

**Tipagem de frameworks interferem no desempenho da  
API?**

**Anderson Mateus da Silva**

**Disciplina:** Projeto e Implementação de Sistemas Distribuídos

**Prof.:** Fernando Antonio Aires Lins

Recife, Abril de 2020

## **1. Introdução**

A revolução tecnológica se dá por meio de ferramentas computacionais, ativos que estão cada vez mais presentes no nosso cotidiano, alguns que sequer percebemos. E quanto mais se passam os anos, quanto mais a tecnologia evolui, novas linguagens de programação e frameworks são criados para

Em linguagens de programação, um sistema de tipos é um conjunto de regras que atribuem uma propriedade chamada de tipo para as várias construções - tais como variáveis, expressões, funções ou módulos - que um programa de computador é composto. Este tipo representa muitas vezes uma descrição aproximada dos valores possíveis que as construções podem armazenar ou computar, ou as mensagens (chamadas de método) que eles vão responder. O objetivo principal de um sistema de tipos é reduzir erros em programas de computador através da definição de interfaces entre diferentes partes de um programa de computador, e em seguida, verificar que as partes tenham sido conectadas de uma maneira consistente. Essa verificação pode acontecer estaticamente (em tempo de compilação), dinamicamente (em tempo de execução) ou como uma combinação destes.[2]

Existem diversos frameworks de desenvolvimento de soluções para serviços distribuídos, alguns deles comumente conhecidos, pela praticidade na utilização ou pela larga escala e reconhecimento em uso comercial/industrial. Alguns destes frameworks são Express, baseado em javascript/node js e Spring, baseado em java.

## **2. Problema a ser tratado**

Dada a existência de diversas linguagens de programação, com diversos tipos distintos, os sistemas computacionais existentes na web são compostos por estas diversos

frameworks que usam como base estas linguagens. Porém, o custo computacional é sempre um quesito prioritário para estes sistemas. Escolher uma conjunto arquitetural que seja bem otimizado às necessidades auxilia a diminuir o custo para manter a própria infraestrutura.

Mas como podemos realmente dizer que um framework, com determinada tipagem, interfere no desempenho de um sistema web?

### **3. Justificativa**

<Nesta seção devem ser descritas as contribuições esperadas do trabalho. Importante também incluir possíveis trabalhos relacionados (caso existam), explicitando qual a diferença entre eles e o seu trabalho >.

Até o ano de 2000 ,não existiam padrão sem como design de API deveria ser.

A integração necessária, uso de protocolos, como SOAP, que era notoriamente conhecido como algo complexo a ser construído, manusear e debugar. Isso mudou quando um grupo de experts, liderados por Roy T. Fielding, mudaram para sempre a concepção de API, pode até se dizer que ele foi o criador da mesma. O trabalho de Fielding consistia em criar padrões que permitiriam dois servidores se comunicarem e trocarem dados em qualquer parte do mundo. Eles então definiram princípios, propriedades e restrições que nomearam “REST”. Sem absorver nenhum estado, sessão, ou cache, utilizando o protocolo HTTP e seus métodos. Rest é baseado em recursos, ou seja, objetos, ‘coisas’ [4].

O modelo REST é um padrão poderoso e extremamente difundido e utilizado no desenvolvimento de sistemas atuais. E definir uma boa estratégia de teste que contemple este padrão é crucial para garantir a qualidade da solução [3].

SOUZA [5] propõe identificar qual das implementações de sockets tem melhor desempenho em que cenários, através de uma análise mais profunda na escolha da API ( Application program interface), realizando uma avaliação de desempenho com três APIs de Sockets: ZeroMQ, NanoMSG e Berkley Sockets.

Com isto, este trabalho visa avaliar através do JMeter o desempenho de serviços API REST implementados com linguagens mais tipadas (Spring) e a mesma API em um serviço menos tipado (Express), buscando assim ponderar se a tipagem dos frameworks interferem no desempenho da API.

## **4. Objetivos**

### **Objetivo Geral:**

- Discorrer uma análise sobre como tipagens diferentes de código podem influenciar no desempenho de um sistema distribuído.

### **Objetivos Específicos:**

- a) Definir um modelo de aplicação e codificar a mesma em duas tipagens distintas;
- b) Executar uma bateria de testes utilizando uma ferramenta específica para análise de performance em ambas aplicações;
- c) Documentar e analisar os resultados obtidos através dos teste executados.

## **5. Etapas do Projeto**

Nesta seção estão descritas as etapas de desenvolvimento deste projeto:

- 1) Investigar e selecionar uma database a ser utilizada para o projeto;
- 2) Implementar API REST em linguagem mais tipada (SPRING);
- 3) Implementar API REST em linguagem menos tipada (EXPRESS);
- 4) Executar testes com a ferramenta;
- 5) Avaliar os resultados;
- 6) Escrever o relatório contendo as considerações sobre o experimento.

## **6. Referências Bibliográficas**

- [1] SEBESTA, Robert W. **Conceitos de Linguagens de Programação-11**. Bookman Editora, 2018.
- [2] Wikipédia - Sistemas de Tipos - acessado em 1/4/2020 - link: [https://pt.wikipedia.org/wiki/Sistema\\_de\\_tipos](https://pt.wikipedia.org/wiki/Sistema_de_tipos)
- [3] FELLER, Nadja Jandt. Estendendo rest-unit: geração baseada em U2TP de drivers e dados de teste para RESTful Web Services. 2010.
- [4] JUNIOR, ANTONIO RICARDO MEDRONHA STEIN; DOS SANTOS, Maicon. ARQUITETURA REST API E DESENVOLVIMENTO DE UMA APLICAÇÃO WEB SERVICE. PROJETOS E RELATÓRIOS DE ESTÁGIOS, v. 1, n. 1, p. 1-59, 2019.
- [5] SOUZA, Samuel Pereira de. Análise de Desempenho de APIs de Sockets. 2016.