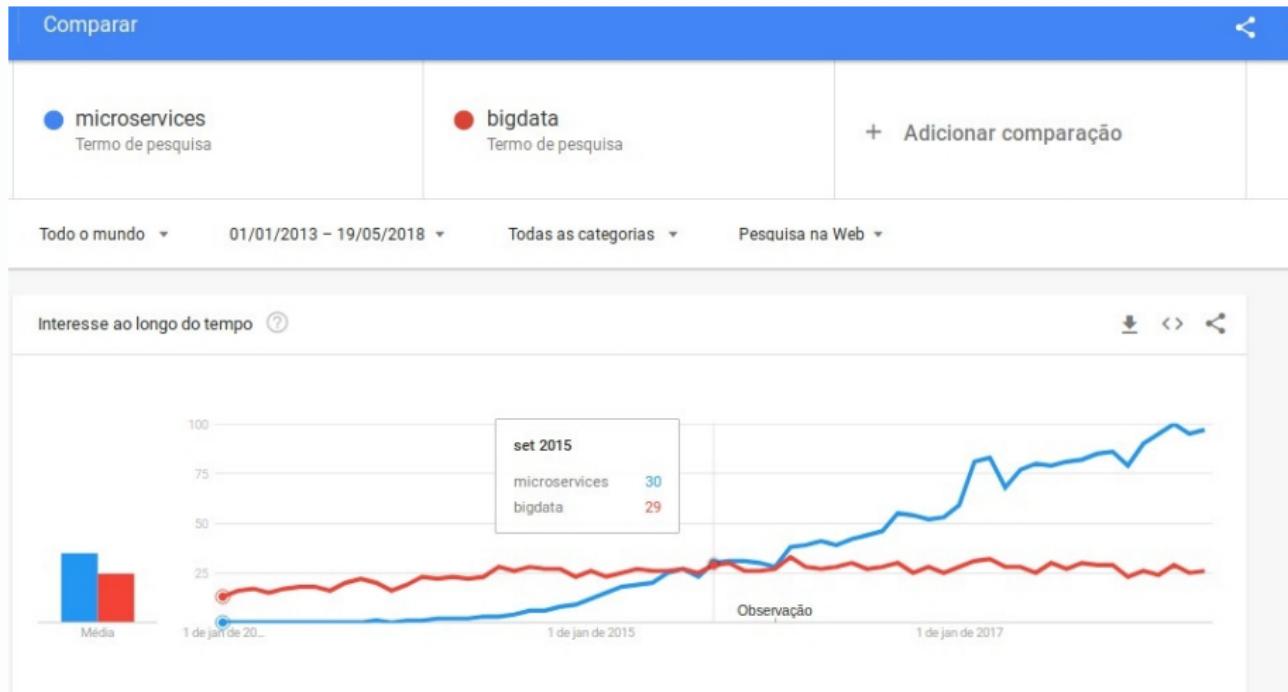
Pesquisa na Web ▾

Interesse ao longo do tempo



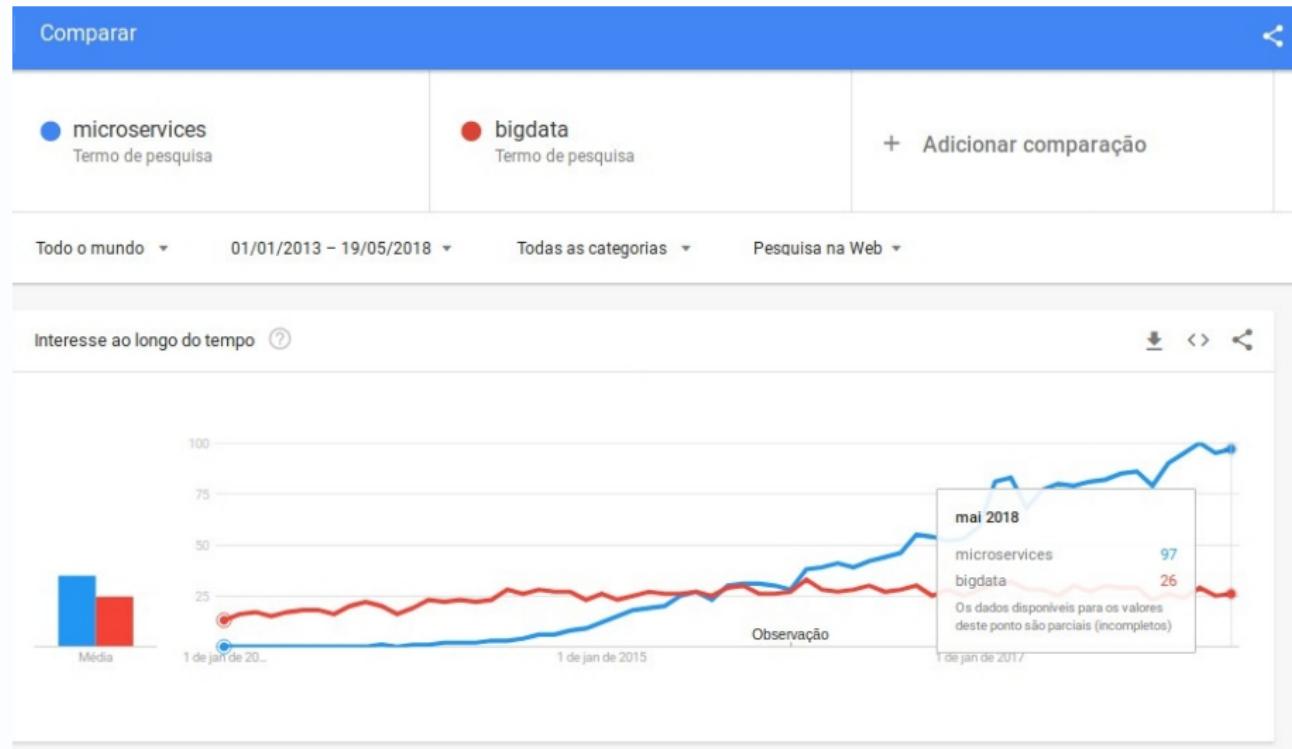


Microsserviços x Big Data





Microsserviços x Big Data



- Foi inicialmente discutido em uma conferência de arquitetos de software em 2011 (Fowler e Lewis, 2015).
- Projeto de aplicações separadas em componentes com o objetivo de torná-las melhor gerenciáveis! (Camargo et al., 2016)





- É uma abordagem derivada de SOA complementada por **Computação na nuvem** e práticas **DevOps**.
- Microsserviços são serviços refinados projetados para realizar funcionalidades específicas.
- SOA fornece serviços mantidos sob a mesma estrutura, integrando aplicações monolíticas.
- A diferença está no **nível de granularidade!**



- Arquitetura **nativa da nuvem** caracterizada pelo desenvolvimento de uma única aplicação como uma **suite de serviços**, cada um rodando em seu próprio processo e se comunicando por meio de mecanismos leves, geralmente através de uma API que utiliza o protocolo **HTTP** (Fowler, 2015).

Arquitetura Monolítica x Microsserviços

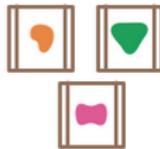
Uma aplicação monolítica coloca toda sua funcionalidade em um único processo...



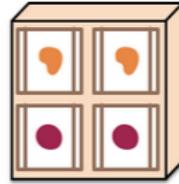
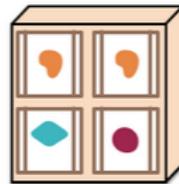
... e escala replicando a aplicação monolítica em vários servidores



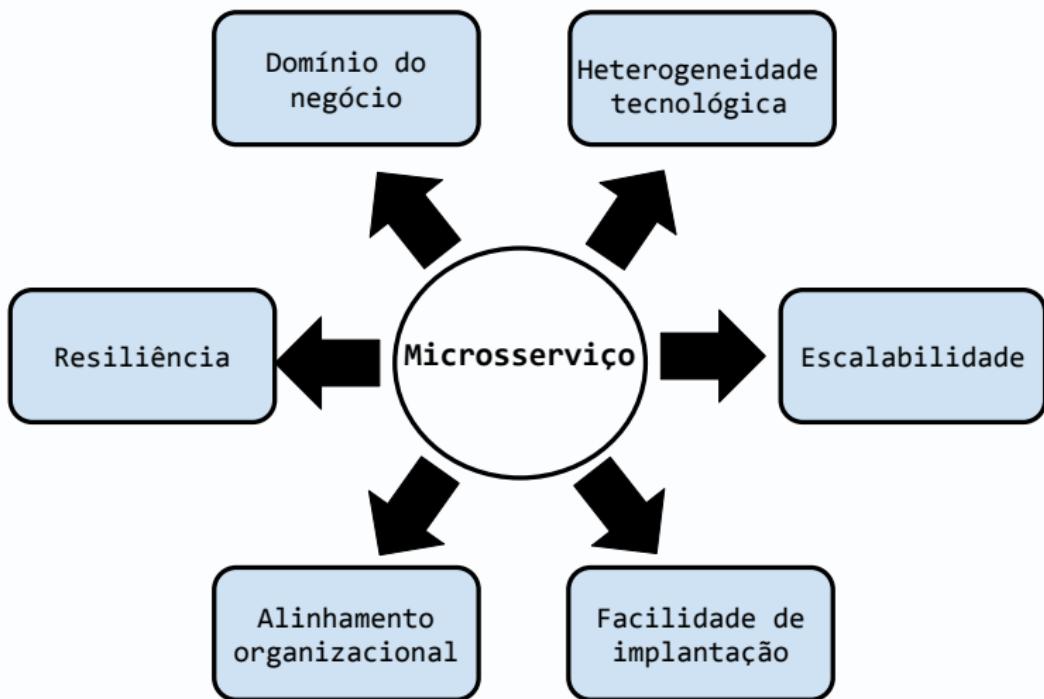
Uma arquitetura em microsserviços põe cada elemento de uma funcionalidade em um serviço separado ...



... e escala distribuindo estes serviços entre os servidores, replicando quando necessário.



Características dos Microsserviços (Newman, 2015)





Modelados em torno do domínio do negócio

- Cada microsserviço deve ser projetado com um contexto **limitado** do domínio do negócio.
- Ou seja, devem executar **funcionalidades específicas!**
- Focados na **Alta coesão e Baixo acoplamento!**



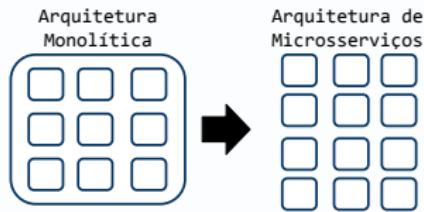
Modelados em torno do domínio do negócio

- Cada microsserviço deve ser projetado com um contexto **limitado** do domínio do negócio.
- Ou seja, devem executar **funcionalidades específicas!**
- Focados na **Alta coesão e Baixo acoplamento!**



Modelados em torno do domínio do negócio

- Em geral, cada módulo de uma arquitetura monolítica transforma-se em um microsserviço.

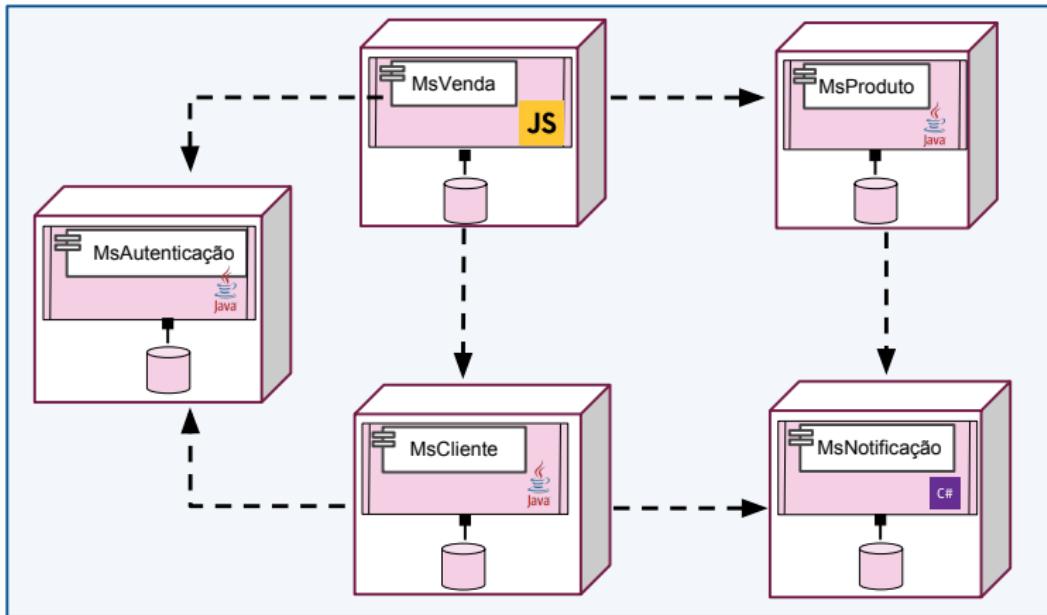


Como ficaria nosso sistema de vendas projetado na arquitetura de microsserviços?



Controle de Vendas com Microsserviços

Aplicação de Controle de Vendas

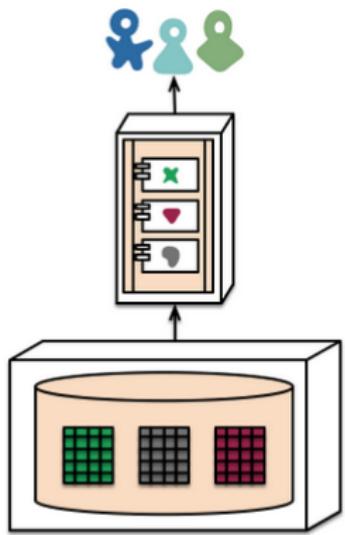




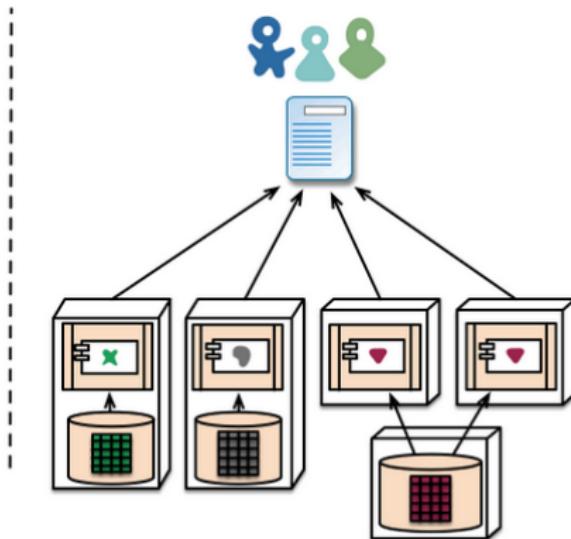
Heterogeneidade tecnológica

- Hospedados em diferentes servidores;
- Implementados em diferentes linguagens de programação;
- Utilizar diferentes *frameworks*;
- Gerenciarem sua própria base de dados.

Heterogeneidade tecnológica



Ambiente monolítico - único banco de dados



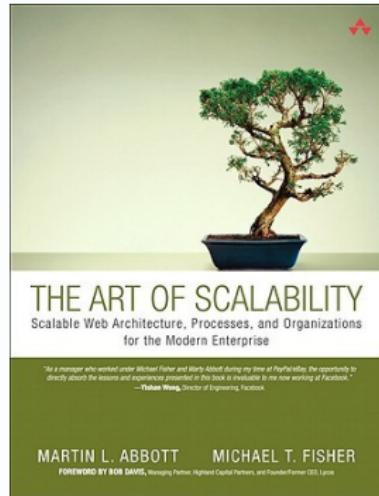
Microsserviços - banco de dados por aplicação

Fonte: <https://martinfowler.com/articles/microservices.html>



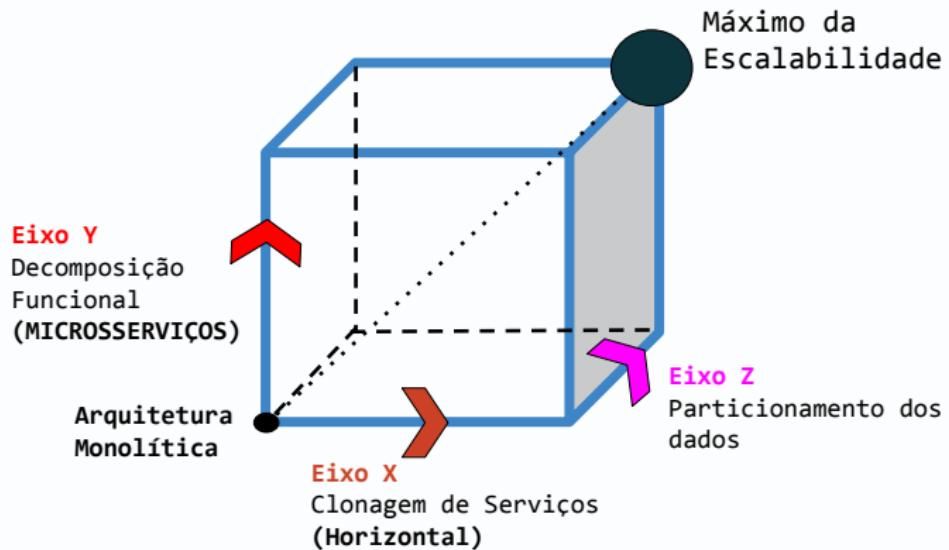
Escalabilidade

- Cubo da escalabilidade: proposto por Martin Abbott e Michael Fisher.
- Representam o máximo da escalabilidade (**X, Y e Z**).
- Reduz o custo da aplicação, escalando apenas os serviços necessários.





Cubo da Escalabilidade





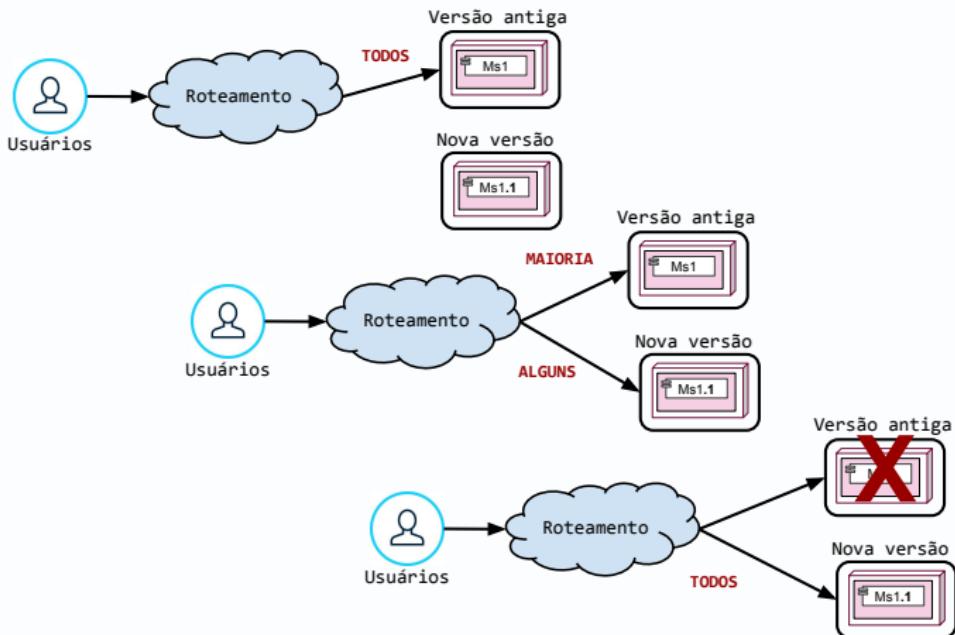
Escalabilidade

- **Eixo X:** Aumento da capacidade de disponibilidade da aplicação.
- **Eixo Z:** Cada servidor executa uma cópia idêntica do conjunto de dados. Base de dados NoSQL (**Roteamento de chamadas**).
- **Eixo Y:** Aumento da complexidade de desenvolvimento entre entidades distintas (vários microsserviços).



Facilidade de Implantação

- Modificar um microserviço não impacta nos outros micro-serviços;
- Interfaces bem definidas (REST);
- O cliente não sabe qual instância do serviço está sendo utilizada naquele momento!
- Desenvolvidos; Implantados; Mantidos; Escalados, Testados, etc.



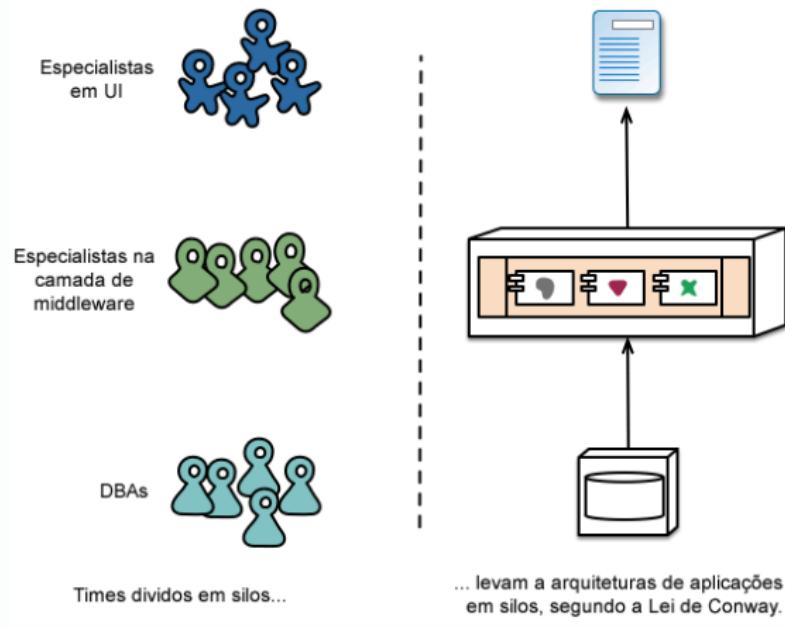


Alinhamento Organizacional

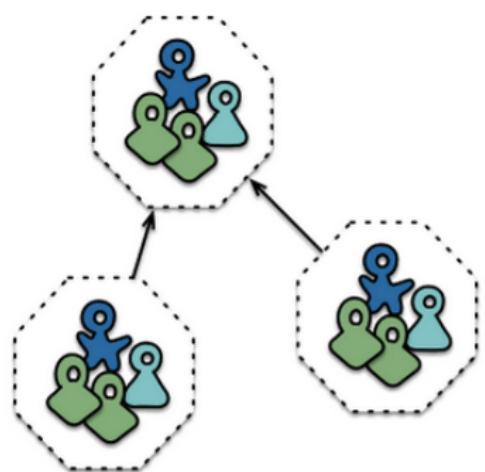
- Alinhar a arquitetura da aplicação e a organização.
- Organizar os times ao redor das áreas de negócio.



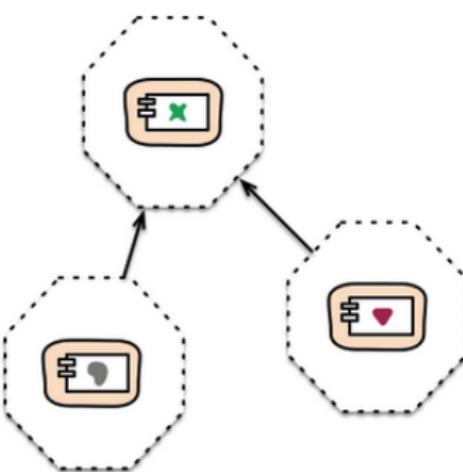
Lei de Conway - Melvyn Conway, 1967



Times Funcionais



Times multifuncionais ...



... organizados ao redor das áreas do negócio,
segundo a Lei de Conway.

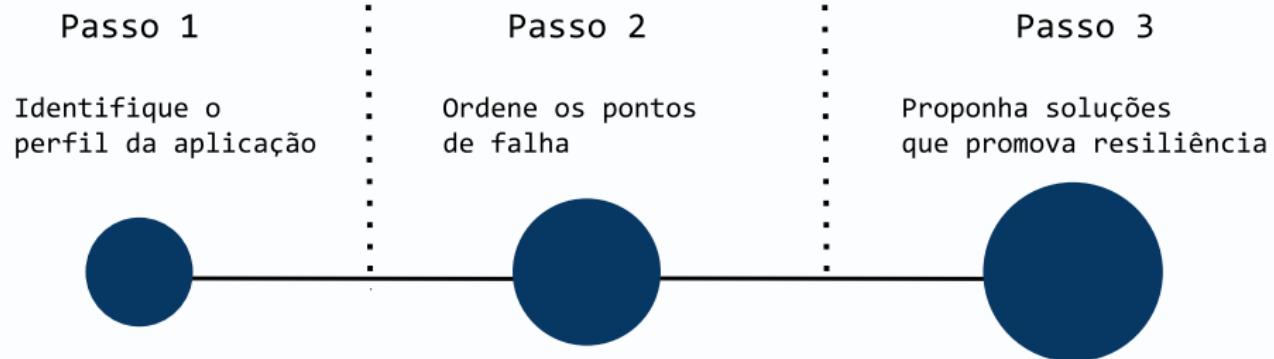


Resiliência

- ¹Propriedade de um corpo de recuperar a sua forma original após sofrer choque ou deformação.
- ²Capacidade de se recobrar facilmente ou se adaptar à má sorte ou às mudanças.

Foco: isolar possíveis falhas causadas por um serviço para que este não afete toda a aplicação

Passos para se obter resiliência em uma aplicação



Fonte: Gustavo Pantuza (desenvolvedor da Globo.com)

Principais problemas

- Lidar com a complexidade adicional de criar e manter um sistema distribuído.
- Complexidade de implantar e gerenciar muitos serviços.

Problemas





Principais problemas

- Descoberta de serviços;
- Roteamento das requisições;
- Composição entre os serviços;
- Balanceamento de carga;
- Tolerância a falhas.





Hoteis.com





- A Spring Cloud possui uma stack completa de micro-serviços desenvolvidos e mantidos pela Pivotal, no qual sua primeira versão foi lançada em 2014.
- *Spring Cloud* fornece uma integração completa entre o Spring Boot e o projeto *Netflix OSS*.





Spring Boot

Ecossistema Spring

Spring
Data

Spring
Security

Spring
Integration

Spring
...

Spring Framework





Meu Primeiro Projeto Spring

- Criar um projeto Spring Boot Starter;
- Habilitar os arquivos de configuração;
- Iniciar a aplicação.



Primeiro Projeto

The screenshot shows the Spring Tool Suite (STS) interface. The title bar indicates "workspace-sts-3.9.4.RELEASE - Spring Tool Suite". The main menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. A toolbar with various icons is visible above the central workspace. On the left, there's a "Pac" (Project Explorer) view showing a single project named "Spring Starter Project". The "File" menu is open, showing options like New, Show In, Copy, Paste, Import..., Export..., Refresh, and F5. The "New" submenu is highlighted and displays a list of project types: Spring Starter Project, Import Spring Getting Started Content, Spring Legacy Project, Java Project, Static Web Project, Dynamic Web Project, Maven Project, Project..., Aspect, Package, Class, Interface, Enum, Annotation, JUnit Test Case, Source Folder, Java Working Set, Spring Bean Configuration File, Spring Web Flow Definition File, Folder, File, Untitled Text File, Example..., and Other... The "Console" view on the right shows log output from 05:20:06 to 05:22:06, detailing the startup and shutdown of a Netflix Discovery Client and StandardService.

```
05:20:06 com.netflix.discovery.DiscoveryClient : Saw local status
05:20:06 com.netflix.discovery.DiscoveryClient : DiscoveryClient_1
05:20:06 com.netflix.discovery.DiscoveryClient : DiscoveryClient_1
05:20:06 o.s.c.support.DefaultLifecycleProcessor : Stopping beans in scope
05:20:06 o.s.j.e.AnnotationBeanExporter : Unregistering JMX
05:20:06 o.s.j.e.AnnotationBeanExporter : Shutting down DI
05:20:06 com.netflix.discovery.DiscoveryClient : Unregistering ..
05:20:06 com.netflix.discovery.DiscoveryClient : DiscoveryClient_1
05:20:06 com.netflix.discovery.DiscoveryClient : Completed shutdown
05:20:06 o.apache.catalina.core.StandardService : Stopping service
05:20:06 o.a.c.c.[Tomcat].[localhost].[] : Destroying Spring
```



Primeiro Projeto

The screenshot shows the Spring Tool Suite interface with the title bar "Activities Spring Tool Suite" and the date/time "qui 16:55". A central dialog box titled "New Spring Starter Project" is open. The dialog contains the following fields:

- Service URL: `https://start.spring.io`
- Name: `MeuPrimeiroProjetoSpring`
- Use default location
- Location: `/home/elena/Documents/workspace-sts-3.9.4.RELEASE/` (with a "Browse" button)
- Type: `Maven` (with a dropdown arrow)
- Packaging: `Jar` (with a dropdown arrow)
- Java Version: `8` (with a dropdown arrow)
- Language: `Java` (with a dropdown arrow)
- Group: `com.app`
- Artifact: `MeuPrimeiroProjetoSpring`
- Version: `0.0.1-SNAPSHOT`
- Description: `Meu Primeiro Projeto Spring Boot`
- Package: `com.example.app`
- Working sets:
 - Add project to working sets (with a "New..." button)
 - Working sets: (with a dropdown arrow and "Select..." button)

At the bottom of the dialog are buttons for `< Back`, `Next >`, `Cancel`, and `Finish`. To the right of the dialog, there is a status bar with a green power icon and a terminal window showing log output related to Spring services.



Primeiro Projeto

The screenshot shows the Spring Tool Suite interface with the title bar "Spring Tool Suite" and the date "qui 17:30". A central dialog box is titled "New Spring Starter Project Dependencies".
The "Available:" section lists various Spring Boot dependencies. Under "Selected:", three checkboxes are checked:

- Eureka Discovery
- Zuul
- Zuul

The "Selected:" list contains:

- X Config Client
- X Eureka Discovery
- X Zuul

At the bottom of the dialog are buttons for "Make Default", "Clear Selection", and "Finish". Below the dialog, a status bar shows log output:

```
2018-08-20 17:30:30.965:INFO::main:[main]ettySpringApplication : Startin
2018-08-20 17:30:30.965:INFO::main:[main]ettySpringApplication : No acti
2018-08-20 17:30:30.965:INFO::main:[main]logApplicationContext : Refresh
2018-08-20 17:30:30.965:INFO::main:[main]BeanExporter : Registr
2018-08-20 17:30:30.965:INFO::main:[main]ettySpringApplication : Started
2018-08-20 17:30:30.965:INFO::main:[main]logApplicationContext : Closing
2018-08-20 17:30:30.965:INFO::main:[main]BeanExporter : Unregis
```

The left side of the interface features a toolbar with icons for file operations like Open, Save, and Run, along with a "Packag" view and a "Servers" tab. The right side shows a workspace with a message "An outline is not available".



Primeiro Projeto

Activities Spring Tool Suite ▾ qui 16:56
workspace-sts-3.9.4.RELEASE - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

Pac Ser Bo

MeuPrimeiroProjetoSpring [b]
src/main/java
src/main/resources
src/test/java
JRE System Library [JavaSE]
Maven Dependencies
src
target
mvnw
mvnw.cmd
pom.xml

An outline is not available.

Console Progress Problems

```
<terminated> /usr/lib/jvm/java-8-oracle/bin/java (24 de maio de 2018 16:52:06)
2018-05-24 16:54:23.731  WARN 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Saw local status
2018-05-24 16:54:23.732 INFO 10619 --- [infoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_!
2018-05-24 16:54:23.751 INFO 10619 --- [infoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_!
2018-05-24 16:54:23.754 INFO 10619 --- [on(2)-127.0.0.1] o.s.c.support.DefaultLifecycleProcessor : Stopping beans in
2018-05-24 16:54:23.778 INFO 10619 --- [on(2)-127.0.0.1] o.s.j.e.AnnotationBeanExporter : Unregistering JMX
2018-05-24 16:54:23.782 INFO 10619 --- [on(2)-127.0.0.1] o.s.j.e.AnnotationBeanExporter : Unregistering JMX
2018-05-24 16:54:23.788 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Shutting down DI:
2018-05-24 16:54:26.791 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Unregistering ..
2018-05-24 16:54:26.808 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : DiscoveryClient_!
2018-05-24 16:54:26.818 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Completed shut do
2018-05-24 16:54:26.871 INFO 10619 --- [on(2)-127.0.0.1] o.apache.catalina.core.StandardService : Stopping service
2018-05-24 16:54:26.873 INFO 10619 --- [ost-startStop-2] o.a.c.c.[Tomcat].[localhost].[] : Destroying Spring
```



Primeiro Projeto

The screenshot shows the Spring Tool Suite interface with the following details:

- Top Bar:** Activities, Spring Tool Suite ▾ workspace-sts-3.9.4.RELEASE - MeuPrimeiroProjetoSpring/src/main/java/com/example/app/MeuPrimeiroProjetoSpringApplication.java - Spring Tool Suite. Date: qui 16:56.
- File Menu:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Left Sidebar (Project Explorer):** Shows the project structure:
 - MeuPrimeiroProjetoSpring [boot]
 - src/main/java
 - com.example.app
 - MeuPrimeiroProjetoSpringApplication
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml
- Central Area (Code Editor):** Displays the code for `MeuPrimeiroProjetoSpringApplication.java`.

```
1 package com.example.app;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class MeuPrimeiroProjetoSpringApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(MeuPrimeiroProjetoSpringApplication.class, args);
10    }
11 }
12 }
```
- Bottom Area (Console):** Shows the terminal output.

```
<terminated> /usr/lib/jvm/java-8-oracle/bin/java (24 de maio de 2018 16:52:06)
2018-05-24 16:54:23.731  WARN 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Saw loc
2018-05-24 16:54:23.732 INFO 10619 --- [infoReplicator-0] com.netflix.discovery.DiscoveryClient : Discove
2018-05-24 16:54:23.751 INFO 10619 --- [infoReplicator-0] com.netflix.discovery.DiscoveryClient : Discover
2018-05-24 16:54:23.754 INFO 10619 --- [on(2)-127.0.0.1] o.s.c.support.DefaultLifecycleProcessor : Stoppi
2018-05-24 16:54:23.778 INFO 10619 --- [on(2)-127.0.0.1] o.s.j.e.a.AnnotationBeanExporter : Unregis
2018-05-24 16:54:23.782 INFO 10619 --- [on(2)-127.0.0.1] o.s.j.e.a.AnnotationBeanExporter : Stoppin
2018-05-24 16:54:23.786 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Unregis
2018-05-24 16:54:26.791 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Shuttin
2018-05-24 16:54:26.809 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Discove
2018-05-24 16:54:26.818 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Complet
2018-05-24 16:54:26.871 INFO 10619 --- [on(2)-127.0.0.1] o.apache.catalina.core.StandardService : Stoppi
2018-05-24 16:54:26.873 INFO 10619 --- [ost-startStop-2] o.a.c.c.Tomcat@localhost:[/] : Destroy
```



Arquivos de Configuração (global)

The screenshot shows the Spring Tool Suite interface with the following details:

- Title Bar:** workspace-sts-3.9.4.RELEASE - MeuPrimeiroProjetoSpring/src/main/resources/application.yml - Spring Tool Suite
- File Menu:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbars:** Standard, Navigator, Properties, and others.
- Package Explorer:** Shows the project structure:
 - MeuPrimeiroProjetoSpring [boot]
 - src/main/java
 - com.example.app
 - MeuPrimeiroProjetoSpringApplication
 - src/main/resources
 - application.yml
 - src/test/java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml
- Central Editor:** Displays the content of application.yml.
- Console View:** Shows log output from the terminal window.
- Right-hand Side:** Includes a "Quick Access" panel and a "MeuPrin" icon.



Arquivos de Configuração (local)

The screenshot shows the Spring Tool Suite interface with the title "workspace-sts-3.9.4.RELEASE - MeuPrimeiroProjetoSpring/src/main/resources/application.yml - Spring Tool Suite". The application.yml file is open in the editor. A context menu is open over the file, with the "File" option highlighted.

The context menu options include:

- New
- Go Into
- Open in New Window
- Open Type Hierarchy F4
- Show In Shift+Alt+W
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete
- Remove from Context
- Build Path
- Source Shift+Alt+S
- Refactor Shift+Alt+T
- Import...
- Export...
- Refresh F5
- Assign Working Sets...
- Run As
- Debug As
- Profile As

The right-hand side of the interface shows the "MeuPrimeiroProjetoSpringApplication.java" file with some code and a list of annotations and methods:

```
etflix.discovery.DiscoveryClient : Saw loc
etflix.discovery.DiscoveryClient : Discover
etflix.discovery.DiscoveryClient : Discover
.e.support.DefaultLifecycleProcessor : StopIn
.e.a.AnnotationBeanExporter : Unregis
.e.a.AnnotationBeanExporter : Unregis
etflix.discovery.DiscoveryClient : Shuttin
etflix.discovery.DiscoveryClient : Unregis
etflix.discovery.DiscoveryClient : Discove
etflix.discovery.DiscoveryClient : Complet
che.catalina.core.StandardService : Stoppin
.c.C.[Tomcat].[localhost].[] : Destroy
```



Arquivos de Configuração (local)

The screenshot shows the Spring Tool Suite (STS) interface with the following details:

- Title Bar:** workspace-sts-3.9.4.RELEASE - MeuPrimeiroProjetoSpring/src/main/resources/bootstrap.yml - Spring Tool Suite
- Toolbar:** Activities, Spring Tool Suite, pt, Quick Access.
- File Menu:** File, Edit, Navigate, Search, Project, Run, Window, Help.
- Project Explorer (Left):** Shows the project structure:
 - Packages
 - Servers
 - Boot Data
 - MeuPrimeiroProjetoSpring [boot]
 - src/main/java
 - com.example.app
 - MeuPrimeiroProjetoSpringApplication
 - src/main/resources
 - application.yml
 - bootstrap.yml
 - src/test/java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml
- Central Area:** Three tabs are open: MeuPrimeiroProjetoSpringApplication.java, application.yml, and bootstrap.yml.
- Console Tab:** Shows log output from the application.

```
<terminated> /usr/lib/jvm/java-8-oracle/bin/java (24 de maio de 2018 16:52:06)
2018-05-24 16:54:23.731  WARN 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient      : Saw locator: Discov
2018-05-24 16:54:23.732 INFO 10619 --- [infoReplicator-0] com.netflix.discovery.DiscoveryClient      : Discov
2018-05-24 16:54:23.751 INFO 10619 --- [infoReplicator-0] com.netflix.discovery.DiscoveryClient      : Stoppin
2018-05-24 16:54:23.754 INFO 10619 --- [on(2)-127.0.0.1] o.s.c.support.DefaultLifecycleProcessor : Stoppin
2018-05-24 16:54:23.778 INFO 10619 --- [on(2)-127.0.0.1] o.s.j.e.a.AnnotationBeanExporter      : Stoppi
2018-05-24 16:54:23.782 INFO 10619 --- [on(2)-127.0.0.1] o.s.j.e.a.AnnotationBeanExporter      : Unregis
2018-05-24 16:54:23.784 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient      : Unregis
2018-05-24 16:54:26.791 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient      : Shuttin
2018-05-24 16:54:26.800 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient      : Discov
2018-05-24 16:54:26.818 INFO 10619 --- [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient      : Complet
2018-05-24 16:54:26.871 INFO 10619 --- [on(2)-127.0.0.1] o.apache.catalina.core.StandardService : Stoppin
2018-05-24 16:54:26.873 INFO 10619 --- [ost-startStop-2] o.a.c.c.C.[Tomcat].[localhost].[/]          : Destroy
```
- Bottom Buttons:** Writable, Insert, 1:1.



Primeiro método para realização da requisição

The screenshot shows the Spring Tool Suite (STS) interface. The title bar indicates "workspace-sts-3.9.4.RELEASE - MeuPrimeiroProjetoSpring/src/main/java/com/example/app/MeuPrimeiroProjetoSpringApplication.java - Spring Tool Suite". The left sidebar shows the project structure for "MeuPrimeiroProjetoSpring [boot]" with packages like "com.example.app", "src/main/java", "src/main/resources", and "src/test/java". The main editor window displays the code for "MeuPrimeiroProjetoSpringApplication.java":

```
1 package com.example.app;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.web.bind.annotation.GetMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 @SpringBootApplication
9 @RestController
10 public class MeuPrimeiroProjetoSpringApplication {
11
12     @RequestMapping(value="mensagem")
13     private String getMensagem() {
14         return "SECOM: o melhor evento de computação do MINAS!";
15     }
16
17     public static void main(String[] args) {
18         SpringApplication.run(MeuPrimeiroProjetoSpringApplication.class, args);
19     }
20 }
```

The right sidebar shows a file tree with "MeuPrimeiroProjetoSpring" selected. The bottom pane shows the "Console" output:

```
<terminated> /usr/lib/jvm/java-8-oracle/bin/java (24 de maio de 2018 16:52:06)
2018-05-24 16:54:23.731  WARN 10619 ... [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Saw loc
2018-05-24 16:54:23.732 INFO 10619 ... [infoReplicator-0] com.netflix.discovery.DiscoveryClient : Discover
2018-05-24 16:54:23.751 INFO 10619 ... [infoReplicator-0] com.netflix.discovery.DiscoveryClient : Discov
2018-05-24 16:54:23.754 INFO 10619 ... [on(2)-127.0.0.1] o.s.c.support.DefaultLifecycleProcessor : Stoppin
2018-05-24 16:54:23.778 INFO 10619 ... [on(2)-127.0.0.1] o.s.j.e.a.AnnotationMBeanExporter : Unregis
2018-05-24 16:54:23.782 INFO 10619 ... [on(2)-127.0.0.1] o.s.j.e.a.AnnotationMBeanExporter : Unregis
2018-05-24 16:54:23.786 INFO 10619 ... [on(2)-127.0.0.1] com.netflix.discovery.DiscoveryClient : Shuttin
```



Subindo a aplicação

The screenshot shows the Spring Tool Suite interface with the following details:

- File Bar:** Activities, Spring Tool Suite, workspace-sts-3.9.4.RELEASE - MeuPrimeiroProjetoSpring/src/main/java/com/example/app/MeuPrimeiroProjetoSpringApplication.java - Spring Tool Suite.
- Central Area:** Code editor showing the `MeuPrimeiroProjetoSpringApplication.java` file with the following code:

```
package com.example.app;

import framework.boot.SpringApplication;
import framework.boot.autoconfigure.SpringBootApplication;
import framework.web.bind.annotation.RequestMapping;
import framework.web.bind.annotation.RestController;

@SpringBootApplication
public class MeuPrimeiroProjetoSpringApplication {

    @RequestMapping(value = "mensagem")
    public String getMensagem() {
        return "ECONOMIA: o melhor evento de computação de MINAS!";
    }

    public static void main(String[] args) {
        SpringApplication.run(MeuPrimeiroProjetoSpringApplication.class, args);
    }
}
```
- Context Menu:** A context menu is open over the first line of the code (`package com.example.app;`). The menu items include: New, Go Into, Open in New Window, Open Type Hierarchy, Show in, Copy, Copy Qualified Name, Paste, Delete, Remove from Context, Build Path, Refactor, Import..., Export..., Refresh, Close Project, Close Unrelated Projects, Assign Working Sets..., Run As, Debug As, Profile As, Validate, and Restore from Local History... . The **Run As** option is highlighted.
- Run As Submenu:** A submenu for "Run As" is displayed, listing the following options: Run on Server, Java Application, JUnit Test, Maven build, Maven build..., Maven clean, Maven generate-sources, Maven install, Maven test, Spring Boot App (highlighted in orange), Spring DevTools Client, and Run Configurations... .
- Right Panel:** Shows the package explorer with the path `com.example.MeuPrimeiroProjetoSpringApplication` selected. It also displays a list of recent files and a "Run" section with several entries.



Execução na porta padrão 8080

The screenshot shows the Spring Tool Suite interface with the following details:

- Title Bar:** Activities - Spring Tool Suite - workspace-sts-3.9.4.RELEASE - MeuPrimeiroProjetoSpring/src/main/java/com/example/app/MeuPrimeiroProjetoSpringApplication.java - Spring Tool Suite
- Toolbar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Left Sidebar (Project Explorer):** Shows the project structure with the package `MeuPrimeiroProjetoSpring [boot]` selected.
- Middle Area (Code Editor):** Displays the Java code for `MeuPrimeiroProjetoSpringApplication.java`. The code includes imports for `com.example.app` and `org.springframework.boot.SpringApplication`, and a main class definition with annotations like `@EnableDiscoveryClient` and `@PortValue("8080")`.
- Bottom Area (Console):** Shows the application logs output from the terminal. The logs indicate the application is running on port 8080 and registering with Eureka.

```
MeuPrimeiroProjetoSpring - MeuPrimeiroProjetoSpringApplication [Spring Boot App] /usr/lib/jvm/java-8-oracle/bin/java (24 de m
    .n.d.provider.DiscoveryJerseyProvider : Using JSON decoding codec LegacyJacksonJson
    .n.d.provider.DiscoveryJerseyProvider : Using XML encoding codec XStreamXml
    .n.d.provider.DiscoveryJerseyProvider : Using XML decoding codec XStreamXml
    .n.d.s.r.ows.ConfigClusterResolver : Resolving eureka endpoints via configuration
    .com.netflix.discovery.DiscoveryClient : Disable delta property : false
    .com.netflix.discovery.DiscoveryClient : Single vip registry refresh property : null
    .com.netflix.discovery.DiscoveryClient : Force full registry fetch : false
    .com.netflix.discovery.DiscoveryClient : Application is null : false
    .com.netflix.discovery.DiscoveryClient : Registered Applications size is zero : true
    .com.netflix.discovery.DiscoveryClient : Application version is -1: true
    .com.netflix.discovery.DiscoveryClient : Getting all instance registry info from the eureka server
    .com.netflix.discovery.DiscoveryClient : The response status is 200
    .com.netflix.discovery.DiscoveryClient : Starting heartbeat executor: renew interval is: 30
    .n.discovery.InstanceInfoReplicator : InstanceInfoReplicator onDemand update allowed rate per min is 4
    .com.netflix.discovery.DiscoveryClient : Discovery Client initialized at timestamp 1527193984817 with init:
    .s.s.c.n.e.s.EurekaServiceRegistry : Registering application unknown with eureka with status UP
    .com.netflix.discovery.DiscoveryClient : Saw local status change event StatusChangeEvent [timestamp=152719:
    .com.netflix.discovery.DiscoveryClient : DiscoveryClient UNKNOWN/192.168.0.28: registering service...
    .com.netflix.discovery.DiscoveryClient : DiscoveryClient UNKNOWN/192.168.0.28 - registration status: 204
    .s.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
    .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 8080
    .e.a.MeuPrimeiroProjetoSpringApplication : Started MeuPrimeiroProjetoSpringApplication in 7.106 seconds (JVM
```

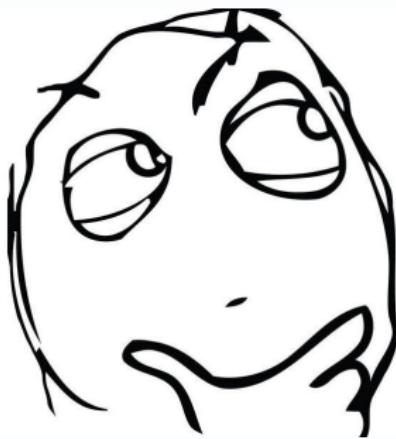


Aplicação no Ar!

A screenshot of a Linux desktop environment, likely elementary OS, showing a Firefox browser window. The browser title bar reads "localhost:8080/mensagem". The main content area of the browser displays the text "SECOM: O melhor evento de computação de minas!". The desktop interface includes a dock on the left with icons for Activities, Eureka, MinicursoMicrosserviço, and a terminal, along with a vertical dock bar at the bottom. The top bar shows the date and time as "qui 17:33" and the application name "Mozilla Firefox".

SECOM: O melhor evento de computação de minas!

Vamos discutir os principais problemas de aplicações na nuvem e sua relação com os frameworks do Netflix?

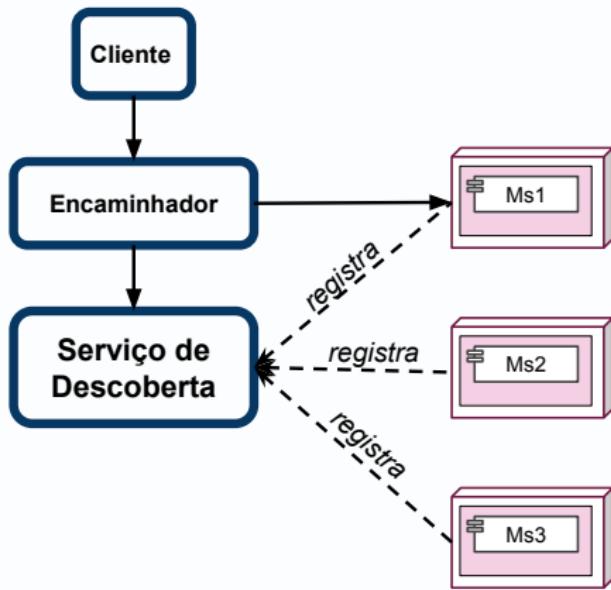




Projeto Netflix OSS

- Descoberta dos microserviços – Netflix Eureka;
- Roteamento – Netflix Zuul;
- Comunicação http – Cliente *Feign* e *RestTemplate*.
- Balanceamento de carga –Netflix Ribbon;

Descoberta de Serviços





Quem faz a descoberta dos microsserviços ?

Eureka!



Usando efetivamente os serviços do Netflix!

- Clone o repositório:
<https://github.com/elenaaugusta94/MinicursoSECOM.git>
- Importe o projeto do microsserviço **EurekaServer** no SpringBoot.



Vamos autenticar nosso microsserviço no Eureka?

Autenticar o microsserviço no Eureka

- Atualizar o arquivo bootstrap.yml
- Atualizar o arquivo application.yml

```
spring:  
  application:  
    name: Meu primeiro projeto
```

Figura 1: Bootstrap.yml

```
server:  
  port: 9099
```

Figura 2: Application.yml



Autenticar o microsserviço no Eureka

- Para executar o serviço do **Eureka**, basta:
- Abra o terminal na pasta do microsserviço.
- Execute o comando: **mvn spring-boot:run**

Na aplicação, faça:

The screenshot shows a code editor with several tabs at the top: 'MeuPrimeiroProjetoSp' (highlighted), 'EurekaServerApplicati...' (disabled), 'bootstrap.yml', and 'application.yml'. The main pane displays the following Java code:

```
1 package com.example.app;
2
3 import org.springframework.boot.SpringApplication;
4
5 @RestController
6 @SpringBootApplication
7 @EnableEurekaClient
8 public class MeuPrimeiroProjetoSpringApplication {
9
10     @RequestMapping(value="mensagem")
11     public String getMensagem() {
12         return "SECOM: O melhor evento de computação de minas!";
13     }
14
15     public static void main(String[] args) {
16         SpringApplication.run(MeuPrimeiroProjetoSpringApplication.class, args);
17     }
18 }
```

A vertical red bar highlights the line containing the annotation `@EnableEurekaClient`. The code editor interface includes standard features like tabs, file navigation, and a status bar at the bottom.



Eureka em execução

The screenshot shows the Eureka UI running in a Firefox browser. The main content area is divided into several sections:

- System Status**:
 - Environment: test
 - Data center: default
- DS Replicas**:

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
No instances available			
- General Info**:

Name	Value
total-avail-memory	400mb
environment	test
num-of-cpus	4



Eureka em execução

The screenshot shows a Firefox browser window with the title "Eureka - Mozilla Firefox". The address bar displays "localhost:8761". The main content area shows the "System Status" and "DS Replicas" sections of the Eureka interface.

System Status

Environment	test
Data center	default

Current time	2018-05-24T18:40:34 -0300
Uptime	00:01
Lease expiration enabled	false
Renew threshold	3
Renews (last min)	0

DS Replicas

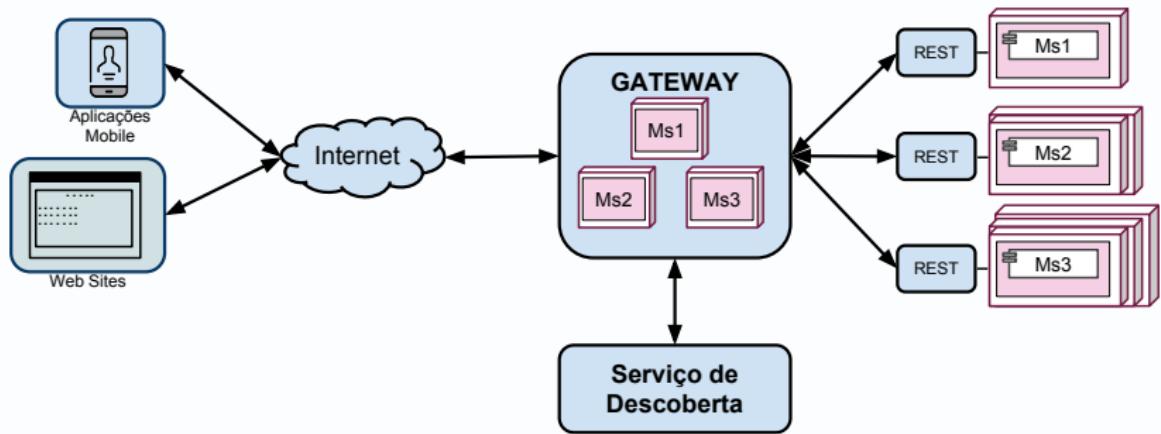
Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
MEU PRIMEIRO PROJETO	n/a (1)	(1)	UP (1) - 192.168.0.28:Meu primeiro projeto:9099

General Info

Name	Value
total-avail-memory	400mb
environment	test
Show Applications	4

Roteamento da comunicação: Zuul

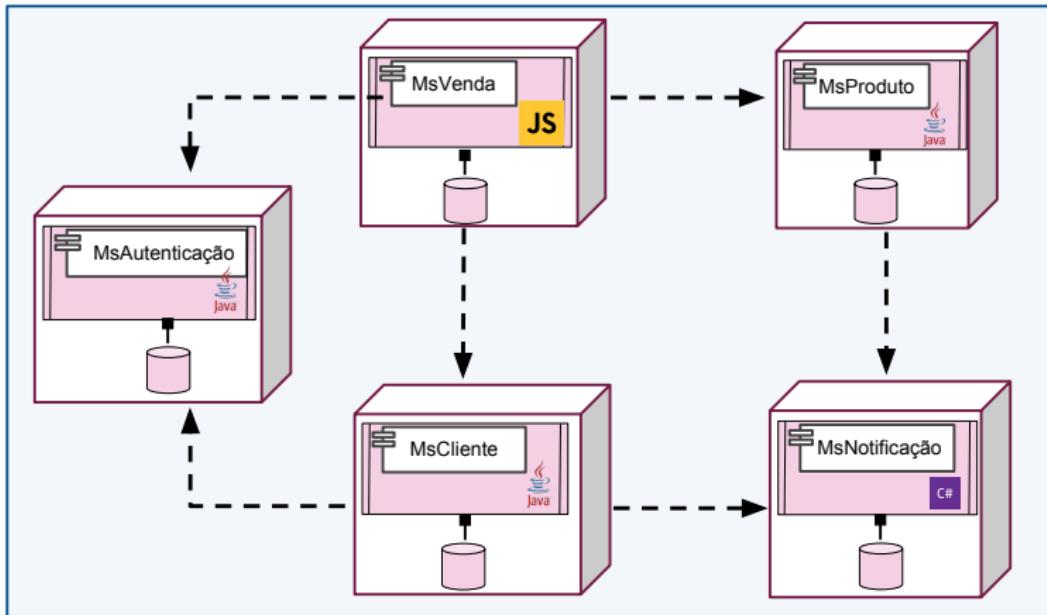




Vamos implementar aqueles
microsserviços do sistema de
controle de vendas?

Controle de Vendas com Microsserviços

Aplicação de Controle de Vendas





Encaminhamento das requisições do sistema de controle de vendas.

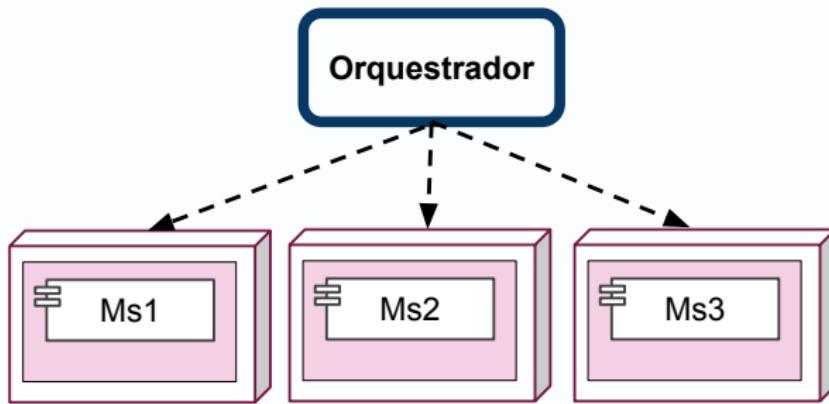
```
zuul:
  prefix:
    routes:
      cliente:
        path: /cliente/**
        serviceId: MsCliente

      product:
        path: /produto/**
        serviceId: MsProduto

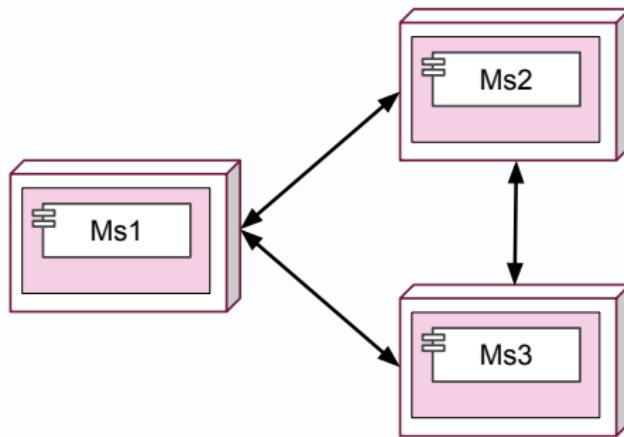
      venda:
        path: /venda/**
        serviceId: MsVenda

      autenticacao:
        path: /autenticacao/**
        serviceId: MsAutenticacao
```

Composição entre os serviços - Orquestração



Composição entre os serviços - Coreografia





Composição entre os serviços

- Cliente Feign e
- Spring RestTemplate.



Cliente Feign

- Definição de uma interface de comunicação:

```
@Autowired  
private InterfaceAutenticacao autenticacao; @FeignClient(value="MsAutenticacao")  
public interface InterfaceAutenticacao {
```



RestTemplate

- Definição de um objeto do tipo RestTemplate:

```
@LoadBalanced
@Bean
public RestTemplate restTemplate(RestTemplateBuilder builder) {
    return builder.build();
}
```

```
@RequestMapping(value = "venda/getProdutos", method = RequestMethod.GET)
@ResponseBody
public ArrayList getProducts() {
    String url = "http://MsProduto/produto/obterProdutos";
    ArrayList response = restTemplate.getForObject(url, ArrayList.class);
    return response;
}
```



- Elena A. Araujo, Elder Rodrigues Jr., Arthur F. Pinto, and Ricardo Terra. (2017). Em busca de uma abordagem de conformidade arquitetural para arquitetura de microsserviços. In: 5th Workshop de Visualização, Evolução e Manutenção de Software (VEM), pages 68-75.
- Fowler, M. (2014). Microservices Resource Guide. Acesso em: 28 maio. 2017. Disponível em: <https://martinfowler.com/microservices/>.
- Granchelli, G., Cardarelli, M., Francesco, P. D., Malavolta, I., Iovino, L., and Salle, A. D. (2017b). Towards recovering the software architecture of microservice-based systems. In 1th International Conference on Software Architecture Workshops (ICSAW), pages 46–53.



- **Resiliência:** <https://www.infoq.com.br/presentations/resiliencia-com-microservices-cache-distribuido-feedback-e-tuning>
- **Spring Cloud :** <http://projects.spring.io/spring-cloud/>
- **Spring Cloud Netflix:** <https://cloud.spring.io/spring-cloud-netflix/>
- **Amazon AWS:** <https://aws.amazon.com/pt/devops/what-is-devops/>
- **Camargo et al.:** C. M. Aderaldo, N. C. Mendonça, C. Pahl, and P. Jamshidi. Benchmark requirements for microservices architecture research. In 1th International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE), pages 8–13, 2017
- **Fowler:** M. Fowler and J. Lewis. Microservices: a definition of this new architectural term, 2014. Acesso em: 29 jun. 2017. Disponível em: <https://martinfowler.com/articles/microservices.html>



Obrigada!!!