

# Aprendizado de Máquina em Service Mesh para Gerenciamento de Configuração

Pablo Yuri S. Souza<sup>1</sup>, Gessica Franciéle M. Azevedo<sup>1</sup>, Jordi Pujol Ricarte<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{gessica.azevedo,jpricarte,pablo.souza}@inf.ufrgs.br

**Abstract.** *This article extends the implementation of a Service Mesh for configuration management, integrating the K-means machine learning algorithm. Utilizing the World Happiness Report 2021 dataset, patterns among countries are identified, and routing rules are defined based on the generated clusters. It is demonstrated how this approach simplifies network configuration management, highlighting the potential of the Service Mesh.*

**Resumo.** *Este artigo estende a implementação de um Service Mesh para gerenciamento de configuração, integrando o algoritmo de aprendizado de máquina K-means. Utilizando o conjunto de dados do World Happiness Report 2021, identificamos padrões entre os países e definimos regras de roteamento com base nos clusters gerados. Demonstra-se como essa abordagem simplifica o gerenciamento de configuração de rede, destacando o potencial do Service Mesh.*

## 1. Introdução

Este artigo apresenta uma extensão de nossa implementação desenvolvida na parte 1 do trabalho prático, a implementação de um Service Mesh para o gerenciamento de configuração. Dessa forma, aplicamos a técnica de aprendizado de máquina K-means ao nosso serviço, possibilitando o roteamento de requisições conforme os clusters gerados pelo K-means.

O K-means é um algoritmo de aprendizagem não supervisionada que gera k agrupamentos a partir de um conjunto de dados de treino. Em nossa implementação, aplicamos o k-means ao dataset world-happiness-report-2021 distribuído pela plataforma Kaggle, nele temos informações relacionadas ao desenvolvimento social e econômico de diferentes países, como índices de felicidade, PIB per capita, suporte social, expectativa de vida, entre outros. A aplicação do K-means a esse conjunto de dados tem o objetivo de identificar padrões ou agrupamentos naturais entre os países com base nessas características.

Nesse sentido, o nosso trabalho planeja fazer o roteamento de requisições que incluam países para diferentes versões de serviços baseados em clusters gerados pelo algoritmo de aprendizado de máquina K-means.

Inicialmente iremos explorar a configuração e arquitetura de nosso Service Mesh que implementamos usando Kubernetes e Istio descrito na seção 2. Na seção 3, iremos abordar o algoritmo K-means e a nossa implementação. Por fim, discutiremos como podemos expandir as ideias vistas para aplicações reais, criando cenários onde o roteamento de requisições descrito aqui resolverá problemas vistos no dia a dia de um serviço.

## 2. Configuração do Ambiente

A configuração do ambiente segue a mesma da primeira parte deste trabalho, o qual utiliza Minikube para criar um cluster Kubernetes local, Docker como driver, *cores* e memória utilizadas de acordo com a capacidade da máquina.

### 2.1. Implementação de microserviços de exemplo

Seguimos com a implementação do aplicativo BookInfo de exemplo do Istio. Foram aplicados os recursos do BookInfo utilizando os arquivos de configuração “platform/kube/bookinfo.yaml” e “networking/bookinfo-gateway.yaml” que apontavam para os arquivos YAML relativos à configuração da aplicação e do gateway da mesma. O resultado disso é o deploy da aplicação com arquitetura da Figura 2.

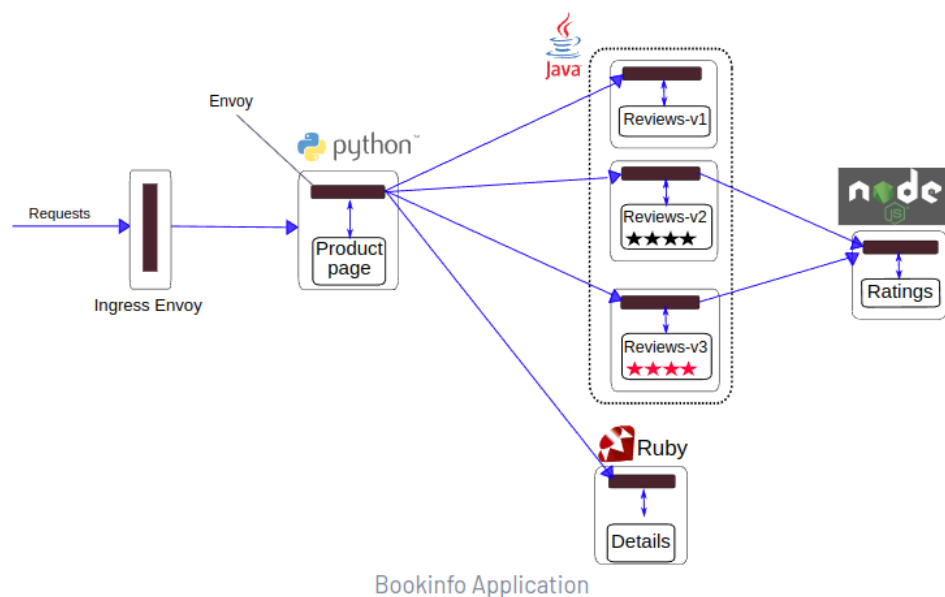


Figura 1. Arquitetura do projeto BookInfo

### 2.2. Mudanças das regras de roteamento de requisição.

Nesta segunda parte, mudamos as regras de roteamento de acordo com determinadas características de cada usuário na hora de realizar o acesso. O parâmetro que define qual regra será aplicada será enviado junto da requisição. As regras seguem sendo configuradas para todos os serviços do BookInfo usando o arquivo “networking/destination-rule-all.yaml”. Utilizamos os chamados “Virtual Services” para especificar regras para um determinado host.

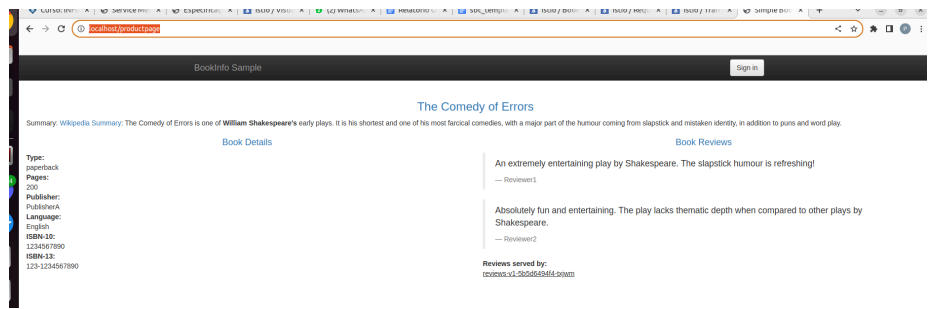


Figura 2. Página de Product Page rodando localmente com microserviço Reviews-v1

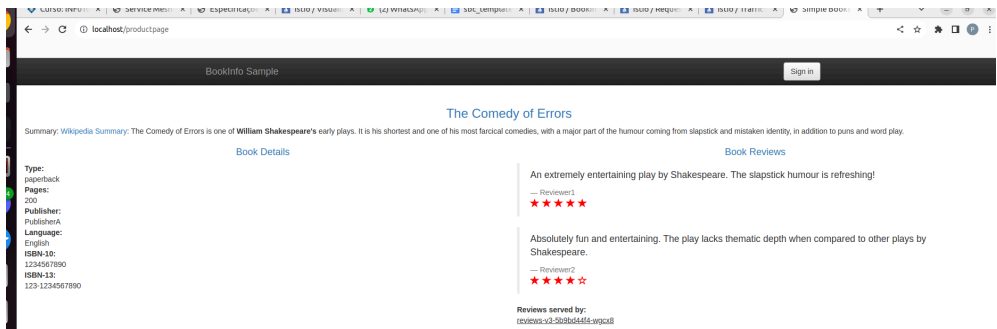


Figura 3. Página de Product Page rodando localmente com microserviço Reviews-v3

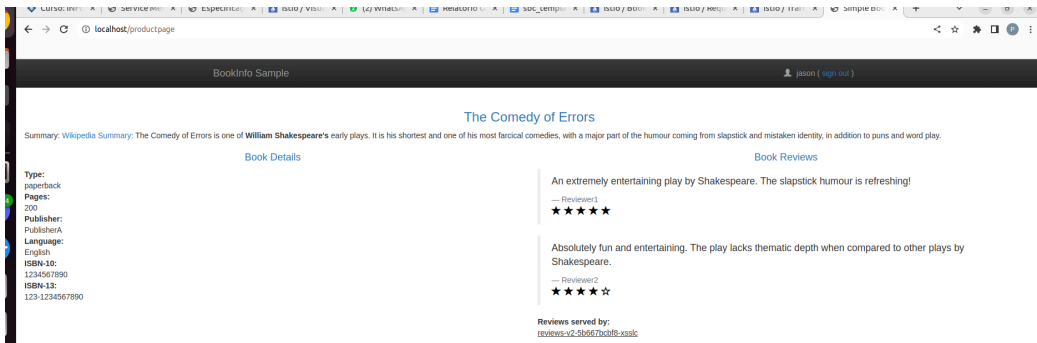


Figura 4. Página de Product Page rodando localmente com microserviço Reviews-v2

### 3. K-Means e nossa implementação

Nesta seção, abordaremos em detalhes a aplicação do algoritmo K-means à nossa implementação de Service Mesh, utilizando o dataset World Happiness Report 2021 para criar clusters representativos de diferentes perfis de países.

Antes de aplicar o K-means, realizamos o pré-processamento dos dados do World Happiness Report 2021 para que seus atributos contribuíssem igualmente. Isso envolveu a normalização de nossas variáveis escolhidas: Logged GDP per capita, Social support, Healthy life expectancy.

Para nossa implementação, escolhemos utilizar 3 clusters. A determinação do número ideal de clusters foi baseada no número de serviços disponíveis nesse contexto. Além disso, utilizamos a biblioteca de aprendizado de máquina em Python, scikit-learn, para implementar o K-means.

Por fim, após gerar os clusters, coletamos a lista de países com seus respectivos clusters e utilizamos ela para realizar o roteamento em nosso Service Mesh.

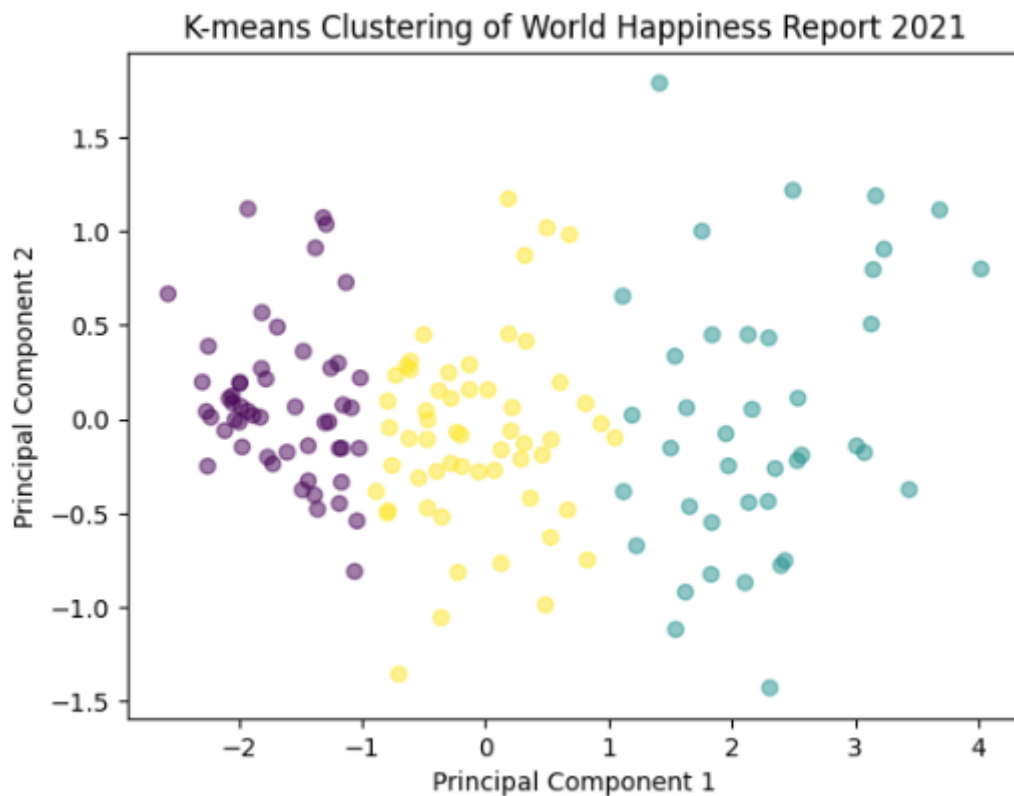


Figura x. Gráfico mostrando nossos clusters

#### **4. Regras de Roteamento:**

Com os países agrupados em 3 grupos, podemos definir regras de roteamento que retorne ao usuário uma versão diferente do programa de acordo com o país informado pelo mesmo. Esse é um processo que pode ser automatizado, gerando um script que forma as regras de roteamento com os resultados do algoritmo de clusterização.

No nosso caso, selecionamos 6 países de cada grupo para, de acordo com o código ISO 3166 A-3 (código de 3 caracteres que representa o país internacionalmente) informado no login, redirecionar para a versão do aplicativo relativa ao grupo no qual o país pertence. Países do primeiro grupo receberam a v1 do programa, as do segundo a v2 e, por fim, o terceiro grupo recebeu a versão 3 do programa.

Em um caso real, podemos dividir usuários ou acessos de acordo com clusters predefinidos, podendo assim retornar versões diferentes de uma aplicação de acordo com características extraídas durante um processo de clusterização. Dessa forma, podemos entregar uma versão especializada de um código para um determinado usuário.

#### **5. Conclusões:**

Em conclusão, este trabalho apresentou uma abordagem da implementação de microsserviços utilizando o exemplo do aplicativo BookInfo no contexto do Istio, em conjunto com a técnica de aprendizagem de máquina, K-means, sob o dataset World Happiness Report 2021. Com isso, foram realizadas mudanças nas regras de roteamento de requisições, personalizando o acesso com base em características específicas de cada país.

Dessa forma, processo descrito acima traz um exemplo simples de como podemos configurar nossa rede para distribuir os acessos a um serviço. Essas são apenas algumas das diversas possibilidades que se abrem quando utilizamos um Service Mesh. Isso mostra como o uso dessa tecnologia pode ser útil para simplificar o gerenciamento de configuração de rede onde, com um arquivo simples e de fácil leitura, podemos criar regras robustas de roteamento.

Podemos expandir a ideia aqui descrita para fazer roteamentos mais complexos, a depender da demanda que chegue até o gerente do projeto.

## References

Istio / Bookinfo Application,

<https://istio.io/latest/docs/examples/bookinfo/#define-the-service-versions>, December.

Istio / Request Routing,

<https://istio.io/latest/docs/tasks/traffic-management/request-routing/>, December.

Istio / Traffic Shifting,

<https://istio.io/latest/docs/tasks/traffic-management/traffic-shifting/>, December.

World Happiness Report 2021,

<https://www.kaggle.com/datasets/ajaypalsinghlo/world-happiness-report-2021>,  
February.

Scikit-learn / KMeans Clustering,

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>, February.