**Documentação de Testes – MS Catálogo**

**1. Introdução**

O objetivo desta documentação é descrever os testes a serem realizados no microserviço de catálogo de produtos (**ms-catalogo**). O sistema é desenvolvido em **Node.js (ESM)** com **Express 5** e utiliza **gRPC** para comunicação com o serviço de produtos.

**2. Escopo dos Testes**

Os testes contemplam:

* Rotas HTTP expostas pelo serviço.
* Comunicação com o serviço gRPC de produtos.
* Tratamento de erros e respostas adequadas.

**3. Ambiente de Testes**

* **Node.js:** >= 18
* **Framework:** Express 5.1.0
* **gRPC:** @grpc/grpc-js 1.13.4
* **Porta padrão do serviço:** 3001
* **Dependência externa:** Serviço gRPC ProdutosService rodando em products: 50051

**4. Casos de Teste**

**4.1 Testes de Saúde do Serviço**

**Caso:** Verificar se a rota de health check responde corretamente.

* **Endpoint:** GET /health
* **Entrada:** Nenhuma
* **Resultado Esperado:**
  + Status: 200
  + Corpo: "OK"

**4.2 Listagem de Produtos – Cenário de Sucesso**

**Caso:** Garantir que a listagem de produtos funciona corretamente.

* **Endpoint:** GET /produtos/
* **Pré-condição:** O serviço gRPC ProdutosService.ListProdutos está disponível e retorna dados válidos.
* **Resultado Esperado:**
  + Status: 200
  + Corpo: JSON contendo a lista de produtos ([{id, nome, preco, ...}]).

**4.3 Listagem de Produtos – Erro de Conexão gRPC**

**Caso:** O serviço gRPC está indisponível.

* **Endpoint:** GET /produtos/
* **Simulação:** Derrubar o container products:50051.
* **Resultado Esperado:**
  + Status: 500
  + Corpo: { "error": "Erro ao listar produtos" }
  + Log no servidor: "Erro ao listar produtos:" seguido do stack trace.

**4.4 Listagem de Produtos – Resposta Vazia**

**Caso:** O gRPC responde sem produtos.

* **Endpoint:** GET /produtos/
* **Mock gRPC:** ListProdutos retorna { produtos: [] }
* **Resultado Esperado:**
  + Status: 200
  + Corpo: []

**4.5 Validação de Estrutura da Resposta**

**Caso:** Validar se a resposta sempre retorna um JSON válido.

* **Endpoint:** GET /produtos/
* **Entrada:** Nenhuma
* **Resultado Esperado:**
  + O Content-Type deve ser application/json.
  + Mesmo em erro, deve retornar JSON válido com chave "error".

**5. Estratégia de Teste**

* **Testes unitários:**
  + Função getProdutos() no catalogo.service.js
  + Função listarProdutos() no catalogo.controller.js
* **Testes de integração:**
  + Rotas /health e /produtos usando **Supertest** ou **Jest**.
  + Simulações com gRPC mockado.
* **Testes end-to-end:**
  + Execução em ambiente Docker com o serviço gRPC ativo e indisponível em diferentes cenários.

**6. Critérios de Aceitação**

* Nenhuma rota deve retornar erro não tratado (stack trace direto para o cliente).
* Logs devem registrar erros de gRPC corretamente.

**7. Riscos e Considerações**

* A indisponibilidade do serviço gRPC de produtos impacta diretamente a listagem.