

Iago Felipe Steigleder Bacci - BP3044246
Gabriela Vieira dos Santos Vaz - BP3044513
João Vitor Ferreti Lippi - BP3029891
Jonas Ribeiro da Rosa - BP304422X
Vinícius Pereira Costa - BP3044289

Trabalho 2 de Arquitetura de Software: Fundamentos de Análise e Projeto Orientados a Objetos

Trabalho apresentado ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Bragança Paulista, como requisito parcial da disciplina de Arquitetura De Software sob orientação do Prof. Dr. Wilson Vendramel.

Trabalho 2 de Arquitetura de Software

Instruções Gerais. Leia com bastante atenção.

- a) Este trabalho se refere a uma lista de exercícios de Análise e Design Arquitetural de Software e corresponde a 50% do processo avaliativo da N2. Tal trabalho possui ao todo 4 grupos de exercícios;
 - b) O trabalho em questão deve ser realizado em grupo de 2 a 5 estudantes;
 - c) O trabalho deve ser entregue em um único arquivo PDF em formato inteligível. Além disso, os exercícios devem ser apresentados conforme enunciados e seções; cada artefato não apresentado na ordem prevista e no seu devido espaço vai acarretar a anulação do exercício correspondente. Se for favorecer a visualização do artefato, cabe ressaltar que links de acesso a arquivos de imagem podem ser inseridos em tal documento. Deve ficar claro que os exercícios com artefatos ilegíveis serão anulados. A fim de se obter uma organização favorável dos artefatos, um sumário deve ser elaborado;
 - d) O trabalho em foco deve ser entregue como tarefa na plataforma Moodle até às 23h59 de 22/06/2025. A entrega da lista por um único integrante do grupo já é suficiente, assim sendo, deve ficar claro que não há a necessidade de todos os integrantes da equipe entregarem tal tarefa;
 - e) O trabalho deve ser verificado e validado pelo próprio grupo antes de ser entregue, dado que não serão aceitas novas entregas, parcial ou totalmente após o prazo final;
 - f) As dúvidas enviadas via e-mail institucional podem ser respondidas uma vez que sejam submetidas até às 23h59 de 19/06/2025;
 - g) Os materiais disponibilizados auxiliam a realização dos exercícios desta lista;
 - h) Estudantes podem ser convidados(as) durante a aula para resolver exercícios com o objetivo de validar a lista realizada;
 - i) Os trabalhos com respostas suspeitas de plágio serão devidamente anulados e zerados. Cada exercício que apresentar a mesma resposta em duas ou mais listas será devidamente anulado e a respectiva pontuação descontada;
 - j) Exceções sobre as instruções supracitadas devem ser tratadas com o próprio professor antecipadamente.
-

Análise e Design de uma Aplicação Web de E-commerce de Livraria

Especificação do Sistema de Software:

Com base nos Atores, Regras de Negócio e Casos de Uso presentes neste documento, realize os exercícios de Análise e Design de Software para uma Aplicação Web de E-commerce de Livraria.

Atores:

- Cliente: esse ator representa os usuários externos interessados na compra de livros;
- Funcionário: esse ator representa os usuários internos responsáveis pela manutenção dos dados dos livros, incluindo o controle da quantidade de exemplares em estoque;
- Sistema de Frete: esse ator representa um sistema/componente externo a ser integrado com a aplicação de software da livraria para calcular o frete a ser pago;
- Sistema de Cartão: esse ator representa um sistema/componente externo a ser integrado com a aplicação de software da livraria para possibilitar o pagamento por meio de cartão de crédito;
- Sistema de Banco: esse ator representa um sistema/componente externo a ser integrado com a aplicação de software da livraria para possibilitar o pagamento via Pix;
- Se necessário, outros atores, inclusive sistemas externos, podem ser identificados.

Regras de Negócio:

- RN01: O livro pode apresentar um dos seguintes status em um determinado tempo: disponível, indisponível ou em FORA DE CIRCULAÇÃO;
- RN02: Há três formatos de livro: a) físico capa dura; b) físico brochura e; c) digital, sendo que o preço de cada tipo de livro deve ser calculado de forma diferente, melhor dizendo, sem desconto, 5% e 10%, respectivamente;
- RN03: Para livros em circulação, o estoque mínimo não pode ser menor do que 2 exemplares. Uma vez que o estoque atinja essa quantidade, um funcionário deve ser notificado via mensagem;
- RN04: O pagamento pode ser realizado por meio de cartão de crédito ou Pix. O pagamento por cartão de crédito pode ser parcelado em até 3 vezes sem juros, porém, se for pago à vista, deve aplicar 3% de desconto no valor total do pedido. O pagamento via Pix, por sua vez, deve ser efetuado à vista e aplicar 8% de desconto no valor total do pedido;
- RN05: O pedido deve apresentar um dos seguintes status em um determinado tempo: EM PROCESSAMENTO, PAGAMENTO PENDENTE, CONFIRMADO, EM TRANSPORTE OU FINALIZADO;
- Se preciso, novas regras de negócio podem ser identificadas.

Casos de Uso:

- CSU01: Pesquisar Livro: este caso de uso representa o processo pelo qual um cliente pode pesquisar livros por título, autor, editora ou categoria. Após selecionar uma forma de pesquisa e informar os parâmetros da pesquisa, o sistema deve exibir uma listagem referente a todos os livros que satisfaçam as condições de busca, contendo o título, autor, editora, categoria, status e preço, além de permitir visualizar os detalhes do livro e adicioná-lo ao carrinho de compras. Se o cliente optar por visualizar os detalhes de um dos livros listados, uma nova página será exibida, apresentando um resumo do livro, seu formato, número de páginas e o ano de publicação. Havendo interesse por algum dos livros apresentados, o cliente pode adicionar o livro ao carrinho de compras, desse modo, uma nova página será exibida, solicitando a confirmação da quantidade de volumes que deseja comprar (o padrão é 1). Cabe frisar que tanto o caso de uso “Exibir Detalhes” quanto o caso de uso “Adicionar ao Carrinho” são extensões do caso de uso de que se trata.
- CSU02: Efetuar Pedido: este caso de uso é referente ao processo de confirmação do pedido. Na primeira etapa, o cliente confirma o pedido (deve haver minimamente um item adicionado ao carrinho), e a página de visualização do carrinho será exibida, assim sendo, o cliente visualiza os livros selecionados e suas respectivas quantidades. Se o cliente ainda não estiver cadastrado no sistema, será preciso antes fazer o cadastro, informando seus dados pessoais. Na segunda etapa, uma vez autenticado pelo sistema, o cliente deve informar o endereço de entrega para o frete ser calculado, e em seguida selecionar a forma de pagamento, podendo ser por meio de cartão de crédito ou Pix. Após isto, o cliente deve finalizar o pedido e a quantidade de livros em estoque deve ser atualizada. Assim que o pedido for finalizado, os itens do carrinho devem ser destruídos. Cabe frisar que os casos de uso “Atualizar Estoque”, “Visualizar Carrinho”, “Calcular Frete” e “Efetuar Pagamento” são inclusões do caso de uso de que se trata e que “Pagar via Cartão” e “Pagar via Pix” são especializações do caso de uso “Efetuar Pagamento”.

- CSU03: Manter Cliente: este caso de uso representa a manutenção dos dados pessoais do cliente. Esse cadastro deve incluir o nome completo, CPF, data de nascimento, e-mail, telefone, endereço (um ou mais). Uma vez que o cliente ainda não possua cadastro no sistema ou seus dados tenham sofrido alguma alteração desde a última compra, é necessário registrar ou alterar seu cadastro. Cabe frisar que o caso de uso em questão é também uma extensão do caso de uso “Efetuar Pedido”.
- CSU04: Visualizar Pedido: este caso de uso se refere ao processo pelo qual um cliente pode acompanhar o pedido recente, assim como, visualizar o histórico de compras. Neste caso, os usuários podem tanto consultar pedidos anteriores quanto o status do pedido atual. Para realizar esse caso de uso, é necessário o cliente estar autenticado pelo sistema.
- CSU05: Manter Livro: este caso de uso se refere à manutenção dos dados do livro. Esse cadastro deve incluir o título, ISBN, número de páginas, ano de publicação, categoria (uma ou mais), formato do livro, status, preço de venda e um pequeno resumo, além da quantidade de exemplares em estoque e informações do autor (nome, data de nascimento e nacionalidade) e da editora (nome, CNPJ, telefone e e-mail para contato). Os casos de uso “Manter Autor” e “Manter Editora” são extensões do caso de uso que se trata.

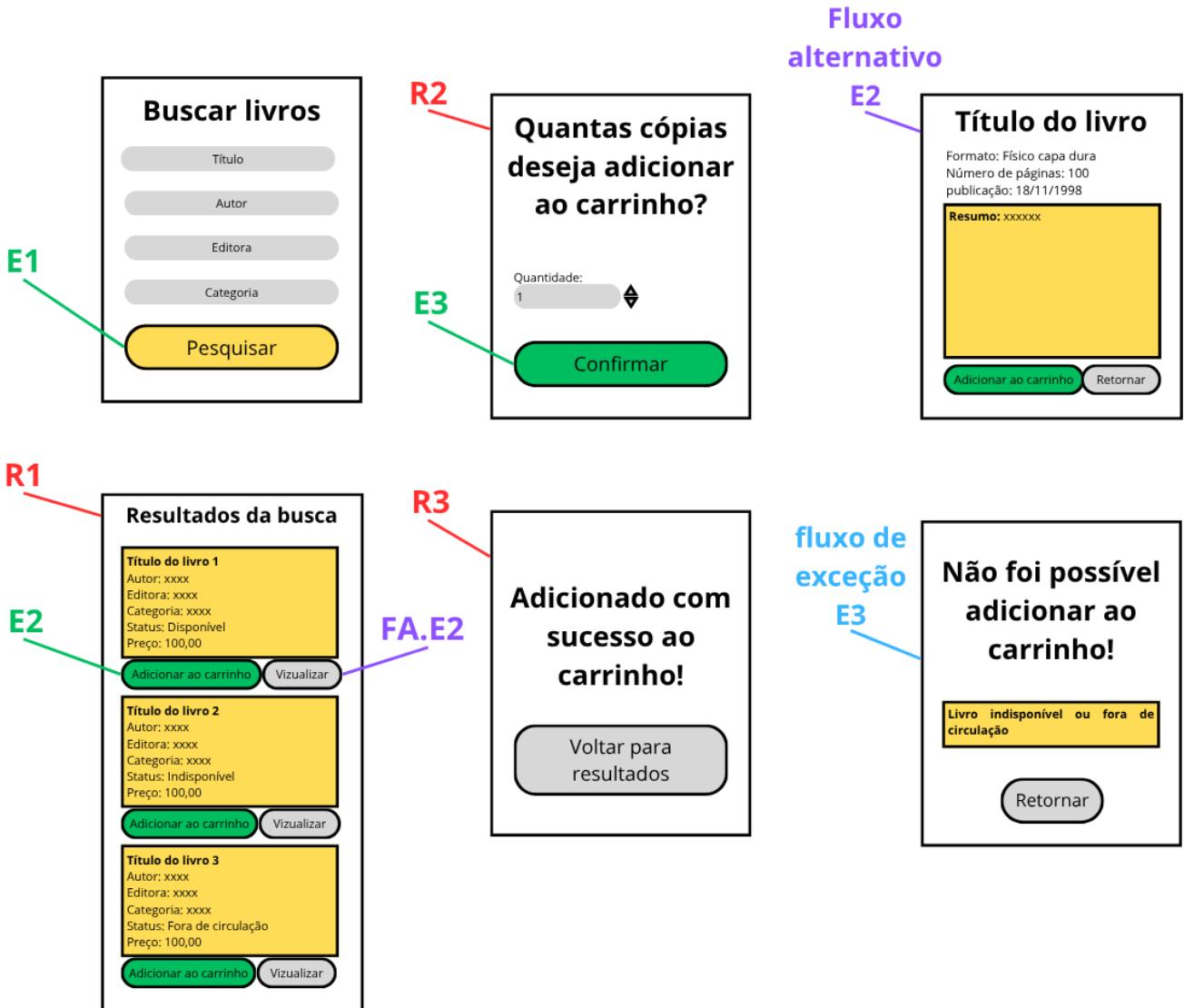
Grupo 1 de Exercícios: Caso de Uso 01, Protótipos de Interface, Eventos de Sistema e VCP (1,5 ponto)

- a) Considerando a especificação do software, descreva textualmente o CSU01, explicitando a sequência de interações entre o ator (esse passo deve ser identificado como estímulo) e o sistema (esse passo deve ser identificado como resposta), para os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template (Quadro 1a) e as regras de negócio previamente descritas. Os diferentes tipos de fluxo devem estar organizados apropriadamente pelo número do passo. (0,5 ponto)

Quadro 1a: Descrição Textual do Caso de Uso

Caso de Uso: Pesquisar livro
Descrição: Cliente pesquisa livro por título, autor, editor ou categoria
Ator Primário: Cliente
Ator(es) Secundário(s), se houver: Software da livraria
Precondições: Estar utilizando o site da livraria
Fluxo Principal:
1) Cliente (Estímulo 1) Digita os parâmetros escolhidos para a busca e clica no botão “Pesquisar” .
2) Sistema (Resposta 1) Exibe uma listagem de livros que satisfazem as condições da busca(Conforme a RN01 e RN02).
3) Cliente (Estímulo 2) Adiciona livros desejados ao carrinho de compras.
4) Sistema (Resposta 2) Exibe uma página solicitando a confirmação da quantidade de volumes que deseja adicionar ao carrinho.
5) Cliente (Estímulo 3) Confirma a quantidade de cópias a adicionar no carrinho.
6) Sistema (Resposta 4) Exibe uma mensagem de que o livro foi adicionado ao carrinho com sucesso.
Fluxo Alternativo (número do passo 4):
Cliente (Estímulo) Clica em “Visualizar” um dos livros listados,
Sistema (Resposta) Uma nova página é exibida, apresentando resumo do livro, formato, número de páginas e ano de publicação.
a) Cliente (Estímulo 1) Adiciona o livro ao carrinho de compras, o caso de uso prossegue a partir do passo 2.
b) Cliente (Estímulo 2) Clica em “Retornar”, o caso de uso retorna ao passo 2.
...
Fluxo de Exceção (número do passo, 3)
Cliente (Estímulo) Adicionar ao carrinho um livro que esteja fora de circulação ou indisponível. Violação da RN01.
Sistema (Resposta) Informa que não é possível adicionar ao carrinho de compras. O caso de uso retorna ao passo 2.
Pós-condições: O cliente pesquisou e adicionou ao carrinho de compras os livros que deseja comprar e suas quantidades.
Regras de Negócio Relacionadas (RN01)
RN01: Garante que apenas livros com status "disponível" podem ser adicionados ao carrinho.

- c) Visando a identificação dos eventos de sistema a partir dos estímulos e respostas reconhecidos nos passos, elabore os protótipos de interface de usuário (baixa ou média fidelidade) para cada evento de sistema reconhecido nesse caso de uso. Com base nos eventos de sistema identificados, associe cada evento à respectiva operação da classe de controle, conforme Tabela 1a. (0,5 ponto)

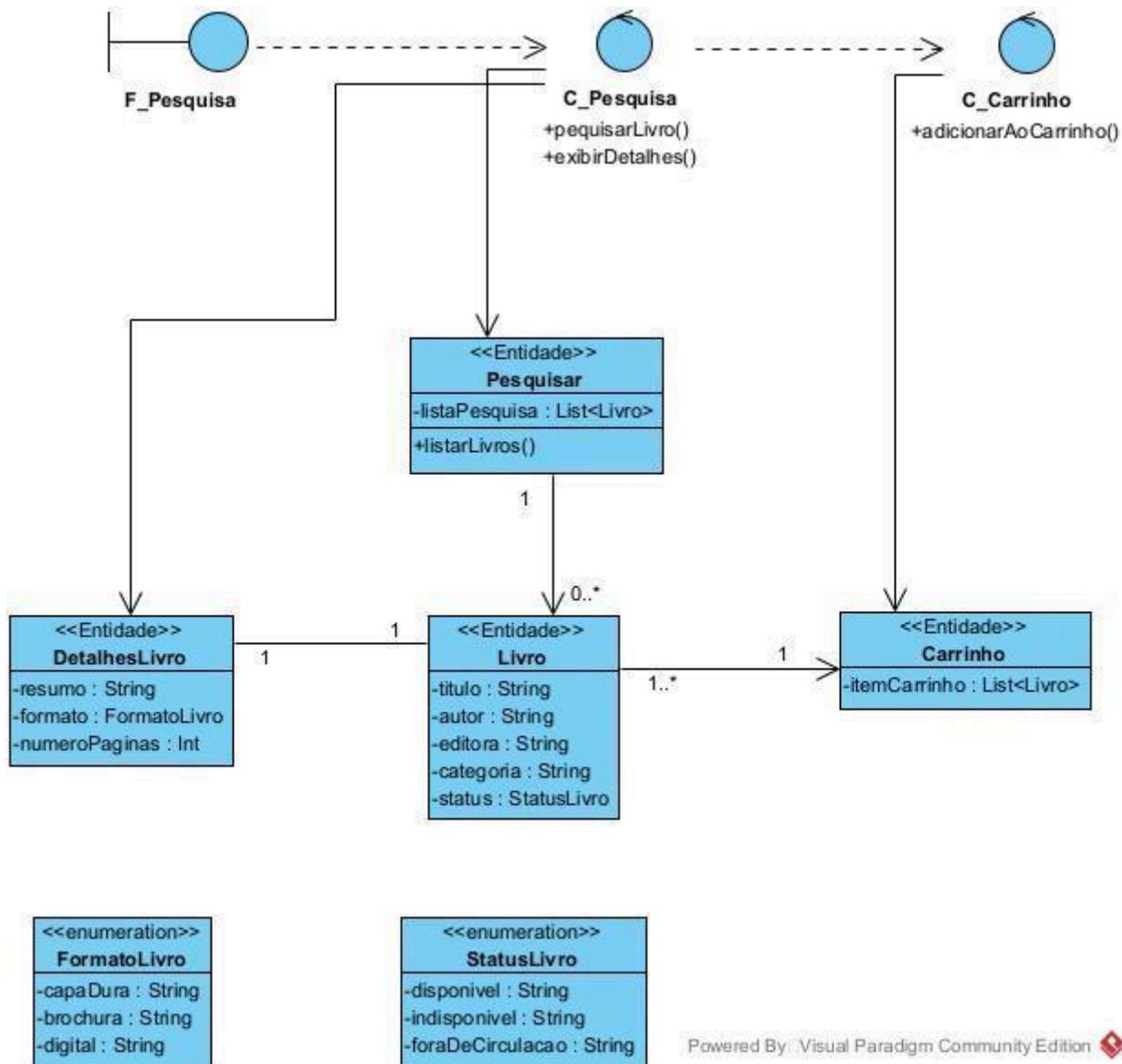


[https://drive.google.com/file/d/11-rZFE7JMdcUhN3bBVSEeyjsHWPCEBXk/view?usp=drive_link]

Tabela 1a: Eventos e Operações da Classe de Controle

Evento	Classe	Operação
Pesquisar	C_Pesquisa	pesquisarLivro()
Vizualizar	C_Pesquisa	exibirDetalhes()
Adicionar ao carrinho	C_Carrinho	adicionarAoCarrinho()
Confirmar	C_Carrinho	adicionarAoCarrinho()

- d) Mediante aplicação da técnica de Análise de Caso de Uso e identificação dos eventos de sistema, modele uma VCP (Visão de Classes Participantes) com categorização BCE para o CSU01. Cabe ressaltar que a classe de controle deve apresentar as devidas operações e as classes de entidade seus atributos. A navegabilidade e as multiplicidades dos relacionamentos devem ser exibidas. (0,5 ponto).



[https://drive.google.com/file/d/1_god2wjcP3Q37RFg8-bupACxlcGN0TpG/view?usp=drive_link]

Explicação:

No diagrama de classe é mostrado como a classe controladora **C_Pesquisa** realiza a pesquisa de acordo com os parâmetros escolhidos pelo cliente através da operação **pesquisarLivro()** que faz a busca por livros que satisfaçam aos parâmetros da busca, e depois, pela classe **Pesquisar** a operação **listarLivros()** exibe uma lista com todos os livros encontrados.

A classe controladora **C_Carrinho** é responsável por ser chamada para adicionar um livro a uma lista dentro da entidade **Carrinho** através da operação **adicionarAoCarrinho()**.

Grupo 2 de Exercícios: Caso de Uso 02, Protótipos de Interface, Eventos de Sistema e VCP (1,5 ponto)

a) Considerando a especificação do software, descreva textualmente o CSU02, explicitando a sequência de interações entre o ator (esse passo deve ser identificado como estímulo) e o sistema (esse passo deve ser identificado como resposta), para os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template (Quadro 2a) e as regras de negócio previamente descritas. Os diferentes tipos de fluxo devem estar organizados apropriadamente pelo número do passo. (0,5 ponto)

Quadro 2a: Descrição Textual do Caso de Uso

Caso de Uso: Efetuar Pedido
Descrição: Este caso de uso descreve o processo pelo qual um cliente finaliza sua compra, confirmando os itens no carrinho, informando dados de entrega e selecionando a forma de pagamento, resultando na criação de um pedido e atualização do estoque.
Ator Primário: Cliente
Ator(es) Secundário(s), se houver: Sistema de Frete, Sistema de Cartão, Sistema de Banco
Precondições: - O cliente possui, no mínimo, um item adicionado ao carrinho de compras. - Os livros no carrinho devem ter o status "disponível" (RN01).
Fluxo o Principal:
1. Cliente (Estímulo): Clica no botão "Confirmar Pedido". 2. Sistema (Resposta): Exibe a página de visualização do carrinho com os livros selecionados, quantidades e preços unitários. Este passo implica a execução do caso de uso "Visualizar Carrinho". 3. Cliente (Estímulo): Após visualizar os itens no carrinho, clica no botão "Continuar". 4. Sistema (Resposta): Verifica se o cliente está autenticado. 5. Cliente (Estímulo): Se o cliente não estiver autenticado, o sistema invoca o caso de uso "Manter Cliente" para autenticação (login ou cadastro). O fluxo do "Efetuar Pedido" é pausado até que o "Manter Cliente" seja concluído com sucesso, resultando na autenticação do cliente. Observação: Falhas dentro do "Manter Cliente" (ex: credenciais inválidas) são tratadas internamente por aquele caso de uso, que exibe suas próprias mensagens de erro e aguarda a correção para permitir a conclusão bem-sucedida da autenticação. 6. Sistema (Resposta): Autentica o cliente e exibe a página de informações de entrega. 7. Cliente (Estímulo): Informa o endereço de entrega completo. 8. Sistema (Resposta): Interage com o Sistema de Frete para calcular as opções de frete e seus respectivos custos, exibindo-as ao cliente. Este passo implica a execução do caso de uso "Calcular Frete". 9. A) Cliente (Estímulo): Seleciona uma opção de frete e a forma de pagamento "Cartão de Crédito". B) Cliente (Estímulo): Seleciona uma opção de frete e a forma de pagamento "Pix". 10. A) Sistema (Resposta): Se "Cartão de Crédito" for selecionado, exibe os campos para dados do cartão e opções de parcelamento (até 3 vezes sem juros) e a opção de pagamento à vista com 3% de desconto (RN04). B) Sistema (Resposta): Se "Pix" for selecionado, informa que o pagamento será à vista com 8% de desconto (RN04). 11. Cliente (Estímulo): Fornece os solicitados(se aplicável, como para cartão de crédito) e clica no botão "Finalizar Pedido" para confirmar e submeter os dados de pagamento.. 12. Sistema (Resposta): Valida os dados de pagamento e invoca o caso de uso "Efetuar Pagamento". Se a forma de pagamento for cartão, é uma especialização "Pagar via Cartão". Se for Pix, é uma especialização "Pagar via Pix". 13. Sistema (Resposta): Após a confirmação do pagamento, atualiza a quantidade de livros em estoque, decrementando-a de acordo com os itens do pedido (RN03). Este passo implica a execução do caso de uso "Atualizar Estoque". Se o estoque de algum livro em circulação atingir 2 exemplares, uma notificação é enviada ao Funcionário (RN03). 14. Sistema (Resposta): Define o status do pedido como "confirmado" (RN05) e destrói os itens do carrinho. 15. Sistema (Resposta): Exibe a página de confirmação do pedido, incluindo o número do pedido e detalhes da compra. Fluxo Alternativo (número do passo, por exemplo, 4): 4: Sistema (Resposta): Verifica se o cliente está autenticado. Se o cliente já estiver autenticado, avança diretamente para o passo 6 do fluxo principal. 12: a) Sistema (Resposta): Caso o pagamento via cartão de crédito seja à vista, aplica 3% de desconto no valor total do pedido (RN04).

b) Sistema (Resposta): Caso o pagamento via Pix seja selecionado, aplica 8% de desconto no valor total do pedido (RN04).

Fluxo de Exceção (número do passo, por exemplo, 2):

12. Sistema (Resposta): Se o pagamento não for aprovado pelo **Sistema de Cartão** ou **Sistema de Banco** (resultado do caso de uso "Efetuar Pagamento"), exibe uma mensagem de erro ao cliente e solicita que tente novamente ou escolha outra forma de pagamento. O status do pedido é definido como "pagamento pendente" (RN05).

Pós-condições:

- Um novo pedido é criado no sistema com o status "confirmado" ou "pagamento pendente" (RN05).
- A quantidade em estoque dos livros comprados é atualizada.
- Os itens do carrinho do cliente são removidos.
- O Funcionário é notificado se o estoque de algum livro em circulação atingir o mínimo de 2 exemplares (RN03).

Regras de Negócio Relacionadas (devem ser redigidas com clareza)

RN01: Garante que apenas livros com status "disponível" podem ser adicionados ao carrinho e incluídos em um pedido.

RN03: Após a finalização do pedido, o sistema verifica se o estoque de livros em circulação atingiu o mínimo de 2 exemplares e, se sim, notifica o funcionário.

RN04: Define as opções de pagamento (cartão de crédito ou Pix), as condições de parcelamento e os descontos aplicáveis (3% para cartão à vista, 8% para Pix à vista).

RN05: Garante que o status do pedido seja atualizado corretamente para "em processamento", "pagamento pendente", "confirmado", "em transporte" ou "finalizado" durante o ciclo de vida do pedido. Neste caso, "confirmado" ou "pagamento pendente" após a tentativa de pagamento.

b) Visando a identificação dos eventos de sistema a partir dos estímulos e respostas reconhecidos nos passos, elabore os protótipos de interface de usuário (baixa ou média fidelidade) para cada evento de sistema reconhecido nesse caso de uso. Com base nos eventos de sistema identificados, associe cada evento à respectiva operação da classe de controle, conforme Tabela 2a. (0,5 ponto).

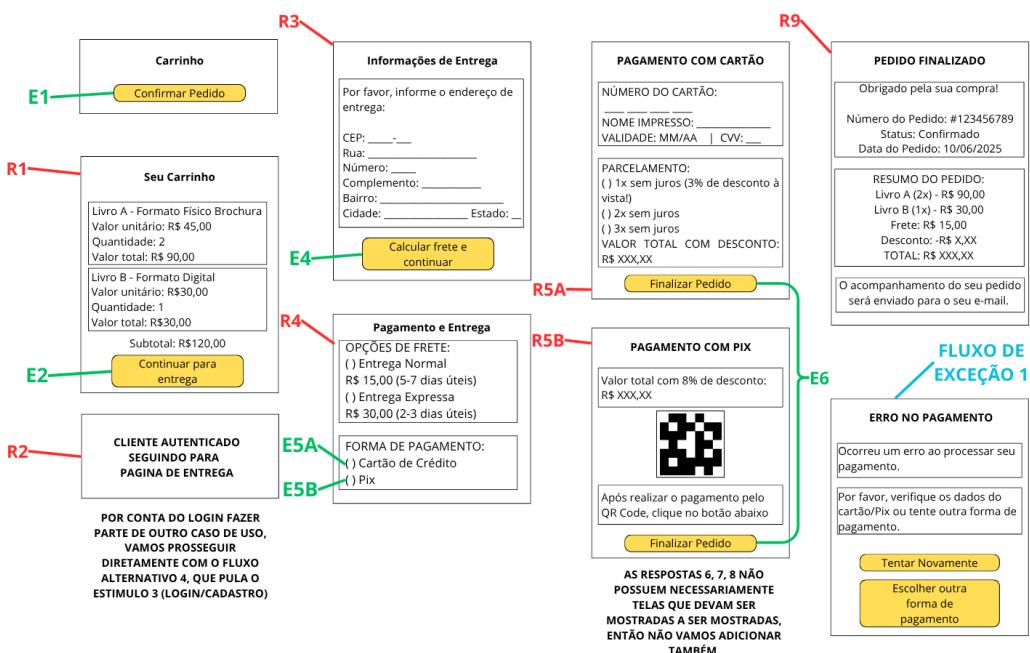
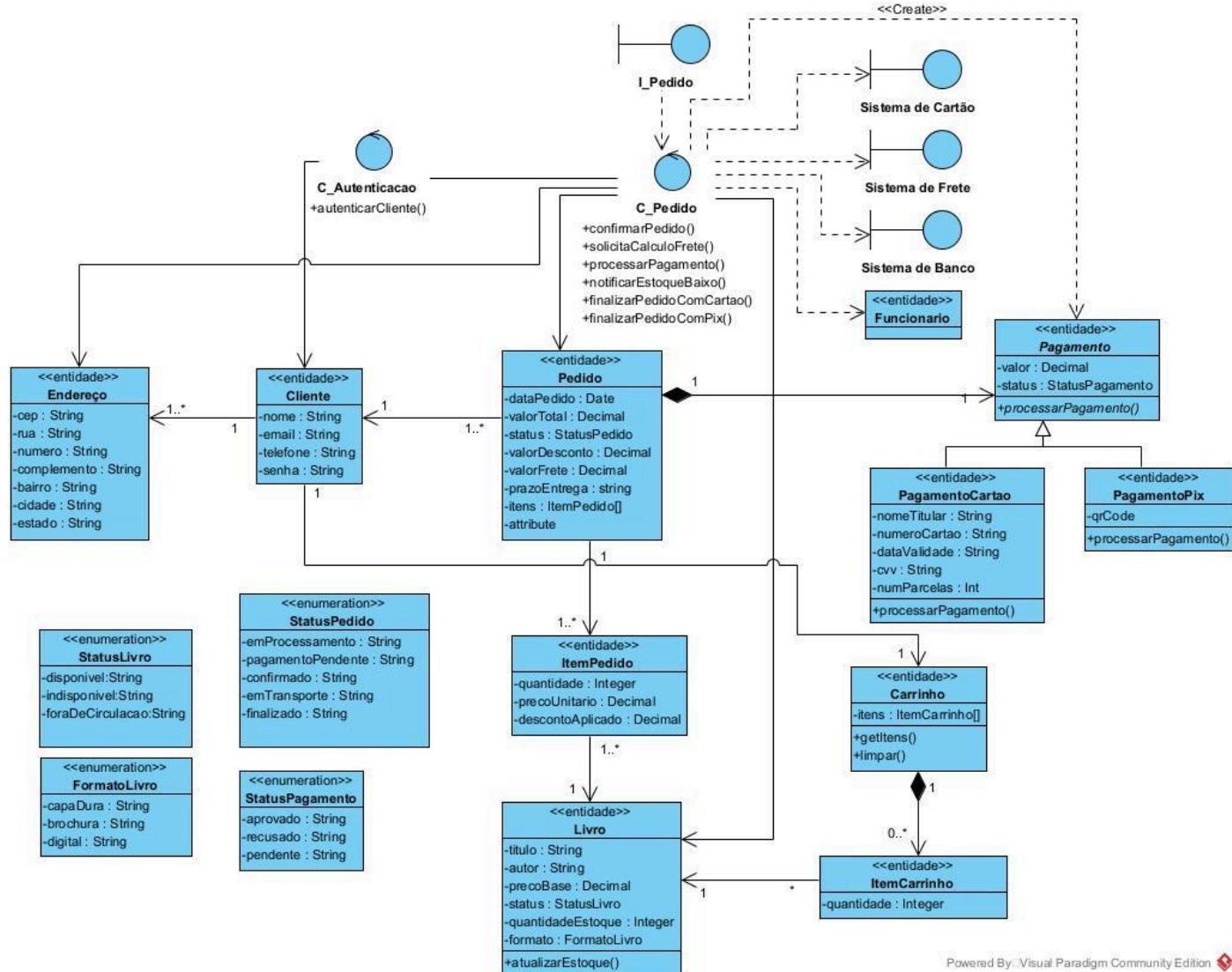


Tabela 2a: Eventos e Operações da Classe de Controle

Evento	Classe	Operação
Continuar para Entrega	C_Autenticação	autenticarCliente()
BCalcular Frete	C_Pedido	calcularFrete()
Finalizar Pedido após selecionar pagamento com cartão	C_Pedido	finalizarPedidoComCartao()
Finalizar Pedido após selecionar pagamento com pix	C_Pedido	finalizarPedidoComPix()
Finalizar Pedido (Independente de qual forma)	C_Pedido	processarPagamento()
Estoque com menos de 2 exemplares	C_Pedido	notificarEstoqueBaixo()
Confirmar pedido	C_Pedido	confirmarPedido()

Mediante aplicação da técnica de Análise de Caso de Uso e identificação dos eventos de sistema, modele uma VCP (Visão de Classes Participantes) com categorização BCE para o CSU02. Cabe ressaltar que a classe de controle deve apresentar as devidas operações e as classes de entidade seus atributos. A navegabilidade e as multiplicidades dos relacionamentos devem ser exibidas. (0,5 ponto)



[https://drive.google.com/file/d/1VjsP2M8maudjtuc0EiMdWL4tDkxCCR/view?usp=drive_link]

Explicação:

O diagrama de classes apresenta um fluxo estruturado de compra em que o Cliente interage com o sistema ao realizar um Pedido, composto por um ou mais livros. Antes disso, os produtos de interesse são armazenados no CarrinhoDeCompras, que pertence exclusivamente ao Cliente e contém múltiplos ItemCarrinho, cada um associado a um Livro específico.

A finalização do pedido é conduzida por dois controladores: o C_Autenticacao, que valida o Cliente, e o C_Pedido, que coordena o restante do processo — cálculo do frete (via Sistema de Frete), processamento do pagamento e criação do Pedido. O Pedido resultante é vinculado a um Cliente e contém ItemPedido, que registra os produtos comprados e seus respectivos preços no momento da compra. O pagamento é representado por uma instância única da classe Pagamento, podendo ser PagamentoCartao ou PagamentoPix, cada um com lógica própria e integração com sistemas externos (Sistema de Cartão ou de Banco).

O Livro possui o método atualizarEstoque(), acionado após a confirmação da compra. Se o estoque atingir um nível crítico, o controlador notifica um Funcionario. O modelo também inclui enums para representar estados e status associados a diferentes classes.

No diagrama de sequência, o fluxo inicia quando o Cliente clica em "Confirmar Pedido". A interface (I_Pedido) envia a solicitação ao C_Pedido, que obtém os itens do carrinho e exibe o resumo. Após a confirmação do cliente, ocorre a autenticação via C_Autenticacao. Com o Cliente validado, o C_Pedido calcula o frete e, conforme a forma de pagamento escolhida, instancia PagamentoPix ou PagamentoCartao, interagindo com os respectivos sistemas.

Após a confirmação do pagamento, o Pedido é criado, os ItemPedido são gerados, o estoque dos livros é atualizado e, se necessário, um Funcionario é alertado. O carrinho é então esvaziado e o cliente recebe a confirmação via interface. Se o pagamento falhar, o sistema redireciona para uma tela de erro, encerrando o fluxo com segurança.

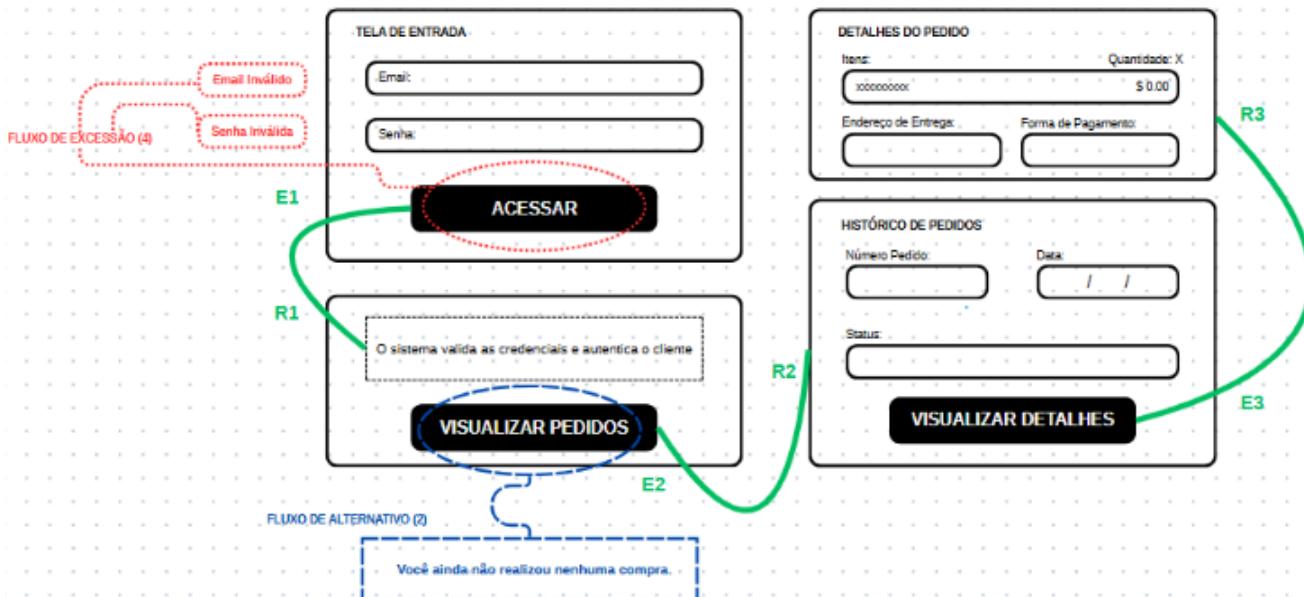
Grupo 3 de Exercícios: Caso de Uso 04, Protótipos de Interface, Eventos de Sistema e VCP (1,5 ponto)

a) Considerando a especificação do software, descreva textualmente o CSU04, explicitando a sequência de interações entre o ator (esse passo deve ser identificado como estímulo) e o sistema (esse passo deve ser identificado como resposta), para os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template (Quadro 3a) e as regras de negócio previamente descritas. Os diferentes tipos de fluxo devem estar organizados apropriadamente pelo número do passo. (0,5 ponto)

Quadro 3a: Descrição Textual do Caso de Uso

Caso de Uso: Visualizar Pedido.
Descrição: Este caso de uso se refere ao processo pelo qual um cliente pode acompanhar o pedido recente, assim como, visualizar o histórico de compras. Neste caso, os usuários podem tanto consultar pedidos anteriores quanto o status do pedido atual. Para realizar esse caso de uso, é necessário o cliente estar autenticado pelo sistema.
Ator Primário: Cliente
Ator(es) Secundário(s), se houver: Nenhum
Precondições: É necessário o cliente estar autenticado pelo sistema.
Fluxo Principal:
1) O cliente insere seu email e senha e solicita o acesso. (Estímulo) 2) O sistema valida as credenciais e autentica o cliente. (Resposta) 3) O cliente acessa a funcionalidade de visualização de pedidos. (Estímulo) 4) O sistema exibe os dados do histórico de compras (número, data e status de cada pedido). (Resposta) 5) O cliente solicita a visualização dos detalhes de um pedido específico. (Estímulo) 6) O sistema exibe as informações completas do pedido selecionado. (Resposta)
Fluxo Alternativo (2): 1) O sistema verifica que o cliente não possui nenhum pedido. 2) O sistema exibe a mensagem: "Você ainda não realizou nenhuma compra."
Fluxo de Exceção (4): 1) O cliente informa suas credenciais de acesso. (Estímulo) 2) O sistema exibe uma mensagem de erro por credencial inválida. (Resposta)
Pós-condições: O cliente autenticado teve acesso a visualização dos pedidos ou do histórico de compras.
Regras de Negócio Relacionadas (devem ser redigidas com clareza) RN05: O pedido deve apresentar um dos seguintes status em um determinado tempo: em processamento, pagamento pendente, confirmado, em transporte ou finalizado;

b) Visando a identificação dos eventos de sistema a partir dos estímulos e respostas reconhecidos nos passos, elabore os protótipos de interface de usuário (baixa ou média fidelidade) para cada evento de sistema reconhecido nesse caso de uso. Com base nos eventos de sistema identificados, associe cada evento à respectiva operação da classe de controle, conforme Tabela 3a. (0,5 ponto).

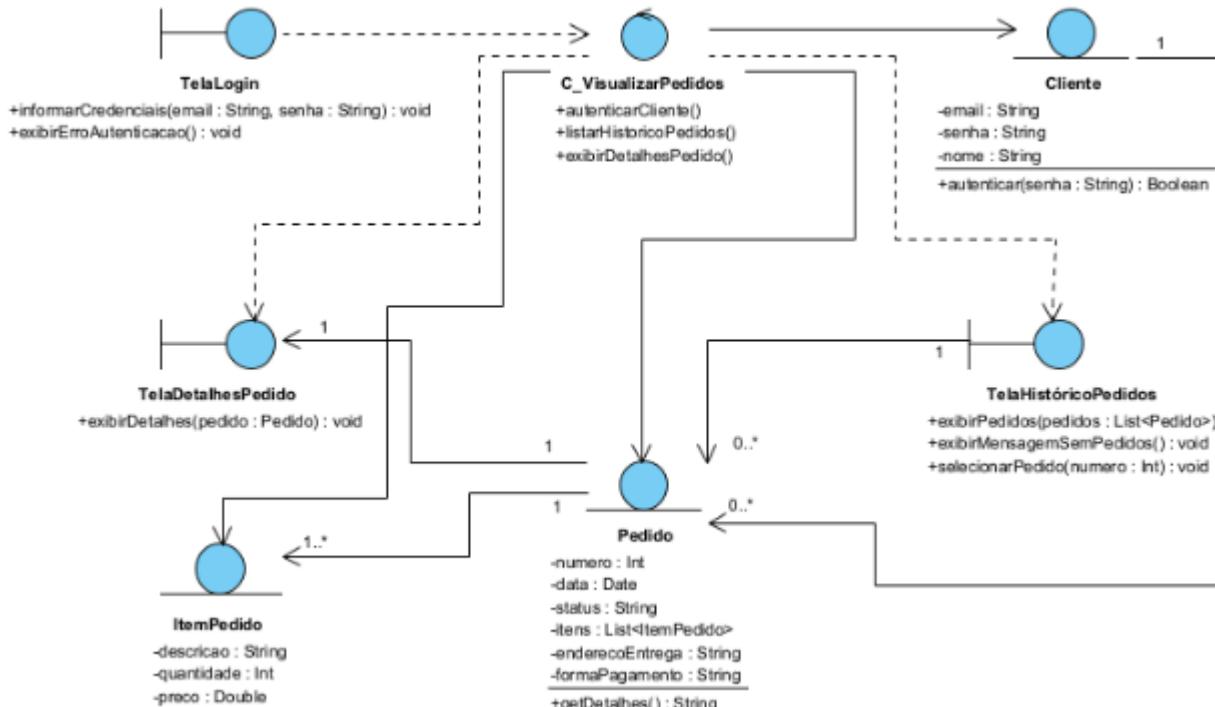


[https://drive.google.com/file/d/1YSFr_qF41NT1eAL-DrLqV9_MSm-2kc1n/view?usp=drive_link]

Tabela 3a: Eventos e Operações da Classe de Controle

Evento	Classe	Operação
Autenticar Cliente	C_VisualizarPedido	autenticarCliente()
Listar Histórico de Compras	C_VisualizarPedido	listarHistoricoPedidos()
Exibir Detalhes Pedido	C_VisualizarPedido	exibirDetalhesPedido()

c) Mediante aplicação da técnica de Análise de Caso de Uso e identificação dos eventos de sistema, modele uma VCP (Visão de Classes Participantes) com categorização BCE para o CSU04. Cabe ressaltar que a classe de controle deve apresentar as devidas operações e as classes de entidade seus atributos. A navegabilidade e as multiplicidades dos relacionamentos devem ser exibidas. (0,5 ponto).



[https://drive.google.com/file/d/1BGPWqP8FthY5fO62HvlSdDPjP-8fLFFt/view?usp=drive_link]

Explicação:

O diagrama de classes do caso de uso "Visualizar Pedido" descreve um fluxo iniciado pela autenticação do Cliente. O processo começa na TelaLogin, onde o usuário informa e-mail e senha. A classe controladora C_VisualizarPedidos gerencia essa etapa, chamando o método autenticar() da entidade Cliente. Se a autenticação falhar, o sistema exibe uma mensagem de erro (fluxo de exceção 4).

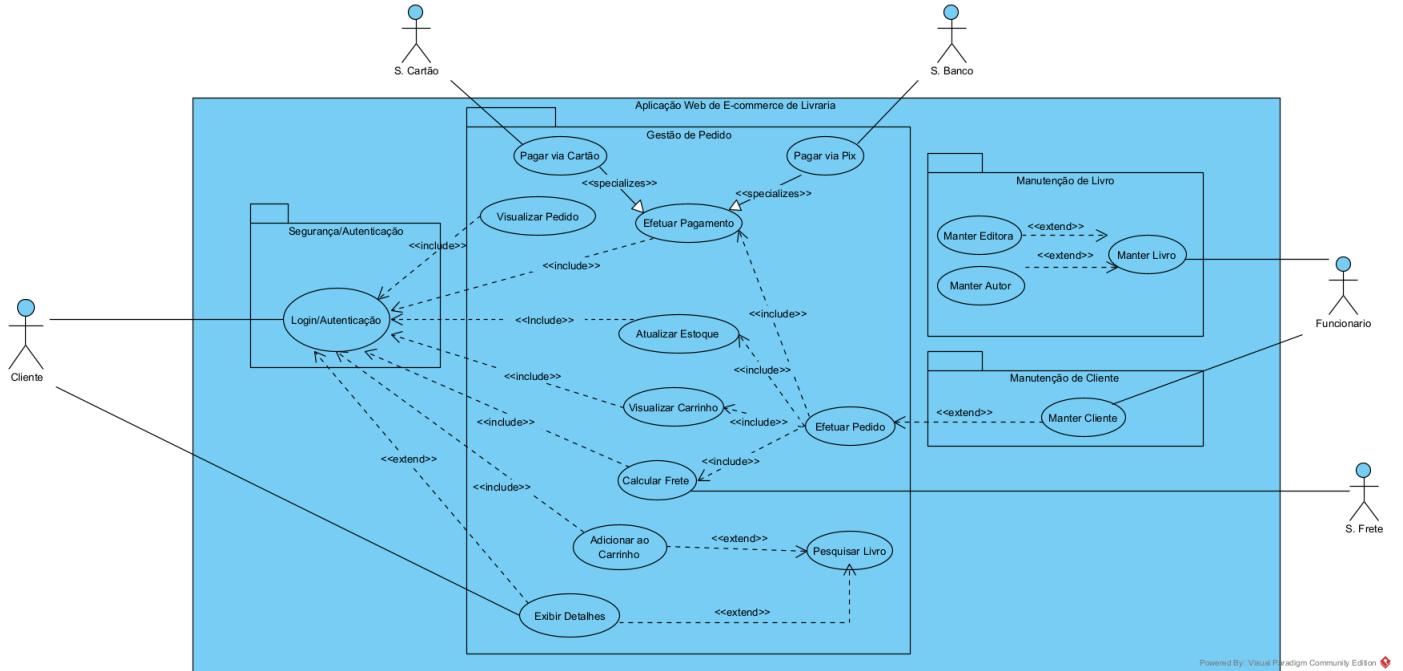
Com a autenticação bem-sucedida, o Cliente acessa a TelaHistoricoPedidos, que exibe todos os pedidos realizados. A listagem é obtida por meio da operação listarHistoricoPedidos(), executada pelo controlador, que consulta a entidade Pedido. Caso não existam pedidos, a interface exibe uma mensagem informativa (fluxo alternativo 2).

Ao selecionar um pedido específico, a interface solicita ao controlador a execução da operação exibirDetalhesPedido(), que recupera da entidade Pedido todas as informações relevantes, incluindo os ItemPedido, que contêm descrição, quantidade e preço de cada produto. Cada pedido está associado a um ou mais itens (relação 1..*).

A entidade Pedido também armazena dados essenciais como número, status, data, itens, endereço de entrega e forma de pagamento, permitindo que o cliente acompanhe o andamento da compra. Os status possíveis seguem a regra de negócio RN05, sendo: "Em Processamento", "Pagamento Pendente", "Confirmado", "Em Transporte" e "Finalizado".

Grupo 04 de Exercícios: Visões Arquiteturais com base no Modelo 4+1 (6,0 pontos)

a) A fim de ilustrar a visão “+1” (visão de cenário/caso de uso) da aplicação, modele um Diagrama de Casos de Uso com base em todos os atores e os casos de uso listados na Especificação do Sistema de Software, frisando que as descrições textuais também podem suportar tal diagrama. Os casos de uso incluídos, estendidos e especializados também devem ser representados. Ademais, os casos de uso devem ser minimamente agrupados nos seguintes pacotes: Gestão de Pedido, Manutenção de Livro e Manutenção de Cliente. Explique de modo objetivo sua representação diagramática em aproximadamente cinco linhas. (0,5 ponto)

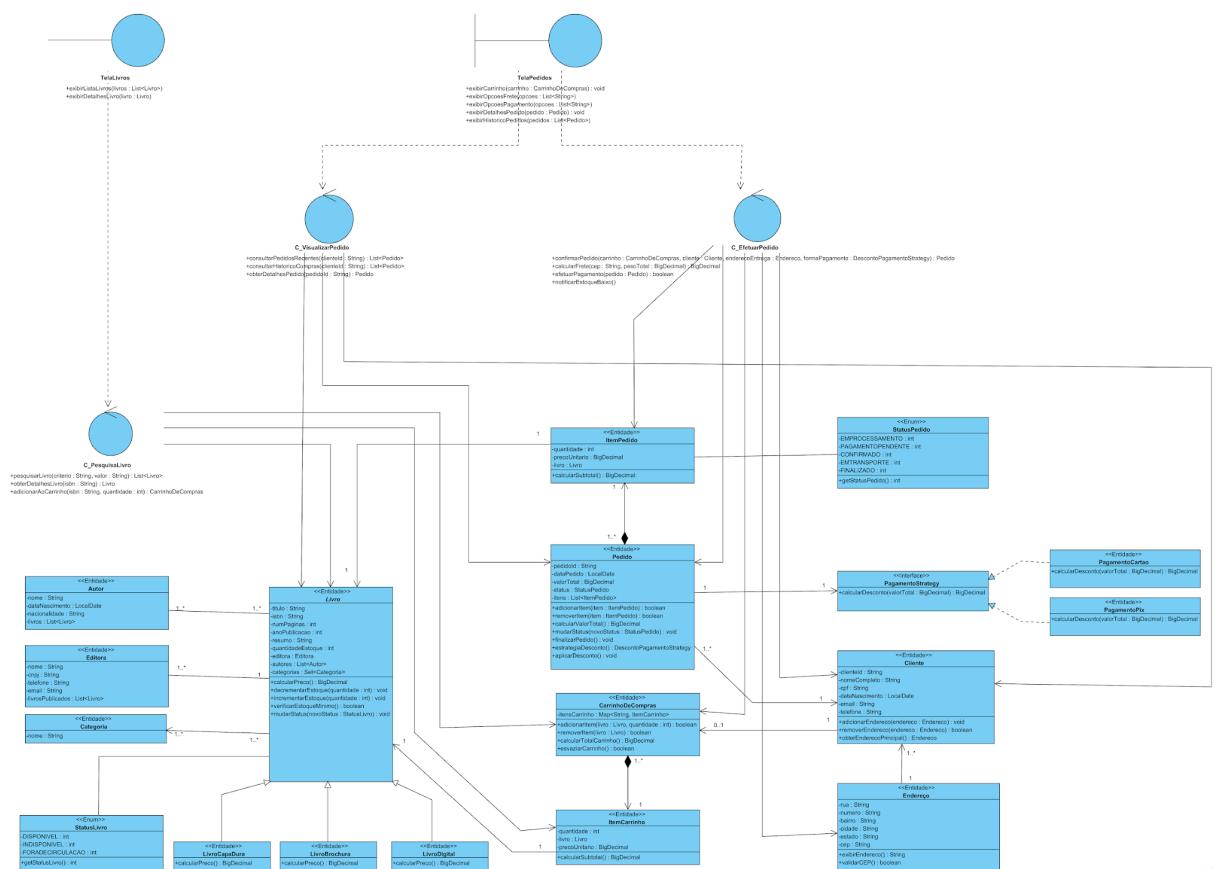


[https://drive.google.com/file/d/1c_F_yYouQDFe-bWKo7VcrCEGNueQbSK7/view?usp=drive_link]

Explicação:

Este diagrama de casos de uso ilustra as funcionalidades do sistema de e-commerce de livraria e suas interações com os atores. Os casos de uso estão organizados em pacotes de domínio (Segurança/Autenticação, Gestão de Pedido, Manutenção de Livro, Manutenção de Cliente), promovendo a separação de responsabilidades. No centro, temos o Cliente, que realiza atividades essenciais como pesquisar, adicionar livros ao carrinho e efetuar pedidos, que por sua vez envolvem pagamentos (via cartão ou Pix) e a atualização do estoque, além de processos de login e segurança. Em paralelo, o Funcionário gerencia o catálogo de livros e os dados dos clientes, garantindo a integridade do sistema. A imagem delineia claramente as dependências entre as funcionalidades através das relações <<include>> (para inclusões obrigatórias, como o cálculo de frete no pedido) e <<extend>> (para extensões opcionais, como a exibição de detalhes do livro após uma pesquisa), enquanto o <<specialize>> mostra as diferentes formas de pagamento.

b) Visando ilustrar a visão arquitetural lógica (estrutural) da aplicação, modele um diagrama de classes de projeto com categorização BCE ou DDD ou MVC, reunindo todas as VCPs refinadas. Tal diagrama de classes deve representar os detalhes das classes de controle, especificamente operações e seus parâmetros, quando houver, e os detalhes das classes de entidade, especificamente atributos e seus tipos. O diagrama deve exibir os relacionamentos com navegabilidade (unidirecional ou bidirecional) e tipo de dependência (estrutural ou não estrutural), as classes parametrizadas para representar as coleções. Mediante abstração da RN02, modele adequadamente a relação de gen/espec e o princípio de polimorfismo. Mediante abstração da RN04, modele adequadamente o padrão de projeto Strategy com interface(s). Explique de modo objetivo sua representação diagramática em aproximadamente cinco linhas, enfatizando a solução para atender a RN02 e a RN04, além de justificar o motivo de cada coleção escolhida, conforme Tabela 4a. A alocação de responsabilidades (atributos e operações) deve atender aos princípios de coesão e acoplamento de maneira equilibrada, desse modo, justifique a classe mais e a menos coesa, como também, a classe mais e a menos acoplada, de acordo com a Tabela 4b. (1,5 ponto).



https://drive.google.com/file/d/1IT295a8JkCJJJa2zI32ZsGHfwcrPdIllog/view?usp=drive_link

Explicação:

O diagrama de classes apresentado, resultado do refinamento das VCPs e da aplicação de princípios de design, ilustra a estrutura lógica da aplicação de e-commerce de livraria, adotando a categorização BCE (Boundary-Control-Entity) para organizar as responsabilidades das classes. As entidades representam os conceitos persistentes do domínio, como Livro, Cliente, Pedido, Autor, Editora, Categoria, Endereço, ItemPedido, ItemCarrinho, CarrinhoDeCompras, além dos enums StatusLivro e StatusPedido. Essas classes encapsulam dados e comportamentos relevantes ao negócio. As classes de controle, como ControlePesquisarLivro, ControleEfetuarPedido, ControleManterCliente, ControleVisualizarPedido e ControleManterLivro, orquestram a lógica de negócio dos casos de uso, mediando a comunicação entre as interfaces (representadas pelas classes de fronteira, como TelaLivro e TelaPedido, ou adaptadores para sistemas externos como Sistema de Frete e Sistema de Cartão) e as entidades. Embora os estereótipos <> não estejam explicitamente presentes, essas interfaces são previstas na arquitetura.

As entidades possuem atributos com visibilidade privada e tipos de dados adequados, como String, BigDecimal, LocalDate e int, com o uso apropriado de BigDecimal para valores monetários, garantindo

precisão. A convenção do atributo id: String em todas as entidades reforça a identificação única e a persistência. As operações refletem comportamentos do domínio, como calcularPreco() em Livro, adicionarItem() em Pedido e calcularSubtotal() em ItemPedido. As classes de controle, embora não detalhadas no diagrama, desempenham papel central na coordenação dos fluxos de cada caso de uso. Os relacionamentos entre as classes são descritos com multiplicidades e setas de navegabilidade, indicando os sentidos das associações. Há uso de coleções parametrizadas, como List<Autor> em Livro, List<ItemPedido> em Pedido e Map<String, ItemCarrinho> em CarrinhoDeCompras, escolhidas conforme os requisitos de ordem, unicidade ou eficiência de acesso. A bidirecionalidade entre algumas classes, como Livro e Autor/Editora/Categoria, e entre Pedido e Cliente, indica que ambas podem navegar entre si, geralmente por meio de atributos de coleção.

O modelo atende a regras de negócio importantes. A RN02, que trata de generalização/especialização e polimorfismo, é atendida pela hierarquia de Livro (classe abstrata) e suas subclasses concretas LivroCapaDura, LivroBrochura e LivroDigital, com o método calcularPreco() implementado de forma polimórfica para refletir as particularidades de cada formato. A RN04 é implementada com o uso do Padrão Strategy, por meio da interface DescontoPagamentoStrategy, com implementações concretas nas classes PagamentoCartao e PagamentoPix. A aplicação do desconto é delegada à interface, permitindo a extensão para novos tipos de pagamento sem alterar o código existente.

Quanto à coesão e acoplamento, há uma distribuição equilibrada de responsabilidades. Classes como Endereço e os enums StatusLivro e StatusPedido apresentam alta coesão e baixo acoplamento, com responsabilidades únicas e poucas dependências. As entidades principais, como Livro, Cliente e Pedido, também demonstram forte coesão ao encapsularem integralmente os aspectos de seu domínio. Em contrapartida, o ControleEfetuarPedido possui menor coesão e maior acoplamento, por interagir com diversas entidades e serviços externos no processo de finalização do pedido, o que é aceitável em controladores de casos de uso mais complexos, conforme discutido na literatura.

Por fim, o diagrama reflete aderência aos princípios SOLID. O SRP (Single Responsibility Principle) é visível na alta coesão das entidades; o OCP (Open-Closed Principle) e o DIP (Dependency Inversion Principle) são aplicados na estratégia de desconto da RN04; o LSP (Liskov Substitution Principle) é atendido pela hierarquia de Livro; e o ISP (Interface Segregation Principle) está implícito na separação de responsabilidades entre entidades, controladores e fronteiras. Esses elementos demonstram um design orientado a objetos bem fundamentado e preparado para manutenção e extensão.

Tabela 4a: Classes Parametrizadas (Coleções)

Classe Consumidora	Coleção com seu tipo de elemento	Justificativa para escolha da coleção
Livro	List<Autor>	Eu posso ter um conjunto de autores publicando um livro e List permite duplicatas (se um autor for listado de forma diferente ou coautoria ou pseudônimo) e mantém a ordem de inserção.
Livro	Set<Categoria>	Um livro pode pertencer a várias categorias, mas uma categoria não deve se repetir para o mesmo livro.
Autor	List<Livro>	Um autor pode ter escrito vários livros, e a List é adequada para armazenar essa coleção de obras em ordem cronológica(ordem de inserção)
Editora	List<Livro>	Uma editora publica vários livros. A List permite armazenar o catálogo de livros da editora, mantendo a ordem.
Cliente	List<Endereco>	Um cliente pode ter múltiplos endereços (residencial, comercial, cobrança), e a ordem ou a capacidade de adicionar/remover específicos (via índice ou objeto) torna a List uma boa escolha.
Cliente	List<Pedido>	Um cliente pode realizar vários pedidos ao longo do tempo.
Pedido	List<ItemPedido>	A List mantém a ordem em que os itens foram adicionados ao pedido e permite duplicatas de objetos ItemPedido. Foco no registro histórico sequencial dos itens da compra finalizada

CarrinhoDeCompras	Map<String, ItemCarrinho>	Como o foco é na manipulação eficiente e única de itens para uma compra em progresso, com acesso rápido por identificador, o Map é ideal para um carrinho. Permite acesso rápido (adição, remoção, atualização de quantidade) a um ItemCarrinho específico sem precisar percorrer toda a coleção, e garante que cada livro (chave) apareça apenas uma vez no carrinho, atualizando sua quantidade.
-------------------	---------------------------	--

Tabela 4b: Acoplamento e Coesão

Princípios	Classe	Justificativa
Classe mais coesa	StatusLivro / StatusPedido (Enums)	Uma classe Enum representa um conjunto fixo de valores constantes e relacionados, com a única responsabilidade de definir esses estados discretos e, eventualmente, fornecer métodos utilitários simples. Não contém lógica de negócio nem gerencia dados mutáveis, servindo exclusivamente para tipagem e representação clara de valores possíveis para um determinado contexto.
Classe menos coesa	C_EfetuarPedido	Da maneira que foi modelado o exercício torna essa classe menos coesa por sua natureza de orquestradora central do processo de finalização de um pedido. Ela coordena diversas responsabilidades de diferentes domínios, como gerenciamento de carrinho, cálculo de frete, aplicação de pagamento e atualização de estoque. Essa ampla gama de interações com múltiplas entidades e serviços, embora necessária, confere a ela múltiplos motivos para mudar, impactando sua coesão interna.
Classe mais acoplada	C_EfetuarPedido	Devido à sua função de orquestrador central do caso de uso "Efetuar Pedido", esta classe interage diretamente com o maior número de outras classes (entidades e interface), tornando-a a mais dependente de outros componentes do sistema.
Classe menos acoplada	StatusLivro / StatusPedido (Enums)	Uma classe Enum é praticamente auto-suficiente. Ela não depende de outras classes para sua existência ou funcionalidade. Outras classes (Livro, Pedido) dependem dela para seus atributos de status, mas a dependência é unilateral e mínima, pois a Enum não invoca métodos ou acessa dados de outras classes.

Classe	Coesão	Acoplamento
Autor	Muito alta	Baixo
Editora	Muito alta	Baixo
Livro	Muito alta	Médio-Baixo
Categoria	Muito alta	Baixo
StatusLivro (Enum)	Máxima	Mínimo
Cliente	Muito alta	Baixo-Médio
Endereco	Máxima	Mínimo
Pedido	Alta	Médio-Alto
StatusPedido (Enum)	Máxima	Mínimo
ItemPedido	Muito alta	Baixo-Médio

CarrinhoDeCompras	Alta	Médio
ItemCarrinho	Muito alta	Baixo-Médio
DescontoPagamentoStrategy (Interface)	Máxima	Mínimo
PagamentoCartao	Máxima	Baixo
PagamentoPix	Máxima	Baixo
ControlePesquisarLivro	Média-Alta	Médio-Alto
ControleManterCliente	Alta	Médio
ControleManterLivro	Alta	Médio-Alto
ControleVisualizarPedido	Alta	Médio-Alto
ControleEfetuarPedido	Baixa	Alto

Ainda levando em consideração o diagrama de classes de projeto, justifique no Quadro 4a de forma clara e condizente se as classes atendem ou não aos princípios SOLID. (1,0 ponto)

Quadro 4a: Análise dos Princípios SOLID

Princípio	Justificativa (de forma clara)
SRP	As Classes Autor, Editora, Livro, Cliente, ItemPedido, CarrinhoDeCompras, ItemCarrinho aderem bem ao SRP. Cada uma possui uma única e bem definida responsabilidade e encapsula seus dados e comportamentos específicos de domínio.
OCP	O Princípio Aberto/Fechado é evidenciado por meio de duas estruturas: O uso de herança e polimorfismo com a classe abstrata Livro e suas subclasses permite a adição de novos formatos sem alterar o código existente. A interface DescontoPagamentoStrategy permite introduzir novas regras de desconto (como para boleto) por meio de novas classes, sem modificar a lógica da classe Pedido.
LSP	As subclasses de Livro (LivroCapaDura, LivroBrochura, LivroDigital) podem substituir a superclasse Livro em qualquer contexto (ex: em uma lista de Livros), e o método calcularPreco() funcionará corretamente, pois todas aderem ao contrato da superclasse, garantindo a coerência do sistema
ISP	A interface DescontoPagamentoStrategy atende ao ISP. Ela é coesa por conter apenas o método calcularDesconto(), o único necessário para suas implementações (PagamentoCartao, PagamentoPix) e para a classe Pedido que a utiliza. Isso evita forçar classes a implementar métodos irrelevantes
DIP	O diagrama atende ao DIP. A classe Pedido (módulo de alto nível) depende da interface DescontoPagamentoStrategy (uma abstração), e não das classes concretas PagamentoCartao ou PagamentoPix (módulos de baixo nível). Isso assegura que ambos dependem de abstrações, facilitando a flexibilidade e testabilidade do sistema.

- c) Com o intuito de ilustrar a visão arquitetural de processo (comportamental) e refinar os aspectos dinâmicos e comportamentais do sistema, modele um diagrama de sequência com categorização BCE para o CSU02, representando também dois quadros de interação para fazer referência ao diagrama principal, um para o caso de uso “Calcular Frete” e outro para o caso de uso “Efetuar Pagamento” e um terceiro quadro de interação para representar um fluxo de controle para aplicar o desconto no valor total do pedido conforme

forma de pagamento. Vale ressaltar que os multiobjetos também devem ser representados adequadamente. Explique de modo objetivo sua representação diagramática em aproximadamente cinco linhas. (1,5 ponto).

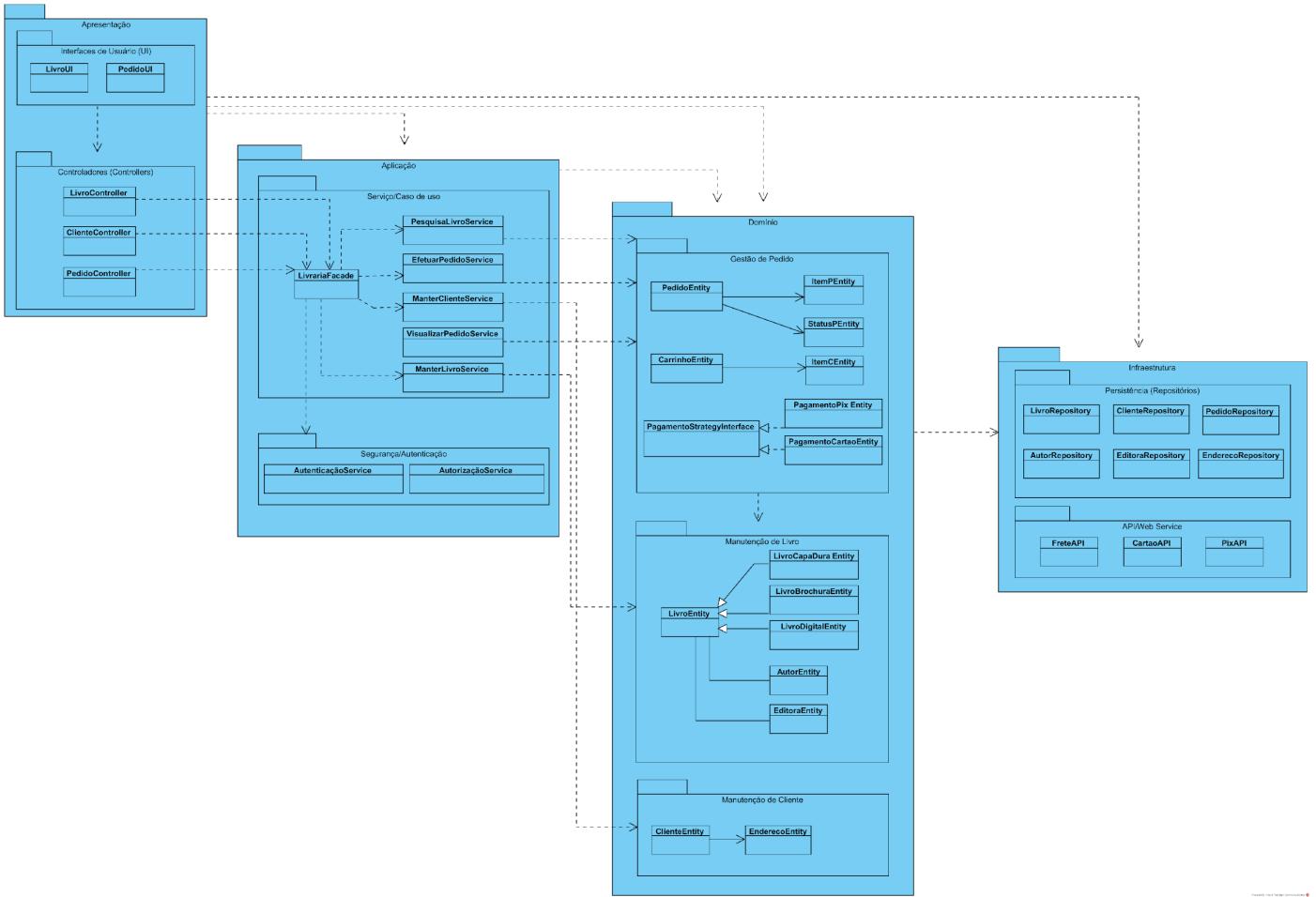
https://drive.google.com/file/d/1awY6wdgdywPt7XYip56mA5dc-YeU4ntM/view?usp=drive_link

Explicação:

Este diagrama de sequência representa o fluxo do caso de uso "Efetuar Pedido". O processo inicia quando o Cliente confirma a intenção de compra (mensagem 1), levando o sistema a buscar os itens do Carrinho e autenticar o usuário (mensagens 1.1 a 2.2). Em seguida, a complexidade do cálculo de frete é abstraída pelo quadro de referência ref: Calcular Frete, que representa a interação com o ator externo Sistema de Frete.

O ponto central do fluxo é a finalização do pedido (mensagem 4), que invoca o caso de uso ref: Efetuar Pagamento. Este quadro encapsula toda a lógica de processamento, seja por Cartão ou Pix, e a aplicação das regras de desconto correspondentes. Após o retorno bem-sucedido do pagamento, um fragmento alt [pagamentoAprovado] governa a etapa de confirmação: o objeto Pedido é criado (mensagem 4.3), e a criação dos itens é representada de forma concisa pela mensagem <<create>> para o multiobjeto (mensagem 4.6). Finalmente, o estoque é atualizado, o carrinho é limpo e o cliente recebe a confirmação, enquanto o cenário de falha no pagamento direciona para uma mensagem de erro, completando o ciclo.

- d) A fim de representar a visão arquitetural lógica e de implementação, combinando padrões de arquitetura Cliente-Servidor e Camadas e diretrizes de Clean Architecture, represente as camadas lógicas por meio de pacotes e faça a alocação de cada classe no devido pacote. Por causa da quantidade versus legibilidade das notações, as classes alocadas devem exibir apropriadamente seu nome e estereótipo. Além disso, este diagrama deve representar minimamente: i) uma camada de apresentação com duas partições (pacotes), sendo uma para as interfaces de usuário e outra para os controladores; ii) uma camada de aplicação com duas partições (pacotes), sendo uma para controlar a lógica específica da aplicação (serviço/caso de uso) e outra para controlar a autenticação e autorização de usuário; iii) uma camada de domínio (negócio) com no mínimo três partições (pacotes) de classes de domínio (entidades), sendo uma para as classes relacionadas à gestão de pedido, outra para as classes referentes à manutenção de livro e uma terceira para as classes relacionadas à manutenção de cliente e; iv) uma camada de infraestrutura com duas partições (pacotes), sendo uma para as classes de persistência (repositórios) e outra para os sistemas externos (API/Web Service). Complemente tal visão arquitetural com a modelagem do padrão de projeto Façade e seus devidos serviços. (1,5 ponto).



[https://drive.google.com/file/d/19uSNm0cVEX63gIHl8vSbvZ6HnPJ50kvI/view?usp=drive_link]

Explicação:

Este diagrama de pacotes representa a visão arquitetural lógica e de implementação da aplicação de e-commerce de livraria, combinando padrões Cliente-Servidor e Camadas, alinhado com diretrizes de Clean Architecture. As camadas lógicas são modeladas por pacotes hierárquicos, e cada classe é alocada ao seu devido pacote, exibindo apropriadamente seu nome e estereótipo para legibilidade.

A arquitetura representa minimamente:

- Uma camada de Apresentação, com partições para Interfaces de Usuário (UI) (contendo classes como LivroUI, PedidoUI) e Controladores (Controllers) (como LivroController, PedidoController, ClienteController).
- Uma camada de Aplicação, subdividida em Serviço/Caso de uso (onde residem os serviços de aplicação como PesquisaLivroService, EfetuarPedidoService, ManterClienteService, VisualizarPedidoService, ManterLivroService) e Autenticação/Autorização (com AutenticacaoService, AutorizacaoService).
- Uma camada de Domínio, com três partições para classes de entidade: Gestão de Pedido (incluindo PedidoEntity, CarrinhoEntity e o Padrão Strategy para pagamento DescontoPagamentoStrategy), Manutenção de Livro (com LivroEntity e suas especializações, AutorEntity, EditoraEntity, CategoriaEntity), e Manutenção do Cliente (com ClienteEntity, EnderecoEntity).
- E uma camada de Infraestrutura, contendo Persistência (Repositórios) (com classes como LivroRepository, ClienteRepository) e API/Web Service (com FreteAPI, CartaoAPI, PixAPI para sistemas externos).

Complementarmente, o padrão de projeto LivrariaFacade foi modelado dentro da camada de Aplicação (Serviço/Caso de uso), expondo serviços de alto nível para simplificar a interação das camadas superiores.

Grupo 05 de Exercícios (Opcional): Apresentação do Grupo 4 de Exercícios (1,0 ponto na Nota 2)

- a) Gravar uma apresentação de aproximadamente 15 minutos para explicar os diagramas do Grupo 4 de Exercícios.

[Link de acesso ao vídeo]