

4-Etapa de Construção-Projeto

4.1-UML-Diagramas de interação

São modelos que descrevem como grupos de objetos colaboram em algum comportamento. Tipicamente capturam o comportamento de um único caso de uso e mostram um certo número de classes/objetos e as mensagens que são passadas entre eles no escopo do caso de uso. Existem dois tipos de diagramas de interação: diagramas de sequência e diagramas de colaboração. Eles são semanticamente idênticos.

4.1.1 – Diagramas de sequência

Representam as interações entre classes/objetos em uma disposição horizontal dos mesmos. As classes/objetos pertencentes ao escopo de um caso de uso são representados no topo de uma linha vertical pontilhada, como mostrado na Figura 4.1. A linha vertical pontilhada é chamada linha da vida. Ela representa a vida do objeto durante a interação. Cada mensagem é representada por uma seta entre as linhas da vida de dois objetos. A ordem em que estas mensagens ocorrem é mostrada de cima para baixo. Cada mensagem é rotulada no mínimo com seu nome, podendo ser incluídos argumentos e informação de controle. Quando existe uma seta de um objeto para ele mesmo, denominamos auto-delegação, significando que uma operação do próprio objeto é invocada.. Na Figura 4.1 temos algumas condições que indicam quando uma mensagem é enviada, ou seja, a mensagem só é enviada quando a condição for *true*. A figura mostra também um marcador de iteração (*) o qual mostra que a mensagem é enviada várias vezes.

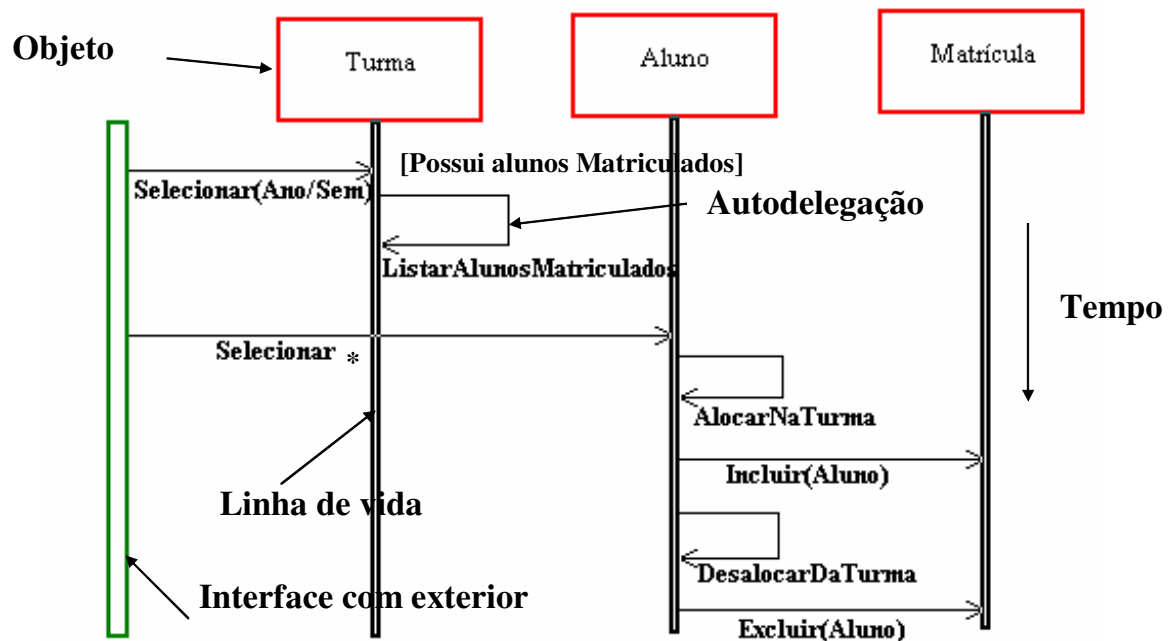


Figura 4.1-Notação para diagramas de seqüência

4.1.2-Diagramas de colaboração

Neste diagrama temos os objetos mostrados como ícones. As setas indicam as mensagens enviadas no caso de utilização.

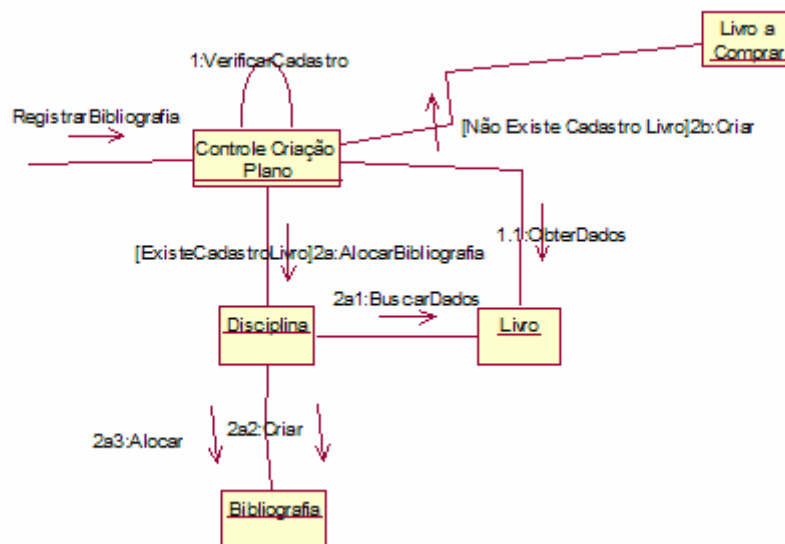


Figura 4.2-Notação para diagramas de colaboração

Com relação à numeração da seqüência de mensagens, algumas observações devem ser feitas:

- O número de seqüência da mensagem a precede e é seguido por ":";
- Quando uma mensagem está subordinada a outra (ou seja, é enviada de dentro da operação solicitada pela mensagem-*mensagem aninhada*), a numeração possui o formato *número da mensagem principal.número da mensagem subordinada*. Por exemplo, na Figura 4.2 1.1:*ObterDados* está subordinada a 1:*VerificarCadastro*.
- Em caminhos condicionais mutuamente exclusivos, aloca-se uma letra após o número de seqüência de cada mensagem condicional. Convencionalmente, a primeira letra é *a*, na outra condição *b* e assim por diante. Na Figura 4.2 temos dois caminhos condicionais mutuamente exclusivos: [Existe cadastro livro] e [Não Existe Cadastro Livro] que condicionam as mensagens numeradas respectivamente por *2a:* e *2b:*
- Se houver mais de uma mensagem derivada de uma condição em um mesmo nível, colocamos a letra entre parêntesis, precedida pelo número sequencial. Por exemplo, a mensagem 1 (1a) é emitida antes de 1(2a) e ambas estão dentro da mesma condição.

4.2-Atividade definir operações

Esta atividade consiste em definirem-se as operações que farão parte de cada classe de controle do sistema. Via de regra, classes de controle controlam casos de uso e suas operações são derivadas da lista de eventos do caso de uso. Uma operação comporta um certo número de eventos de um caso de uso que descrevem as regras de negócio implementadas pela operação.

Suponha como exemplo, a descrição do seguinte caso de uso chamado *Pagar Contas*:

Ação do Usuário	Resposta do Sistema
1. Usuário entra com AGÊNCIA, CONTA e SENHA.	2. Se os dados informados estiverem corretos, exibe a tela para entrada do código de barras, caso contrário exibe mensagem de erro e retorna à tela inicial.
3. Usuário entra com código de barras.	4. Sistema verifica se os números digitados estão corretos. Em caso afirmativo obtém o nome do credor, valor e data de vencimento. 5. Sistema verifica se a data de vencimento é menor que a data de hoje. Se não for, exibe uma mensagem de erro e volta à tela inicial. Se for mostra uma tela com o nome do credor, valor e data de vencimento da conta e aguarda confirmação.
6. Usuário confirma dados.	7. Sistema verifica se o cliente possui saldo suficiente. Se tiver registra pagamento da conta, debitando a

	mesma na conta do usuário e creditando na conta do credor. Em seguida um recibo é impresso.
--	---

Os eventos 1 e 2 referem-se à autenticação do usuário. Portanto comportarão uma operação a qual chamaremos *FazerAutenticação*.

Os eventos 3,4 e 5 referem-se ao processo de obter, verificar e depurar os dados de uma conta. Chamaremos a operação correspondente de *ObterDadosConta*

Finalmente, as operações 6 e 7 completam o caso de uso, gravando as informações da conta que foi paga. Chama-la-emos *RegistrarPagamento*.

A seqüência em que as operações são invocadas é mostrada em diagrama de seqüência o qual denominamos *diagrama de seqüência de operações*. As mensagens correspondentes à chamada das operações saem da interface e entram na classe de controle à qual estão alocados. Portanto, um caso de uso sempre vai possuir uma classe de controle capaz de controlar seu fluxo de eventos. Uma classe de controle pode ser comum a mais de um caso de uso, mas a situação mais comum é um caso de uso para cada classe de controle.

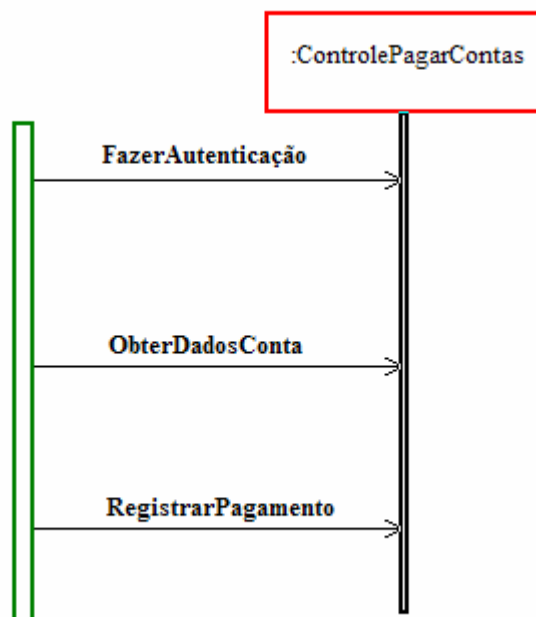


Figura 4.3-Diagrama de seqüência de operações do caso de uso Pagar Contas

4.3-Atividade Detalhar Operações

Nesta atividade, cada operação definida na atividade anterior é minuciosamente detalhada no *Descritivo de Operação*. Ele descreve o que a operação se compromete a realizar. Um Descritivo de Operação possui o seguinte formato:

Nome: nome da operação e parâmetros

Responsabilidades: descrição informal das responsabilidades que a operação deve cumprir.

Referências cruzadas: Referências a funções do sistema, use cases, etc.

Notas: notas de projeto, algoritmos, etc.

Exceções: descrição de caminhos de exceções

Saída: saídas esperadas desta operação

Pré-condições: hipóteses ou assertivas acerca do estado do sistema antes da execução da operação.

Pós-condições: conjunto de situações que descrevem o estado do sistema após a execução da operação. Categorias:

- *Criação de instância:* a execução da operação acarreta a criação instâncias de classes.
- *Modificação de atributos:* a execução da operação acarreta a alteração de atributos de instâncias.
- *Criação de associação:* a execução da operação acarreta a associação entre instâncias

No exemplo, temos as seguintes Descrições de Operações;

Nome: FazerAutenticação (conta: integer, agencia: integer, senha: integer)

Responsabilidades:

- Verificar dados de autenticação.
- Exibir tela para entrada de cód. De barras.

Notas: Usar acesso muito rápido a BD.

Referências cruzadas: Caso de uso: PagarContas;

Exceções: Se a senha, a conta corrente e/ou a agência não forem válidos, indicar erro.

Pré-condições: Senha, conta corrente e agência digitadas.

Pós-condições:

- Criar Instância de usuário
- Criar Instância de conta corrente de usuário
- Associar usuário à sua conta corrente

Nome: ObterDadosConta (CodBarras: integer)

Responsabilidades:

- Verificar se os números digitados no cód. De barras estão corretos.
- Obter o nome do credor, valor e data de vencimento.
- Verificar se a data de vencimento é menor que a data de hoje.
- Mostrar uma tela com o nome do credor, valor e data de vencimento da conta.

Notas: Usar acesso muito rápido a BD.

Referências cruzadas: Caso de uso: PagarContas;

Exceções: Se o UPC não for válido e/ou a data de vencimento for maior que a data de hoje, indicar erro.

-Pré-condições: Código de barras digitado.

-Pós-condição:

- Criar Instância de credor.
- Criar Instância de conta corrente de credor.
- Associar a instância de credor à instância da conta corrente do credor

Nome: RegistrarPagamento ()

Responsabilidades:

- Verificar se o cliente possui saldo suficiente.
- Registrar pagamento da conta,
- Debitar a mesma na conta do usuário
- Creditar na conta do credor.
- Imprimir recibo

Notas: Usar acesso muito rápido a BD.

Referências cruzadas: Caso de uso: PagarContas;

Exceções: Se houver problemas no crédito da conta do credor ou débito na conta do usuário, indicar erro.

Pré-condições: Dados de usuário e código de barras devem estar corretos até o momento.

Pós-condições:

- Criar instância de Pagamento
- Criar instância de Recibo
- Atualizar campo saldo da conta do usuário
- Atualizar campo saldo da conta do credor

4.4-Atividade Produzir Diagramas de Colaboração

4.4.1-Gerar a primeira versão

1-Criar um diagrama de colaboração para cada operação de sistema, tendo a mesma como mensagem inicial.

2-Usando as responsabilidades e pós condições das Descrições de Operações e a descrição do caso de uso como referência, desenhe um sistema de objetos que interagem entre si para executar tarefas.

Para as Descrições de Operações temos os seguintes diagramas de colaboração:

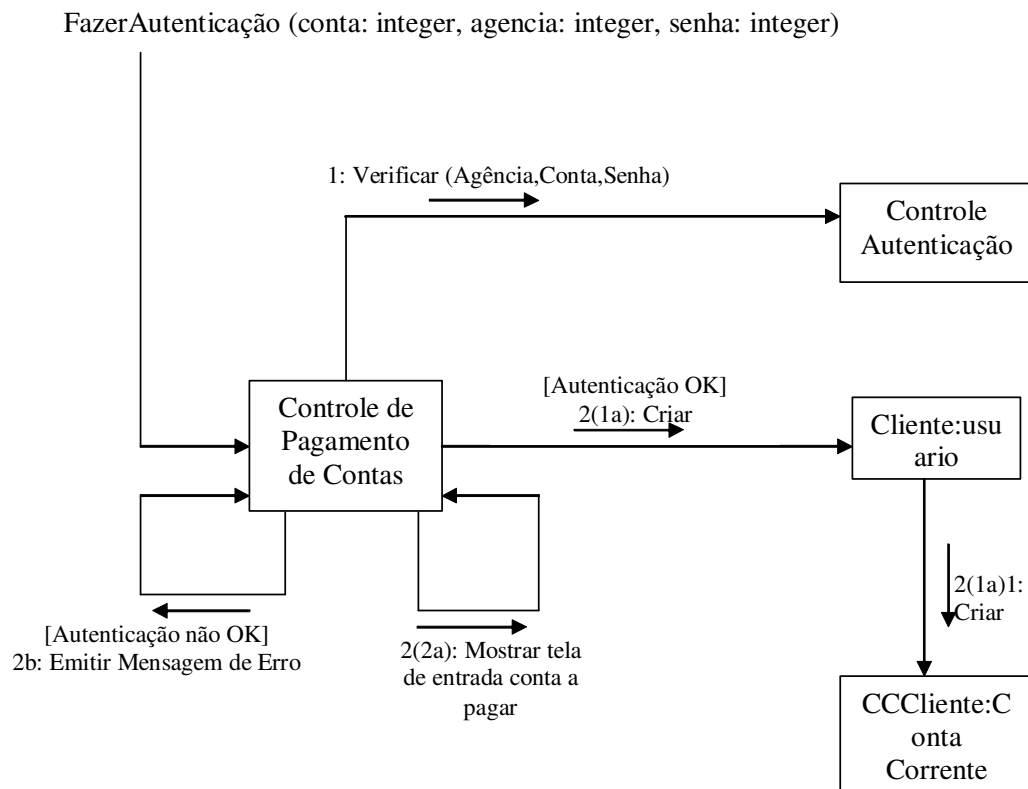


Figura 4.4-Diagrama de Colaboração da operação FazerAutenticação()

ObterDadosConta (CodBarras: integer)

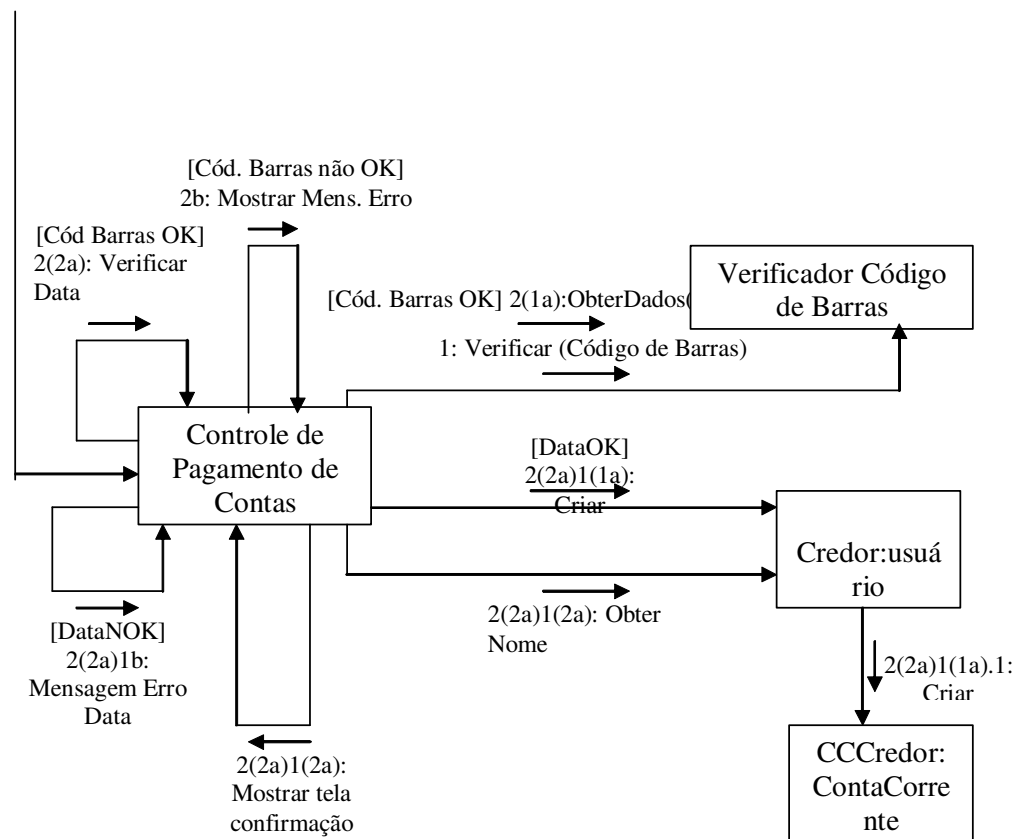


Figura 4.5-Diagrama de Colaboração da operação `ObterDadosConta()`

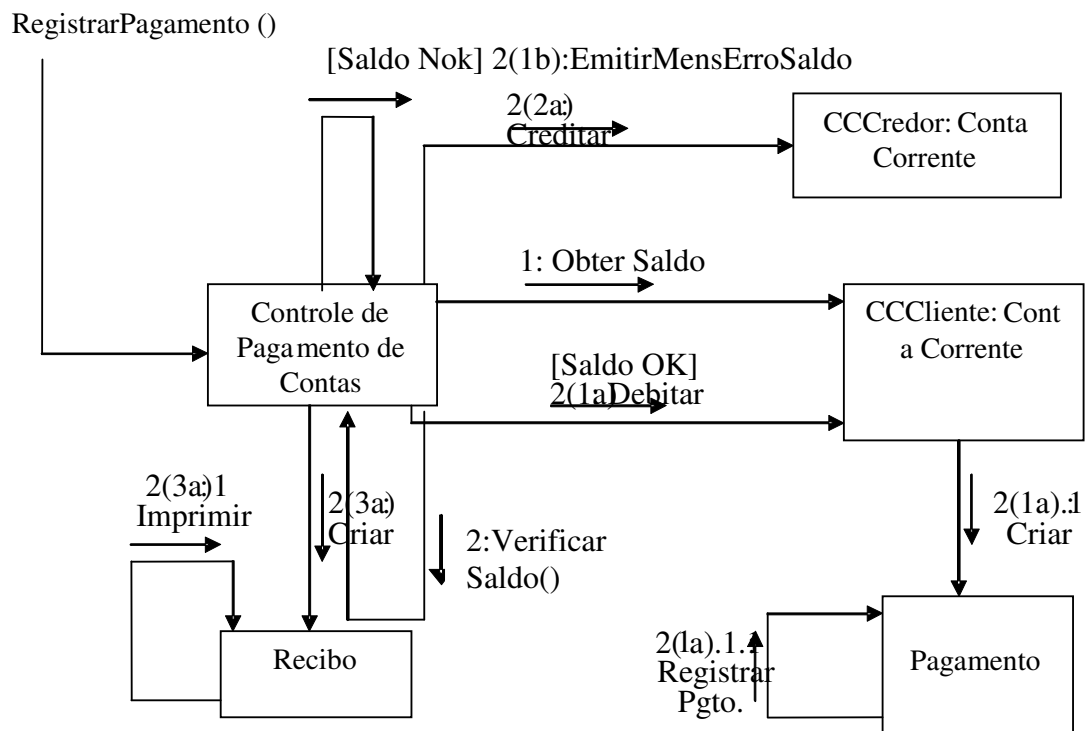


Figura 4.6-Diagrama de Colaboração da operação RegistrarPagamento()