



# Estácio

**Missão Prática | Nível 5 | Mundo 3 - RPG0018 – Por que não paralelizar**

**Luciano Bento de Sousa Filho - Matrícula: 202302068766**

**Campus Shopping América - Goiás**

**Nível 5: Por que não paralelizar – Número da Turma – 2024.1**

## Objetivo da Prática

Nesse exercício, é esperado que o estudante demonstre as habilidades básicas no uso prático de Threads em ambientes cliente e servidor.

## Conclusão

- **A.** Como funcionam as classes Socket e ServerSocket?

**R:** As classes Socket e ServerSocket são fundamentais para a comunicação em rede usando a linguagem de programação Java. Elas permitem que programas estabeleçam conexões, enviem e recebam dados pela rede. A classe Socket é usada para criar um cliente que se conecta a um servidor. A classe ServerSocket é usada para criar um servidor que espera por conexões de clientes.

- **B.** Qual a importância das portas para a conexão com servidores?

**R:** As portas são um componente essencial das redes de computadores, permitindo a multiplexação de conexões, a diferenciação de serviços, a aplicação de políticas de segurança e o gerenciamento eficiente do tráfego de rede. Sem a utilização de portas,

seria impossível organizar e gerenciar as comunicações entre clientes e servidores de maneira eficiente e segura.

- **C.** Para que servem as classes de entrada e saída `ObjectInputStream` e `ObjectOutputStream`, e por que os objetos transmitidos devem ser serializáveis?

**R:** As classes `ObjectInputStream` e `ObjectOutputStream` são essenciais para a serialização e desserialização de objetos em Java. A serialização permite que objetos sejam transmitidos entre diferentes partes de um programa ou entre diferentes programas, preservando seu estado. Implementar a interface `Serializable` é crucial para que os objetos possam ser processados por esses streams.

- **D.** Por que, mesmo utilizando as classes de entidades JPA no cliente, foi possível garantir o isolamento do acesso ao banco de dados?

**R:** Ao utilizar classes de entidades JPA no cliente, o isolamento do acesso ao banco de dados é mantido através de uma arquitetura de software bem projetada que inclui camadas de abstração, serviços intermediários, uso de DTOs, controle de acesso e encapsulamento da lógica de negócios. Isso garante que o cliente não tenha acesso direto ao banco de dados e que todas as interações sejam mediadas pelo servidor, proporcionando segurança e integridade na manipulação dos dados.