

RELATÓRIO FINAL – SISTEMA DE GERENCIAMENTO DE ESTOQUE PARA LIVRARIA

ALGORITMOS E PROGRAMAÇÃO

Prof: **ARLEY CARLOS CARNEIRO DOS SANTOS COSTA**

Alunos:

GABRIEL ORDONEZ DOS REIS

RAFAEL OLIVEIRA DA SILVA

TAINO SAMUEL LIMA RIBEIRO

VICTOR EDUARD RODRIGUES CABRAL

Resumo Executivo

Este relatório detalha o desenvolvimento de um sistema de software projetado para automatizar o gerenciamento de estoque de uma livraria. A solução, implementada em Python, aborda a necessidade crítica de registrar entradas de livros, controlar o estoque por gênero, monitorar níveis mínimos para evitar rupturas e gerar relatórios analíticos. O documento abrange desde a definição dos requisitos de negócio, passando pelo fluxograma e pseudocódigo, até a comprovação de testes funcionais e sugestões para futuras evoluções do sistema. O projeto cumpre todos os objetivos iniciais, entregando uma ferramenta funcional e eficiente para otimizar a gestão de estoque.

1. REQUISITOS DE NEGÓCIO

Contexto do Problema

A livraria enfrenta desafios significativos no controle e gerenciamento de seu estoque de livros. Atualmente, a empresa não possui um sistema automatizado para:

- **Registrar entradas de novos livros** de forma organizada e sistemática.
- **Controlar quantidades por gênero literário** para otimizar o mix de produtos.
- **Monitorar níveis mínimos de estoque** para evitar rupturas e perdas de vendas.
- **Analisar a distribuição percentual** dos gêneros no estoque total.
- **Gerar relatórios gerenciais** para tomada de decisões estratégicas.

Necessidades Identificadas

O software desenvolvido visa solucionar as seguintes necessidades críticas:

1. **Automatização do Registro:** Permitir o cadastro rápido e padronizado de entradas de livros.
2. **Controle de Estoque Mínimo:** Estabelecer parâmetros mínimos por gênero para evitar desabastecimento.
3. **Análise Estatística:** Calcular automaticamente porcentagens de distribuição por gênero.
4. **Relatórios Gerenciais:** Gerar informações consolidadas para apoio à gestão.
5. **Validação de Dados:** Garantir a integridade das informações inseridas no sistema.

Benefícios Esperados

- Redução de erros manuais no controle de estoque.
- Melhoria na tomada de decisões baseada em dados.
- Otimização do mix de produtos por gênero.
- Prevenção de rupturas de estoque.
- Aumento da eficiência operacional.

2. FLUXOGRAMA DA SOLUÇÃO

O fluxograma visual da solução, que representa a lógica completa do algoritmo, está disponível no arquivo fluxograma.svg. O diagrama inclui:

- Processo de inicialização do sistema.
- Loop de entrada de dados para os livros.
- Validações de título, gênero e quantidade.
- Cálculos estatísticos de estoque total e porcentagens.
- Verificação de estoque mínimo por gênero.
- Geração do relatório final.

Principais Fluxos do Sistema

1. **Fluxo Principal:** Entrada → Validação → Armazenamento → Cálculos → Relatório.
2. **Fluxos de Validação:** Verificações de dados com retorno para correção.
3. **Fluxo de Loop:** Repetição controlada para as entradas obrigatórias.
4. **Fluxo de Saída:** Geração e exibição do relatório consolidado.

3. PSEUDOCÓDIGO

ALGORITMO GerenciamentoEstoqueLivraria

INÍCIO

// Definir constantes de estoque mínimo por gênero

CONSTANTE ESTOQUE_MINIMO_FICCAO = 10

CONSTANTE ESTOQUE_MINIMO_NAO_FICCAO = 8

CONSTANTE ESTOQUE_MINIMO_ROMANCE = 12

CONSTANTE ESTOQUE_MINIMO_MISTERIO = 6

CONSTANTE ESTOQUE_MINIMO_FANTASIA = 15

CONSTANTE ESTOQUE_MINIMO_BIOGRAFIA = 5

CONSTANTE ESTOQUE_MINIMO_HISTORIA = 7

CONSTANTE ESTOQUE_MINIMO_CIENCIA = 9

// Inicializar variáveis

DECLARAR lista_livros COMO LISTA VAZIA

DECLARAR contador_entrada COMO INTEIRO = 1

// Exibir instruções do sistema

ESCREVER "Sistema de Gerenciamento de Estoque - Livraria"

ESCREVER "Você deverá registrar exatamente 2 entradas de livros"

// Loop para registrar duas entradas

ENQUANTO contador_entrada <= 2 FAÇA

 ESCREVER "=== ENTRADA", contador_entrada, "==="

 // Solicitar e validar título

 REPETIR

 ESCREVER "Digite o título do livro:"

 LER titulo

 SE titulo VAZIO ENTÃO

 ESCREVER "Erro: O título não pode estar vazio"

 FIM SE

 ATÉ titulo NÃO VAZIO

 // Solicitar e validar gênero

 ESCREVER "Gêneros disponíveis:"

 ESCREVER "1. Ficção, 2. Não-ficção, 3. Romance, 4. Mistério"

 ESCREVER "5. Fantasia, 6. Biografia, 7. História, 8. Ciência"

 REPETIR

 ESCREVER "Escolha o gênero (1-8):"

```

LER opcao_genero
SE opcao_genero NÃO ESTÁ EM [1,2,3,4,5,6,7,8] ENTÃO
    ESCREVER "Erro: Opção inválida"
FIM SE
ATÉ opcao_genero VÁLIDA

// Converter opção para nome do gênero
genero = CONVERTER_OPCAO_PARA_GENERO(opcao_genero)

// Solicitar e validar quantidade
REPETIR
    ESCREVER "Digite a quantidade de exemplares:"
    LER quantidade
    SE quantidade <= 0 OU NÃO É NÚMERO ENTÃO
        ESCREVER "Erro: Digite um número positivo válido"
    FIM SE
ATÉ quantidade VÁLIDA

// Armazenar dados do livro
livro = CRIAR_REGISTRO(titulo, genero, quantidade)
ADICIONAR livro À lista_livros

ESCREVER "Livro registrado com sucesso!"
contador_entrada = contador_entrada + 1
FIM ENQUANTO

// Calcular estoque total
estoque_total = 0
PARA CADA livro EM lista_livros FAÇA
    estoque_total = estoque_total + livro.quantidade
FIM PARA

// Agrupar por gênero e calcular estatísticas
DECLARAR generos COMO DICIONÁRIO VAZIO
PARA CADA livro EM lista_livros FAÇA
    SE livro.genero EXISTE EM generos ENTÃO
        generos[livro.genero] = generos[livro.genero] + livro.quantidade
    SENÃO
        generos[livro.genero] = livro.quantidade
    FIM SE
FIM PARA

// Calcular porcentagens e verificar estoque mínimo

```

```

DECLARAR estatisticas COMO DICIONÁRIO VAZIO
PARA CADA genero, quantidade EM generos FAÇA
    porcentagem = (quantidade / estoque_total) * 100
    estoque_minimo = OBTER_ESTOQUE_MINIMO(genero)
    acima_minimo = quantidade >= estoque_minimo

    estatisticas[genero] = CRIAR_ESTATISTICA(
        quantidade, porcentagem, estoque_minimo, acima_minimo
    )
FIM PARA

// Exibir relatório final
ESCREVER "=== RELATÓRIO DE ESTOQUE - LIVRARIA ==="
ESCREVER "LIVROS REGISTRADOS:"
PARA CADA livro EM lista_livros FAÇA
    ESCRIVER livro.titulo, "(", livro.genero, ") -", livro.quantidade, "exemplares"
FIM PARA

ESCREVER "ESTOQUE TOTAL:", estoque_total, "exemplares"
ESCREVER "DISTRIBUIÇÃO POR GÊNERO:"
PARA CADA genero, stats EM estatisticas FAÇA
    ESCRIVER genero, ":"
    ESCRIVER " - Quantidade:", stats.quantidade, "exemplares"
    ESCRIVER " - Porcentagem:", stats.porcentagem, "% do estoque total"
    ESCRIVER " - Estoque mínimo:", stats.estoque_minimo, "exemplares"
    SE stats.acima_minimo ENTÃO
        ESCRIVER " - Status: Adequado"
    SENÃO
        ESCRIVER " - Status: Baixo"
    FIM SE
FIM PARA

ESCREVER "PROCESSAMENTO CONCLUÍDO"
FIM

```

4. ALGORITMO DESENVOLVIDO

O código-fonte completo da solução está disponível no arquivo main.py. O algoritmo foi desenvolvido em Python seguindo as melhores práticas.

Principais Características do Código

- **Validação Robusta:** Verificações completas de entrada de dados.
- **Constantes Bem Definidas:** Estoques mínimos configuráveis.
- **Tratamento de Erros:** Loops de validação para entradas inválidas.
- **Interface Amigável:** Mensagens claras e formatação organizada.

5. COMPROVAÇÃO DE TESTES FUNCIONAIS

Cenário de Teste 1: Entrada Padrão

- **Entrada:**
 - Livro 1: "O Senhor dos Anéis" (Fantasia), 25 exemplares.
 - Livro 2: "1984" (Ficção), 15 exemplares.
- **Saída Esperada:** Relatório com estoque total de 40 exemplares, indicando status "Adequado" para ambos os gêneros.

Cenário de Teste 2: Estoque Baixo

- **Entrada:**
 - Livro 1: "Steve Jobs" (Biografia), 3 exemplares.
 - Livro 2: "Sherlock Holmes" (Mistério), 7 exemplares.
- **Saída Esperada:** Relatório com estoque total de 10 exemplares, indicando status "Baixo" para Biografia e "Adequado" para Mistério.

Validação dos Requisitos

- **Registro de Entradas:** Sistema registra 2 entradas corretamente.
- **Estoque Mínimo:** Constantes aplicadas para os 8 gêneros.
- **Cálculo do Estoque Total:** Soma automática das quantidades.
- **Exibição de Dados:** Porcentagens calculadas e exibidas corretamente.
- **Validação de Entrada:** Sistema rejeita dados inválidos.

6. SUGESTÕES DE MELHORIAS FUTURAS

1. **Expansão do Número de Entradas:** Permitir registro de múltiplos livros com opções de adicionar, editar e remover.
2. **Persistência de Dados:** Salvar e carregar dados de um arquivo (CSV) ou banco de dados.
3. **Relatórios Avançados:** Gerar gráficos visuais e permitir a exportação para PDF/CSV.
4. **Interface Web:** Criar uma aplicação web com dashboards interativos.
5. **Integração com APIs:** Conectar com APIs de livrarias para busca de informações e sincronização com sistemas de vendas.
6. **Backup e Recuperação:** Criar um sistema automático de backup e recuperação de dados.

CONCLUSÃO

O sistema desenvolvido atende completamente aos requisitos funcionais especificados, fornecendo uma solução robusta e eficiente para o gerenciamento básico de estoque. A implementação seguiu boas práticas. As sugestões de melhorias oferecem um caminho claro para a evolução do sistema.

Arquivos do Projeto

- main.py - Código-fonte principal
- main_explicado.py - Código-fonte com todas as linhas comentadas
- fluxograma.svg - Fluxograma visual da solução
- relatorio_completo.md - O relatório original