Centro Universitário de Excelência
Sistemas de Informação

Segunda Etapa do Projeto Lu Delivery: Refatoração de Código

Autor: Anna Beatriz Lima
Ian Salomão
Valéria Soares

Feira de Santana, 2025

Agenda

Mostrar a evolução do sistema de pedidos em Kotlin, agora modularizado, com funções, validações e centralização de dados.

1. Objetivos da refatoração:
Explicar os motivos da refatoração

2. Estrutura do projeto:
Mostrar como o sistema está organizado: classes de dados, controle centralizado e módulos separados.

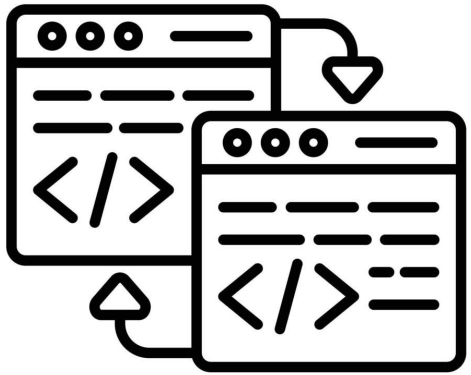
3. Principais funções implementadas:
Apresentar as funções-chave do sistema: cadastro, atualização e consulta de itens e pedidos, e funções auxiliares

4. Fluxo de Cadastro de Itens:
Demonstrar como a estrutura funcional foi implementada

5. Considerações finais:
Destacar os resultados da refatoração

Objetivos da Refatoração

- Separação de Responsabilidades
- Estruturação de Dados

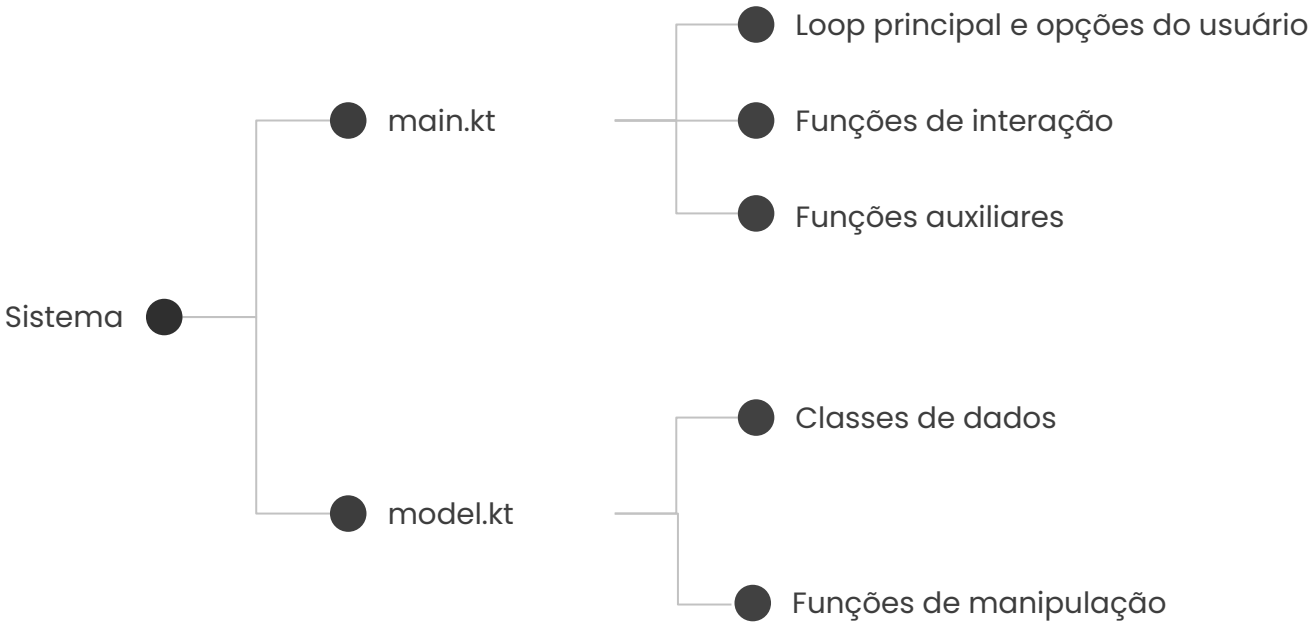


Objetivos da Refatoração

Código Anterior	x	Código Atual
Estrutura Linear		Funções
Código Direto		Validação

Segunda Etapa do Projeto Lu Delivery:
Refatoração de Código

Estrutura do Projeto



Segunda Etapa do Projeto Lu Delivery:
Refatoração de Código

Principais Funções Implementadas

- Para Interação com o usuário
  - **menuCadastrarItem() ->** Permite adicionar novos itens ao cardápio.
  - **menuAtualizarItem() ->** Atualiza nome, descrição, preço ou estoque de um item.
  - **menuCadastrarPedido() ->** Cria um novo pedido com itens e subtotal.
  - **menuAtualizarPedido() ->** Altera o status de um pedido existente.
  - **menuBuscarPedidoPorStatus() ->** Consulta pedidos filtrados por status.

Principais Funções Implementadas

- Para Manipulação (Lógica de negócio)
  - **cadastrarItem() ->** Cria e valida novos itens no menu.
  - **buscarItem() ->** Localiza item pelo código.
  - **atualizarItem() ->** Modifica informações de um item existente.
  - **cadastrarPedido() ->** Registra o pedido
  - **atualizarStatusPedido() ->** Altera o status do pedido (usando enum).

Principais Funções Implementadas

- Funções para Exibição. Ex.: **exibirResumoPedido()**
- Funções Auxiliares. Ex.: **perguntarSeContinua()**

Fluxo de Cadastrar Itens

```
main.kt
└── menuCadastrarItem()
├── exibirCabecalho("CADASTRAR ITEM")
├── lerTexto() → nome, descrição
├── lerDouble() → preço
├── lerInteiro() → estoque
├── cadastrarItem(nome, desc, preco, estoque)   // chama função do model.kt
│      ↓
│      model.kt
│      └── cadastrarItem()
│            ├── valida dados
│            ├── cria novo ItemMenu
│            ├── adiciona em SystemControl.itensMenu
│            └── retorna ItemMenu cadastrado
│
├── println("Item cadastrado com sucesso!")
└── perguntarSeContinua() → repete ou volta ao menu
```

Segunda Etapa do Projeto Lu Delivery:
Refatoração de Código

Considerações finais

- A refatoração torna o sistema mais organizado
- Facilita o crescimento do sistema

Referências

- Fonte: Vecteezy, 2025. Disponível em: https://static.vecteezy.com/ti/vetor-gratis/p1/7571308-code-refactoring-line-icon-vetor.jpg. Acesso em: 30 set. 2025.
- SEALSTORE. Serviço de manutenção. Disponível em: https://www.sealstore.com.br/servico-de-manutencao. Acesso em: 30 set. 2025.
- BOTELHO, Adriana. Considerações finais. Portefólio das Aprendizagens. 19 jun. 2015. Disponível em: https://portefoliodasaprendizagens.wordpress.com/wp-content/uploads/2015/06/11355637_1056287487716137_173100881_n.jpg. Acesso em: 30 set. 2025.