Pontifícia Universidade Católica do Paraná

Documentação

Sistema de Gerenciamento de Inventário em Haskell

Emily Pontes Fontana

Curitiba 2025

—

Visão do Projeto

Este projeto é um sistema completo de gerenciamento de inventário desenvolvido em Haskell, implementando operações CRUD (Create, Read, Update, Delete) com auditoria detalhada de todas as transações. Além disso, esse projeto segue conforme as regularidades apresentadas no arquivo inventário.ra2 da disciplina de Programação Lógica e Funcional - 4U.

Estruturas de Dados Principais - conforme especificação da atividade

O sistema é construído sobre estruturas de dados bem definidas que modelam o domínio do problema. O item representa cada elemento do inventário, contendo identificadores. O inventário como um todo é implementado como um mapa que associa IDs aos respectivos itens, permitindo acesso eficiente. Para o sistema de auditoria, foram definidos tipos específicos que categorizam as ações (crud) e os status das operações (Sucesso ou Falha). Cada entrada de log registra temporalmente a operação realizada, com detalhes específicos e o resultado obtido, proporcionando rastreabilidade completa de todas as transações efetuadas no sistema.

- Item:

itemID: String (Identificador único)
nome: String (Nome do produto)
quantidade: Int (Quantidade em estoque)
categoria: String (Categoria do produto)

- Tipo de inventário (Mapa de ID para Item)

```
type Inventario = Map String Item
```

- Ações suportadas pelo sistema

```
data AcaoLog = Add | Remove | Update | QueryFail
  deriving (Show, Read, Eq)
```

- Status das operações

```
data StatusLog = Sucesso | Falha String
  deriving (Show, Read, Eq)
```

- LogEntry:

timestamp: UTCTime (Data/hora da operação)
acao: AcaoLog (Tipo de ação - Add/Remove/Update)
detalhes: String (Detalhes específicos)
status: StatusLog (Sucesso ou Falha)

Descrição dos Tipos de Ação Suportados(CRUD):

- Add - Adicionar novo item
- Remove - Remover quantidade do estoque
- Update - Atualizar quantidade existente

Registro pedido conforme instruções da atividade após execução e teste do projeto::

Inventário.dat

ID | Nome | Quantidade | Categoria
1 | CAMA | 2 | MOBILIA
2 | tenis | 5 | calçado
3 | lapis | 21 | material
4 | GARRAFA | 12 | OBJETO
5 | suco | 100 | bebida
6 | hamburguer | 59 | comida
8 | ovo | 12 | comida
9 | boneca | 2 | brinquedo
11 | HAMBURGUER | 97 | COMIDA
15 | refrigerante | 8 | bebida
20 | vela | 3 | velas
90 | remedio | 33 | medicamento

Total: 12 itens no inventário

Funcionalidades Principais

As funcionalidades principais incluem a gestão completa de itens, com operações de adição (que valida ID único, quantidade positiva e campos não vazios), remoção (que verifica existência e estoque suficiente) e atualização (que permite modificar quantidades existentes). O sistema de auditoria registra com precisão temporal todas as operações, persistindo tanto sucessos quanto falhas em arquivo dedicado.

Gestão de Itens

Adição: Válida ID único, quantidade positiva, campos não vazios
Remoção: Verifica existência e estoque suficiente
Atualização: Permite modificar quantidades

Sistema de Auditoria

Registro timestamp de todas as operações
Log de sucessos e falhas
Persistência em arquivo Auditoria.log

Relatórios

Estatísticas de operações (sucessos/falhas)
Item mais movimentado
Detalhamento de erros

Estrutura de Arquivos

- main.hs: Código fonte principal
- Inventario.dat: Base de dados do inventário
- Auditoria.log: Log de transações e auditoria

Validações Implementadas

Além do ID único para novos itens, outras validações implementadas constam a não permissão de quantidades negativas, a verificação de estoque antes de remoções e a obrigatoriedade de preenchimento de campos.

Tratamento de Erros

No código como forma de trativa de erros implementamos mensagens descritivas para usuário, e uma forma de manter controle sobre atos falhos é a utilização de logging de todas as falha.

Registro e Estudo de Operações Registradas no Auditoria.log:

- Esses dados nos permitem manter controle e rastreabilidade sobre o uso do sistema:

    - Total de operações: 31
    - Sucessos: 20 operações
    - Falhas: 11 operações

___

Causas de Falha Relatadas no Projeto:

- Foram registradas as seguintes falhas durante a execução do projeto:

    - Item já existe no inventário (5 ocorrências)
    - Item não encontrado (4 ocorrências)
    - Estoque insuficiente (2 ocorrências)

—

Como é o fluxo do progama:

- Menu interativo de opções é apresentado para o usuário realizar alguma das escolhas mostradas a seguir:

Menu de Opções:
1 - Adicionar item
2 - Remover item
3 - Atualizar item
4 - Listar inventario
5 - Relatorio
0 - Sair

- Descritivo do Fluxo:

    1. Sistema carrega estado anterior automaticamente
    2. Operações são validadas em tempo real
    3. Alterações são persistidas automaticamente
    4. Log detalhado é gerado para auditoria

Conclusão

Este projeto exemplifica de maneira prática vários conceitos fundamentais da programação funcional. A recursão é empregada no loop principal do sistema, onde a função chama a si mesma continuamente, passando o estado atualizado do inventário a cada iteração. O backtracking, que foi tratado em outros projetos como o prolog, não é utilizado no sistema, uma vez que as operações são baseadas em estado persistente e não há necessidade de mecanismos de tentativa e erro ou desfazer ações. Todas as operações, uma vez validadas e executadas, são definitivas e registradas no sistema de auditoria.

O código além disso utiliza funções de alta ordem como map, filter e fold para processamento de coleções, e o uso de monads para controle de efeitos colaterais, como IO para operações de entrada/saída e Either para tratamento funcional de erros.