

Documento de Especificação de Casos de Uso

Sistema de Gestão de Estoque e Produtos

Autores: [Matheus Lopes]

Data: 24 de Julho de 2025

Versão: 1.0

Sumário

1. Introdução	3
2. Diagrama de Casos De Uso	4
3. Especificação de Caso de uso	5
3.1. Fazer Login (UC-01).....	5
3.2. Ver Produtos (UC-02)	6
3.3. Ver Estoque (UC-03).....	7
3.4. Imprimir Relação Produtos PDF	8
3.5. Adicionar Produto	9
3.6. Excluir Produto	10
3.7. Editar Produto	11
3.8. Registrar Entrada de Estoque.....	12
3.9. Registrar Saída de Estoque	13
3.10. Exportar Relatório de Estoque XLSX.....	14
3.11. Organizar Estoque	14
3.13. Ordenar Estoque por Ordem Alfabética	15
3.14. Gerenciar Usuários.....	16
3.18. Gerenciar Naturezas	18
3.19. Gerenciar Fornecedores	18
3.20. Registrar Entrada Rápida de Estoque por Código	19
3.21. Visualizar Dashboard	20
3.22. Registrar Saída Rápida de Estoque por Código	20
3.23. Gerar Relatórios	21
4. Diagrama de Classes	22
5. Documento de Requisitos Não-Funcionais	23
1. Introdução	23
2. Requisitos Não Funcionais Detalhados	23
6. Diagrama de Entidade e Relacionamentos.....	27
7. Modelo Lógico do Banco de Dados.....	28
8. Modelo Físico do Banco de Dados(Sql Script)	29
9. Glossário	32

1. Introdução

1.1 Objetivo Do Documento

Este documento detalha as funcionalidades do Sistema de Gestão de estoque, especificando os requisitos funcionais através de casos de uso e descrevendo, o funcionamento do sistema em alinhamento com o banco de dados, e interações entre os atores e o sistema.

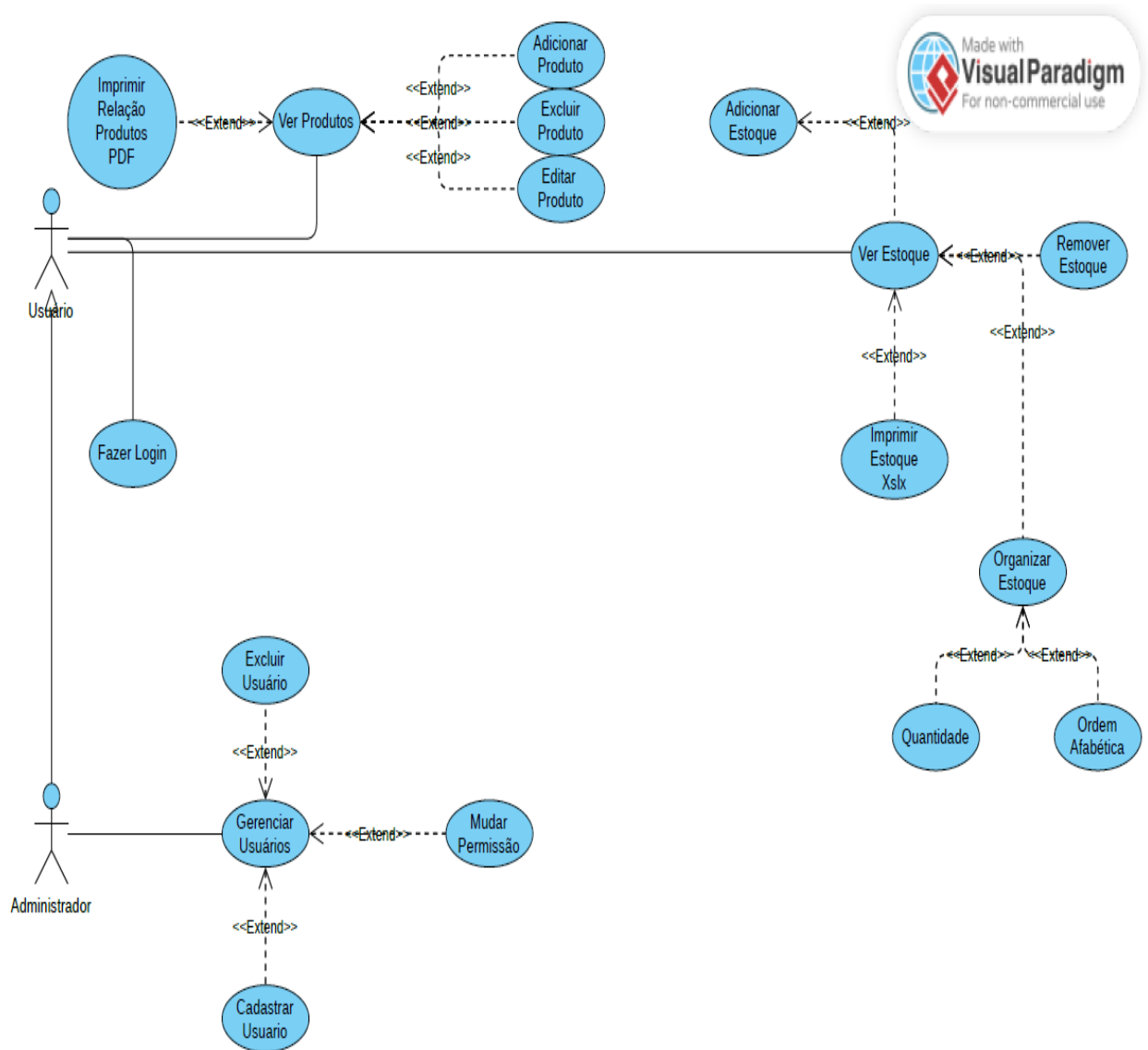
1.2 Escopo Do Sistema

O Sistema permitira o gerenciamento de um catálogo de produtos, o controle de seus respectivos estoques e a administração de contas de usuários com diferentes níveis de permissão.

1.3 Atores:

- Administrador: Responsável por gerenciar todo o sistema, incluindo produtos, estoques e contas de usuários.
- Usuário: Um ator com permissões limitadas, capaz de visualizar produtos e estoques.

2. Diagrama de Casos De Uso



Acima segue o diagrama de caso de uso, onde o autor, “Usuário” pode interagir com vários casos de uso, e alguns possuem extensões, o autor “Administrador” tem acesso a todos os casos de uso do autor “Usuário” e ao caso de uso “Gerenciar Usuário” com suas respectivas extensões.

3. Especificação de Caso de uso

3.1. Fazer Login (UC-01)

- **ID:** UC-01
- **Nome:** Fazer Login
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite que um ator se autentique no sistema para obter acesso às suas funcionalidades.
- **Pré-condições:** O ator deve possuir uma conta ativa no sistema.
- **Pós-condições (Sucesso):** O ator é autenticado, um token de acesso é gerado, e a janela principal da aplicação é exibida.
- **Pós-condições (Falha):** O acesso é negado e o sistema permanece na tela de login.
- **Fluxo Principal:**
 1. O ator abre a aplicação e visualiza a tela de login.
 2. O ator insere o seu `login` e `senha` nos campos apropriados.
 3. O ator clica no botão "Entrar".
 4. O sistema envia as credenciais para a API (`POST /api/login`).
 5. A API valida as credenciais contra a base de dados.
 6. A API retorna um token de acesso JWT.
 7. A aplicação guarda o token, fecha a janela de login e exibe a `JanelaPrincipal`.
- **Fluxos Alternativos:**
 - **FA-01: Credenciais inválidas:** No passo 5, se o login ou a senha estiverem incorretos ou o usuário estiver inativo, a API retorna um erro 401. O sistema exibe uma mensagem de "Credenciais inválidas".
 - **FA-02: Campos vazios:** No passo 2, se o ator clicar em "Entrar" com campos vazios, o sistema exibe uma mensagem de erro localmente.

3.2. Ver Produtos (UC-02)

- **ID:** UC-02
 - **Nome:** Ver Produtos
 - **Ator(es):** Usuário, Administrador
 - **Descrição:** Permite visualizar a lista de todos os produtos cadastrados, com opções de busca e acesso às funcionalidades de gestão.
 - **Pré-condições:** O ator deve estar autenticado.
 - **Pós-condições (Sucesso):** A lista de produtos é exibida na tela.
-
- **Fluxo Principal:**
 1. O ator clica na opção "Produtos" na navegação.
 2. O sistema faz um pedido à API (GET /api/produtos).
 3. A API retorna a lista de produtos.
 4. O sistema exibe os produtos numa tabela com as colunas Código, Nome, Descrição, Preço, Código B, Código C, Fornecedores e Naturezas.
A tabela não permite edição direta.
-
- **Fluxos Alternativos:**
 - **FA-01: Busca em tempo real:** Conforme o ator digita na barra de pesquisa, o sistema aguarda uma breve pausa e refaz o pedido à API com um parâmetro de busca (ex: ?search=termo), atualizando a tabela com os resultados filtrados.

3.3. Ver Estoque (UC-03)

- **ID:** UC-03
- **Nome:** Ver Estoque
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite visualizar o saldo de estoque atual para todos os produtos.
- **Pré-condições:** O ator deve estar autenticado.
- **Pós-condições (Sucesso):** A lista de saldos de estoque é exibida.

Fluxo Principal:

1. O ator clica na opção "Estoque" na navegação.
2. O sistema faz um pedido à API (GET /api/estoque/saldos).
3. A API calcula o saldo de cada produto a partir das movimentações e retorna a lista.
4. O sistema exibe os dados numa tabela com as colunas Código, Nome do Produto e Saldo Atual. A tabela não permite edição direta.

3.4. Imprimir Relação Produtos PDF

- **ID:** UC-04
- **Nome:** Imprimir Relação Produtos PDF
- **Ator(es):** Usuário, Administrador
- **Descrição:** Gera um relatório em PDF da lista de produtos.
- **Pré-condições:** O ator deve estar na tela de Relatórios.
- **Fluxo Principal:**
 1. O ator acede à tela "Relatórios".
 2. O ator clica no botão "Exportar Produtos (PDF)".
 3. O sistema faz um pedido à API (GET `/api/relatorios/produtos?formato=pdf`).
 4. A API gera o ficheiro PDF e o front-end aciona o download para o usuário.

3.5. Adicionar Produto

- **ID:** UC-05
- **Nome:** Adicionar Produto
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite o cadastro de um novo produto no sistema, incluindo as suas associações.
- **Pré-condições:** O ator deve estar autenticado.
- **Fluxo Principal:**
 1. Na tela "Produtos", o ator clica em "Adicionar Novo".
 2. O sistema abre a `FormularioProdutoDialog` em modo de adição. O formulário exibe campos para todos os dados do produto e listas de seleção para `Fornecedores` e `Naturezas`.
 3. O ator preenche os dados e seleciona as associações.
 4. O ator clica em "Salvar".
 5. O sistema envia os dados para a API (`POST /api/produtos`).
 6. Após a criação do produto, o sistema envia pedidos adicionais para criar as associações (`POST /api/produtos/<id>/fornecedores`, etc.).
 7. Ao receber a confirmação de sucesso, o formulário fecha e a tabela de produtos é atualizada.
- **Fluxos Alternativos:**
 - **FA-01: Verificação de código em tempo real:** Ao preencher o campo "Código", o sistema verifica a sua disponibilidade na API e exibe um status visual.
 - **FA-02: Adição rápida de item relacionado:** O ator pode clicar num botão "+" ao lado das listas de seleção para abrir um formulário rápido de cadastro de um novo Fornecedor ou Natureza.

3.6. Excluir Produto

- **ID:** UC-06
- **Nome:** Excluir Produto
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite a remoção de um produto do sistema.
- **Pré-condições:** O ator deve estar autenticado.
- **Fluxo Principal:**
 1. Na tela "Produtos", o ator seleciona um produto e clica em "Excluir".
 2. O sistema exibe uma caixa de diálogo a pedir confirmação.
 3. Ao confirmar, o sistema envia um pedido `DELETE` para `/api/produtos/<id>`.
 4. A API verifica se o produto não possui histórico de movimentações.
 5. Se a verificação passar, a API remove o produto e as suas associações.
 6. A tabela de produtos é atualizada.
- **Fluxos Alternativos:**
 - **FA-01: Produto com histórico de estoque:** No passo 4, se o produto tiver movimentações, a API retorna um erro. O sistema exibe uma mensagem clara ao usuário a informar que a exclusão não é permitida.

3.7. Editar Produto

- **ID:** UC-07
- **Nome:** Editar Produto
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite a alteração dos dados de um produto existente.
- **Pré-condições:** O ator deve estar autenticado.
- **Fluxo Principal:**
 1. Na tela "Produtos", o ator seleciona um produto e clica em "Editar".
 2. O sistema abre a `FormularioProdutoDialog` em modo de edição, pré-preenchida com os dados atuais do produto, incluindo as associações de fornecedores e naturezas já marcadas.
 3. O ator altera os dados e clica em "Salvar".
 4. O sistema envia os dados para a API (`PUT /api/produtos/<id>`).
 5. A API atualiza o produto e sincroniza as associações (remove as antigas, adiciona as novas).
 6. O formulário fecha e a tabela de produtos é atualizada.

3.8. Registrar Entrada de Estoque

- **ID:** UC-08
- **Nome:** Registrar Entrada de Estoque
- **Ator(es):** Usuário, Administrador
- **Descrição:** Cria um registo de transação para a entrada de unidades de um produto.
- **Pré-condições:** O ator deve estar autenticado.
- **Fluxo Principal:**
 1. O ator acede a uma tela de movimentação (ex: Entrada Rápida).
 2. O ator especifica o produto e a quantidade a ser adicionada.
 3. Ao confirmar, o sistema envia os dados para a API (`POST /api/estoque/entrada`).
 4. A API cria um novo registo na tabela `mov_estoque` com `tipo = 'Entrada'`, associando o usuário logado como responsável.

3.9. Registrar Saída de Estoque

- **ID:** UC-09
- **Nome:** Registrar Saída de Estoque
- **Ator(es):** Usuário, Administrador
- **Descrição:** Cria um registo de transação para a saída de unidades de um produto.
- **Pré-condições:** O ator deve estar autenticado.
- **Fluxo Principal:**
 1. O ator acede a uma tela de movimentação (ex: Saída Rápida).
 2. O ator especifica o produto, a quantidade a ser retirada e um motivo obrigatório.
 3. Ao confirmar, o sistema envia os dados para a API (`POST /api/estoque/saida`).
 4. A API primeiro verifica se o saldo de estoque é suficiente.
 5. Se for, a API cria um novo registo na tabela `mov_estoque` com `tipo = 'Saida'`.
- **Fluxos Alternativos:**
 - **FA-01: Estoque insuficiente:** No passo 4, se a quantidade a retirar for maior que o saldo, a API retorna um erro e o sistema exibe uma mensagem de "Estoque insuficiente".

3.10. Exportar Relatório de Estoque XLSX

- **ID:** UC-10
- **Nome:** Exportar Relatório de Estoque XLSX
- **Ator(es):** Usuário, Administrador
- **Descrição:** Gera um relatório em Excel (XLSX) do inventário atual.
- **Pré-condições:** O ator deve estar na tela de Relatórios.
- **Fluxo Principal:**
 1. O ator acede à tela "Relatórios" e seleciona a opção de Inventário.
 2. O ator clica no botão "Exportar para Excel".
 3. O sistema faz um pedido à API (GET `/api/relatorios/inventario?formato=xlsx`).
 4. A API gera o ficheiro e o front-end aciona o download.

3.11. Organizar Estoque

- **ID:** UC-11
- **Nome:** Organizar Estoque
- **Descrição:** Este caso de uso representa a ação do usuário de reordenar a visualização da tabela de estoque. É um agrupador para os casos de uso de ordenação específicos.

3.12. Ordenar Estoque Por Quantidade

- **ID:** UC-12
- **Nome:** Ordenar Estoque Por Quantidade
- **Ator(es):** Usuário, Administrador

- **Descrição:** Permite reordenar a tabela de estoque com base na coluna "Saldo Atual".
- **Fluxo Principal:**
 1. Na tela "Estoque", o ator clica no cabeçalho da coluna "Saldo Atual".
 2. A interface reordena os dados exibidos de forma ascendente.
 3. Um segundo clique inverte a ordem para descendente.

3.13. Ordenar Estoque por Ordem Alfabética

- **ID:** UC-13
- **Nome:** Ordenar Estoque por Ordem Alfabética
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite reordenar a tabela de estoque com base na coluna "Nome do Produto".
- **Fluxo Principal:**
 1. Na tela "Estoque", o ator clica no cabeçalho da coluna "Nome do Produto".
 2. A interface reordena os dados exibidos de forma alfabética (A-Z).
 3. Um segundo clique inverte a ordem (Z-A).

3.14. Gerenciar Usuários

- **ID do Caso de Uso:** UC-14
- **Nome do Caso de Uso:** Gerenciar Usuários
- **Ator(es):** Administrador
- **Descrição:** Permite que o Administrador visualize, crie, edite e desative/reactive contas de usuário, controlando o acesso ao sistema.
- **Pré-condições:**
 1. O ator deve estar autenticado com a permissão de "Administrador".
- **Fluxo Principal (Visualizar e Adicionar):**
 1. O Administrador acede à tela "Gerenciar Usuários".
 2. O sistema exibe uma tabela com todos os usuários, mostrando as colunas Nome, Login, Permissão e Status ("Ativo" ou "Inativo").
 3. O Administrador clica em "Adicionar Novo".
 4. O sistema abre a `FormularioUsuarioDialog`.
 5. O Administrador preenche os dados (Nome, Login, Senha, Permissão) e salva.
 6. A API cria o novo usuário e a tabela na interface é atualizada.
- **Fluxos Alternativos e de Exceção:**
 - **FA-01: Editar um Usuário:**
 1. No passo 2, o Administrador seleciona um usuário na tabela e clica em "Editar".
 2. O sistema abre a `FormularioUsuarioDialog` pré-preenchida com os dados do usuário. O campo de senha fica em branco.
 3. O Administrador altera os dados desejados. Se o campo de senha for preenchido, ela será atualizada.
 4. Ao salvar, a API (`PUT /api/usuarios/<id>`) atualiza os dados e a tabela é recarregada.
 - **FA-02: Desativar um Usuário Ativo:**
 1. No passo 2, o Administrador seleciona um usuário "Ativo" e clica em "Desativar".
 2. O sistema exibe uma caixa de diálogo a pedir confirmação.
 3. Ao confirmar, a API (`DELETE /api/usuarios/<id>`) executa a exclusão lógica (soft delete).
 4. A tabela é atualizada, e o status do usuário muda para "Inativo".
 - **FA-03: Reativar um Usuário Inativo:**

1. No passo 2, o Administrador seleciona um usuário "Inativo". O botão "Desativar" pode mudar para "Reativar".
2. O Administrador clica em "Reativar".
3. Ao confirmar, a API (`PUT /api/usuarios/<id>`) altera o status do usuário para `ativo = true`.
4. A tabela é atualizada, e o status do usuário muda para "Ativo".

3.18. Gerenciar Naturezas

- **ID:** UC-18
- **Nome:** Gerenciar Naturezas
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite o CRUD completo para as naturezas dos produtos.
- **Fluxo Principal:**
 1. O ator acede à tela "Naturezas".
 2. O sistema exibe uma tabela com as naturezas existentes.
 3. O ator pode Adicionar, Editar ou Excluir uma natureza através dos botões de ação, que abrem um formulário de diálogo para a inserção/edição do nome. A exclusão é impedida se a natureza estiver em uso.

3.19. Gerenciar Fornecedores

- **ID:** UC-19
- **Nome:** Gerenciar Fornecedores
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite o CRUD completo para os fornecedores.
- **Fluxo Principal:**
 1. O ator acede à tela "Fornecedores".
 2. O sistema exibe uma tabela com os fornecedores existentes.
 3. O ator pode Adicionar, Editar ou Excluir um fornecedor através dos botões de ação, que abrem um formulário de diálogo para a inserção/edição do nome. A exclusão é impedida se o fornecedor estiver em uso.

3.20. Registrar Entrada Rápida de Estoque por Código

- **ID do Caso de Uso:** UC-20
- **Nome do Caso de Uso:** Registrar Entrada Rápida de Estoque por Código
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite o registro otimizado de entradas de estoque através da inserção de um código de produto.
- **Fluxo Principal:**
 1. O ator acede à tela "Entrada Rápida".
 2. O sistema exibe um campo para o "Código do Produto" e um botão "Verificar".
 3. O ator digita um código e clica em "Verificar" (ou pressiona Enter).
 4. O front-end faz um pedido `GET` para `/api/produtos/codigo/<codigo>`.
 5. Se o produto existir, o sistema exibe o nome do produto para confirmação visual e habilita o campo "Quantidade" e o botão "Registrar Entrada".
 6. O ator insere a quantidade e clica em "Registrar Entrada" (ou pressiona Enter).
 7. O front-end faz um pedido `POST` para `/api/estoque/entrada`.
 8. O sistema exibe uma mensagem de sucesso e limpa a tela, pronta para a próxima entrada.
- **Fluxos Alternativos e de Exceção:**
 - **FA-01: Código do produto não encontrado:** No passo 4, se o código não existir, o sistema exibe uma mensagem de erro clara e mantém o foco no campo de código.

3.21. Visualizar Dashboard

- **ID do Caso de Uso:** UC-21
- **Nome do Caso de Uso:** Visualizar Dashboard
- **Ator(es):** Usuário, Administrador
- **Descrição:** Apresenta uma visão geral e resumida do estado do sistema, servindo como tela inicial.
- **Fluxo Principal:**
 1. Após o login, a tela do Dashboard é exibida.
 2. O sistema exibe a logo da aplicação.
 3. O front-end faz pedidos à API para buscar dados agregados.
 4. O sistema exibe os dados em cartões de KPI (Indicadores-Chave):
 - **Total de Produtos:** Contagem de todos os produtos cadastrados.
 - **Total de Fornecedores:** Contagem de todos os fornecedores.
 - **Valor Total em Estoque:** Soma do `(saldo_atual * preço)` de cada produto.
 5. O sistema exibe botões de atalho grandes e visuais para "Registrar Entrada" e "Registrar Saída", que levam diretamente às telas correspondentes.

3.22. Registrar Saída Rápida de Estoque por Código

- **ID do Caso de Uso:** UC-22
- **Nome do Caso de Uso:** Registrar Saída Rápida de Estoque por Código
- **Ator(es):** Usuário, Administrador
- **Descrição:** Permite o registro otimizado de saídas de estoque, espelhando a funcionalidade da entrada rápida.
- **Fluxo Principal:**
 1. O ator acede à tela "Saída Rápida".
 2. O ator digita um código de produto e clica em "Verificar".
 3. O front-end faz um pedido `GET` para `/api/produtos/codigo/<codigo>`.

4. Se o produto existir, o sistema exibe o nome e o saldo atual para referência, e habilita os campos "Quantidade a Retirar", "Motivo da Saída" (obrigatório) e o botão "Registrar Saída".
5. O ator preenche os campos e clica em "Registrar Saída".
6. O front-end faz um pedido `POST` para `/api/estoque/saida`.
7. O sistema exibe uma mensagem de sucesso e limpa a tela.

- **Fluxos Alternativos e de Exceção:**

- **FA-01: Estoque insuficiente:** No passo 5, se a quantidade a retirar for maior que o saldo atual, o sistema exibe um erro e não permite o registro.

3.23. Gerar Relatórios

- **ID do Caso de Uso:** UC-23

- **Nome do Caso de Uso:** Gerar Relatórios

- **Ator(es):** Usuário, Administrador

- **Descrição:** Permite a geração e exportação de relatórios consolidados do sistema.

- **Fluxo Principal (Relatório de Inventário):**

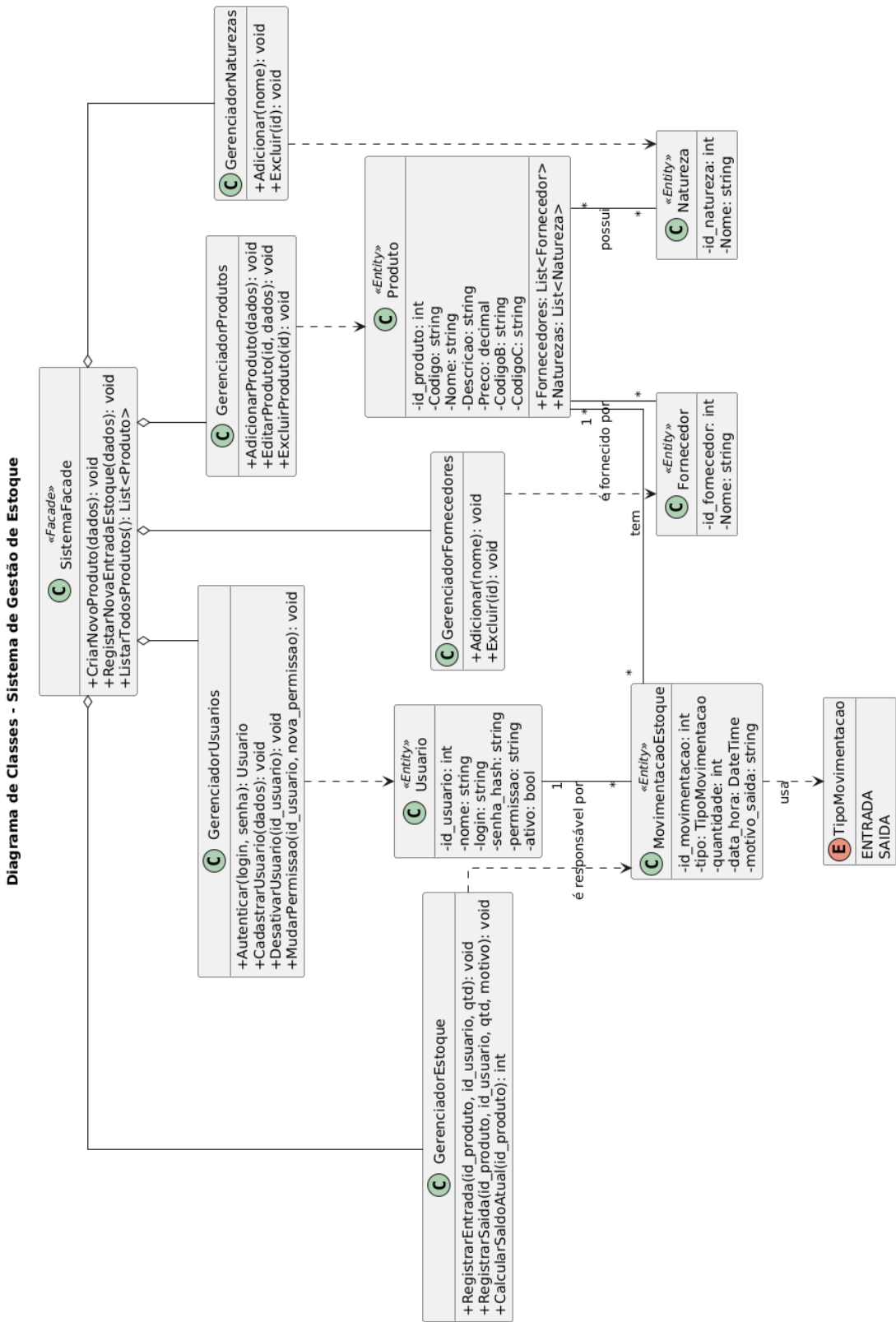
1. O ator acede à tela "Relatórios" e seleciona a aba "Inventário Atual".
2. O sistema exibe uma pré-visualização da tabela de inventário (Produto, Saldo, Preço Unit., Valor Total).
3. O ator clica em "Exportar para PDF" ou "Exportar para Excel".
4. O front-end faz um pedido à API (ex: `GET /api/relatorios/inventario?formato=pdf`).
5. A API gera o ficheiro e o front-end aciona o download.

- **Fluxos Alternativos e de Exceção:**

- **FA-01: Relatório de Histórico de Movimentações:**

1. O ator seleciona a aba "Histórico de Movimentações".
2. O sistema exibe filtros por período (data de início/fim) e por tipo (Todas, Entradas, Saídas).
3. O ator aplica os filtros e o sistema exibe uma pré-visualização da tabela de movimentações.
4. O fluxo de exportação segue o mesmo do Fluxo Principal.

4. Diagrama de Classes



5. Documento de Requisitos Não-Funcionais

1. Introdução

Este documento detalha os Requisitos Não Funcionais (RNFs) do Sistema de Gestão de Estoque. Enquanto os requisitos funcionais descrevem *o que* o sistema faz (ex: cadastrar um produto), os requisitos não funcionais descrevem *como* o sistema executa essas funções, definindo os seus atributos de qualidade, restrições e características operacionais.

2. Requisitos Não Funcionais Detalhados

2.1. Usabilidade

A usabilidade define a facilidade e a eficiência com que um utilizador pode interagir com o sistema.

- **RNF-US-01: Interface Gráfica Intuitiva:** O sistema deve apresentar uma interface gráfica de utilizador (GUI) clara, com navegação consistente através de um painel lateral e uma barra de menus superior.
- **RNF-US-02: Identificação Visual de Ações:** As ações primárias (Adicionar, Editar, Excluir) devem ser visualmente distintas, utilizando ícones e cores padronizadas (Verde para adição, Azul para edição, Vermelho para exclusão/perigo) para facilitar o reconhecimento rápido.
- **RNF-US-03: Navegação por Teclado:** O sistema deve permitir a execução das principais operações (login, verificação de códigos, submissão de formulários) através da tecla "Enter", minimizando a dependência do rato e agilizando o fluxo de trabalho.
- **RNF-US-04: Feedback em Tempo Real:** O sistema deve fornecer feedback imediato para ações críticas, como a verificação de existência de um código de produto durante o cadastro, informando o utilizador sobre o status ("Disponível" ou "Já existe") antes da submissão do formulário.
- **RNF-US-05: Estilo Visual Coeso:** Toda a aplicação deve seguir um tema visual ("Modern XP") consistente, definido num arquivo de estilo centralizado (`style.qss`), garantindo que todos os componentes, janelas e diálogos tenham uma aparência profissional e unificada.

2.2. Desempenho

O desempenho refere-se à capacidade de resposta e eficiência do sistema sob carga de trabalho.

- **RNF-DE-01: Arquitetura Cliente-Servidor:** A arquitetura desacoplada garante que o processamento pesado e as consultas à base de dados sejam realizados no back-end, mantendo a interface do utilizador (front-end) leve e responsiva.
- **RNF-DE-02: Tempo de Resposta da Interface:** As interações na interface, como a filtragem e ordenação de tabelas, devem ser executadas no lado do cliente para fornecer uma resposta instantânea, sem a necessidade de novas chamadas à rede.
- **RNF-DE-03: Carregamento Inicial:** O tempo de arranque da aplicação na sua versão empacotada (`--onefile`) pode ser de vários segundos devido à descompressão inicial. Uma alternativa (`--onedir`) está disponível para inicialização quase instantânea.

2.3. Segurança

A segurança define a capacidade do sistema de proteger os dados e controlar o acesso.

- **RNF-SE-01: Autenticação de Utilizador:** Todo o acesso às funcionalidades do sistema (exceto a tela de login) deve exigir autenticação prévia através de login e senha.
- **RNF-SE-02: Autorização por Nível de Permissão:** O sistema deve implementar um controlo de acesso baseado em funções (Role-Based Access Control). Funções críticas, como a gestão de utilizadores, devem ser restritas a utilizadores com o nível de "Administrador".
- **RNF-SE-03: Comunicação Segura por Token:** A comunicação entre o cliente (front-end) e o servidor (back-end) deve ser protegida. Após o login, cada pedido à API deve incluir um Token JWT (JSON Web Token) para validar a sessão e a identidade do utilizador.
- **RNF-SE-04: Armazenamento Seguro de Senhas:** As senhas dos utilizadores nunca devem ser armazenadas em texto plano. Elas devem ser guardadas na base de dados como um hash criptográfico (usando `werkzeug.security`).

- **RNF-SE-05: Preservação de Dados (Soft Delete):** A desativação de um utilizador não deve resultar na sua exclusão física da base de dados, garantindo que o histórico de movimentações associado a esse utilizador seja preservado para fins de auditoria.

2.4. Confiabilidade

A confiabilidade refere-se à robustez do sistema e à sua capacidade de lidar com erros.

- **RNF-CO-01: Tratamento de Erros:** O sistema não deve "crashar" ou exibir erros técnicos indecifráveis para o utilizador final. Todas as operações críticas (chamadas de API, manipulação de dados) devem ser envolvidas em blocos de tratamento de exceções (`try/except`).
- **RNF-CO-02: Mensagens de Erro Amigáveis:** Em caso de falha (ex: falha de conexão com a API, dados inválidos), o sistema deve exibir uma mensagem de erro clara e compreensível numa caixa de diálogo, orientando o utilizador.
- **RNF-CO-03: Validação de Dados:** A entrada de dados deve ser validada tanto no front-end (para feedback imediato) quanto no back-end (como garantia de segurança) para prevenir o registo de dados inconsistentes ou incompletos na base de dados.

2.5. Manutenibilidade

A manutenibilidade define a facilidade com que o sistema pode ser modificado, corrigido ou melhorado.

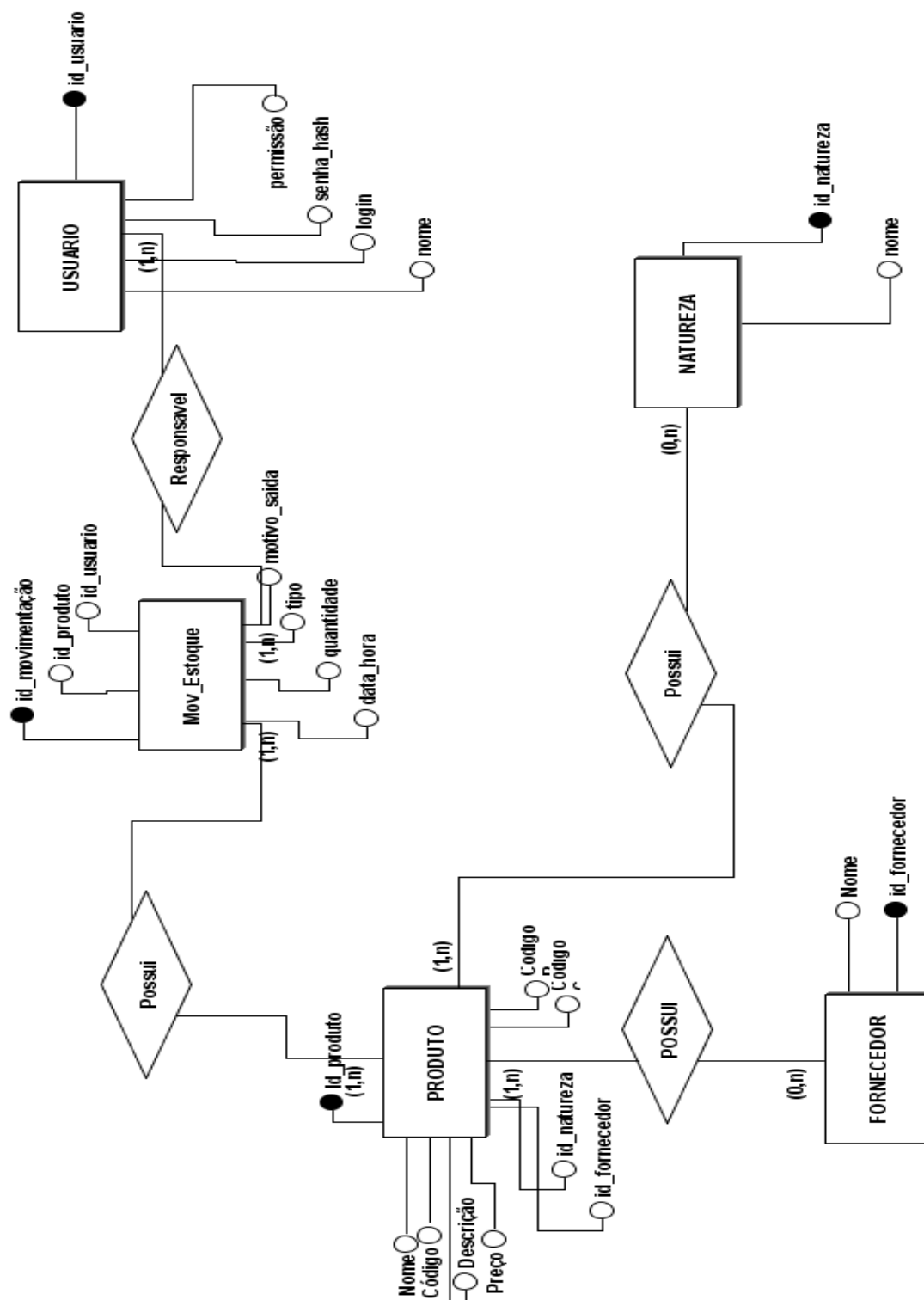
- **RNF-MA-01: Separação de Responsabilidades:** O código está claramente dividido entre o back-end (lógica de negócio e acesso a dados) e o front-end (lógica de apresentação), facilitando a manutenção e o desenvolvimento paralelo.
- **RNF-MA-02: Código Modular:** A interface gráfica é construída com componentes modulares e reutilizáveis (ex: `ProdutosWidget`, `KPICardWidget`, `QuickAddDialog`), permitindo que cada "tela" seja mantida de forma independente.
- **RNF-MA-03: Estilo Centralizado:** Todo o design visual da aplicação está concentrado num único arquivo (`style.qss`), permitindo que alterações na

aparência sejam feitas num único local, sem a necessidade de modificar o código Python.

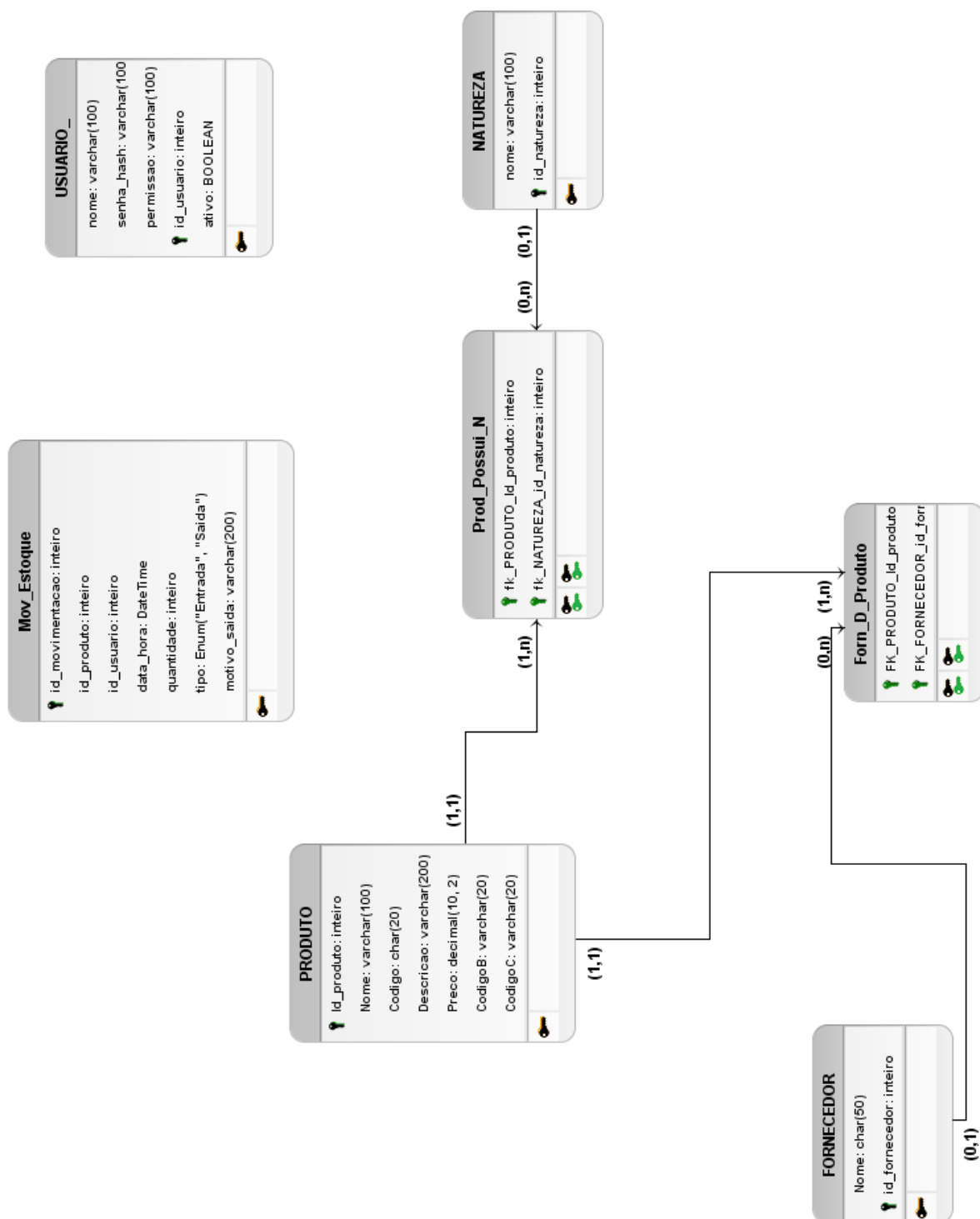
2.6. Portabilidade e Compatibilidade

- **RNF-PO-01: Sistema Operacional:** A aplicação cliente foi desenvolvida e testada para o sistema operacional Windows.
- **RNF-PO-02: Distribuição:** O sistema pode ser empacotado num único ficheiro executável (.exe) através do PyInstaller, eliminando a necessidade de instalar o Python ou quaisquer dependências na máquina do utilizador final.
- **RNF-PO-03: Base de Dados:** O back-end foi projetado para se conectar a um servidor de base de dados MySQL.
- **RNF-PO-04: Acesso em Rede:** O servidor pode ser configurado para aceitar conexões de outros computadores na mesma rede local, permitindo o uso do sistema por múltiplos clientes.

6. Diagrama de Entidade e Relacionamentos



7. Modelo Lógico do Banco de Dados



8. Modelo Físico do Banco de Dados(Sql Script)

```
CREATE TABLE PRODUTO (  
    Id_produto INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Codigo CHAR(20) UNIQUE NOT NULL,  
    Descricao VARCHAR(200),  
    Preco DECIMAL(10, 2) NOT NULL,  
    CodigoB VARCHAR(20),  
    CodigoC VARCHAR(20)  
);
```

```
CREATE TABLE USUARIO (  
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    login VARCHAR(100) UNIQUE NOT NULL,  
    senha_hash VARCHAR(255) NOT NULL,  
    permissao VARCHAR(100) NOT NULL,  
    ativo BOOLEAN NOT NULL DEFAULT TRUE  
);
```

```
CREATE TABLE FORNECEDOR (  
    id_fornecedor INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50) UNIQUE NOT NULL  
);
```

```
CREATE TABLE NATUREZA (  

```

```
id_natureza INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(100) UNIQUE NOT NULL  
);
```

```
CREATE TABLE Mov_Estoque (  
id_movimentacao INT AUTO_INCREMENT PRIMARY KEY,  
id_produto INT NOT NULL,  
id_usuario INT NOT NULL,  
data_hora DATETIME NOT NULL,  
quantidade INT NOT NULL,  
tipo ENUM("Entrada", "Saida") NOT NULL,  
motivo_saida VARCHAR(200),
```

```
FOREIGN KEY (id_produto) REFERENCES PRODUTO(Id_produto) ON DELETE  
RESTRICT,
```

```
FOREIGN KEY (id_usuario) REFERENCES USUARIO(id_usuario) ON DELETE  
RESTRICT
```

```
);
```

```
CREATE TABLE PRODUTO_FORNECEDOR (  
FK_PRODUTO_Id_produto INT NOT NULL,  
FK_FORNECEDOR_id_fornecedor INT NOT NULL,  
PRIMARY KEY (FK_PRODUTO_Id_produto, FK_FORNECEDOR_id_fornecedor)  
);
```

```
CREATE TABLE PRODUTO_NATUREZA (  
fk_PRODUTO_Id_produto INT NOT NULL,  
fk_NATUREZA_id_natureza INT NOT NULL,
```

PRIMARY KEY (fk_PRODUTO_Id_produto, fk_NATUREZA_id_natureza)
);

ALTER TABLE PRODUTO_FORNECEDOR ADD CONSTRAINT
FK_PRODUTO_FORNECEDOR_PRODUTO
FOREIGN KEY (FK_PRODUTO_Id_produto)
REFERENCES PRODUTO (Id_produto)
ON DELETE CASCADE;

ALTER TABLE PRODUTO_FORNECEDOR ADD CONSTRAINT
FK_PRODUTO_FORNECEDOR_FORNECEDOR
FOREIGN KEY (FK_FORNECEDOR_id_fornecedor)
REFERENCES FORNECEDOR (id_fornecedor)
ON DELETE RESTRICT;

ALTER TABLE PRODUTO_NATUREZA ADD CONSTRAINT
FK_PRODUTO_NATUREZA_PRODUTO
FOREIGN KEY (fk_PRODUTO_Id_produto)
REFERENCES PRODUTO (Id_produto)
ON DELETE CASCADE;

ALTER TABLE PRODUTO_NATUREZA ADD CONSTRAINT
FK_PRODUTO_NATUREZA_NATUREZA
FOREIGN KEY (fk_NATUREZA_id_natureza)
REFERENCES NATUREZA (id_natureza)
ON DELETE RESTRICT;

9. Glossário

Termo	Definição
Ator	Uma entidade (pessoa ou sistema externo) que interage com o sistema. No nosso caso: Usuário e Administrador.
Caso de Uso (UC)	Descreve uma sequência de ações que um sistema executa para produzir um resultado observável de valor para um ator. Descreve "o que" o sistema faz.
DER	Acrônimo para Diagrama Entidade-Relacionamento. Um modelo conceitual que representa as entidades de um sistema e como elas se relacionam.
Exclusão Lógica (Soft Delete)	Abordagem de exclusão onde o registo não é fisicamente apagado da base de dados. Em vez disso, um campo (no nosso caso, ativo na tabela USUARIO) é alterado para false, impedindo o acesso ao dado mas preservando o histórico e a integridade referencial.
Integridade Referencial	Uma propriedade dos bancos de dados que garante que um registo "filho" não pode existir sem o seu registo "pai". É mantida através de Chaves Estrangeiras (FKs).
Modelo Físico	A representação de como os dados serão fisicamente armazenados no banco de dados. Para nós, é o script SQL com os comandos CREATE TABLE.
Modelo Lógico	Uma evolução do DER que detalha as tabelas, colunas e os seus tipos de dados, chaves primárias e estrangeiras.
Movimentação Transaccional	Modelo de dados onde cada entrada ou saída de estoque não altera um saldo existente, mas cria um novo registo de transação (Mov_Estoque). O saldo atual é sempre calculado a partir da soma de todas as transações, garantindo um histórico auditável completo.
Natureza	Uma categoria de classificação para um PRODUTO (ex: "Matéria-prima", "Produto Acabado").
On Delete (Regra)	Regra de integridade numa Chave Estrangeira que define o comportamento do banco de dados ao apagar um registo "pai". Discutimos RESTRICT (impede a exclusão) e CASCADE (apaga os filhos junto).
Requisito Não-Funcional (RNF)	Descreve como o sistema deve executar uma função, definindo as suas qualidades. Exemplos: Desempenho, Segurança, Usabilidade.