



# Computação Orientada a Objetos II

Prof. Dr. Rodrigo Duarte Seabra

**Universidade Federal de Itajubá**  
**Bacharelado em Ciência da Computação/Sistemas de Informação**

# Diagrama de Sequência

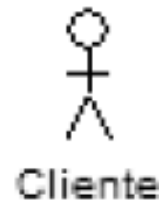
## Aula 06

Prof. Dr. Rodrigo Duarte Seabra

**Universidade Federal de Itajubá**  
**Bacharelado em Ciência da Computação/Sistemas de Informação**

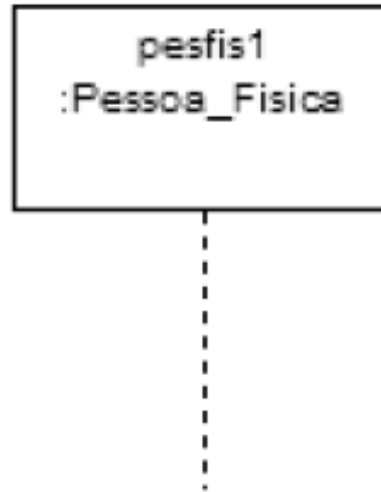
- ✓ Determina a **sequência de eventos** que ocorrem em determinado processo, identificando **quais mensagens devem ser disparadas** entre os elementos envolvidos e **em que ordem**
- ✓ **Baseia-se no diagrama de casos de uso**
- ✓ Em geral, **há um diagrama de sequência para cada caso de uso** declarado
- ✓ Valida e complementa o diagrama de classes

- ✓ Instâncias dos **atores declarados no diagrama de casos de uso**
- ✓ Representam entidades externas que interagem com o sistema e que solicitam serviços gerando eventos que iniciam processos

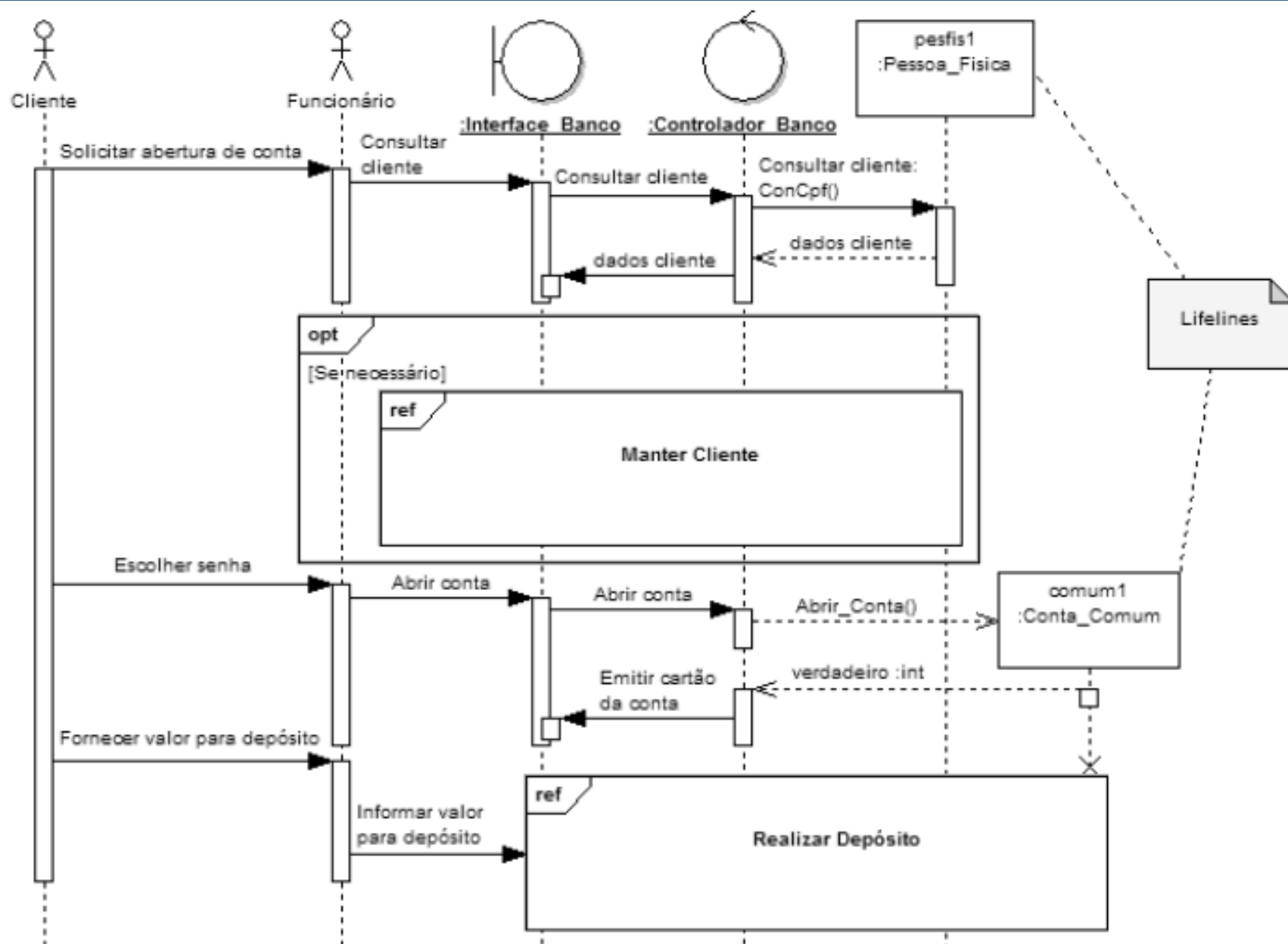


**Exemplo de ator**

- ✓ Participante individual em uma interação
- ✓ Normalmente, refere-se a uma **instância de uma classe (objeto)** que participa de uma interação



**Exemplo de Lifeline**

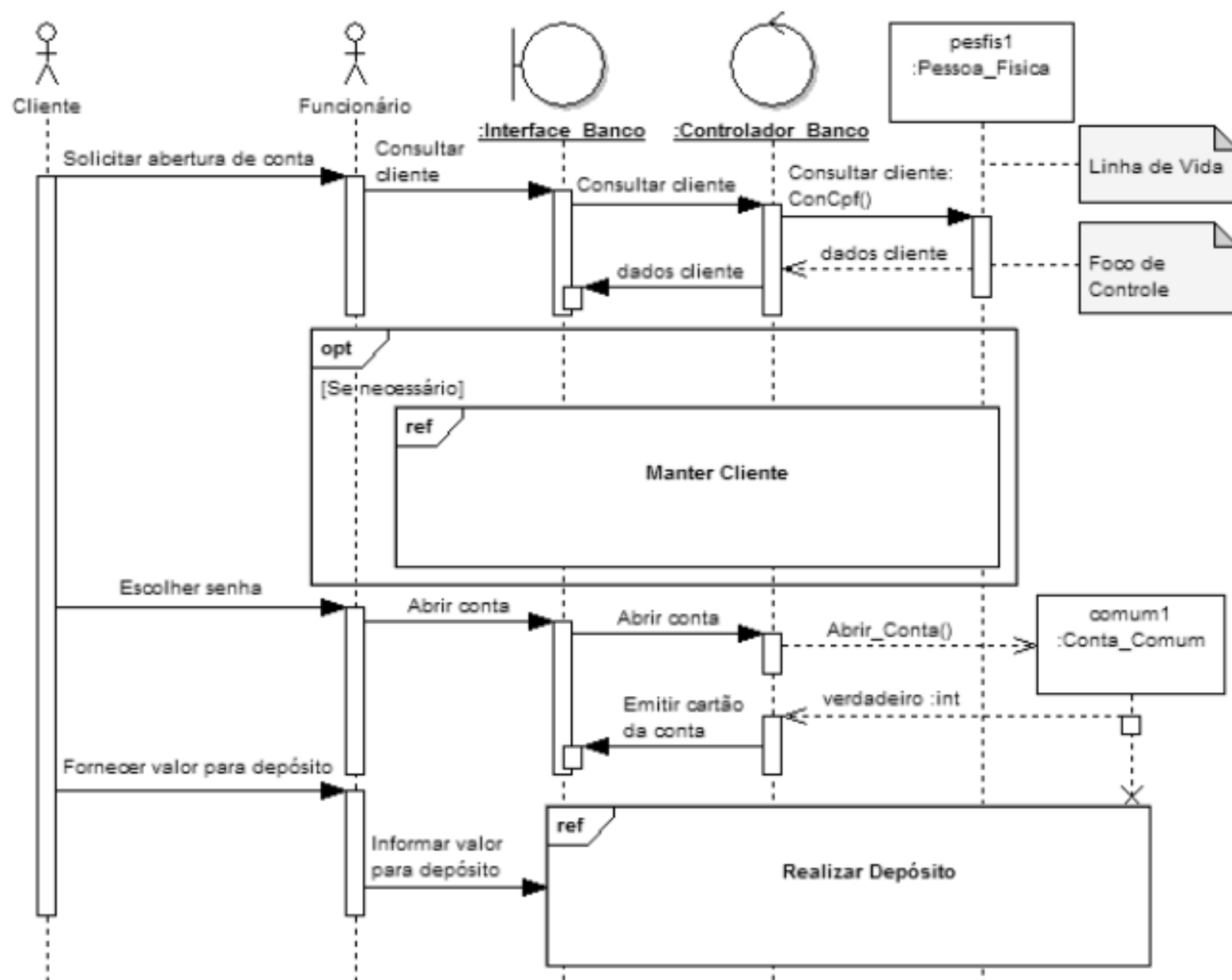


- ✓ Representa o **tempo em que um objeto (lifeline) existe** durante um processo
- ✓ Representadas por linhas finas verticais tracejadas partindo do retângulo que representa o objeto
- ✓ A linha de vida é interrompida com um “X” quando o objeto é destruído
- ✓ Se um objeto for criado ao longo de um processo, sua representação não ocorre no topo do diagrama, mas somente a partir do momento em que forem criados

- ✓ Indica os períodos em que determinado objeto está participando ativamente do processo
- ✓ Representados dentro da linha de vida de um objeto



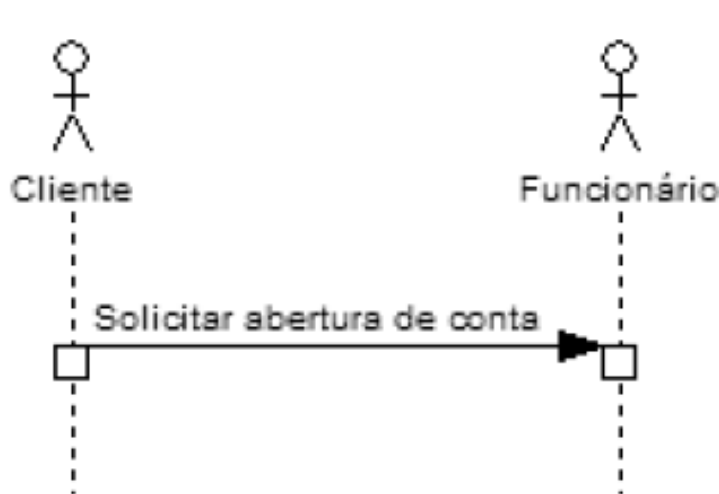
# Foco de Controle ou Ativação



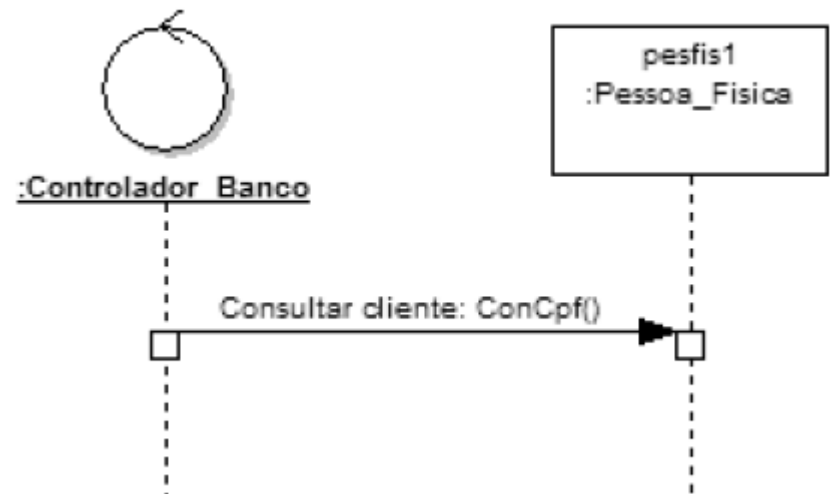
- ✓ Demonstram a **ocorrência de eventos**, que normalmente forçam a **chamada de um método** em algum dos objetos envolvidos no processo
- ✓ Podem ser disparadas entre:
  - **um ator e outro ator**
  - **um ator e um objeto**, onde um ator produz um evento que dispara um método em um objeto
  - **um objeto e outro objeto**, o que constitui a ocorrência mais comum de mensagens
  - **um objeto e um ator**, o que normalmente ocorre quando um objeto envia uma mensagem de retorno em resposta à chamada de um método solicitado, contendo seus resultados

# Mensagens ou Estímulos

- ✓ Representadas por **linhas entre dois componentes, contendo setas** indicando qual componente enviou a mensagem e qual a recebeu
- ✓ Apresentadas na posição horizontal entre as linhas de vida dos componentes e sua ordem sequencial é demonstrada de cima para baixo
- ✓ Os textos contidos nas mensagens identificam qual evento ocorreu e forçou o envio da mensagem e qual método foi chamado

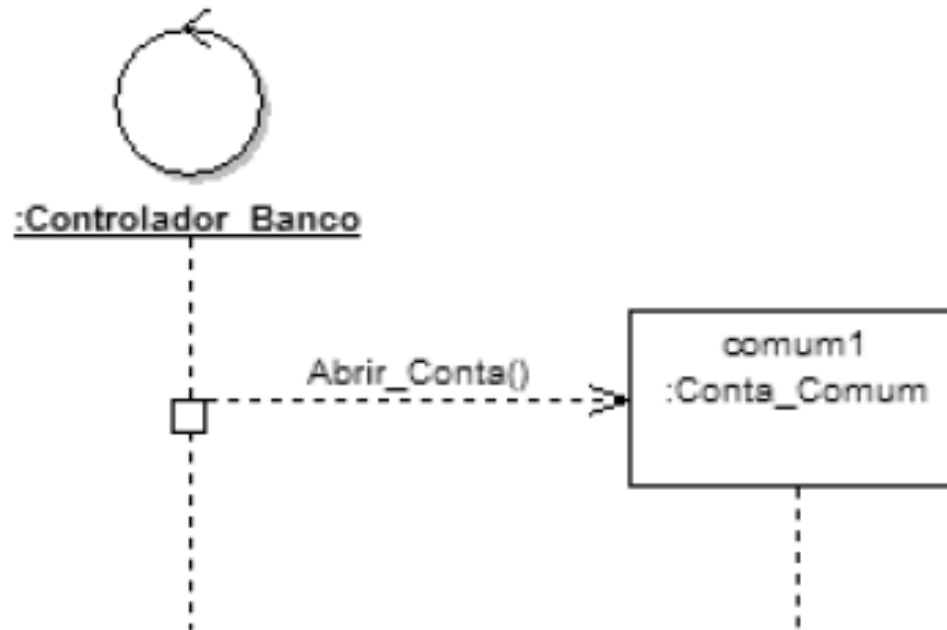


**Mensagem simples entre atores**



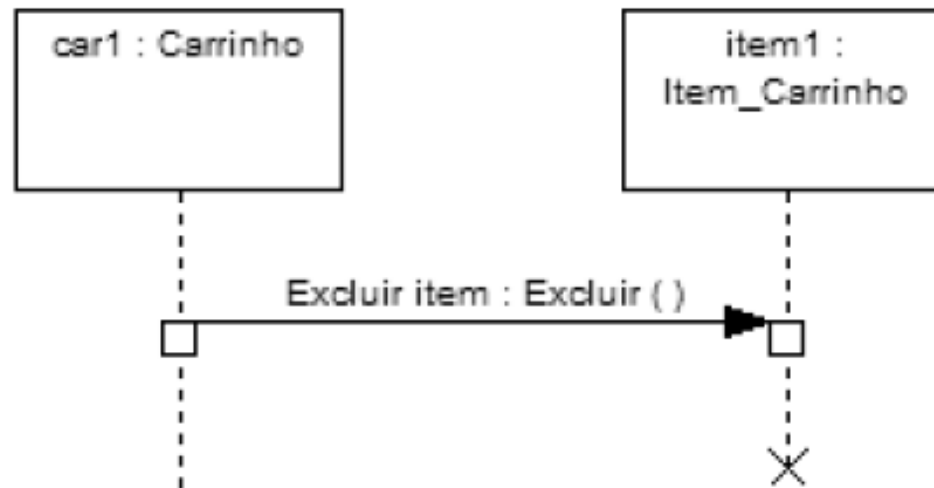
**Mensagem com disparo entre objetos**

- ✓ Quando a mensagem cria um novo objeto, a seta atinge o retângulo que representa o objeto, indicando que a mensagem representa um método construtor



**Mensagem que instancia um novo objeto**

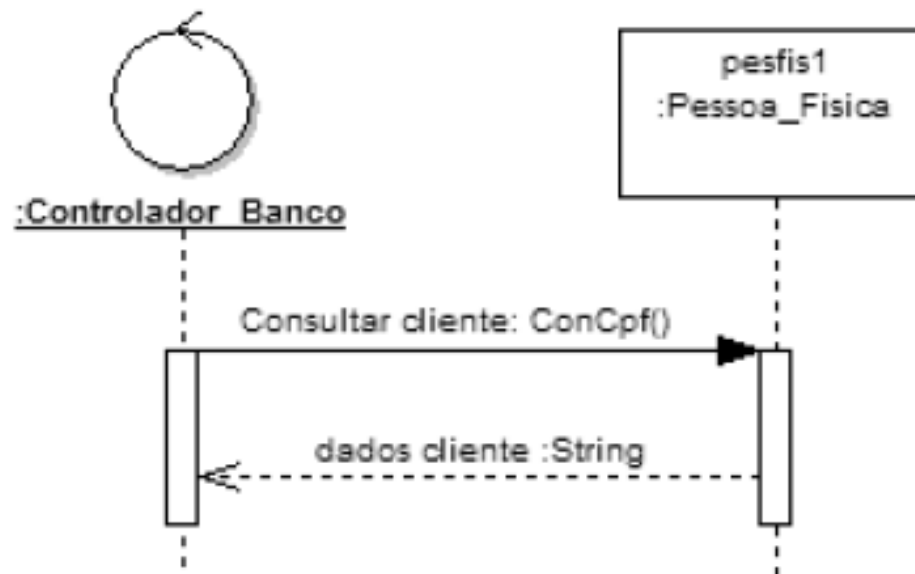
- ✓ Quando uma mensagem elimina um objeto, ela atinge a linha de vida dele e a interrompe com um X



**Mensagem que dispara um método destrutor**

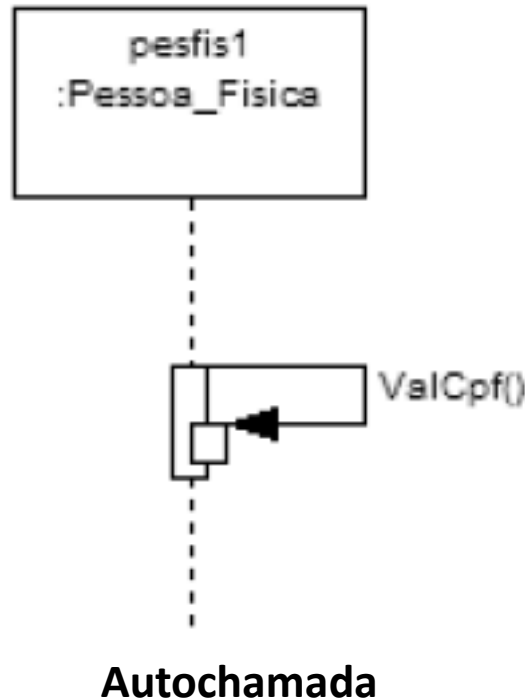
# Mensagens de Retorno

- ✓ Identifica a **resposta a uma mensagem** para o objeto ou ator que a chamou
- ✓ Pode retornar informações específicas do método chamado ou apenas um valor indicando se o método foi executado com sucesso ou não
- ✓ Representadas por uma **linha tracejada contendo uma seta fina** que aponta para o objeto que recebe o resultado do método chamado

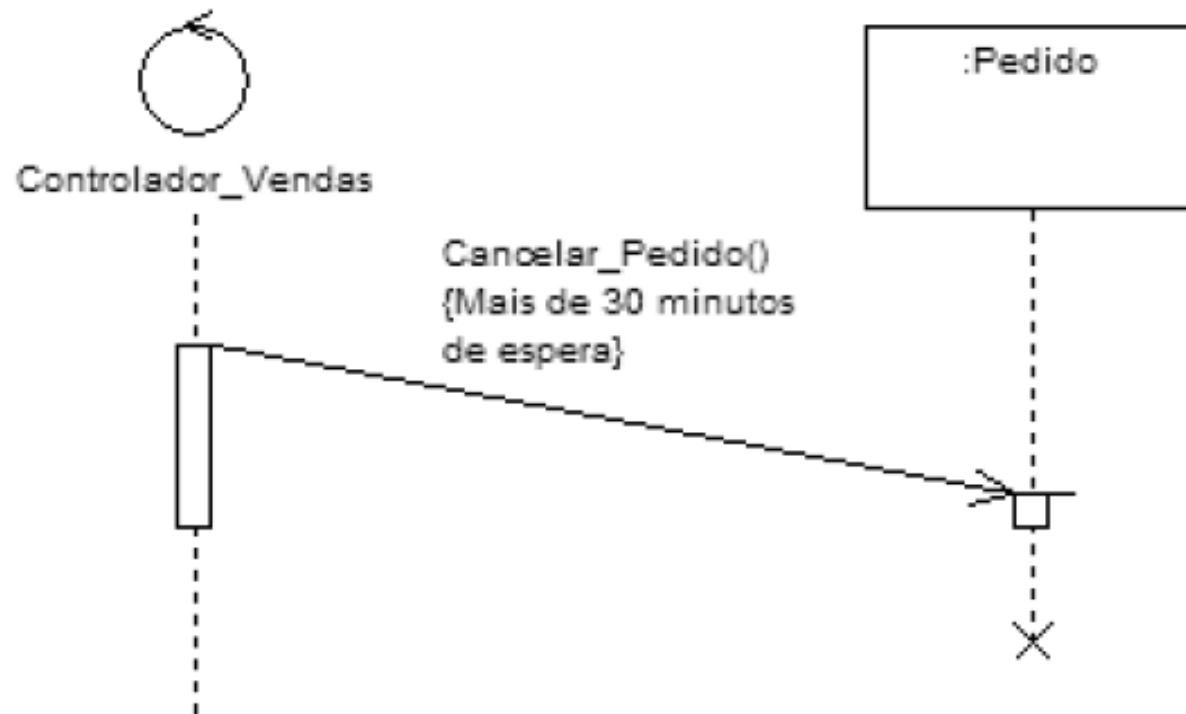


# Autochamadas ou Autodelegações

- ✓ Mensagens que um objeto envia para si mesmo
- ✓ A mensagem parte da linha de vida do objeto e atinge a linha de vida do próprio objeto



- ✓ Quando se quer demonstrar o **tempo que uma mensagem leva em consideração antes de ser disparada**, deve-se usar restrições de duração, e a mensagem é apresentada na diagonal

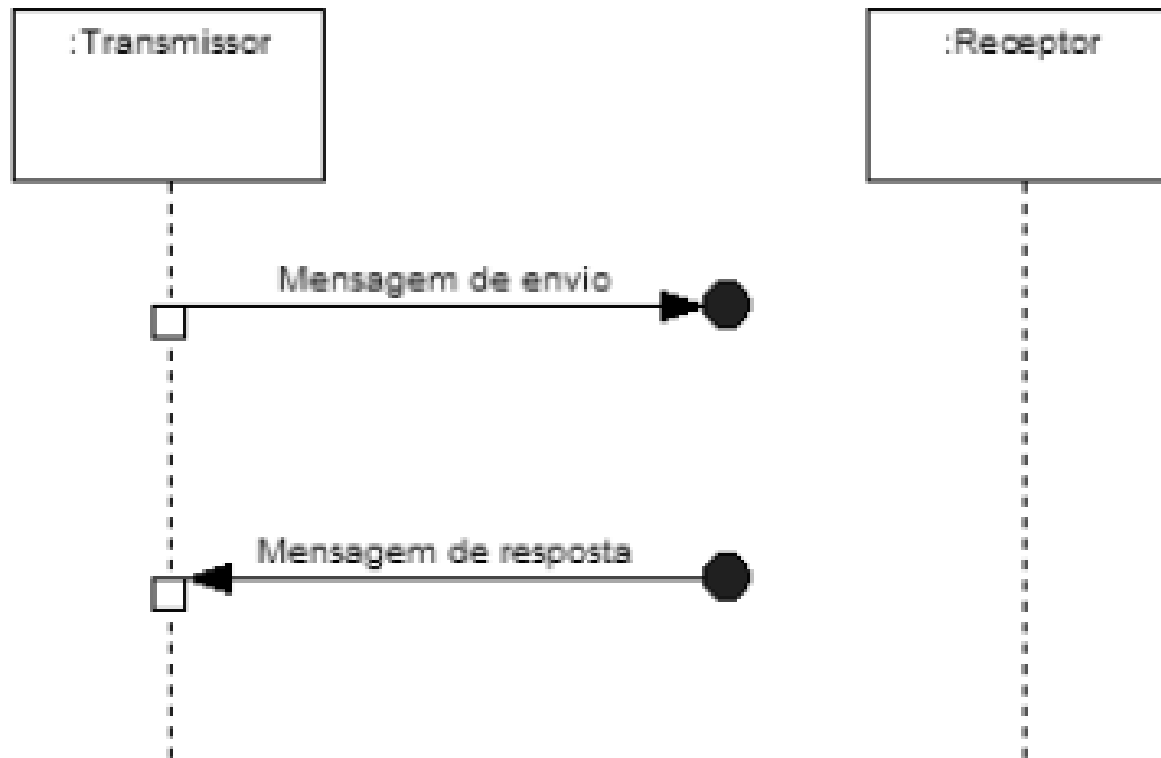


Detalhes de tempo



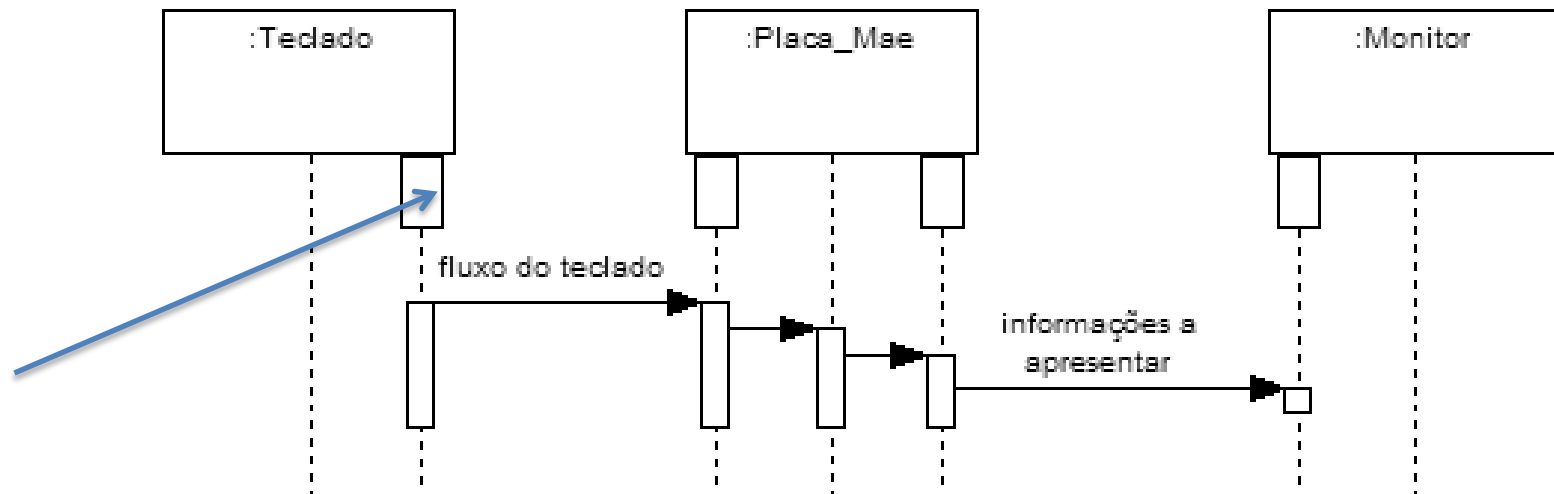
- ✓ **Perdida:** representa uma **mensagem que foi enviada e sua confirmação de recebimento não foi recebida**, podendo significar que a mensagem não chegou ao seu destino
- ✓ **Encontrada:** representa o **recebimento de uma mensagem enviada por um elemento desconhecido ou um elemento não representado no diagrama, ou o recebimento de uma mensagem que fora dada como perdida**, pois seu tempo de espera por resposta poderia ter sido encerrado
- ✓ São representadas por um círculo preenchido
  - **perdida:** o círculo é atingido pela mensagem
  - **encontrada:** mensagem parte do círculo

# Mensagens Perdidas e Mensagens Encontradas



**Exemplo de mensagem perdida e encontrada**

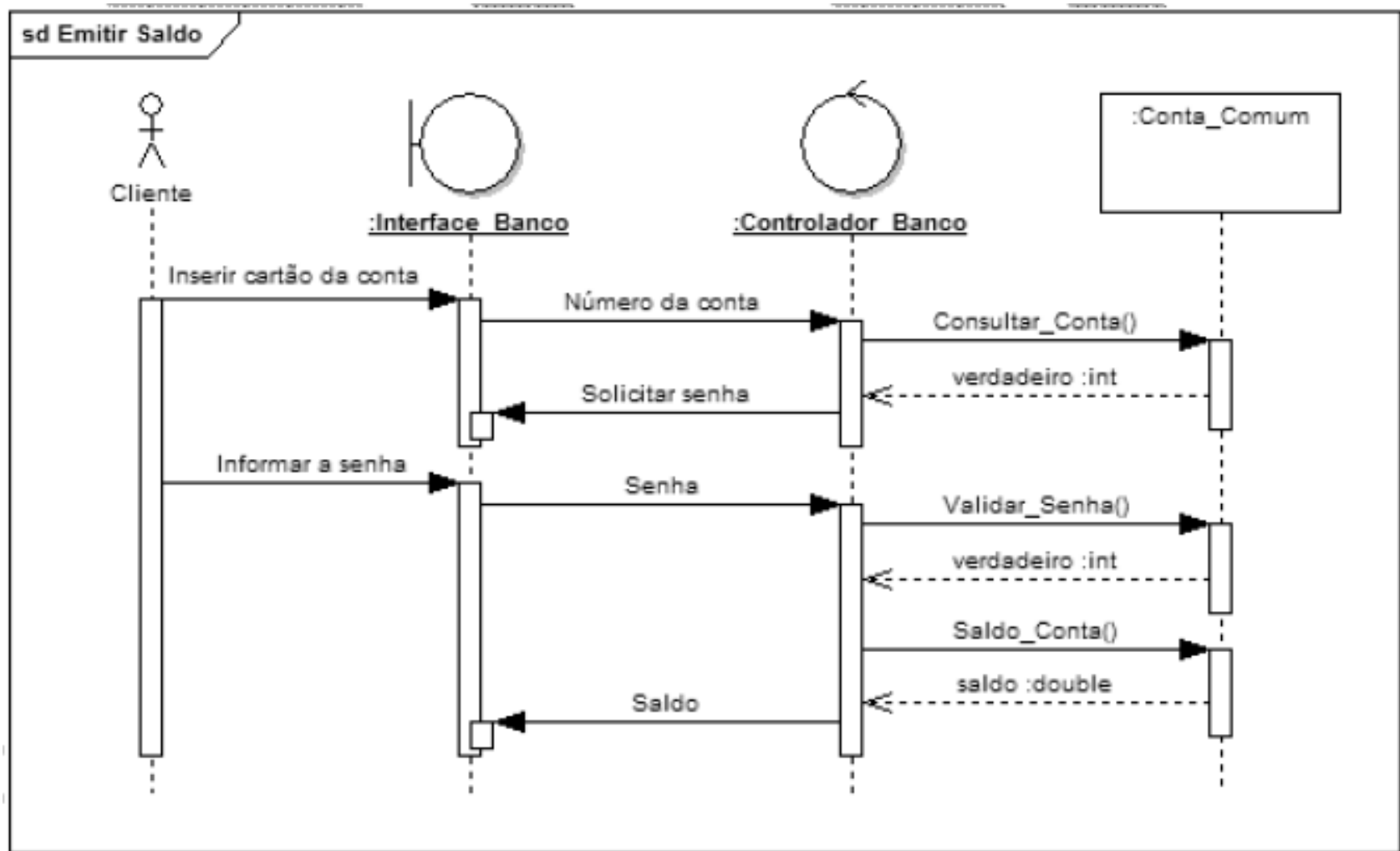
- ✓ É possível representar um objeto no diagrama contendo **instâncias das portas declaradas na classe** a que ele pertence
- ✓ Neste caso, o objeto terá mais de uma linha de vida, o que permite a representação de mensagens externas e internas no objeto



Portas

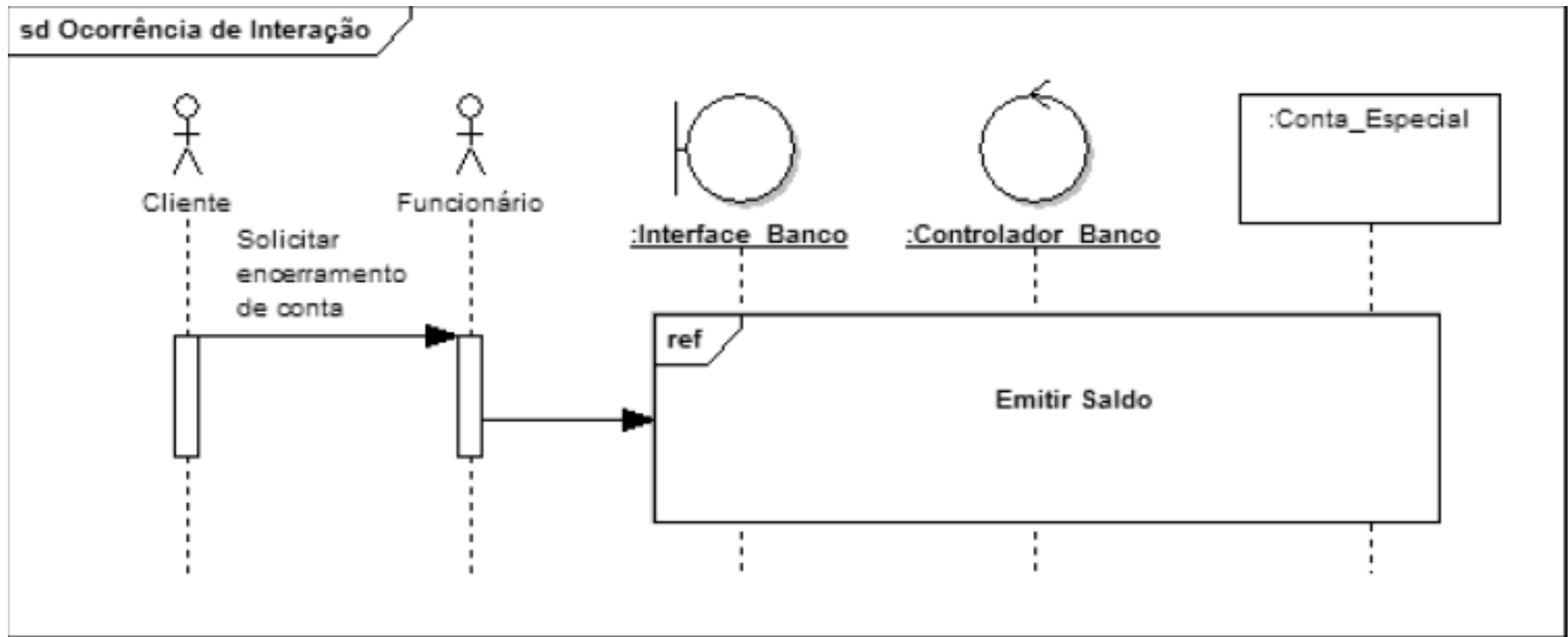
- ✓ **Noções abstratas de unidades de interação geral**
- ✓ Representa uma **parte de uma interação**
- ✓ Retângulo que envolve toda a interação, além de conter uma aba no canto superior esquerdo, contendo um operador que determina qual tipo de diagrama de interação ele se refere

# Fragmentos de Interação



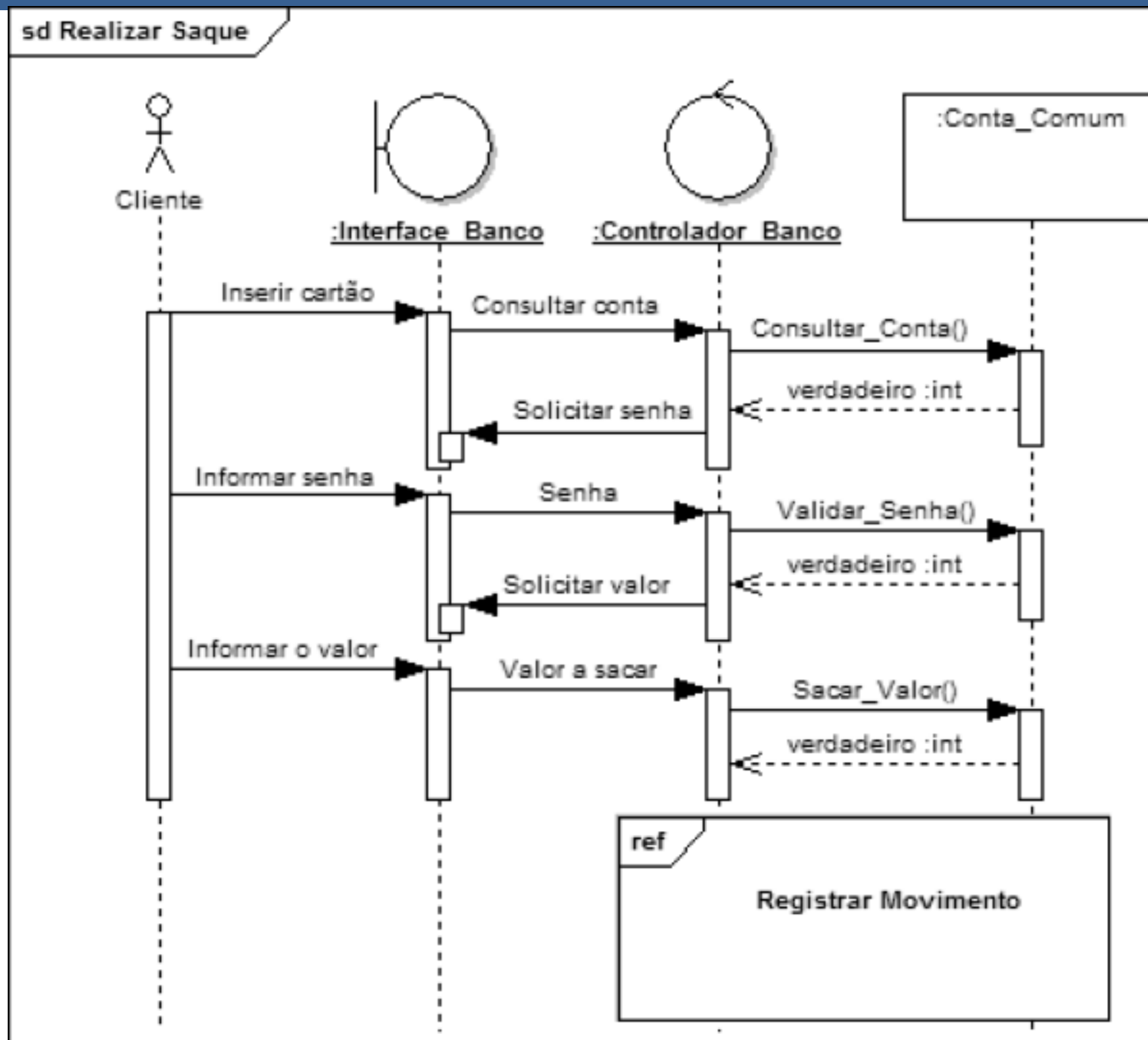
Exemplo de fragmento de interação

- ✓ Uma das principais vantagens do uso de fragmentos de interação caracteriza-se pela possibilidade de se poder referenciá-los por meio do operador **Ref**



Exemplo de ocorrência de interação

- ✓ É possível encontrar usos de interação simplesmente sobrepostos às linhas de vida dos objetos que fazem parte do processo, sem nem ao menos chamá-las por meio de uma mensagem, como se as instruções contidas nos usos de interação fossem adicionadas automaticamente ao diagrama.



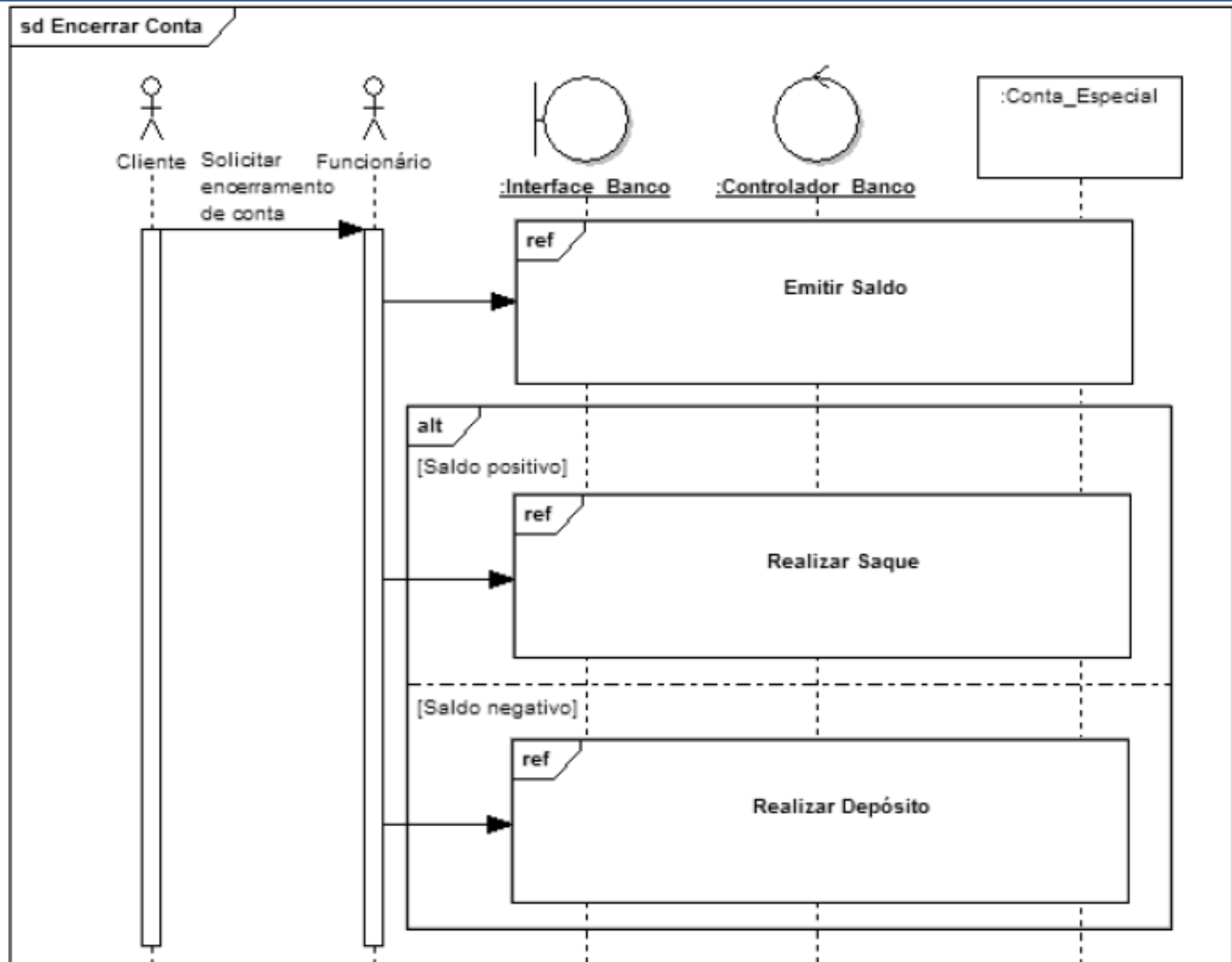
Exemplo de Uso de Interação – Processo de Realizar Saque



- ✓ Permitem uma modelagem semi-independente da parte do diagrama onde deve-se focar **problemas que envolvam testes se-senão, laços ou processamentos paralelos**.
- ✓ Representados por um retângulo que determina a área de abrangência do fragmento no diagrama, além de conterem ainda uma subdivisão em sua extremidade superior esquerda para identificar a descrição do fragmento combinado e seu operador de interação
- ✓ Operadores:
  - *Alt (alternativas)*
  - *Opt (opção)*
  - *Par (paralelo)*
  - *Loop (laço)*
  - *Break (quebra)*
  - *Critical region (região crítica)*

- ✓ Define que o fragmento combinado representa uma **escolha entre dois ou mais comportamentos**

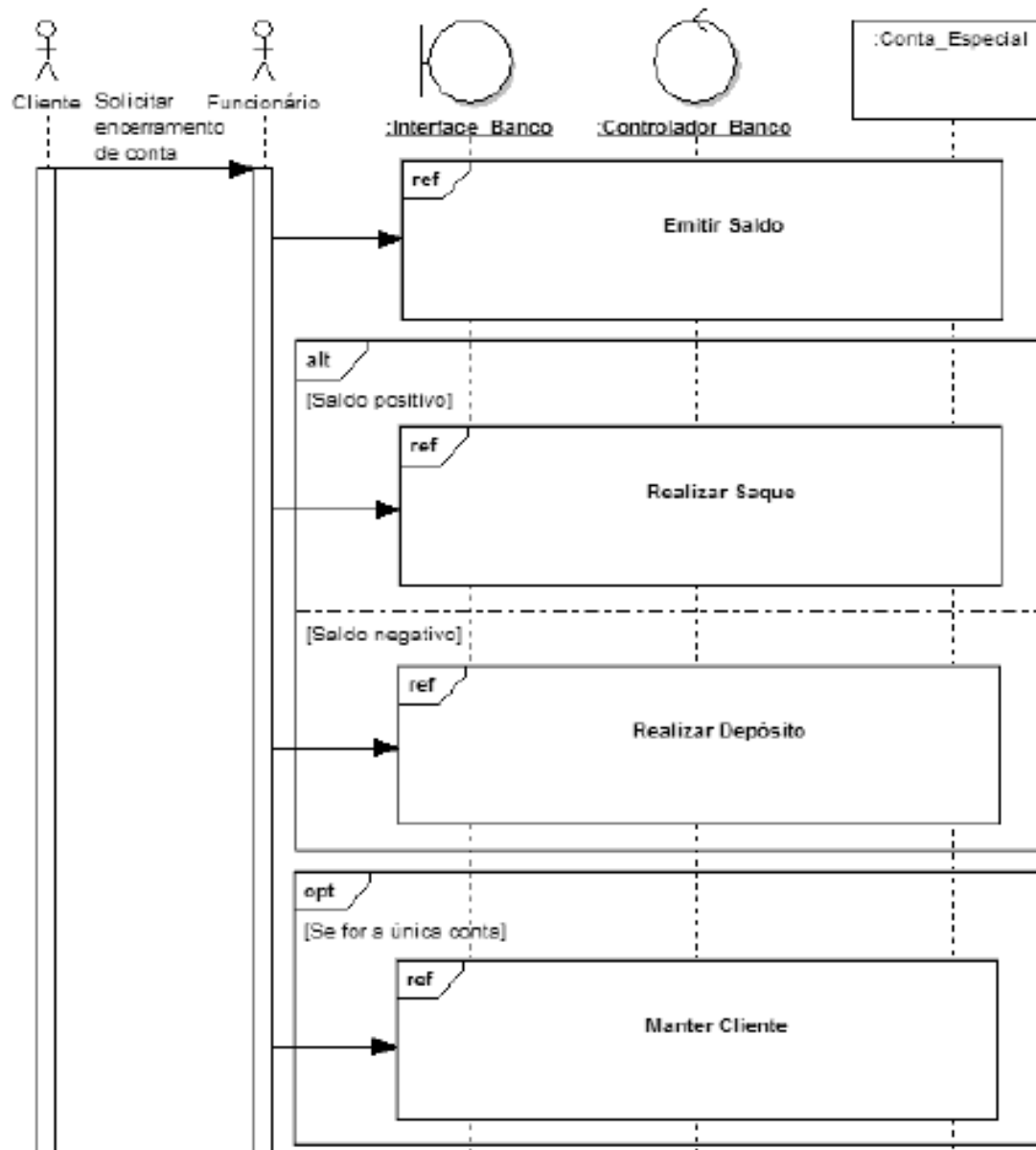
# Operador Alt (Alternativas)



- ✓ Determina que o fragmento combinado representa uma **escolha de comportamento onde esse será ou não executado**

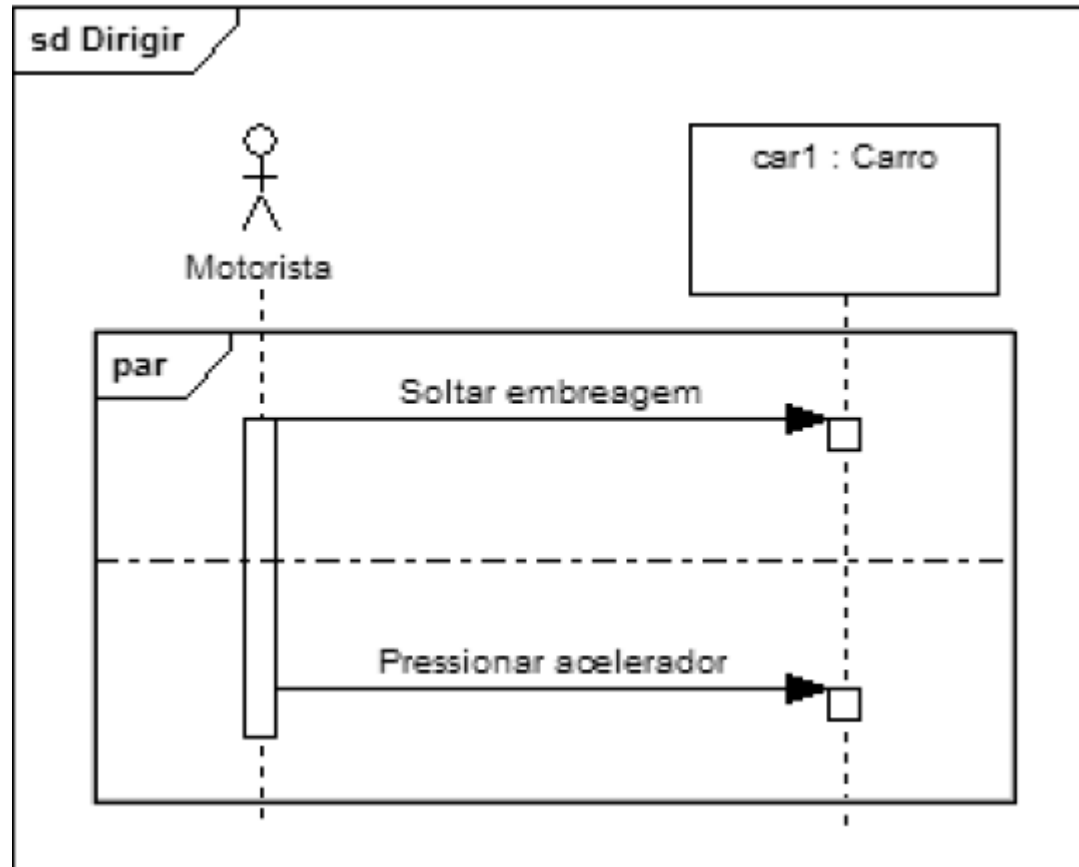
# Operador Opt (Opção)

sd Encerrar Conta



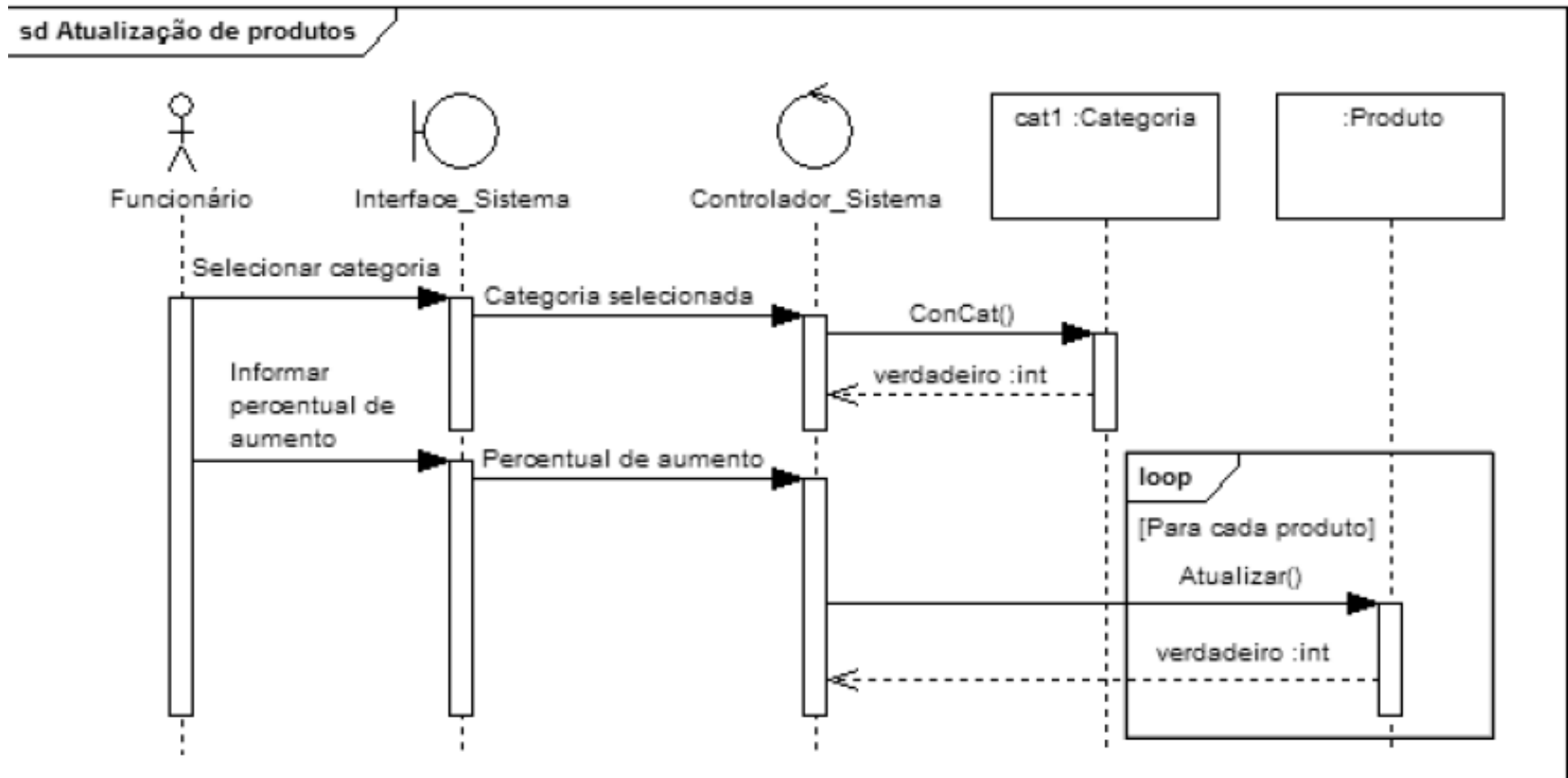
# Operador Par (Paralelo)

- ✓ Determina que o fragmento combinado representa uma **execução paralela de dois ou mais comportamentos**



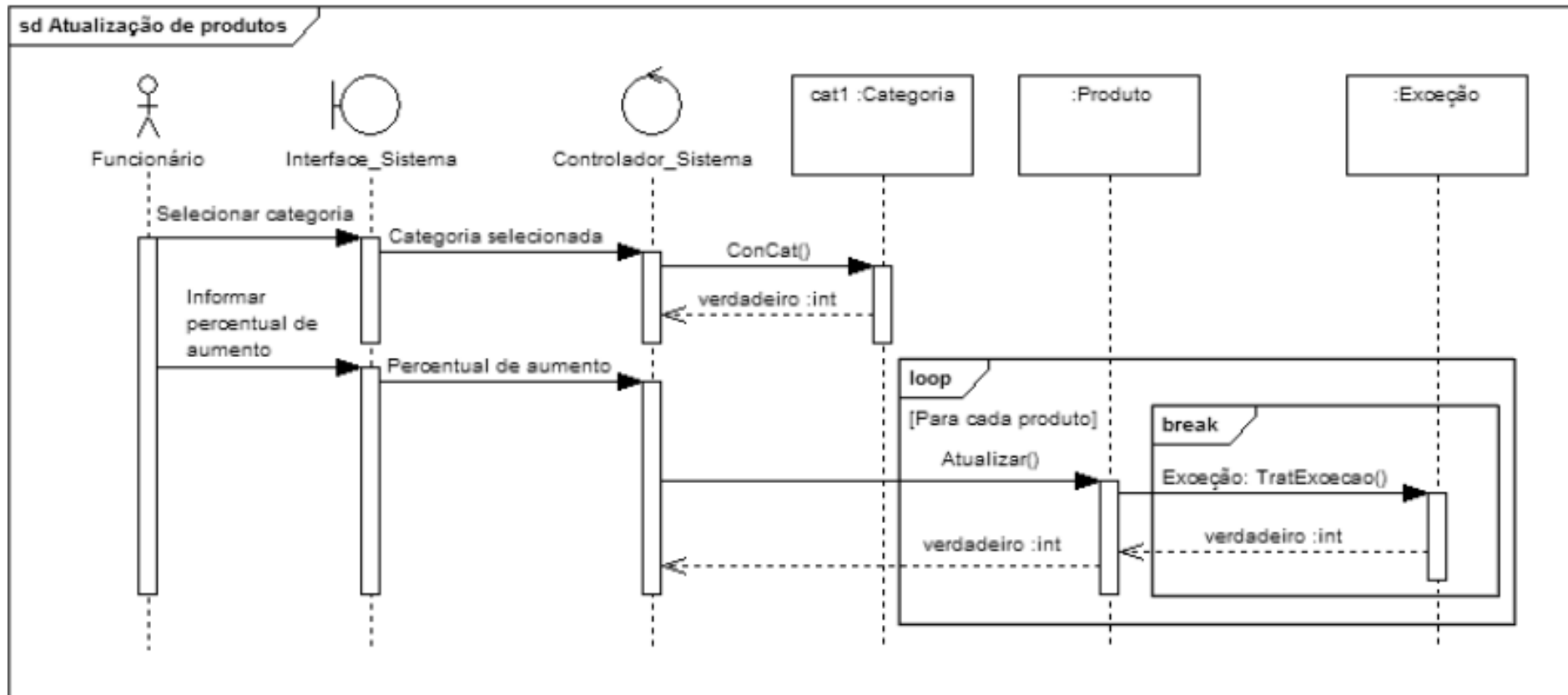
# Operador Loop (Laço)

- ✓ Determina que o fragmento combinado representa **um laço que poderá ser repetido diversas vezes**



# Operador Break (Quebra)

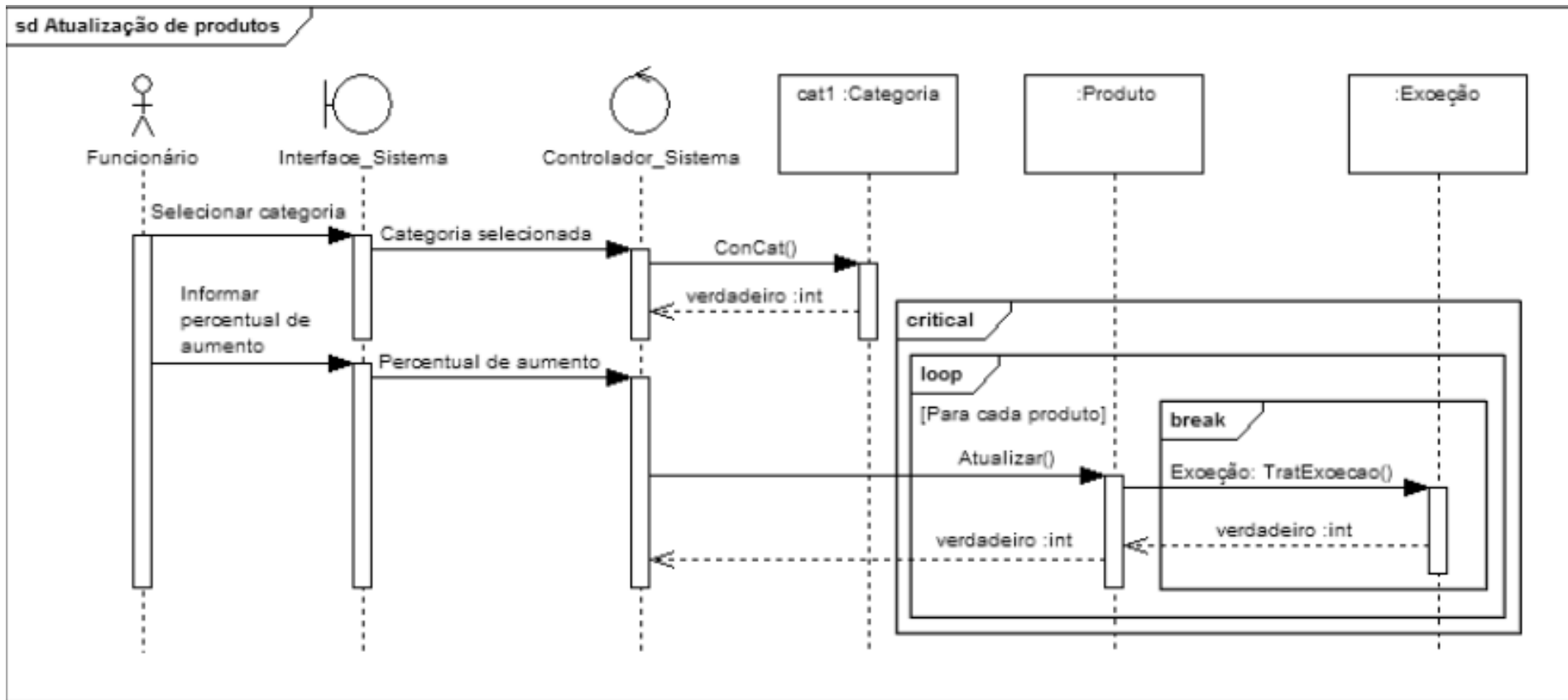
- ✓ Indica uma “**quebra**” na execução normal do processo (usado no tratamento de exceções)





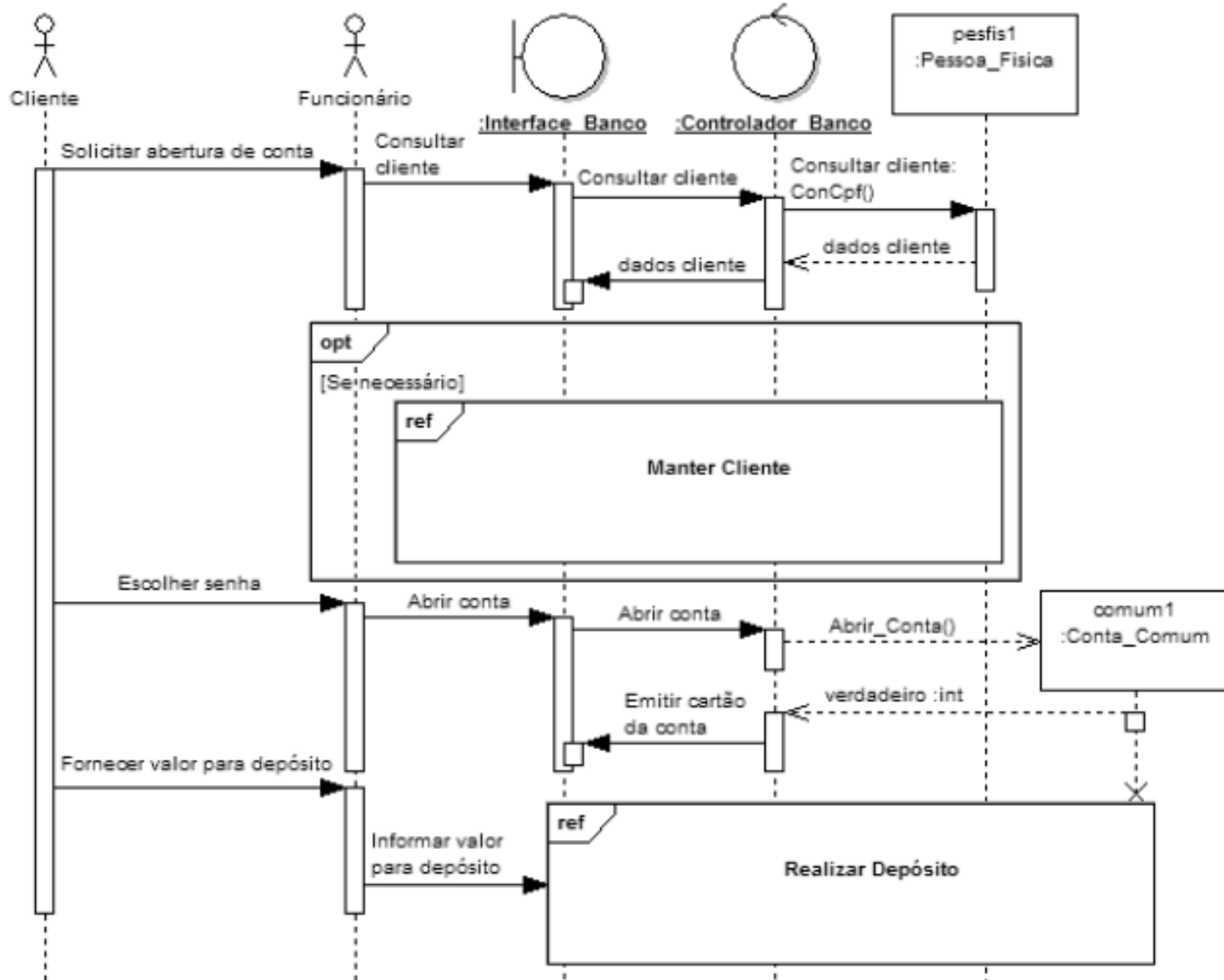
# Operador Critical Region (Região Crítica)

- ✓ Identifica uma **operação atômica** que não pode ser interrompida por outro processo até ser totalmente concluída

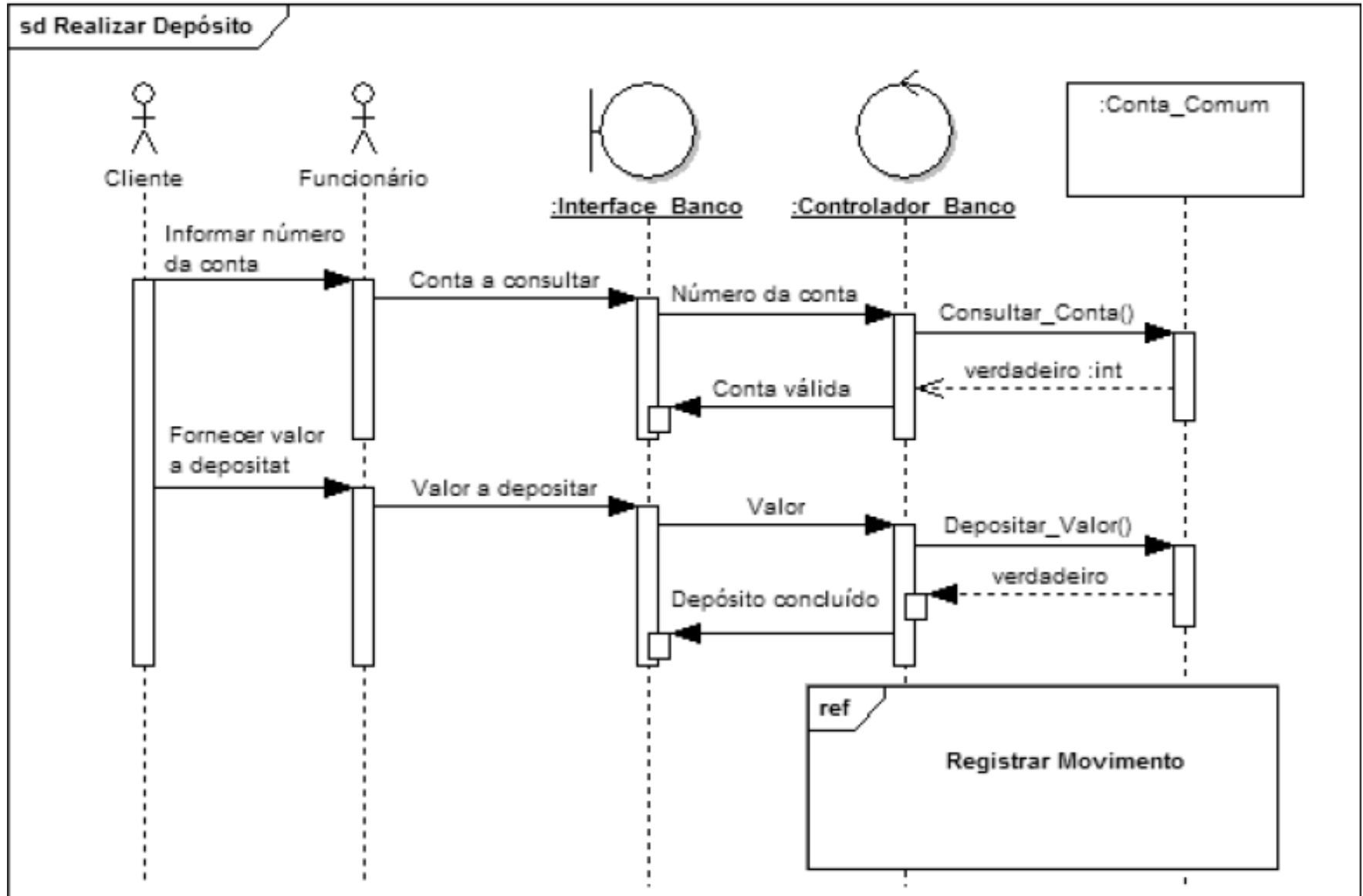


# Exemplo – Abrir Conta Comum

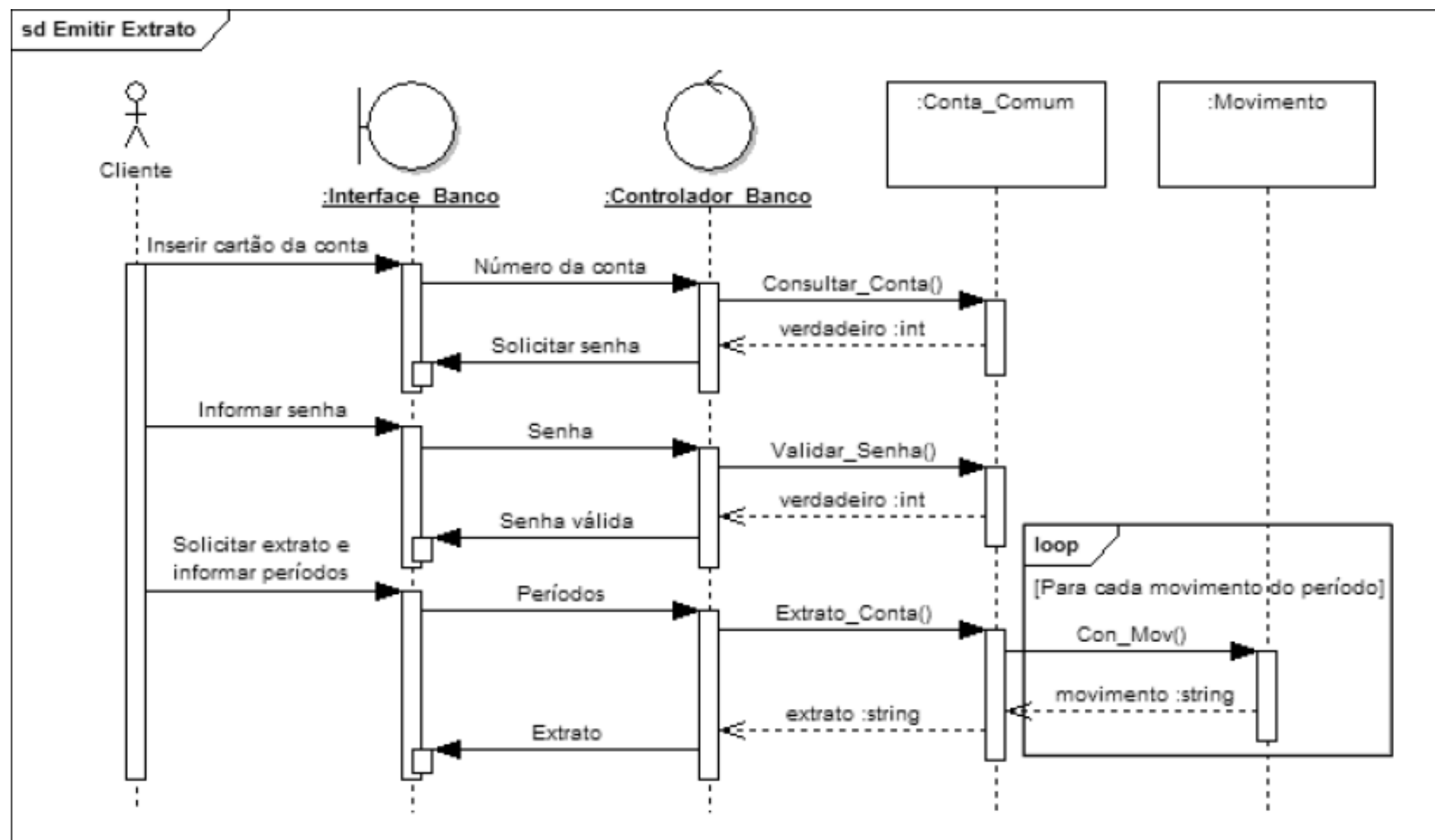
sd Abrir Conta Comum



# Exemplo - Realizar Depósito



# Exemplo – Emitir Extrato

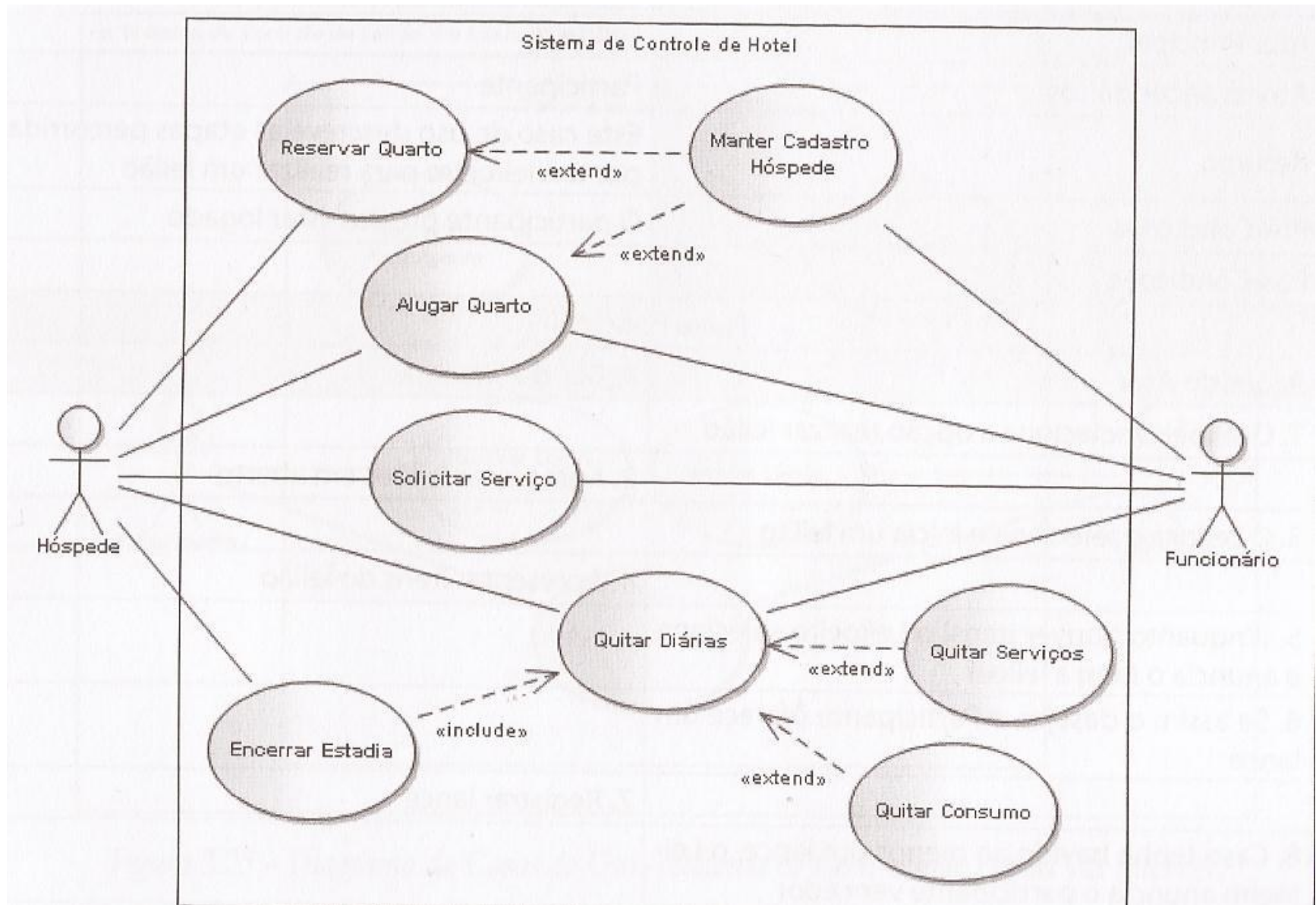


# Exercício – Sistema de Controle de Hotelaria

- ✓ Considere os requisitos do sistema a seguir (Sistema de Controle de Hotelaria) e os diagramas de Casos de Uso e Classes modelados:
  - Os quartos podem ser alugados no momento em que o hóspede chega ao hotel (desde que existam vagas) ou serem reservados via internet
  - Caso seja a primeira vez que aluga quartos, os seus dados tenham mudado, o hóspede deve ser cadastrado antes de finalizar o aluguel do quarto
  - Além do aluguel do quarto, o hotel oferece diversos serviços, como restaurante, lavar e/ou passar roupas etc. Obviamente, qualquer desses serviços, se solicitado, será cobrado na fatura final
  - O hóspede pode também consumir os produtos contidos no frigobar, que também são cobrados pelo hotel

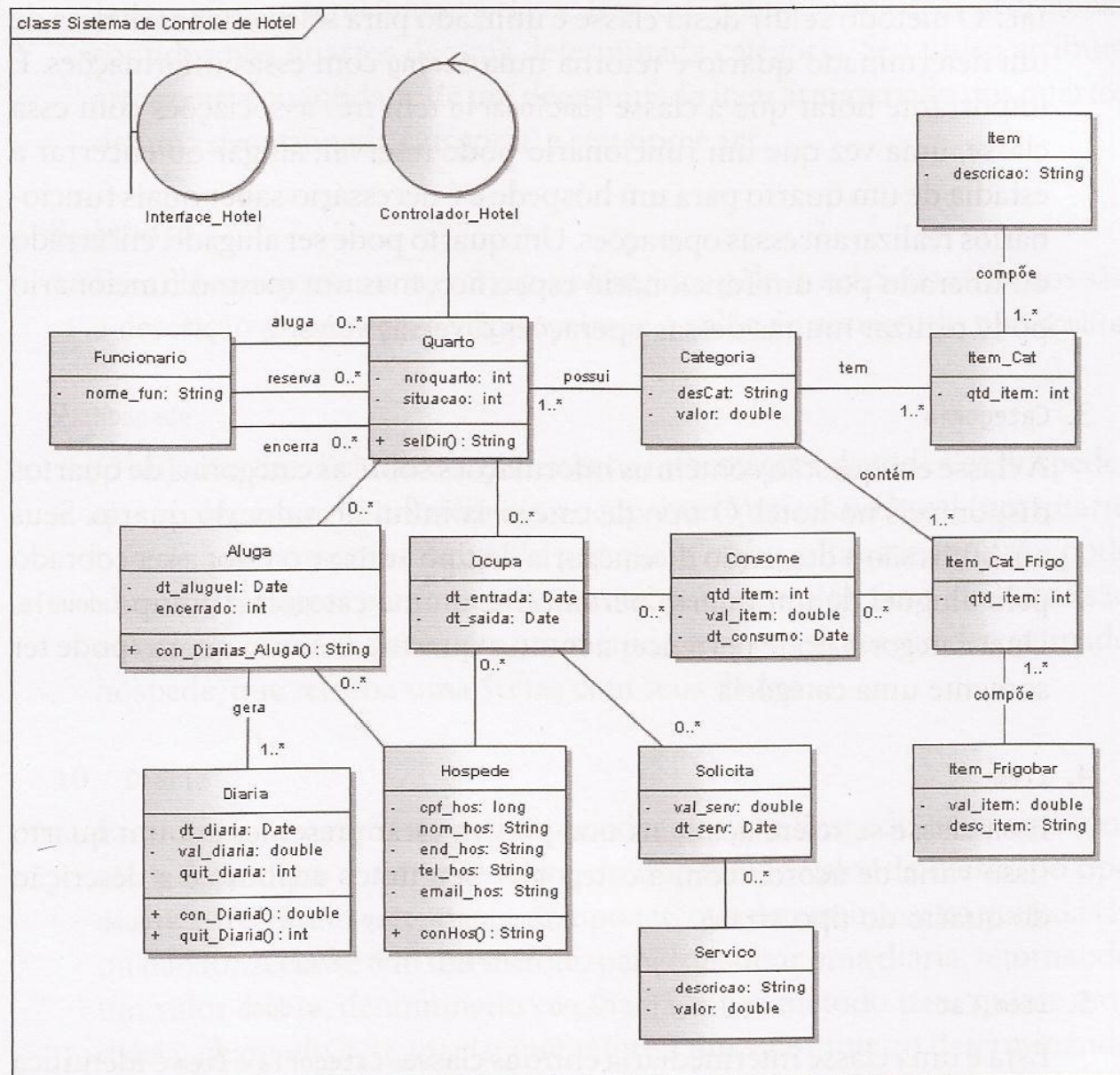
- **As diárias vencem ao meio-dia. A política do hotel exige que as diárias sejam quitadas semanalmente. Quando o cliente for quitar a fatura, quitará não somente as diárias do(s) quarto(s) que alugou, mas também qualquer serviço que tenha solicitado e os itens consumidos no frigobar**
- **O hóspede, depois de quitar a fatura, pode permanecer no hotel ou encerrar sua estadia**
- **Quando for encerrar sua estadia, o hóspede deverá pagar quaisquer serviços e/ou diárias ainda não pagas**

# Exercício – Sistema de Controle de Hotelaria





# Exercício – Sistema de Controle de Hotelaria





## ✓ Processo de Pagamento de Diárias

✓ Desenvolva o diagrama de sequência para o processo de pagamento de diárias, de acordo com os seguintes requisitos:

- O hóspede se dirige ao funcionário e informa os quartos que deseja pagar
- O funcionário, por meio do sistema, deve então buscar todas as diárias ainda não pagas relativas ao quarto e apresentar ao hóspede
- Ao realizar o pagamento, as diárias serão quitadas, podendo o hóspede permanecer no hotel ou encerrar sua estadia
- Caso o hóspede tenha solicitado algum serviço ou consumido algum item de frigobar, deverá pagá-los igualmente
- **ENTREGAR O EXERCÍCIO NA PRÓXIMA AULA (grupos de até 5 alunos)!!!**

- ✓ **Acesse a ferramenta REA-UML e identifique o raciocínio lógico envolvido na modelagem dos diagramas de sequência e de comunicação modelados para o sistema descrito como estudo de caso.**
- ✓ **O link para acesso à ferramenta é: <http://sgvclin.altervista.org/rea-uml/>**