# Desenvolvimento de Plataforma de Reserva de Viagens em Java usando Threads

**Guilherme Martins, Lucas Moura, Mateus Regasi, Priscila Leite, Gustavo Gomes**

Instituto de Computação - Universidade Federal Fluminense (UFF) - Av. Gal. Milton Tavares de Souza, s/nº – Campus da Praia Vermelha – Boa Viagem – Niterói/RJ – Brasil

{guilhermemr,priscilalss,Lucaslimamoura,gustavo_viana,mateusregasi}@id.uff.br

***Abstract:*** *This work describes the development and implementation of a travel platform using threads in Java language to simulate flight and hotel reservations. The proposed system processes a list of users in two different ways: sequentially and in parallel. In the sequential approach, each user is processed one by one, where affordability or budget constraint is based on the available amount. In the parallel approach, the user list is divided into user blocks, allowing for simultaneous bookings, ensuring there are no inconsistencies as new orders are processed.*

***Resumo:*** *Este relatório descreve o desenvolvimento e a implementação de uma plataforma de viagens utilizando threads em linguagem Java para simular a reserva de voos e hotéis. O sistema proposto processa uma lista de usuários de duas formas distintas: de maneira sequencial e de maneira paralela. Na abordagem sequencial, cada usuário é processado um a um, onde a aceitação ou rejeição do orçamento é baseada no valor disponível. Na abordagem paralela, a lista de usuários é dividida em blocos de usuários, permitindo reservas simultâneas, assegurando que não haja inconsistências à medida que novos pedidos são processados.*

## 1. Introdução

O objetivo deste trabalho é a criação de uma plataforma de reserva de viagens e a utilização de threads em um projeto de programação usando a linguagem de programação Java, por meio deste projeto, iremos realizar um estudo sobre threads, como threads afetam o desempenho de um programa, as dificuldades para programar em paralelo, as vantagens e desvantagens de programar em paralelo e vantagens e desvantagens de programar de maneira sequencial.

## 2. Plataforma de Reserva de Viagens

O trabalho consiste em implementar uma plataforma de reserva de viagens, a qual os usuários entram na plataforma e solicitam um orçamento de viagem; depois eles

avaliam e aceitam ou não esse orçamento (considerando valor e requisitos de qualidade dos hotéis). O orçamento envolve escolher o hotel mais barato disponível para a quantidade de estrelas pedidas pelo usuário (exemplo, se pediu no mínimo 3 estrelas, pode ser 3, 4 ou 5, mas não 1 ou 2). O orçamento também envolve pegar o ponto de saída do usuário e encontrar voos mais baratos até chegar no destino. Exemplo: usuário mora no RIO e quer chegar em Palmas/TO (PMW), possivelmente ele precisará encontrar um voo RIO x Belo Horizonte/MG (CNF) e outro CNF x PMW. Para simplificar, iremos considerar no máximo duas conexões. À medida que usuários aceitam ou não o orçamento, os hotéis e voos podem ir se esgotando ou não, impactando assim os próximos usuários.

Foram utilizadas, no total, 6 classes, com intenção de seus nomes serem autoexplicativos. São estas: Csv, Funções, Paralelo, Cliente, Hotel, Voo.

## 3. Programação Sequencial

Programação sequencial é uma forma de programar na qual executa-se instruções (métodos, desvios, repetições, operações e etc…) na ordem em que foram especificados pela pessoa que desenvolveu determinada aplicação, a programação sequencial é a forma mais comum e mais simples de programar, haja vista que, o processo de pensamento, por exemplo, a estruturação de etapas a se concluir para executar uma determinada tarefa, é pensado pelas pessoas naturalmente de forma sequencial, esta forma de desenvolvimento de aplicações possui vantagens e desvantagens, a vantagem consiste em ser natural para as pessoas o processo de desenvolver algoritmos para executar tarefas de forma sequencial, como consequência desta vantagem, temos também a vantagem de ser mais fácil de achar erros e resolvê los, como desvantagem da programação sequencial temos a escalabilidade limitada, ou seja, conforme aumentamos a quantidade de dados e processos que temos em nossa aplicação, pior é o desempenho da aplicação, necessitando de um tempo maior para executar.

## 4. Programação com Threads

Programação com threads ou programação paralela, é uma estratégia usada para o desenvolvimento de aplicações, nesta forma de desenvolvimento não necessariamente executa-se as instruções na ordem em que foram especificadas, com a criação de threads é possível a execução de mais de uma tarefa ao mesmo tempo, portanto, podemos aproveitar os vários processadores presentes no nosso computador, a principal vantagem desta estratégia de desenvolvimento é a diminuição do tempo de execução das aplicações, podemos executar as mesmas tarefas de forma mais rápida, porém, não há só vantagens, o problema da programação com threads (paralela), consiste na dificuldade em fazer um programa eficiente e conciso que executa diversas tarefas ao mesmo tempo, a execução de múltiplas tarefas ao mesmo tempo não é simples de desenvolver,

haja vista que, você precisa fazer com que diversas threads trabalhem em cima de dados que podem ou não ter dependência entre si em determinada parte da execução da tarefa, outra desvantagem consiste em ser mais difícil de encontrar erros e corrigir na programação com threads, pois, com diversas instruções sendo executadas simultaneamente, não é de fácil visualização o local que está ocasionando o problema na execução da aplicação.

## 5 - Resultados Obtidos

Foram realizados experimentos computacionais para medir o tempo de execução das duas abordagens com conjuntos de 100, 1000, 10.000 e 20.000 clientes. Os resultados mostraram que a implementação paralela é mais eficiente do que a sequencial, exceto em cenários específicos onde a sobrecarga de gerenciamento de threads impactou negativamente a performance.

**Tabela 1. Tempo de execução para cada tipo de fluxo baseado no número de iterações para cada tipo de fluxo**

| | 100 clientes | 1.000 clientes | 10.000 clientes | 20.000 clientes |
|---|---|---|---|---|
| Sequencial | 97 ms | 463 ms | 3.318 ms | 38.998 ms |
| Paralela | 130 ms | 437 ms | 1.823 ms | 14.468 ms |

Desta maneira, pode-se observar que quanto maior for a quantidade de iterações, mais vantajoso será a utilização de threads. Porém, quando as iterações são poucas, o tempo de execução utilizando o fluxo paralelo será maior, visto que a criação e destruição das threads gerarão mais custos na implementação do que apenas um programa sequencial, pois para um pequeno número de iterações, essa sobrecarga pode ser significativa em relação ao trabalho que a thread realmente executa.

**Tabela 2. Porcentagem de clientes atendidos para cada tipo de fluxo**

| | 100 clientes | 1.000 clientes | 10.000 clientes | 20.000 clientes |
|---|---|---|---|---|
| Sequencial | 92% | 91% | 61,31% | 52,23% |
| Paralela | 92% | 92,4% | 67,46% | 63,42% |

Podemos observar que, o fluxo paralelo atende mais clientes que o sequencial. Isso acontece, porque o fluxo paralelo permite que múltiplas requisições sejam tratadas simultaneamente, ao contrário do fluxo sequencial, onde as requisições são tratadas uma a uma

## 6 - Conclusão

Observa-se, portanto, que para tarefas de curta duração ou com poucas iterações, a sobrecarga associada ao uso de threads pode superar os benefícios de paralelismo, resultando em tempos de execução maiores do que a execução sequencial. Para melhorar a performance de tarefas paralelizadas em tais cenários, é importante considerar o balanceamento da carga de trabalho, minimizar a comunicação entre threads, e utilizar técnicas de otimização para reduzir a sobrecarga de criação e destruição de threads. Conclui-se que o uso de threads pode significativamente melhorar a performance de sistemas de reserva em tempo real, desde que bem gerenciado.

## 7 - Referências Bibliográficas

W3Schools. "Java Threads". Disponível em: https://www.w3schools.com/java/java_threads.asp. Acesso em: 1 jul. 2024.

DevMedia. "Trabalhando com Threads em Java". Disponível em: https://www.devmedia.com.br/trabalhando-com-threads-em-java/28780. Acesso em: 1 jul. 2024.

MURTA, Leonardo. Programação Orientada a Objetos. 2020.1. Disponível em: https://leomurta.github.io/courses/2020.1/poo.html. Acesso em: 06 jul. 2024.