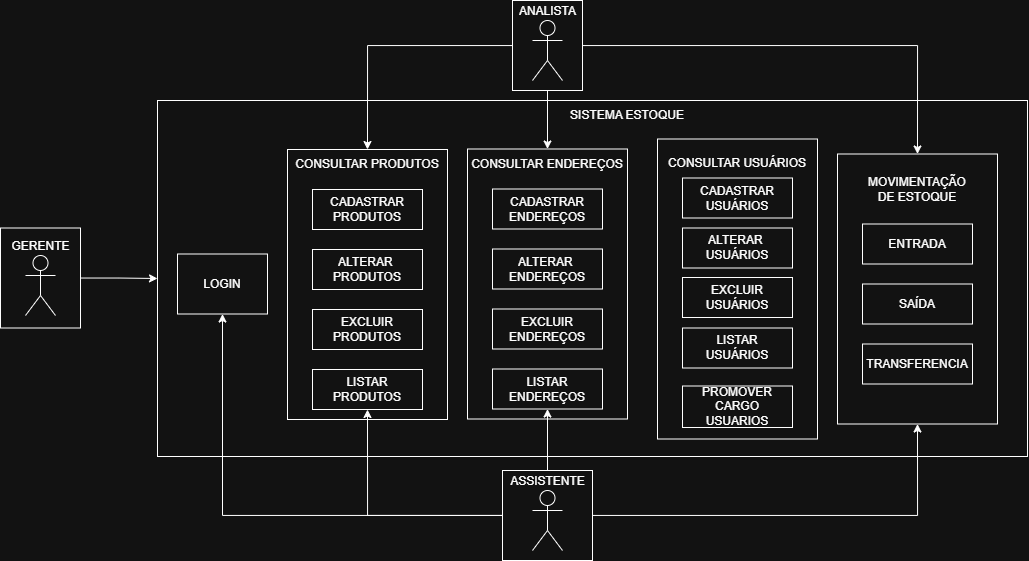
**SISTEMA DE ESTOQUE – LOG WMS**

# **DEFINIÇÃO**

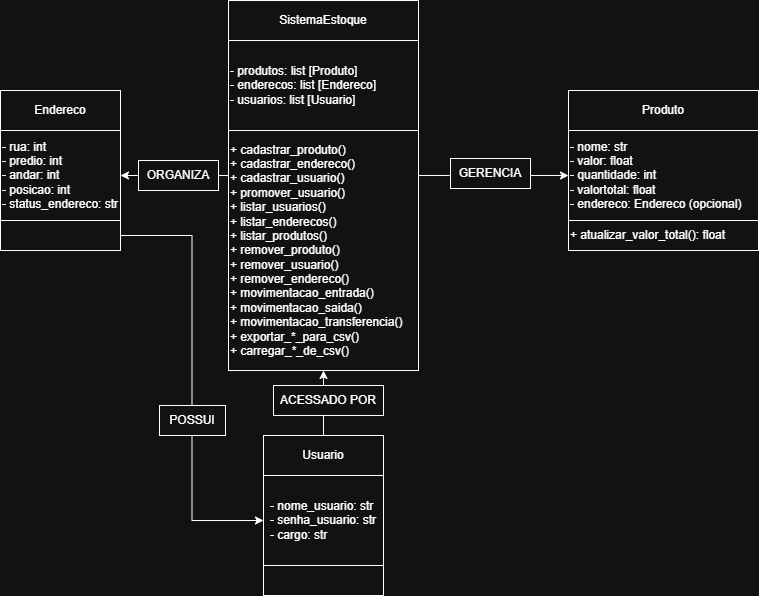
* **Objetivo documentação:** Desenvolver um sistema de controle de estoque para pequenas empresas utilizando controle por endereços e armazenamento de produtos.
* **Introdução Geral sobre o Sistema LOG WMS**: Sistema LOG WMS tem como proposito gerir e controlar estoque com endereçamento WMS, para otimizar e facilitar o processo de armazenagem utilizando ferramentas disponibilizadas no sistema.
* **Público Alvo:** Tem como alvo empresas de pequeno porte que precisam ter um sistema para controlar o estoque e que estão iniciando no mercado de logística.
* **Principais Funcionalidades:** Cadastro de produtos, endereços e usuários, exibição de relatórios de lista de produtos com valores, estoque mínimo e estoque máximo, funções de movimentação de estoque como entrada, saída e transferência de endereço

## **CASOS DE USO**

**Diagrama de casos** de uso mostra as funcionalidades do sistema e seus atores, **Gerente** possui acesso a todas funcionalidades, **Analista** tem acesso a várias funcionalidades do sistema de menos relacionada a usuários, **Assistente** tem apenas acesso a listar produtos/endereços e movimentação de estoque.

## **DIAGRAMA DE CLASSES**

Diagrama de classes mostra todas as classes do sistema e seus relacionamentos, **Endereço** tem como atributos rua, prédio, andar, posição e status de endereço, **Produto** terá os atributos nome, valor, quantidade, valortotal e endereço, como método terá apenas atualizar o valor total, **Usuário** terá as informações como nome, senha e cargo (Gerente, Analista e Assistente) e a classe **SistemaEstoque** terá todas as regras de negócio, como validação de status de endereço, cadastro e alteração de produtos, endereços e usuários e a movimentação de estoque



**TIPO DE ARQUITETURA**

### **Modular Monolítica**

O sistema foi desenvolvido com uma **arquitetura modular monolítica**, onde todas as funcionalidades estão integradas em um único aplicativo, porém organizadas em **módulos separados** de acordo com suas responsabilidades (produto, endereço, usuário e controle de estoque).  
Essa abordagem facilita o desenvolvimento, manutenção e entendimento do código, sem a complexidade de comunicação entre serviços distribuídos.

## **Descrição Geral da Arquitetura**

O sistema é composto pelos seguintes módulos principais:

### 1. **Módulo de Modelo (Entidades)**

Local: **modelo/**  
Responsável por definir as **classes que representam as entidades principais do sistema**.

* **Classe Produto**
  + Atributos: nome, valor, quantidade, valortotal, endereco.
  + Métodos: atualizar\_valor\_total().
  + Responsável por armazenar e gerenciar as informações sobre os produtos cadastrados.
* **Classe Endereco**
  + Atributos: rua, predio, andar, posicao, status\_endereco.
  + Representa os endereços físicos (posições no estoque) onde os produtos podem ser armazenados.
* **Classe Usuario**
  + Atributos: nome\_usuario, senha\_usuario, cargo.
  + Responsável pelo gerenciamento de credenciais e permissões dos usuários do sistema (Gerente único que possui acesso).

### 2. **Módulo de Negócio (Sistema de Estoque)**

Local: **sistema\_estoque.py**  
Contém a **classe principal SistemaEstoque**, que implementa toda a lógica de negócio da aplicação.

Principais responsabilidades:

* Gerenciar cadastros de produtos, endereços e usuários.
* Controlar operações de **entrada**, **saída** e **transferência** de produtos.
* Manter persistência de dados através de arquivos CSV.
* Realizar exportação e importação de informações.
* Executar validações e atualizações de status (por exemplo, endereço livre “L” ou ocupado “O”).

### **3**. **Módulo de Persistência (CSV)**

A persistência dos dados é feita de forma simples, utilizando **arquivos CSV** como base de dados local.  
Cada tipo de dado (produto, endereço e usuário) possui seu próprio arquivo:

* estoque.csv → Produtos
* enderecos.csv → Endereços
* usuarios.csv → Usuários

As funções exportar\_\*\_para\_csv() e carregar\_\*\_de\_csv() são responsáveis por gravar e recuperar as informações desses arquivos.

### **4. Módulo de Interface (Interação com o Usuário)**

Local: **main.py** (ou arquivo principal que contém o menu())

Responsável pela **interface de linha de comando**.  
Implementa o **menu interativo**, que varia conforme o cargo do usuário logado (gerente, analista ou assistente).

Funcionalidades:

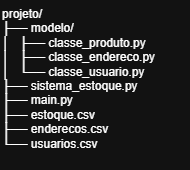
* Sistema de **login e autenticação** de usuários.
* Menus dinâmicos de acordo com o nível de permissão.
* Execução dos métodos do SistemaEstoque.

**Fluxo Geral de Funcionamento**

1. **Inicialização:**
   * O sistema carrega os dados dos arquivos CSV de todos os cadastros.
   * Caso não existam usuários, cria um administrador inicial.
2. **Login:**
   * O usuário se autentica pelo terminal.
   * O sistema identifica o cargo e define as permissões.
3. **Operações:**
   * Usuário acessa menus de **produtos**, **endereços**, **usuários** ou **movimentações**.
   * Todas as ações são executadas por métodos da classe SistemaEstoque.
4. **Persistência:**
   * Após cada alteração, os dados são automaticamente exportados para CSV, garantindo a integridade das informações.

**Tecnologias e Estrutura**

* **Linguagem:** Python 3
* **Persistência:** Arquivos CSV
* **Interface:** Terminal (CLI)
* **Paradigma:** Orientação a Objetos (OO)
* **Organização:**



**INSTRUÇÕES DE EXECUÇÃO DO SISTEMA**

### **Pré-requisitos**

* Python 3.10+ instalado no computador.
* Nenhuma dependência externa necessária (somente bibliotecas padrão).

### **Passos para execução**

1. Baixe o projeto no GitHub.
2. Abra o terminal na pasta do projeto.
3. Execute o comando:
4. python main.py
5. Caso seja a primeira execução, o sistema solicitará a criação de um **usuário administrador (gerente)**.
6. Faça login e acesse o menu de opções para interagir com o sistema.

**Referências Técnicas**

* Documentação oficial do Python: https://docs.python.org/3/
* Conceitos de WMS (Warehouse Management System):  
  https://pt.wikipedia.org/wiki/Sistema\_de\_gerenciamento\_de\_armazenagem