**Relatório de testes manuais unitários, integração e ponto a ponto.**

Logotipo

O conteúdo gerado por IA pode estar incorreto.

Gabriel Barbosa Zanon

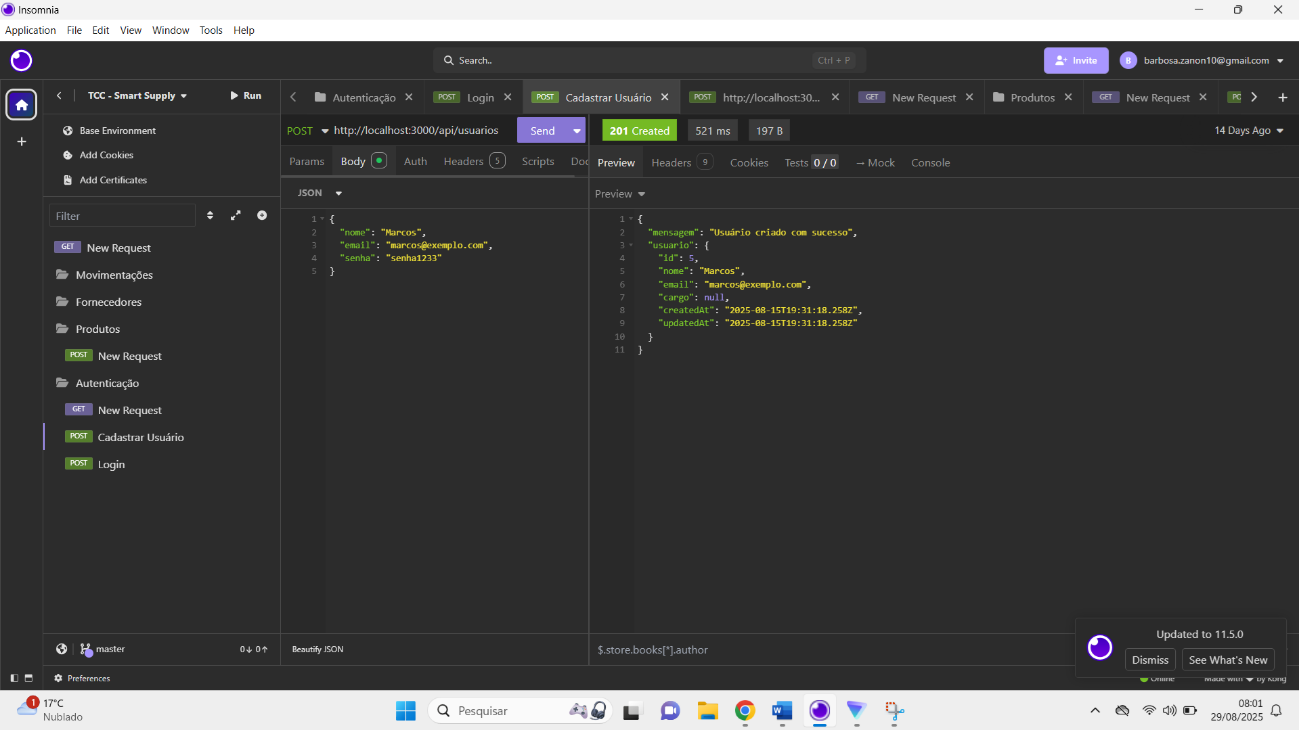
Kauê H. C. Fidelis

Marcos V. Oliveira

Lucas G. Giachetto

Lucas Colombo

**IMPLEMENTAÇÃO JWT AUTENTICAÇÃO DE USUÁRIO**

****

**Retorno: 201 Created**

A requisição retorna os dados de um usuário recém-criado, distinguindo entre **administrador** e **cliente**. Durante o processo, é gerada uma senha única acompanhada de um **token temporário** (com tempo de expiração definido), além do registro de informações como **e-mail** e **nome** do usuário.

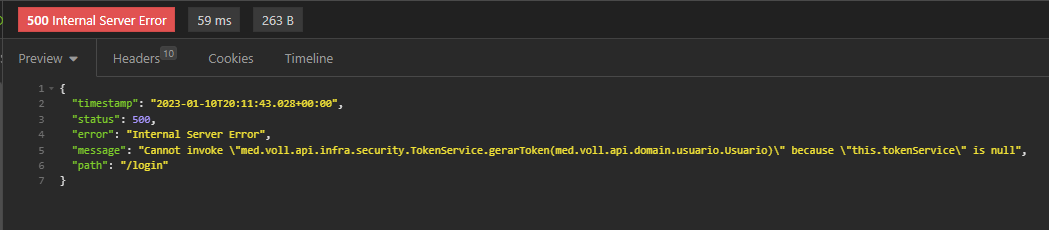
**Retorno: 400 Bad Request**

Isso ocorre quando os dados no processo de requisição são inválidos ou estão incompletos, como **E-mail** repetido ou com informações e caracteres incompletos. Isso retorna **Error 400 Bad Request** na requisição.

**Retorno: 409 Conflict**

Isso ocorre quando já existe um usuário cadastrado com o mesmo e-mail. Justamente por isso, na requisição é retornado **409 Conflict** pois há um conflito de informações no processo de cadastro de **administrador** e **cliente**.

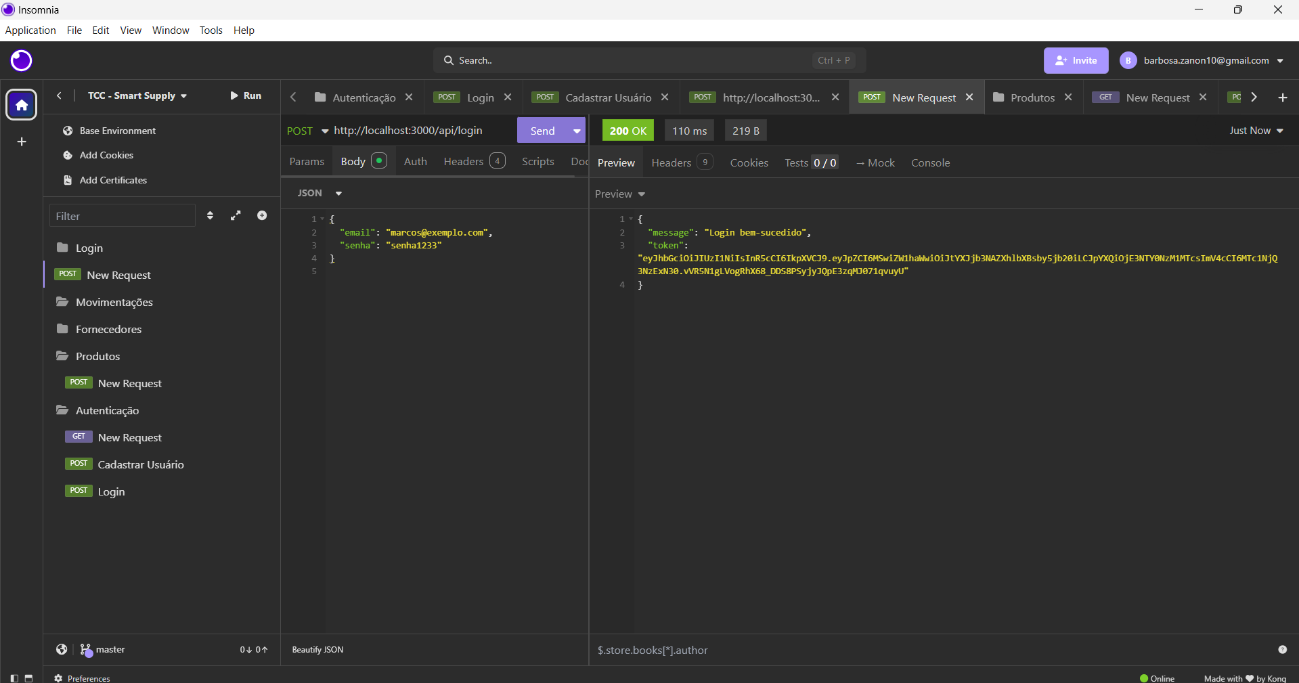
**Retorno: 500 Internal Server Error**



Indica uma falha no servidor durante o processo de criação, possivelmente algo relacionado com a ativação do **Back-End** ou pacote de softwares (ambiente de servidor local no computador).

**LOGIN E GERAÇÃO TOKEN**

**Retorno: 200 Requisição bem sucedida**



A requisição retorna a mensagem “login bem-sucedido” juntamente com o Token de autenticação do usuário criado.

**Retorno: 400 Bad Request**

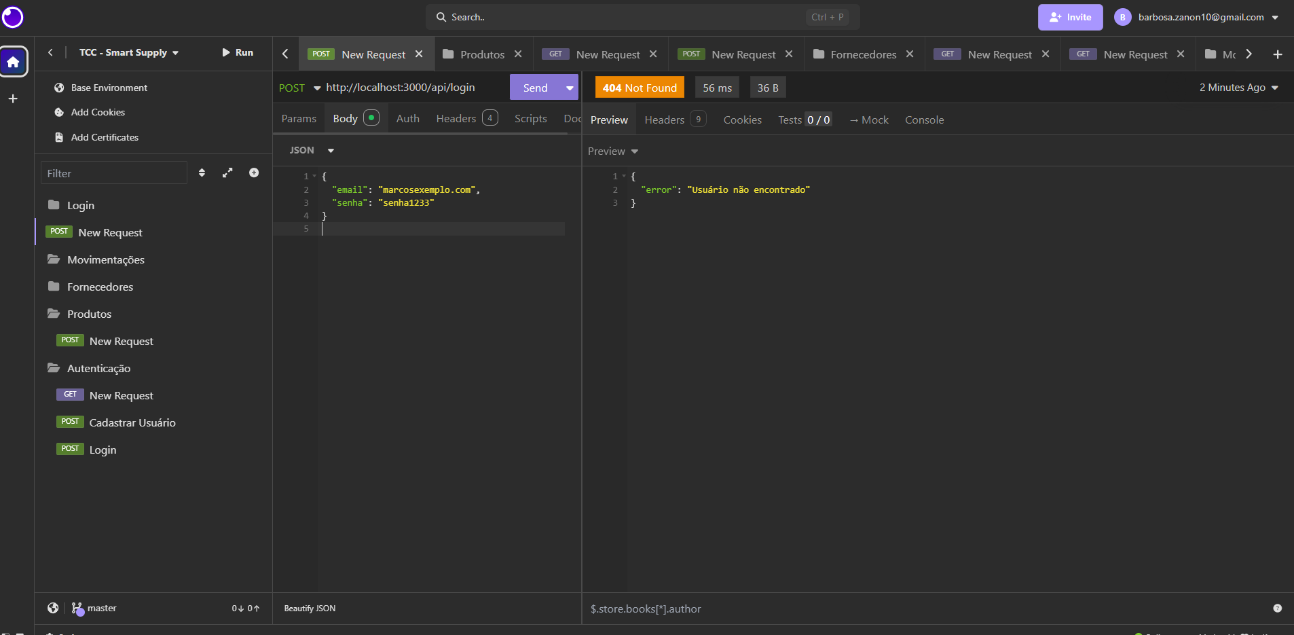
Ocorre quando o cliente não envia todos os dados obrigatórios (ex.: e-mail ou senha ausentes).

**Retorno: 401 Unauthorized**

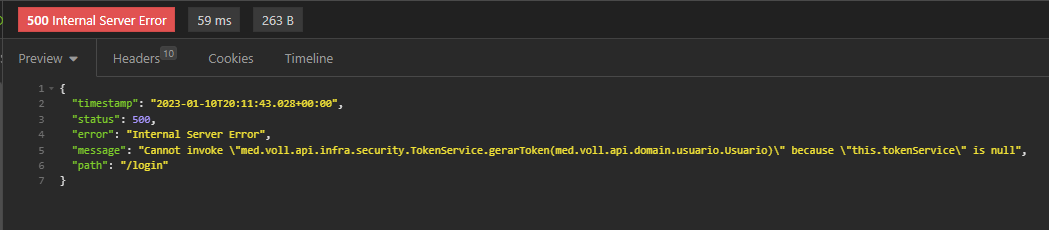
Ocorre quando o usuário fornece credenciais incorretas (e-mail não encontrado ou senha inválida).

**Retorno: 404 Not Found**

Ocorre quando o usuário não é encontrado e suas credenciais não são validadas na requisição.

****

**Retorno: 500 Internal Server Error**



Indica uma falha no servidor durante o processo de criação, possivelmente algo relacionado com a ativação do **Back-End** ou pacote de softwares (ambiente de servidor local no computador).

**TESTES CONTROLLERS**

**CONTROLLER PRODUTOS**

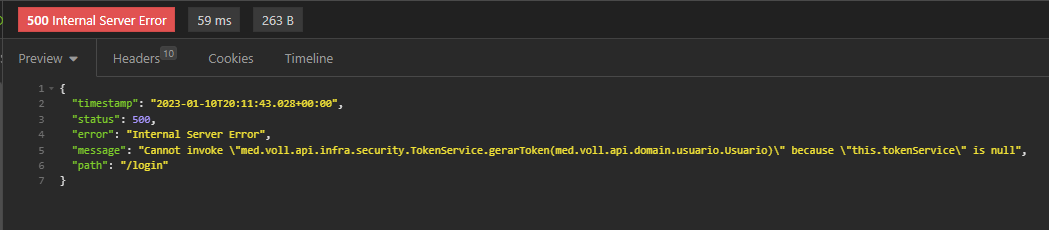
**Endpoints a testar:**

* POST /produtos → criar produto
* GET /produtos → listar produtos
* GET /produtos/:id → buscar produto por ID
* PUT /produtos/:id → atualizar produto
* DELETE /produtos/:id → excluir produto

**Casos de teste sugeridos:**

| **Caso** | **Entrada** | **Esperado** | **Status HTTP** |
| --- | --- | --- | --- |
| Criar produto válido | nome, descricao, codigoBarras, quantidade, localEstoque, fornecedorId | Produto criado no DB, retorna dados do produto | 201 Created |
| Criar produto com codigoBarras duplicado | Mesmo codigoBarras de outro produto | Retorna erro de duplicidade | 400 Bad Request |
| Buscar produto existente | ID válido | Retorna dados do produto | 200 OK |
| Buscar produto inexistente | ID inválido | Retorna mensagem de erro | 404 Not Found |
| Atualizar produto | Alterar quantidade ou descrição | Retorna produto atualizado | 200 OK |
| Atualizar produto com codigoBarras duplicado | Mesmo código de outro produto | Retorna erro | 400 Bad Request |
| Excluir produto existente | ID válido | Produto removido | 204 No Content |
| Excluir produto inexistente | ID inválido | Retorna mensagem de erro | 404 Not Found |

**Retorno: 500 Internal Server Error**



Indica uma falha no servidor durante o processo de criação, possivelmente algo relacionado com a ativação do **Back-End** ou pacote de softwares (ambiente de servidor local no computador).

**CONTROLLER FORNECEDOR**

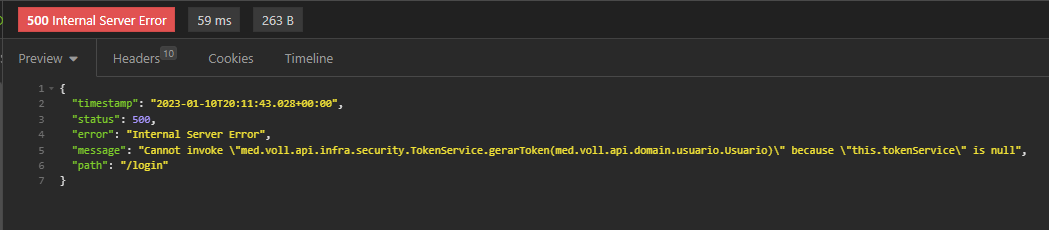
**Endpoints a testar:**

* POST /fornecedores
* GET /fornecedores
* GET /fornecedores/:id
* PUT /fornecedores/:id
* DELETE /fornecedores/:id

**Casos de teste:**

| **Caso** | **Entrada** | **Esperado** | **Status HTTP** |
| --- | --- | --- | --- |
| Criar fornecedor válido | nome, cnpj, contato, email | Fornecedor criado | 201 Created |
| Criar fornecedor com CNPJ duplicado | Mesmo CNPJ | Retorna erro de duplicidade | 400 Bad Request |
| Buscar fornecedor existente | ID válido | Retorna dados do fornecedor | 200 OK |
| Atualizar fornecedor | Alterar nome ou email | Retorna fornecedor atualizado | 200 OK |
| Excluir fornecedor com produtos associados | ID de fornecedor com produtos | Retorna erro ou permite exclusão dependendo da regra | 400/204 |

**Retorno: 500 Internal Server Error**



Indica uma falha no servidor durante o processo de criação, possivelmente algo relacionado com a ativação do **Back-End** ou pacote de softwares (ambiente de servidor local no computador).

**Retorno: 401 Unauthorized**

**Tela de celular

O conteúdo gerado por IA pode estar incorreto.**

**CONTROLLER MOVIMENTACOES**

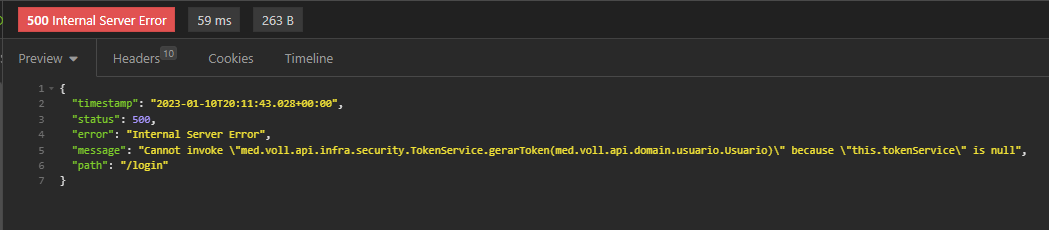
**Endpoints a testar:**

* POST /movimentacoes
* GET /movimentacoes
* GET /movimentacoes/:id

**Casos de teste:**

| **Caso** | **Entrada** | **Esperado** | **Status HTTP** |
| --- | --- | --- | --- |
| Criar movimentação entrada | tipo: 'entrada', quantidade, produtoId | Movimentação criada, estoque atualizado | 201 Created |
| Criar movimentação saída | tipo: 'saida', quantidade, produtoId | Verifica se quantidade não excede estoque | 400 Bad Request (estoque insuficiente) |
| Buscar movimentação existente | ID válido | Retorna movimentação | 200 OK |
| Buscar movimentação inexistente | ID inválido | Retorna erro | 404 Not Found |

**Retorno: 500 Internal Server Error**



Indica uma falha no servidor durante o processo de criação, possivelmente algo relacionado com a ativação do **Back-End** ou pacote de softwares (ambiente de servidor local no computador).

**CONTROLLER USUARIO**

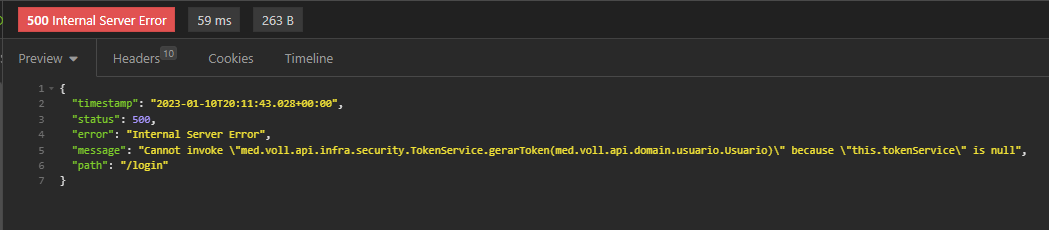
**Endpoints a testar:**

* POST /usuarios → criar usuário
* POST /login → autenticação
* GET /usuarios → listar usuários
* GET /usuarios/:id → buscar usuário por ID
* PUT /usuarios/:id → atualizar usuário
* DELETE /usuarios/:id → excluir usuário

**Casos de teste:**

| **Caso** | **Entrada** | **Esperado** | **Status HTTP** |
| --- | --- | --- | --- |
| Criar usuário válido | nome, email, senha, cargo | Usuário criado | 201 Created |
| Criar usuário com email duplicado | Mesmo email | Retorna erro de duplicidade | 400 Bad Request |
| Login com credenciais corretas | email, senha | Retorna token JWT | 200 OK |
| Login com senha incorreta | email, senha inválida | Retorna mensagem de erro | 401 Unauthorized |
| Atualizar usuário | Alterar nome ou senha | Retorna usuário atualizado | 200 OK |
| Excluir usuário | ID válido | Usuário removido | 204 No Content |
| Buscar usuário inexistente | ID inválido | Retorna erro | 404 Not Found |

**Retorno: 500 Internal Server Error**



Indica uma falha no servidor durante o processo de criação, possivelmente algo relacionado com a ativação do **Back-End** ou pacote de softwares (ambiente de servidor local no computador).

**MATRIZ SWOT (FOFA)**

**Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.**

**DIAGRAMA DE CLASSES (ATUALIZADO)**

**Interface gráfica do usuário, Aplicativo, Word

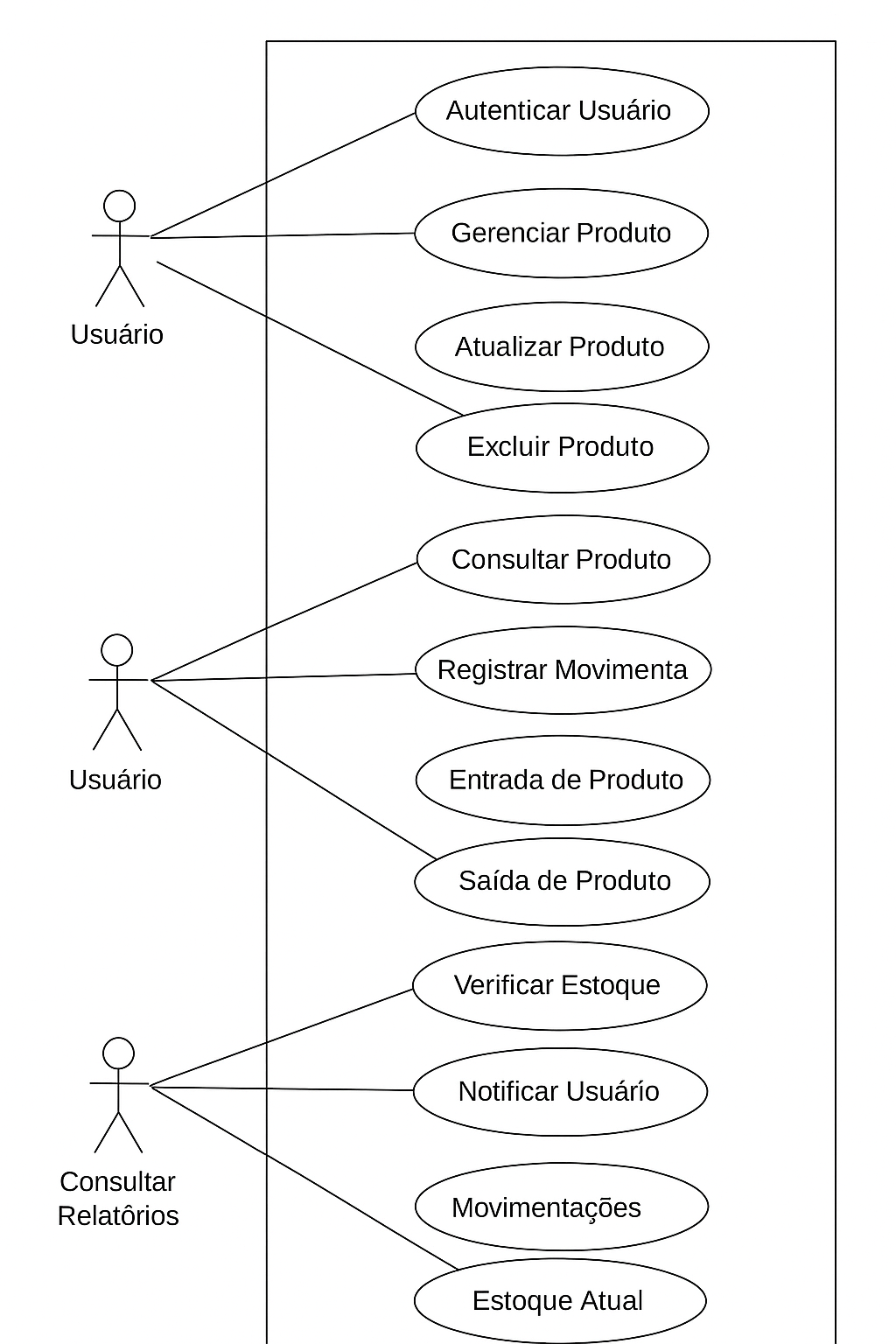
O conteúdo gerado por IA pode estar incorreto.**

**DIAGRAMA DE ATIVIDADES (ATUALIZADO)**

**Diagrama

O conteúdo gerado por IA pode estar incorreto.**

**DIAGRAMA DE CASO DE USO (ATUALIZADO)**



**CONCLUSÃO**

Em conclusão, os testes realizados nos controllers do sistema comprovaram a funcionalidade e a robustez das operações de CRUD para as entidades **Produto, Fornecedor, Movimentacao e Usuario**, garantindo que tanto os fluxos de sucesso quanto os casos de erro sejam tratados corretamente. A validação de dados, o controle de duplicidade e a manutenção das relações entre entidades mostraram-se consistentes, e os endpoints retornaram os códigos HTTP apropriados para cada situação. Com isso, o sistema se mostra confiável, seguro e pronto para suportar operações reais, fornecendo respostas claras e previsíveis para os usuários e integradores da API.