**Desenvolvedor: Ignacio A. C. Neto**

**Data: 03/10/2025**

**Prova: 1 (Sistema gerenciador de comidas nordestinas “Casa do Norte”) Versão em Python**

**ENTREGA 01: Requisitos Funcionais**

|  |  |  |
| --- | --- | --- |
| **Sistema:** Gerenciador de ferramentas. | | |
| **ID** | **Requisito** | **Descrição** |
| RF01 | Cadastro de Comida | Permitir ao usuário cadastrar novas comidas típicas nordestinas, informando nome, descrição, categoria, origem, ingredientes, porção, calorias, quantidade em estoque e estoque mínimo. |
| RF02 | Consulta de Comidas | Permitir ao usuário visualizar a lista de comidas cadastradas em uma tabela, com busca dinâmica através de termo inserido pelo usuário. |
| RF03 | Edição de Comida | Permitir ao usuário editar os dados das comidas existentes, com validação dos campos obrigatórios e alerta em caso de preenchimento incorreto ou ausente. |
| RF04 | Exclusão de Comida | Permitir ao usuário excluir comidas cadastradas, com confirmação antes da exclusão |
| RF05 | Registro de Movimentação | Permitir o registro de movimentações de entrada e saída do estoque, informando comida, tipo de movimentação, data e responsável pela operação. |
| RF06 | Histórico de Movimentações | Registrar e disponibilizar o histórico completo de todas as movimentações de estoque, assegurando rastreabilidade por data e responsável. |
| RF07 | Gestão de Estoque Mínimo | Permitir a configuração de estoque mínimo por comida e emitir alertas automáticos quando o nível atingir ou ficar abaixo do mínimo. |
| RF08 | Autenticação de Usuário | Permitir o login dos usuários do sistema, validando credenciais e informando motivos em caso de falha, com opção de logout. |
| RF09 | Interface Principal | Exibir interface principal com nome do usuário logado, opções para logout, cadastro de comida e gestão de estoque. |
| RF10 | Navegação Intuitiva | Oferecer navegação clara entre interfaces principais do sistema: login, cadastro de comida, gestão de estoque. |
| RF11 | Validação de Dados | Validar os dados inseridos pelo usuário nas interfaces de cadastro, edição e movimentação, exibindo alertas de erro ou campos obrigatórios. |

**ENTREGA 2: Diagrama Entidade Relacionamento (DER)**

**Uma imagem contendo Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.**

**ENTREGA 3: Script de criação e população do banco de dados**

Arquivo: **db\_init.sql** (Encontra-se junto com a codificação)

**ENTREGA 4: Interface de autenticação de usuários (Login)**

**Arquivos:**

1º **db.py** (Conexão e inicialização do banco)

Obs: Rodar arquivo temporário para confirmar que conectou (**teste\_db.py**)

2º **utils.py** (Funções auxiliares)

3º **login.py** (Tela de login)

**ENTREGA 5: Interface principal do sistema (main.py)**

4º **main.py** (Tela principal - Inicializa o sistema)

**ENTREGA 6: Interface cadastro das comidas (comidas.py)**

5º **comidas.py** (Cadastro e exibição de comidas típicas)

**ENTREGA 7: Interface gestão de estoque (estoque.py)**

5º **estoque.py** (Gestão e movimentação de estoque)

**ENTREGA 8: Descritivo de Casos de Teste de Software**

**8.1. Casos de Teste**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID Caso de Teste** | **ID Requisito Funcional** | **Descrição** | **Precondição** | **Passos** | **Resultado esperado** |
| CT01 | Autenticação de usuário | Login com credenciais válidas | Usuário cadastrado e ativo | Entrar com usuário e senha no login | Login bem-sucedido, acesso ao sistema |
| CT02 | Autenticação de usuário | Login com credenciais inválidas | Usuário cadastrado e ativo | Entrar com usuário/senha inválidos | Mensagem de erro, permanência na tela de login |
| CT03 | Cadastro de nova comida | Cadastro de nova comida | Usuário logado | Preencher todos os campos obrigatórios e salvar | Comida registrada, aparece na listagem |
| CT04 | Edição de comida | Edição de comida existente | Usuário logado, comida cadastrada | Editar dados da comida e salvar | Dados atualizados na listagem |
| CT05 | Exclusão de comida | Exclusão de comida existente | Usuário logado, comida cadastrado | Excluir comida da listagem | Comida removida da tabela |
| CT06 | Consulta / listagem de comidas | Busca por comida | Comidas cadastradas | Buscar termo por nome ou descrição | Lista filtrada conforme termo de busca |
| CT07 | Movimentação de estoque (entrada e saída) | Registro de movimentação | Comida existente, usuário logado | Registrar entrada/saída com quantidade e data | Estoque atualizado, histórico registrado |
| CT08 | Alerta de estoque mínimo na gestão | Controle e alerta de estoque mínimo | Comida com estoque próximo do mínimo | Registrar saída que reduz estoque abaixo do mínimo | Alerta visual destacado na linha da tabela |
| CT09 | Histórico de movimentações | Rastreabilidade de processos | Movimentações registradas | Acessar histórico de movimentações na gestão | Histórico completo, dados de usuário e data |
| CT10 | Logout | Logout de usuário | Usuário logado | Clicar no botão "Sair" | Sessão encerrada, retorno à tela de login |

**8.2. Ferramentas e Ambientes de Teste**

* **Ferramentas de Teste:**
  + Testes manuais navegando pela interface Tkinter (funcional).
  + **SQLite Browser** (ex.: DB Browser for SQLite) ou linha de comando sqlite3 para inspeção de dados e verificação.
* Ferramentas do editor (ex.: VS Code) para inspeção de código.
* **Ambiente de Teste**
  + Execução local com **Python 3.12+** (não requer servidor web).
  + Banco de Dados/Versão: SQLite 5.7 ou SQLite 10.4

**ENTREGA 9: Lista de requisitos de infraestrutura**

**SGBD Utilizado:**  
SQLite  
**Versão:**  
SQLite 5.7 ou superior / SQLite 10.4 ou superior

**Sistema Operacional:**  
Windows, Linux ou MacOS  
**Versão:**  
Windows 10 ou superior / Ubuntu 20.04 ou superior / MacOS Monterey ou superior

**Linguagem de Programação:**  
PYTHON 3.12+  
**Versão:**  
PYTHON 3.12+ 7.4 ou superior