

# **Praticando Kotlin: Sistema de Gerenciamento de Pedidos de Restaurante**

**Autores:** Anna Beatriz Lima  
Ian Carneiro  
Valéria Soares

# Agenda

Apresentar como desenvolvemos um sistema de pedidos para restaurantes em Kotlin, operando exclusivamente no console, com uma estrutura linear, sem a utilização de funções externas e com dados apenas em memória.

## Funcionalidades do sistema:

- **1. Cadastrar Item:**
  - Permite adicionar novos produtos ao menu do restaurante.
- **2. Atualizar Item:**
  - Possibilita modificar informações de itens já cadastrados.
- **3. Criar Pedido:**
  - Inicia o processo de montagem de um novo pedido.
- **4. Atualizar Pedido:**
  - Permite alterar o status de um pedido existente.
- **5. Consultar Pedidos:**
  - Exibe informações dos pedidos, com a opção de filtrar por status.

# Laço principal



```
variável opçãoMenu = -1
```

```
enquanto(opçãoMenu != 0){  
    imprimirTelaDeMenu()  
    opçãoMenu = lerEntrada()  
  
    quando(opçãoMenu){  
        1 -> cadastrarItem()  
        2 -> atualizarItem()  
        3 -> criaPedido()  
        4 -> AtualizaPedido()  
        5 -> consultaPedidos()  
        0 -> opçãoMenu = 0  
    }  
}
```

# Cadastro de itens



```
1 -> {  
    imprimir("=== CADASTRAR ITEM ===")  
  
    nome = lerEntrada("Nome do item:")  
    descrição = lerEntrada("Descrição:")  
    preço = lerEntrada("Preço:")  
    estoque = lerEntrada("Estoque:")  
  
    novoItem = criarItem(código, nome, descrição, preço, estoque)  
    adicionarNaLista(listaItensMenu, novoItem)  
    incrementar(proximoCódigo)  
}
```

# Atualização de itens



```
2 -> {  
    imprimir("=== ATUALIZAR ITEM ===")  
    mostrarTodosItens(listaItensMenu)  
  
    códigoEscolhido = lerEntrada("Código do item:")  
  
    para cada item em listaItensMenu {  
        se(item.código == códigoEscolhido) {  
            imprimirMenuDeAtualizacao()  
            opção = lerEntrada("Escolha:")  
  
            quando(opção):  
                1 -> item.nome = lerEntrada("Novo nome:")  
                2 -> item.descrição = lerEntrada("Nova descrição:")  
                3 -> item.preço = lerEntrada("Novo preço:")  
                4 -> item.estoque = lerEntrada("Novo estoque:")  
        }  
    }  
}
```

# Criação de pedido



```
3 -> {  
    imprimir("=== CRIAR PEDIDO ===")  
  
    novoPedido = criarPedido(próximoCódigo)  
    continuarAdicionando = verdadeiro  
  
    enquanto(continuarAdicionando) {  
        mostrarItensDisponiveis(listaItensMenu)  
        códigoItem = lerEntrada("Código do item:")  
  
        para cada item em listaItensMenu {  
            se(item.código == códigoItem) {  
                quantidade = lerEntrada("Quantidade:")  
                adicionarItemAoPedido(novoPedido, item, quantidade)  
                reduzirEstoque(item, quantidade)  
                calcularTotal(novoPedido)  
            }  
        }  
  
        continuarAdicionando = pergunta("Adicionar mais itens?")  
    }  
  
    aplicarCupom(novoPedido)  
    adicionarNaLista(listaPedidos, novoPedido)  
}
```

# Atualização do pedido



```
4 -> {  
    imprimir("=== ATUALIZAR PEDIDO ===")  
    mostrarTodosPedidos(listaPedidos)  
  
    códigoPedido = lerEntrada("Código do pedido:")  
  
    para cada pedido em listaPedidos{  
        se(pedido.código == códigoPedido){  
            imprimirMenuDeStatus()  
            novoStatus = lerEntrada("Novo status:")  
  
            quando(novoStatus):  
                1 -> pedido.status = ACEITO  
                2 -> pedido.status = FAZENDO  
                3 -> pedido.status = FEITO  
                4 -> pedido.status = ESPERANDO_ENTREGADOR  
                5 -> pedido.status = SAIU_PARA_ENTREGA  
                6 -> pedido.status = ENTREGUE  
        }  
    }  
}
```

# Consulta dos Pedidos



```
5 -> {
    imprimir("=== CONSULTAR PEDIDOS ===")
    imprimirMenuDeStatus()

    statusEscolhido = lerEntrada("Status para filtrar:")

    para cada pedido em listaPedidos {
        se(pedido.status == statusEscolhido) {
            mostrarDetalhesPedido(pedido)
        }
    }
}
0 -> {
    opçãoMenu = 0
}
```



# Referências



- **JETBRAINS.** Kotlin Documentation. Kotlinlang.org. Disponível em: <http://kotlinlang.org/docs/home.html>. Acesso em: 22 ago. 2025.