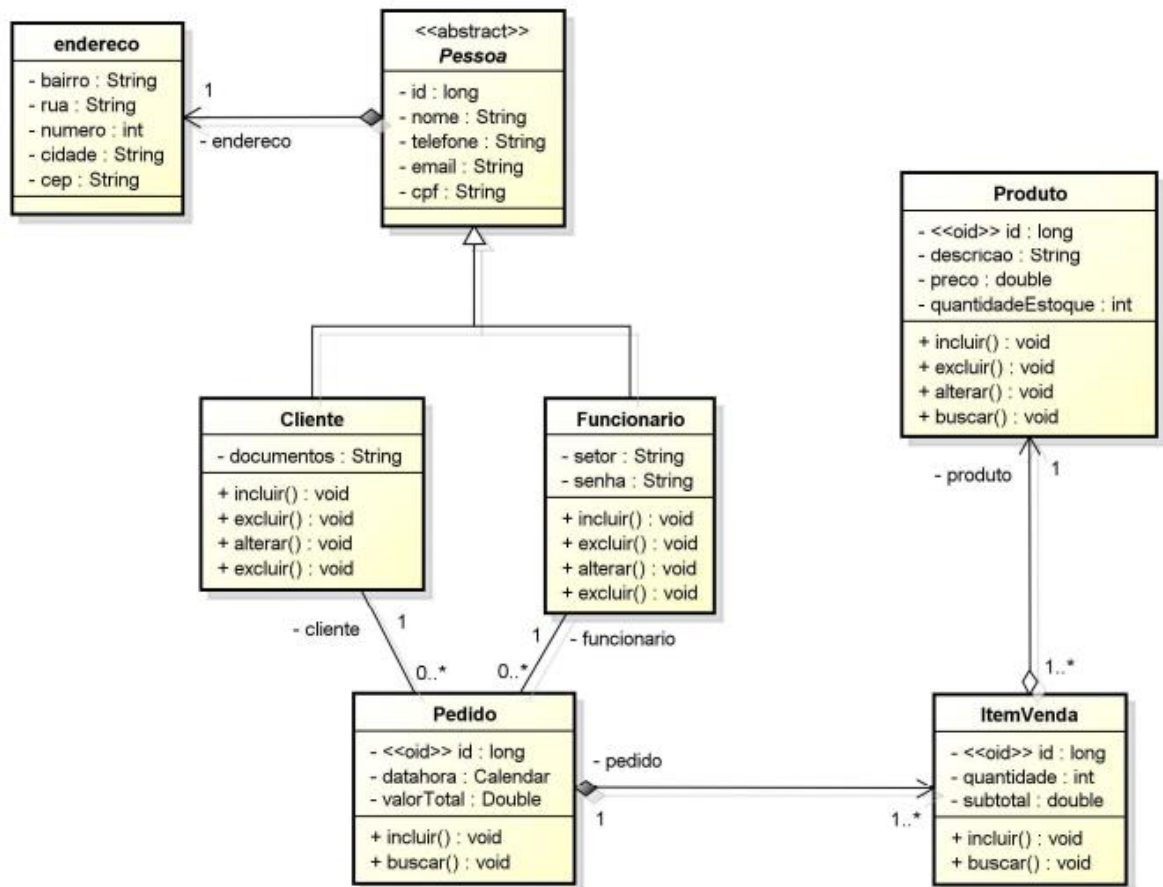


Luis Henrique Barbosa Gomes Filho

Controle de Vendas



Restrições

Atributos gerais Classe: Pessoa

	id	nome	telefone	email	cpf	bairro	rua	numero	cidade	cep
Permite nulo?	Não	Não	Não	Não	Não	Sim	Sim	Sim	Sim	Sim
Permite alteração?	Não	Não	Sim	sim	Não	Sim	Sim	Sim	Sim	Sim
Deverá ser único?	Sim	Sim	Sim	Sim	Sim	Não	Não	Não	Não	Não
Regex	Não	Não	sim	Não	Sim	Não	Não	Não	Não	Não
Tamanho mínimo		4	13	10	14	5			5	9
Tamanho máximo		80	14	80	14	30			30	9
Espaço em branco		Sim	Não	Não	Não	Sim			Sim	Não
Valores negativos?	Não		Não				Não	Não		

Atributos gerais Classe: Cliente

	Permite nulo?	Permite alteração?	Deverá ser único?	Tamanho mínimo	Tamanho máximo	Espaço em branco
documentos	Sim	Não	Não	5	80	Sim

Atributos gerais Classe: Funcionario

	Permite nulo?	Permite alteração?	Deverá ser único?	Tamanho mínimo	Tamanho máximo	Espaço em branco
setor	Não	Sim	Não	5	50	Sim
senha	Não	Sim	Sim	4	50	Não

Atributos gerais Classe: Pedido

	id	cliente	funcionario	datahora	valorTotal
Permite nulo?	Não	Não	Não	Não	Não
Permite alteração?	Não	Não	Não	Não	Não
Deverá ser único?	Sim	Não	Não	Não	Não
Regex	Não	Não	Não	Não	Não
Valores negativos?	Não			Não	Não

Atributos gerais Classe: ItemVenda

	id	pedido	produto	quantidade	subtotal
Permite nulo?	Não	Não	Não	Não	Não
Permite alteração?	Não	Não	Sim	Sim	Sim
Deverá ser único?	Sim	Não	Não	Não	Não
Regex	Não	Não	Não	Não	Não
Valores negativos?	Não			Não	Não

Atributos gerais Classe: Produto

	id	descricao	preco	quantidadeEstoque
Permite nulo?	Não	Não	Não	Não
Permite alteração?	Não	Sim	Sim	Sim
Deverá ser único?	Sim	Não	Não	Não
Regex	Não	Não	Não	Não
Tamanho mínimo		5		
Tamanho máximo		80		
Espaço em branco		Sim		
Valores negativos?	Não			Não

O funcionário será o usuário do sistema, simulando um ponto de vendas e efetuando os registros das vendas.

- **Módulo funcionário-vendedor**

- I. Realiza criação de pedidos solicitados por clientes. Após registrar os dados do pedido, o funcionário confirma o pedido;
- II. Realiza a verificação e edição de seus pedidos em andamento;
- III. Realiza cadastro, consulta e edição dos dados dos clientes;
- IV. Consulta produtos cadastrados, sem excluir ou editar.
- V. O funcionário do setor de vendas não tem acesso a inclusão, edição e alteração dos produtos e de outros funcionários.

- **Módulo funcionário- gerente**

- I. Criação de um funcionário (sendo obrigatório id, nome, cpf, setor e senha) para que eles tenham acesso ao sistema. Pode listar todos seus funcionários, editar informações e excluir;
- II. O gerente pode realizar todas as ações que um vendedor tem acesso;
- III. Realiza a listagem dos clientes, editar das informações e excluir.
- IV. Realiza o controle de estoque adicionado (deve ter um código gerado automaticamente, descrição, preço, descrição e quantidade), editando ou excluindo produtos do cadastro.
- V. Não é permitida a exclusão de produtos atrelado a um item de venda.

feedback realizados na tarefa - tarefa 01

“+ Diagrama

- O nome da classe Pessoa não deve vir dentro de << >> e sim, acima dela deve vir <<abstract>>. Acho que foi isso que queria fazer. Use a opção estereótipo no Astah.
- Coloque os dados referente ao endereço numa classe separada, isso deixará mais clean a classe pessoa. Veja o diagrama do projeto de exemplo da disciplina.
- Os atributos que se referem ao relacionamento, não precisam ser representados dentro da classe.
- O losango preto fica "colado" na classe TODO. Você inverteu.
- Na relação ItemVenda-Produto, não colocaria composição e sim uma relação simples (ou agregação) e adicionaria uma restrição para que não se excluísse um produto atrelado a Item de Venda.

+ Restrições

- OK”

Alterações feitas- tarefa 01

- Realizei as alterações pedidas no comentário.
 - Altere o atributo telefone de Int para String e coloquei restrição de tamanho e regex.
-

Feedback tarefa 02:

- “- Acho que o método equals de Pessoa, Pedido, ItemVenda e Produto poderiam ser a comparação de somente o ID.
- Classe Funcionario não está herdando de Pessoa
- Não vejo motivo de sobrescrever o método equals implementado em Pessoa nas subclasses Cliente e Funcionario
- Implementar as classes de modelo no diretório "main" e não no diretório "test" ”

Correções realizadas tarefa 02

- Realizei as alterações pedidas no comentário.

Feedback semana 3

- “- A anotação @JsonBackReference não pode anotar um atributo do tipo Coleção (veja abaixo). Na vídeoaula cometi esse erro e corrijo em videoaula futura usando @JsonIgnore. Esse problema só será evidenciando quando desenvolvermos a api rest. A solução futura, substituir a anotação @JsonBackReference por @JsonIgnore e excluir a anotação @JsonManagedReference na classe dona da relação.
"Annotation used to indicate that associated property is part of two-way linkage between fields; and that its role is "child" (or "back") link. Value type of the property must be a bean: it can not be a Collection, Map, Array or enumeration. Linkage is handled such that the property annotated with this annotation is not serialized; and during deserialization, its value is set to instance that has the "managed" (forward) link."”

Correções realizadas tarefa 03

- Realizei as alterações pedidas no comentário.

Feedback semana 4

- Coloque @Valid no atributo Pessoa.endereco. Isso fará que ao validar uma Pessoa, também seja validado o endereço.
- Coloque @Valid no atributo Pedido.itensVendas também

Correções realizadas tarefa 04

- Realizei as alterações pedidas no comentário.

Feedback semana 5

- Em `PedidoService.update` você está convertendo um objeto do tipo `ItemVenda` em uma `List<ItemVenda>`. Por isso você pode usar o `List.of(objeto)`. Mas assim, você está dizendo que toda vez que atualizar um pedido ele irá resetar (apagar) a lista de itens, pois você está colocando uma lista vazia. Aqui você deve receber somente o objeto Pedido a ser atualizado, como o Pedido tem um `CascadeType.ALL` para os itens, ele atualiza automaticamente os itens, basta que o objeto Pedido contenha a lista de itens atualizada.
- Mesma coisa acontece em `Produto.update`, não sei o motivo que te levou a querer apagar a relação entre Produto-Itens.
- Quanto ao método "`verificaExclusaoProdutosVendidos`", acho que seria necessário apenas verificar se não há `ItemVenda` relacionado com o Produto. Como Produto já possui uma lista de `ItemVenda` (pois esse relacionamento é bidirecional), basta verificar se essa lista está vazia.

Correções realizadas tarefa 05

- Acredito ter concertado `Pedidoservice.update` e `Produto.update` fazendo eles receberem apenas um objeto
- Produto possui relação com itens(acredito que tinha tirado sem perceber)
- Alterei o método "`verificaExclusaoProdutosVendidos`",

Feedback semana 6

-OK.

Feedback semana 7

-OK.