Correção das questões da Prova

**Aluno:** Jacson de Vargas Erthal

Questão 1.

Alternativa correta: D) Conceitos, associações e atributos.

Questão 2. Cliente faz pedido online acessando o sistema e selecionando os produtos que deseja comprar. O cliente adiciona os produtos ao carrinho e verifica o carrinho. O cliente confirma o pedido e insere os dados de pagamento. A loja processa o pagamento, confirma o pedido e envia um e-mail de confirmação para o cliente. O cliente acompanha o status do pedido. O Funcionário da loja gerencia os pedidos e atualiza o status do pedido. O cliente acessa o sistema e executa a opção "Acompanhar Pedido”. Loja exibe o status atual do pedido (processando, em trânsito, entregue). A loja permite que o cliente visualize as informações do pedido (produtos, data de entrega, etc.). O funcionário da loja gerencia os pedidos executando a opção "Gerenciar Pedidos". A loja exibe uma lista de todos os pedidos. O funcionário seleciona um pedido e pode visualizar as informações do pedido (produtos. cliente, etc.). 0 funcionário pode atualizar o status do pedido (processando, em trânsito, entregue). O Funcionário da loja atualiza ○ status do pedido. O funcionário executa a opção "Atualizar Status do Pedido". A loja exibe uma lista de todos os pedidos. O funcionário seleciona um pedido e atualiza o status do pedido (processando, em trânsito, entregue).

Questão 3.

Alternativa correta: A) III e IV.

Questão 4.

Alternativa correta: E) Requisito não funcional.

Justificativa: Na descrição do problema, é mencionada a necessidade de verificar a identidade do cliente por meio da utilização de dados biométricos. Nesse contexto, a utilização de dados biométricos é um recurso que define como o sistema deve funcionar e não está diretamente relacionado a regras de negócios específicas, padrões de arquitetura ou requisitos funcionais.

Os requisitos não funcionais descrevem features do sistema que não estão diretamente relacionados às funcionalidades específicas, mas sim a propriedades como desempenho, segurança, usabilidade e confiabilidade.

Caso de uso em PlantUML

@startuml

left to right direction

actor Cliente

actor Funcionário

actor Loja

rectangle Sistema {

(Enviar E-mail de Confirmação) <-- Loja

(Processar Pagamento) <-- Loja

cliente --> (Confirmar Pedido)

cliente --> (Adicionar Produtos)

cliente --> (Acompanhar Pedido)

cliente --> (Inserir Dados de Pagamento)

funcionario --> (Gerenciar Pedidos)

(Adicionar Produtos) <.. (Confirmar Pedido) : <<include>>

(Inserir Dados de Pagamento) <.. (Confirmar Pedido) : <<include>>

(Confirmar Pedido) <.. (Processar Pagamento) : <<include>>

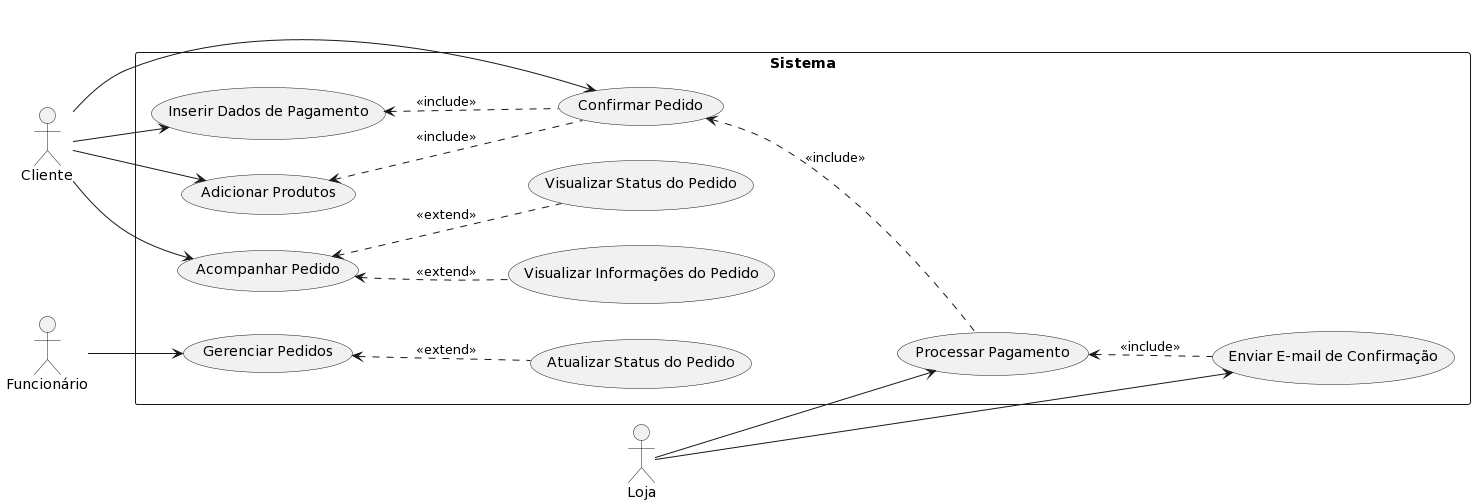
(Processar Pagamento) <.. (Enviar E-mail de Confirmação) : <<include>>

(Acompanhar Pedido) <.. (Visualizar Status do Pedido) : <<extend>>

(Acompanhar Pedido) <.. (Visualizar Informações do Pedido) : <<extend>>

(Gerenciar Pedidos) <.. (Atualizar Status do Pedido) : <<extend>>

}

@enduml

Modelo de Domínio em PlantUML

@startuml

class Cliente {

- codCliente: Integer

- CPF : Double

- nome: String

- email: String

- endereço: String

}

class Produto {

- codProduto: Integer

- descrição: String

- preço: Double

}

class Pedido {

- codPedido : Integer

- codCliente : Integer

- codProduto : Integer

- status: String

- valorTotal: Double

- localDeEntrega : String

- dataDeEntrega : Date

}

class Pagamento {

- codPagamento

- codPedido: Integer

- valorTotal: Double

}

class Loja {

- codFuncionário : Integer

- codCliente : Integer

- codProduto : Integer

- codPedido : Integer

- codPagamento : Integer

- nome : String

- CNPJ : Double

}

class Funcionário {

- codFuncionário : Integer

- nome : String

- CPF : Double

}

class Status\_Pedido {

- codStatus : Integer

- codPedido : Integer

- situaçãoDoPedido : String

- localizaçãoDoPedido : String

}

Pedido <|-- Status\_Pedido

Loja "1" --> "\*" Funcionário : possui

Loja "1" --> "\*" Pedido : possui

Loja "1" --> "\*" Produto : possui

Loja "1" --> "\*" Cliente : possui

Loja "1" --> "\*" Pagamento : recebe

Cliente "\*" --> "\*" Pagamento : realiza

Cliente "\*" --> "\*" Pedido : solicita

Cliente "\*" --> "\*" Status\_Pedido: acompanha

Pedido "1" --> "\*" Produto : contém

Pedido "1" --> "\*" Pagamento : possui

Funcionário "\*" --> "\*" Pedido : gerencia

Funcionário "\*" --> "\*" Status\_Pedido: atualiza

@enduml

