

**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE QUIXADÁ**



**Relatório**

**SGP - Sistema de Gerenciamento de Produtos**

Disciplina: Sistemas Distribuídos – 2024.2

Equipe:

DAVID GABRIEL DE ALBUQUERQUE ALVES

FELIPE FEITOSA TEIXEIRA

GISLENO RODRIGUES DE ALENCAR SILVA JÚNIOR

Quixadá - CE.

## 1. Visão geral do serviço do remoto

O SGP (Sistema de Gerenciamento de Produtos) tem como objetivo facilitar a administração e controle dos produtos de uma organização, permitindo o acesso e gerenciamento de informações relacionadas aos produtos cadastrados. O sistema oferece funcionalidades, como adicionar, editar, listar e remover produtos, permitindo a consulta e alteração de estoque.

## 2. Descrição dos Métodos Remotos

Os seguintes métodos remotos estão disponíveis no sistema:

- **Adicionar Produto:** Permite registrar novos produtos no sistema, capturando informações como código, nome e preço.
- **Listar Produtos:** Retorna uma lista completa de todos os produtos registrados no sistema e suas informações como o preço.
- **Editar Produto:** Permite a alteração das informações de um produto existente, utilizando seu identificador único.
- **Remover Produto:** Exclui um produto do sistema com base em seu identificador.

## 3. Descrição dos Dados transmitidos

A modelagem dos dados trocados entre cliente e servidor segue um padrão estruturado conforme os arquivos .proto.

Os dados incluem:

- **Produto:**
  - **id (int32):** Identificador único do produto.
  - **nome (string):** Nome do produto.
  - **preço (float):** Preço do produto.

- **Mensagem de Resposta:**
  - **status (int32):** Código de status (200 para sucesso, 400 para erro, etc.).
  - **mensagem (string):** Texto explicativo sobre o resultado da operação.

As operações de adicionar, editar e remover produtos são encapsuladas em mensagens pré-definidas.

## 4. Descrição das classes implementadas nos lados Cliente e Servidor

No lado do **servidor**, as principais classes incluem:

- **ServidorUDP.java:** Responsável por gerenciar o socket UDP, recebendo e processando as requisições enviadas pelos clientes. Implementa a lógica de conexão e gerencia as respostas enviadas.
- **ProdutoServente.java:** Implementa a lógica central do sistema, incluindo as operações de cadastro, consulta, edição e remoção de produtos. Essa classe faz a validação dos dados e garante a integridade das informações armazenadas.
- **ProdutoDespachante.java:** Atua como um intermediário, recebendo requisições do servidor e encaminhando para as classes responsáveis pelo processamento.
- **ProdutoEsqueleto.java:** Implementa a interface entre o cliente e o servidor, garantindo que as chamadas remotas sejam corretamente interpretadas e encaminhadas para os serviços apropriados.

No lado do **cliente**, destacam-se:

- **client.py:** Implementa a interface do cliente para interação com o servidor, permitindo que os usuários realizem operações sobre os produtos. Essa classe lida com a entrada do usuário e exibe os resultados obtidos.

- **proxy\_cliente.py:** Atua como um intermediário entre o cliente e o servidor, traduzindo chamadas de métodos em mensagens de rede e processando as respostas recebidas.
- **udp\_client.py:** Responsável pelo envio de mensagens via UDP, garantindo que as solicitações do cliente sejam corretamente enviadas ao servidor.

## 5. Descrever o modelo de falhas

### 1. Possíveis Falhas no Serviço Distribuído:

- **Falhas de comunicação:** Perda de pacotes devido ao uso do protocolo UDP, causando retransmissões desnecessárias e duplicidade de mensagens.
- **Falhas no servidor:** O servidor pode falhar por sobrecarga, quedas inesperadas ou erros internos.
- **Falhas no cliente:** O cliente pode enviar requisições malformadas, resultando em erros de processamento.

### 2. Tratamentos de Falhas Implementados

- **Timeouts e retransmissão:** Implementado um mecanismo de timeout para reenvio de requisições perdidas.
- **Identificação de mensagens duplicadas:** Cada mensagem tem um identificador único para evitar processamento repetitivo.
- **Monitoramento de servidor:** O sistema registra logs para análise e correção de erros.

### 3. Tratamento de Exceções em Nível de Aplicação

- **Erros de conexão:** O cliente tenta se reconectar automaticamente antes de falhar.
- **Erros de requisição:** Caso dados inválidos sejam enviados, uma mensagem de erro é retornada ao cliente informando a causa.

## 6. Correções

Durante a apresentação do projeto, foi apontado que não estava sendo demonstrado adequadamente os mecanismos de tratamento de falhas, especificamente em relação à retransmissão de mensagens e à prevenção de duplicidades.

**Para corrigir essa lacuna, realizamos as seguintes ações:**

Um vídeo foi gravado para demonstrar na prática o funcionamento do sistema após testes do tratamento de falhas. O vídeo pode ser acessado pelo seguinte link: <https://youtu.be/1dGSPUfl0r8>