

Universidade Aberta de Portugal
Aluno: Bruno Ricardo de Sá Ferreira
Número de Estudante: 2201529
Data de entrega: 25/Maio/2024
Disciplina: Linguagens de programação
Curso: Engenharia Informática
Docente: Professor RICARDO BAPTISTA
Monitor: RÚDI GUALTER OLIVEIRA

Relatório Linguagens de Programação

A função main inicializa o ambiente Prolog, configura o Scanner para interação com o usuário, exibe o menu principal, direciona para submenus com base na escolha do usuário, e garante entradas válidas até a opção de saída ser selecionada.

Conexões:

Método: registrarVenda

Conexões e Chamadas:

Método Java Main: store.lerClientes()

Store: O método lerClientes da classe Store envia uma consulta Prolog.

Prolog: Recupera todos os registros de clientes (cliente/4) disponíveis na base de dados Prolog.

Método Java Main: store.lerItens()

Store: O método lerItens da classe Store envia uma consulta Prolog.

Prolog: Recupera todos os registros de itens (item/5) disponíveis no inventário do sistema Prolog.

Método Java Main: realizarVenda(store, carrinho)

Store: Utiliza vários métodos da classe Store para calcular totais de venda, aplicar descontos, calcular custos de envio e registrar a venda.

Prolog: Envolve múltiplas consultas Prolog para:

Calcular o total dos itens no carrinho (getItemTotalFromProlog).

Calcular os descontos aplicáveis (getDiscountsFromProlog).

Calcular o custo de envio (getShippingCostFromProlog).

Registrar a venda final (registrarVenda), gravando um novo fato purchase/7 no sistema Prolog.

Método: realizarVenda

Conexões e Chamadas:

Método Java Main: store.getItemTotalFromProlog(carrinho.getItems())

Store: Consulta Prolog para calcular o total dos itens no carrinho.

Prolog: Calcula o total dos itens com base nos registros item/5.

Método Java Main: store.getDiscountsFromProlog(carrinho.getItems(), carrinho.getCliente().getAnosLealdade())

Store: Consulta Prolog para calcular os descontos aplicáveis.

Prolog: Calcula os descontos baseados nos registros discount/2 e loyalty_discount/2.

Método Java Main: store.getShippingCostFromProlog(carrinho.getCliente().getDistrito())

Store: Consulta Prolog para calcular o custo de envio.

Prolog: Calcula o custo de envio com base nos registros shipping_cost/2.

Método Java Main: store.registrarVenda(carrinho.getCliente().getId(), data, carrinho.getItems())

Store: Envia uma consulta Prolog para registrar a venda.

Prolog: Grava um novo fato purchase/7 no sistema Prolog.

Método: verVendasPorData

Conexões e Chamadas:

Método Java Main: store.getVendasPorData(data)

Store: Consulta Prolog para obter as vendas por data.

Prolog: Recupera registros purchase/7 relacionados à data especificada.

Método: verVendasPorCliente

Conexões e Chamadas:

Método Java Main: store.getVendasPorCliente(clienteID)

Store: Consulta Prolog para obter as vendas por cliente.

Prolog: Recupera registros purchase/7 relacionados ao ID do cliente especificado.

Método: verVendasPorDistrito

Conexões e Chamadas:

Método Java Main: store.getVendasPorDistrito(distrito)

Store: Consulta Prolog para obter as vendas por distrito.

Prolog: Recupera registros purchase/7 relacionados ao distrito especificado.

Método: verTotaisPorDistrito

Conexões e Chamadas:

Método Java Main: store.getTotaisPorDistrito(distrito)

Store: Consulta Prolog para obter os totais de vendas, descontos e custos de envio por distrito.

Prolog: Recupera registros totais_por_distrito/5.

Método: verTotaisPorData

Conexões e Chamadas:

Método Java Main: store.getTotaisPorData(data)

Store: Consulta Prolog para obter os totais de vendas, descontos e custos de envio por data.

Prolog: Recupera registros totais_por_data/5.

Método: verDistritoMaisDescontos

Conexões e Chamadas:

Método Java Main: store.getDistritoMaisDescontos()

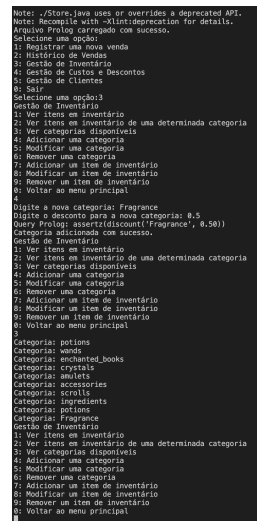
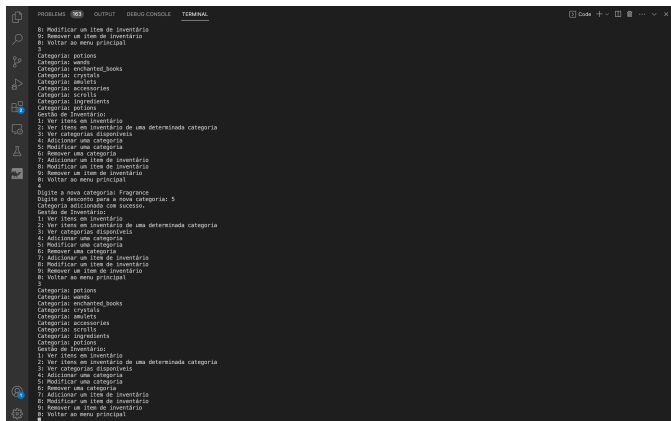
Store: Consulta Prolog para determinar o distrito com mais descontos.

Prolog: Calcula o distrito com base nos registros de descontos aplicados.

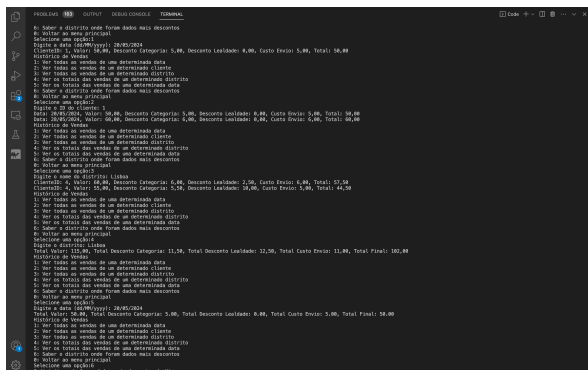
Prints de Testes

Menu Principal Opção 1.

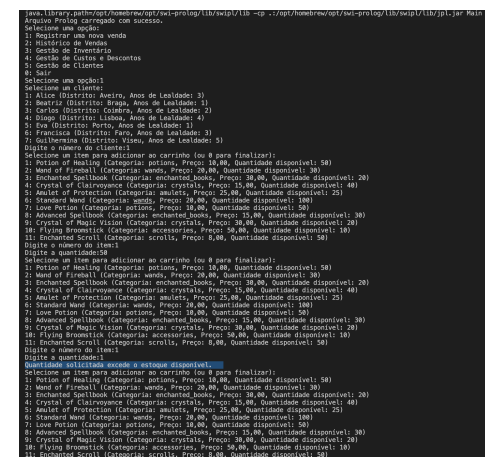
Adicionar categoria



Menu principal opção 2



Menu principal opção 3



A lógica, dos métodos é bastante semelhante então pode-se esperar que estejam funcionais. Não consegui passar os restantes testes, pelo limite de 4 páginas.

Modificar o cliente adicionado, na opção 5 do menu principal e remoção do mesmo

```
ID: 5, Nome: Eva, Distrito: Porto, Anos de Lealdade: 1
ID: 6, Nome: Francisca, Distrito: Faro, Anos de Lealdade: 3
ID: 7, Nome: Guilhermina, Distrito: Viseu, Anos de Lealdade: 5
Gestão de Clientes:
1: Ver todos os clientes
2: Ver todos os clientes de um determinado distrito
3: Ver todas as vendas de um determinado cliente
4: Ver todos os clientes com lealdade superior a determinado valor
5: Adicionar cliente
6: Modificar cliente
7: Remover cliente
0: Voltar ao menu principal
5
Digite o ID do novo cliente: 13
Digite o nome do novo cliente: Bruno
Digite o distrito do novo cliente: Lisboa
Digite os anos de lealdade do novo cliente: 23
Query Prolog: assertz(cliente(13, 'Bruno', 'Lisboa', 23))
Cliente adicionado com sucesso.
ID: 1, Nome: Alice, Distrito: Aveiro, Anos de Lealdade: 3
ID: 2, Nome: Beatriz, Distrito: Braga, Anos de Lealdade: 1
ID: 3, Nome: Carlos, Distrito: Coimbra, Anos de Lealdade: 2
ID: 4, Nome: Diogo, Distrito: Lisboa, Anos de Lealdade: 4
ID: 5, Nome: Eva, Distrito: Porto, Anos de Lealdade: 1
ID: 6, Nome: Francisca, Distrito: Faro, Anos de Lealdade: 3
ID: 7, Nome: Guilhermina, Distrito: Viseu, Anos de Lealdade: 5
ID: 13, Nome: Bruno, Distrito: Lisboa, Anos de Lealdade: 23
Gestão de Clientes:
1: Ver todos os clientes
2: Ver todos os clientes de um determinado distrito
3: Ver todas as vendas de um determinado cliente
4: Ver todos os clientes com lealdade superior a determinado valor
5: Adicionar cliente
6: Modificar cliente
7: Remover cliente
0: Voltar ao menu principal
6
Digite o ID do cliente a ser modificado: 13
Digite o novo nome do cliente: Ferreira
Digite o novo distrito do cliente: Lisboa
Digite os novos anos de lealdade do cliente: 32
Query Prolog: assertz(cliente(13, 'Ferreira', 'Lisboa', 32))
Cliente modificado com sucesso.
ID: 1, Nome: Alice, Distrito: Aveiro, Anos de Lealdade: 3
ID: 2, Nome: Beatriz, Distrito: Braga, Anos de Lealdade: 1
ID: 3, Nome: Carlos, Distrito: Coimbra, Anos de Lealdade: 2
ID: 4, Nome: Diogo, Distrito: Lisboa, Anos de Lealdade: 4
ID: 5, Nome: Eva, Distrito: Porto, Anos de Lealdade: 1
ID: 6, Nome: Francisca, Distrito: Faro, Anos de Lealdade: 3
ID: 7, Nome: Guilhermina, Distrito: Viseu, Anos de Lealdade: 5
ID: 13, Nome: Ferreira, Distrito: Lisboa, Anos de Lealdade: 32
Gestão de Clientes:
1: Ver todos os clientes
2: Ver todos os clientes de um determinado distrito
3: Ver todas as vendas de um determinado cliente
4: Ver todos os clientes com lealdade superior a determinado valor
5: Adicionar cliente
6: Modificar cliente
7: Remover cliente
0: Voltar ao menu principal
1: Ver todos os clientes
2: Ver todos os clientes de um determinado distrito
3: Ver todas as vendas de um determinado cliente
4: Ver todos os clientes com lealdade superior a determinado valor
5: Adicionar cliente
6: Modificar cliente
7: Remover cliente
0: Voltar ao menu principal
7
Digite o ID do cliente a ser removido: 13
Cliente removido com sucesso.
Gestão de Clientes:
1: Ver todos os clientes
2: Ver todos os clientes de um determinado distrito
3: Ver todas as vendas de um determinado cliente
4: Ver todos os clientes com lealdade superior a determinado valor
5: Adicionar cliente
6: Modificar cliente
7: Remover cliente
0: Voltar ao menu principal
1
ID: 1, Nome: Alice, Distrito: Aveiro, Anos de Lealdade: 3
ID: 2, Nome: Beatriz, Distrito: Braga, Anos de Lealdade: 1
ID: 3, Nome: Carlos, Distrito: Coimbra, Anos de Lealdade: 2
ID: 4, Nome: Diogo, Distrito: Lisboa, Anos de Lealdade: 4
ID: 5, Nome: Eva, Distrito: Porto, Anos de Lealdade: 1
ID: 6, Nome: Francisca, Distrito: Faro, Anos de Lealdade: 3
ID: 7, Nome: Guilhermina, Distrito: Viseu, Anos de Lealdade: 5
Gestão de Clientes:
1: Ver todos os clientes
2: Ver todos os clientes de um determinado distrito
3: Ver todas as vendas de um determinado cliente
4: Ver todos os clientes com lealdade superior a determinado valor
5: Adicionar cliente
6: Modificar cliente
7: Remover cliente
0: Voltar ao menu principal
```

O código é uma continuidade do e-folio anterior, com base na estrutura passada.

Baseie-me no feedback do professor para fazer algumas melhorias, e acredito que neste consegui tratar melhor os erros e exceções.

Preferi não modularizar tanto o código, por perceber melhor, e foi mais fácil para ir identificando os erros, de formatação e atualização dos dados que foram os mais frequentes.