



DISCIPLINA

# ANÁLISE DE SISTEMAS

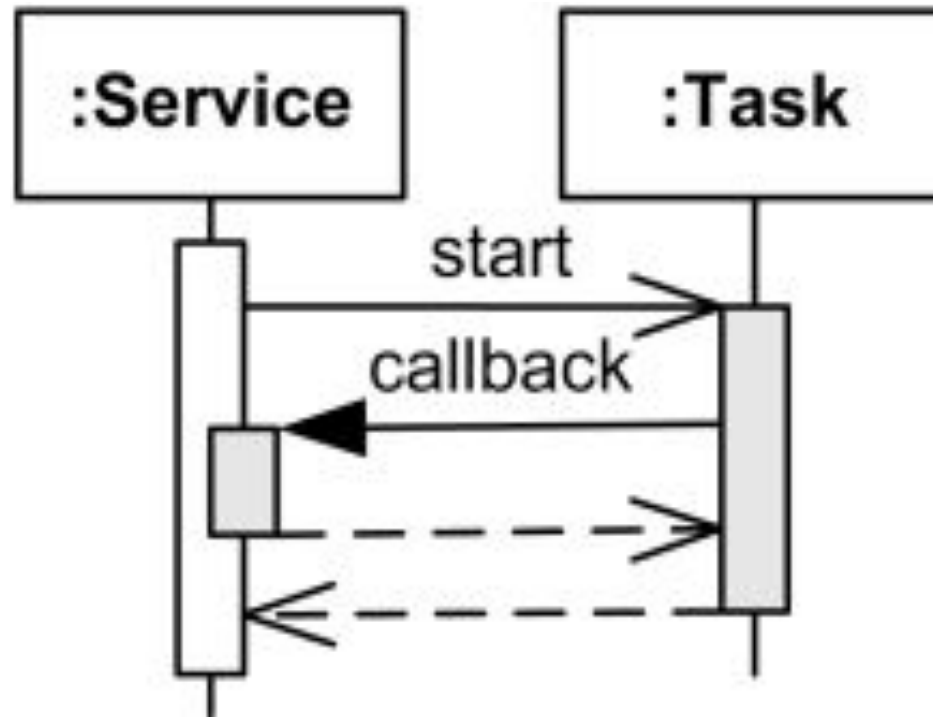
---

# ORIENTADO A OBJETOS

**AULA 10**

Profº. Me. Flávio Henrique Fernandes Volpon  
[flavio.volpon@docente.unip.br](mailto:flavio.volpon@docente.unip.br)

# Diagrama de Sequência

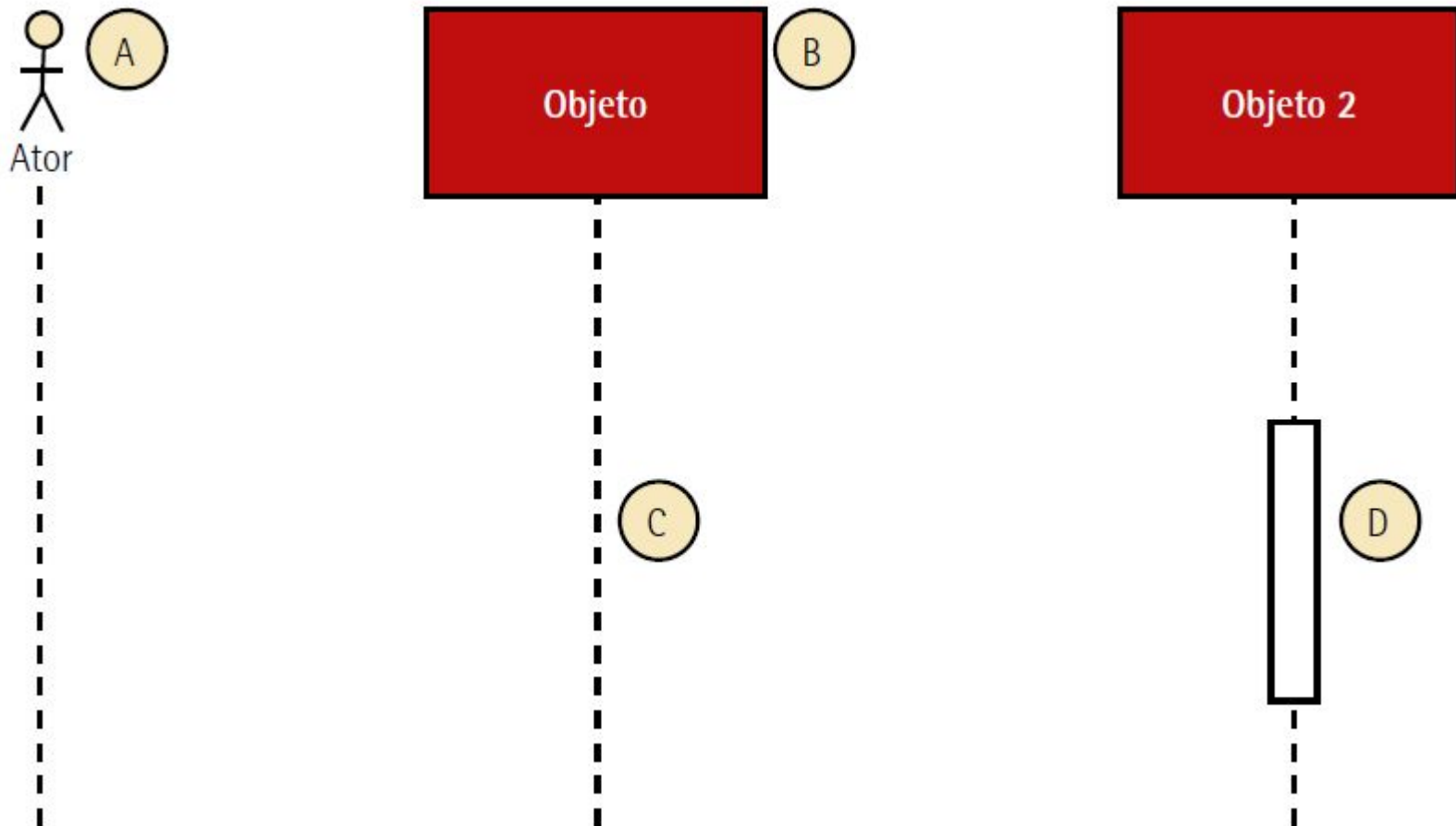


# Diagrama de Sequência

- O **diagrama de sequência** da **UML** representa a **interação** de um conjunto de **objetos**, a **troca de mensagens** entre eles para resolver um problema específico.
- Nele podemos **visualizar** a **troca de mensagens** de forma **sequencial** e encaixada em uma **linha de tempo**.
- **Podemos** representar **todos** os **objetos e atores** que fazem parte do **cenário** do problema que desejamos representar.
- Idealmente, **representamos um cenário específico** de um problema que queremos resolver, **podemos** fazer uma **quebra** por **casos de uso** ou por **regras de negócio**, a depender da complexidade.

# Diagrama de Sequência

A figura abaixo mostra os principais elementos de um diagrama de sequência:



Principais elementos do diagrama de sequência da UML

# Diagrama de Sequência

O quadro abaixo mostra os conceitos do diagrama de sequencia representado pela figura anterior:

Letra	Conceito
A	Representa o ator, envolvido no contexto. Como vimos, um ator pode enviar uma mensagem para um objeto. Essa mensagem tem o significado de evento.
B	O retângulo representa um objeto que faz parte da execução do cenário. Note que estamos falando de objetos e não de classes, pois quando estamos representando a troca de mensagens, no aspecto dinâmico do sistema, o objeto já está instanciado. Obrigatoriamente, os objetos do diagrama de sequência devem constar no modelo de classes.
C	A linha tracejada abaixo dos atores e dos objetos representa a linha de tempo
D	O retângulo posicionado abaixo do objeto significa que naquele espaço de tempo iniciou-se o ciclo de vida do objeto. Como vimos, um objeto é criado, utilizado e destruído, e nesse espaço de tempo representado pelo retângulo o objeto está ativo, ou seja, está criado, e está apto a receber mensagens.

## Conceitos e elementos do diagrama de sequência

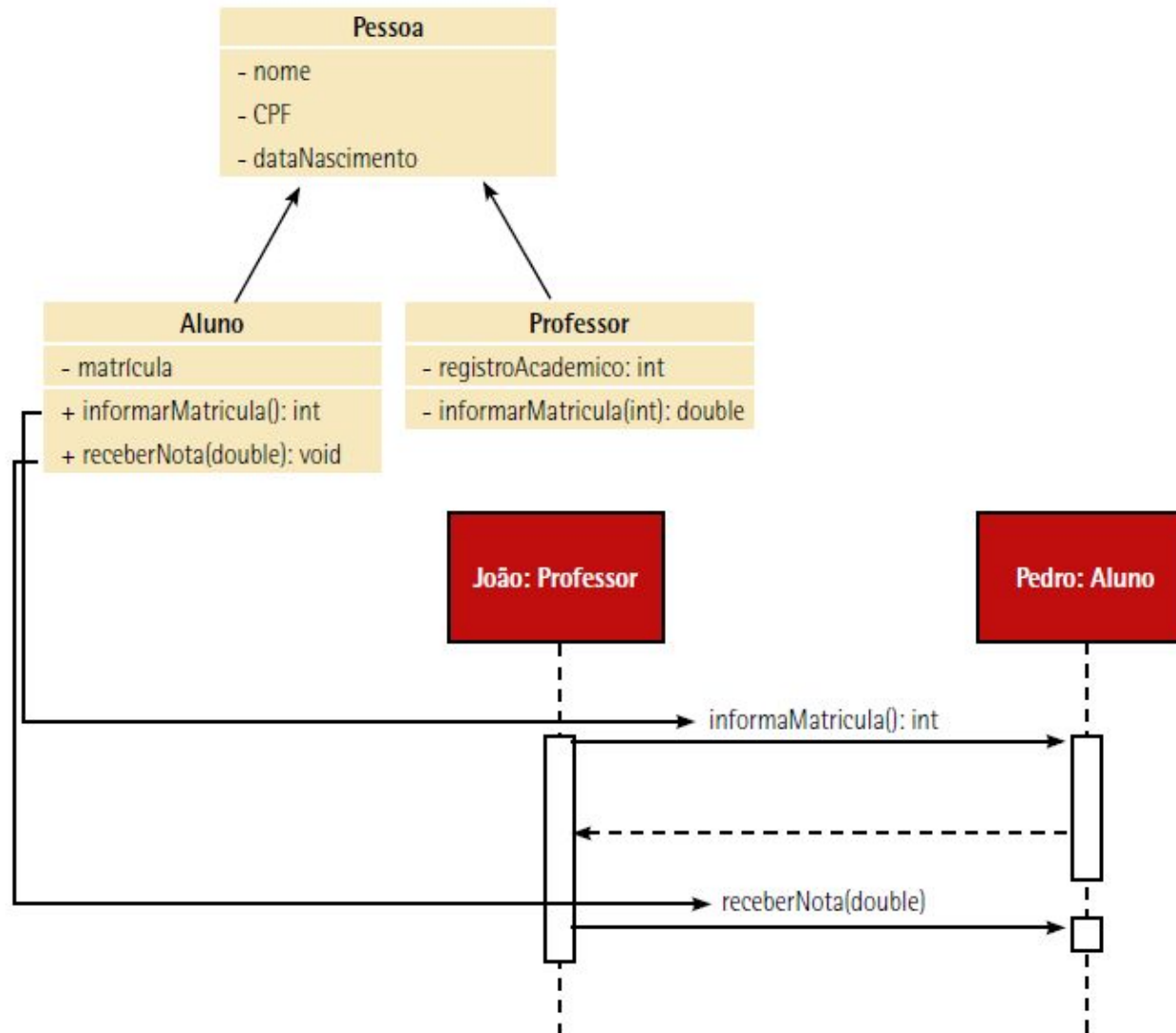
# Diagrama de Sequência

- A próxima figura mostra um **exemplo de troca de mensagens, representado pelo diagrama de sequência, associado ao modelo de classes.**
- Note que **temos duas classes, Aluno e Professor, representadas no diagrama de classes.**
- Essas **classes originaram dois objetos** no diagrama de sequência:
  - ✓ Pedro (Aluno)
  - ✓ e João (Professor).

# Diagrama de Sequência

- A classe **Aluno** possui dois métodos públicos: **informarMatricula** e **receberNota**
- Eles são representados no diagrama de sequência como mensagens enviadas de professor (João) para aluno (Pedro).
- Note ainda, que o nome da mensagem segue o mesmo padrão da declaração do método na classe envolvida, bem como os parâmetros ou argumentos e retornos dos métodos.

# Diagrama de Sequência



Exemplo de troca de mensagens no diagrama de sequência



# Diagrama de Sequência

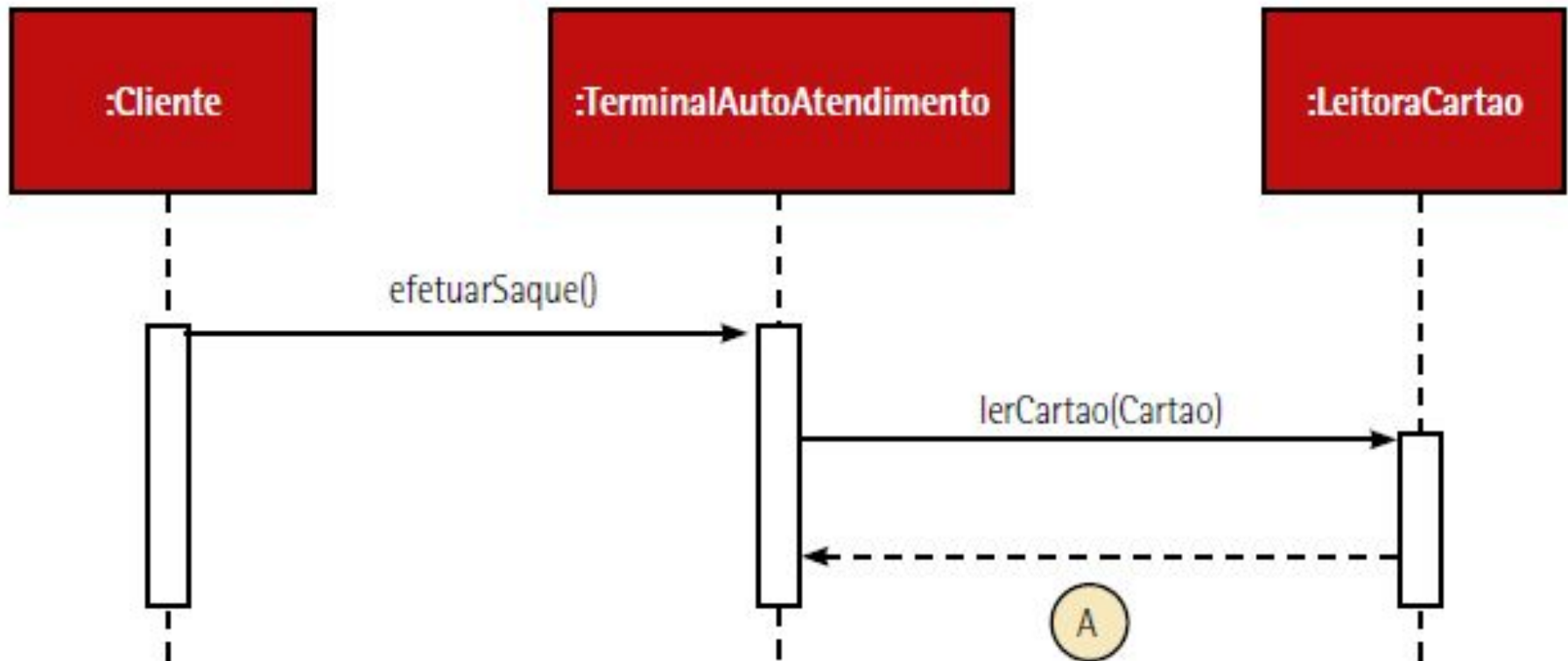
## MENSAGENS SÍNCRONAS

- Criam uma **dependência** do **estado** do **objeto** que **enviou** uma **mensagem** com o **estado** do **objeto** que a **recebe**.
- O objeto que **enviou** a **mensagem** **não** tem seu **estado** **alterado**, **não executa** nenhuma **ação**, **até** que o **objeto** que **recebeu** a mensagem **permita**.
- Fazendo uma **analogia** com **funções algorítmicas**, e como uma **chamada de função com retorno**, no qual o **algoritmo** que **chamou** a função **não executa** nenhuma **ação** até que a **função finalize** seu **processamento** e **retorna** um **valor**.

# Diagrama de Sequência

## MENSAGENS SÍNCRONAS

Em O.O., o exemplo mais comum é a chamada de um método com retorno, como mostra o exemplo a seguir.

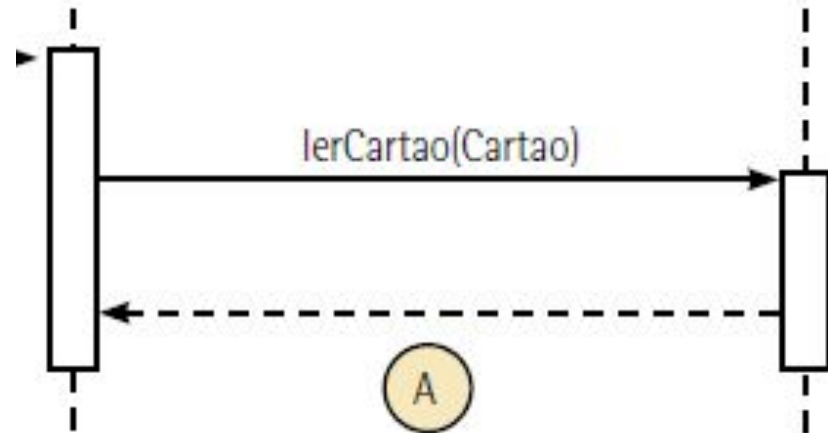


Exemplo de mensagens síncronas no diagrama de

# Diagrama de Sequência

## MENSAGENS SÍNCRONAS

- O **ponto** indicado com a letra **A**, a **seta tracejada**, indica o **retorno** da mensagem **lerCartao**.
- Se olharmos sob o ponto de vista do **sincronismo**, o objeto **TerminalAutoAtendimento** não pode executar nenhuma **ação** até que não haja o **retorno** da mensagem por parte do objeto **LeitoraCartao**.



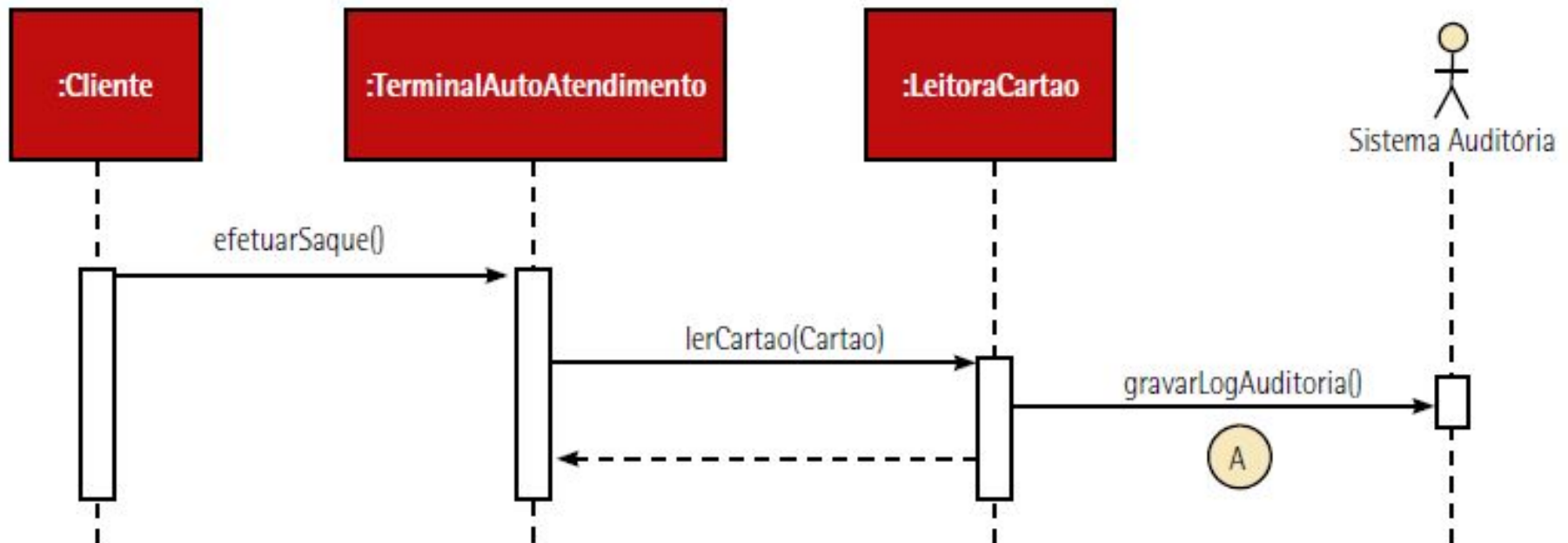
# Diagrama de Sequência

## MENSAGENS ASSÍNCRONAS

- Ao **contrário** das mensagens **síncronas**, as **assíncronas** **não criam** uma **dependência** do **estado** do **objeto** que **enviou** a mensagem com o **estado** do **objeto** que a **recebe**.
- Ou seja, o **objeto** que **enviou** a mensagem **pode executar qualquer ação independentemente** da **reação** do objeto que **recebeu** a mensagem.
- A figura a seguir mostra um exemplo de mensagem assíncrona no diagrama de sequência.

# Diagrama de Sequência

## MENSAGENS ASSÍNCRONAS

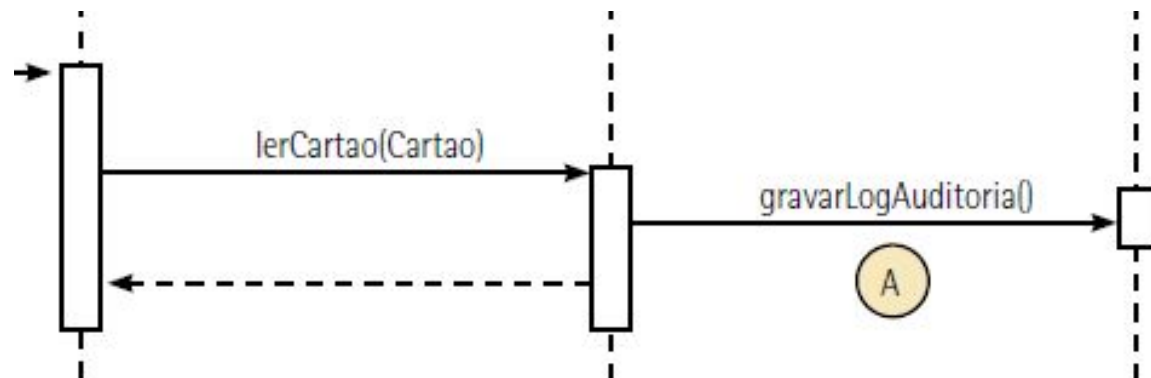


Exemplo de mensagens assíncronas no diagrama de sequência

# Diagrama de Sequência

## MENSAGENS ASSÍNCRONAS

- O **ponto** indicado com a letra **A**, a **seta contínua** com flecha na ponta, **indica** que a mensagem **gravarLogAuditoria** é **assíncrona**.
- Se olharmos sob o ponto de vista do **sincronismo**, o objeto **LeitoraCartao** pode **executar qualquer ação independentemente** do ator externo Sistema Auditoria.



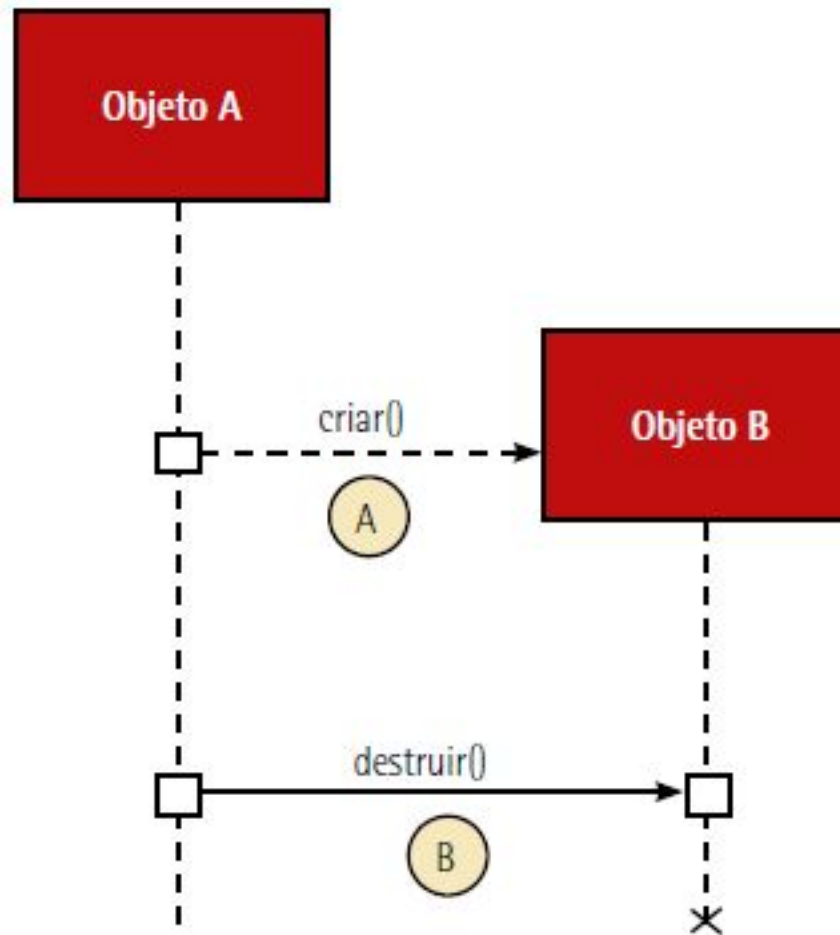
# Diagrama de Sequência

## CRIAÇÃO E DESTRUIÇÃO DE OBJETOS

- Como **vimos** no **ciclo** de **vida** de um **objeto**, existem **dois métodos**, **construtores** e **destrutores**, que tem como **objetivo criar e remover** um **objeto** da estrutura de **memória** de um sistema.
- A **figura a seguir** mostra a **representação** da **criação** e **destruição** de um objeto.
- No caso, temos o **Objeto A criando o Objeto B**, como indicado pela letra **A**, e temos o mesmo **Objeto A destruindo o Objeto B**, como indicado pela letra **B**.
- Note que a **remoção**, por completo, de um **objeto** da **memoria** e **representado** por um **X** ao **fim da sua linha de tempo**, indicando que o **ciclo de vida** do objeto se encerrou.

# Diagrama de Sequência

## CRIAÇÃO E DESTRUIÇÃO DE OBJETOS



Exemplo de representação de construção e destruição de objetos



# Diagrama de Sequência

## ESTRUTURAS DE DECISÃO E DE REPETIÇÃO

- Assim como em **algoritmos**, temos **estruturas de decisão** e **laços de repetição**.
- A interpretação é a mesma que damos em lógica de programação, ou seja:
  - Em **estruturas de decisão**, uma **mensagem só é enviada** para um objeto **se** uma determinada **condição for satisfeita**.
  - Em um **laço de repetição**, uma **mensagem é enviada sucessivas vezes**, **tantas quantas** forem **especificadas no laço**.

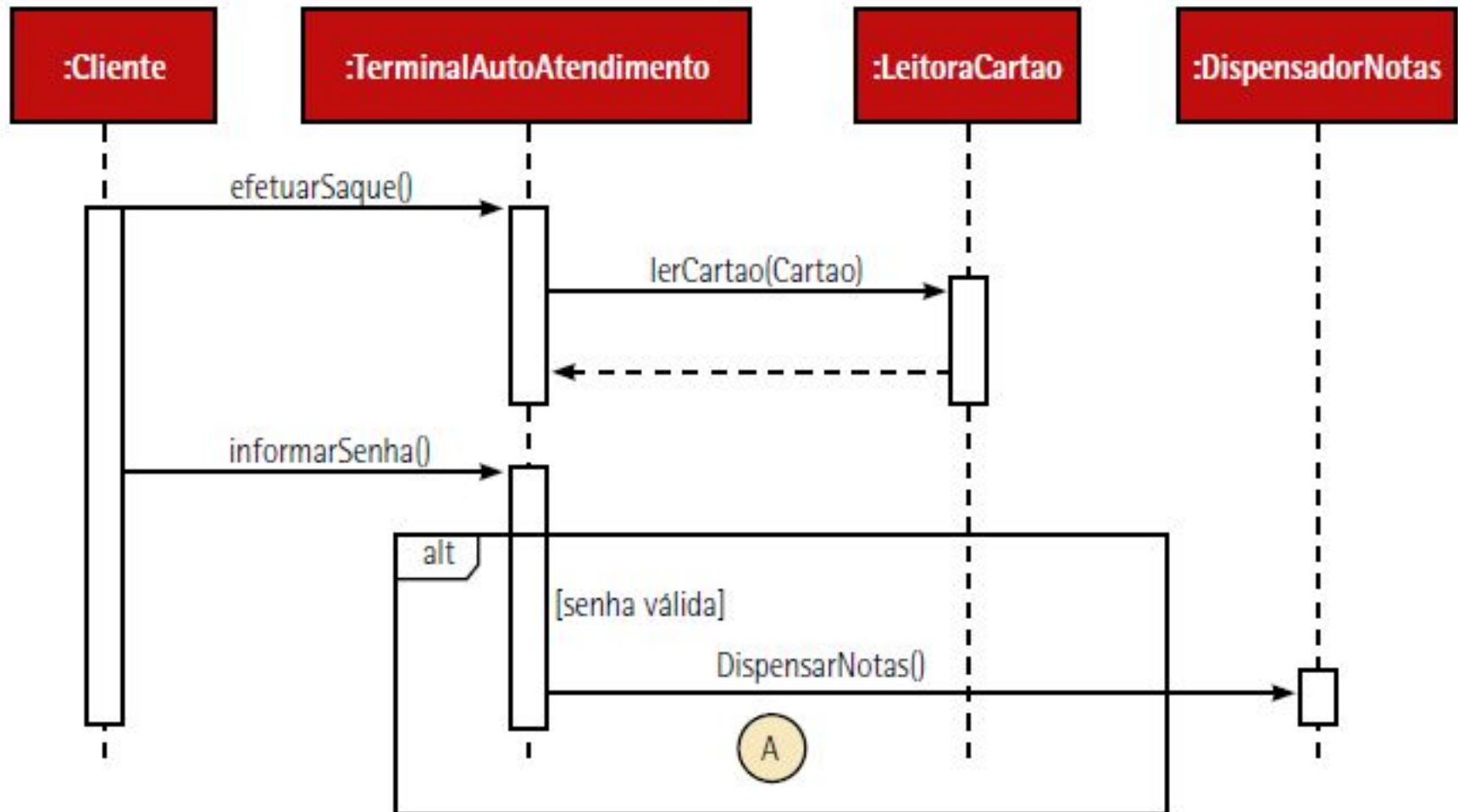
# Diagrama de Sequência

## ESTRUTURA DE DECISÃO

- A figura a seguir mostra um **exemplo** de estrutura de decisão.
- No **diagrama** de sequência, temos o **elemento operador de controle**, que pode ser **observado** no **ponto** indicado com a **letra A**.
- O indicativo alt no **operador de controle** significa **fluxo alternativo**, ou seja, só será executado se uma condição for satisfeita, no caso, a **senha informada** deve ser **correta**.

# Diagrama de Sequência

## ESTRUTURA DE DECISÃO



Exemplo de fluxo alternativo em diagrama de sequência

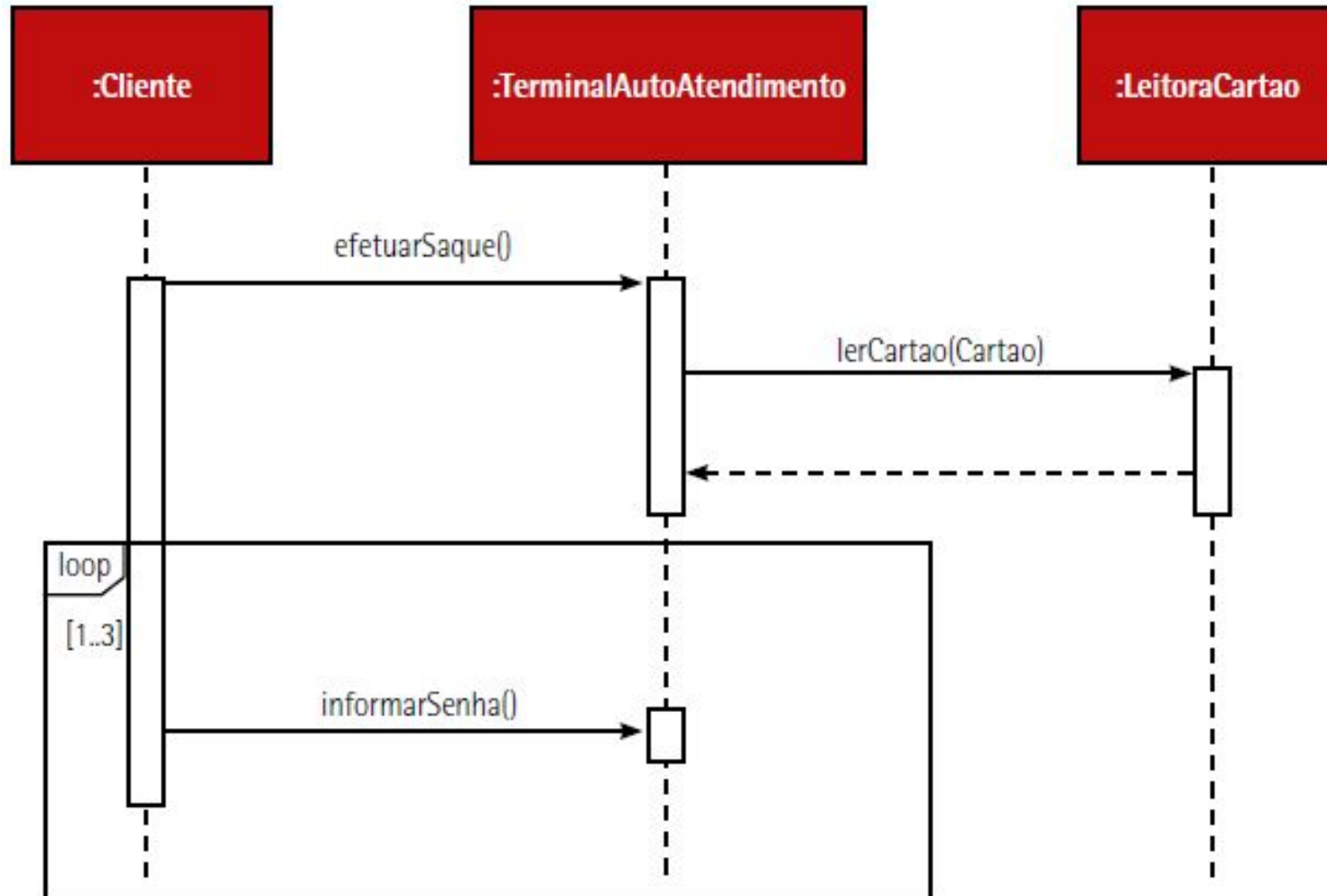
# Diagrama de Sequência

## ESTRUTURA DE REPETIÇÃO

- A próxima **figura** mostra um **exemplo** de **laço de repetição**.
- O **indicativo loop** no **operador** de **controle**, indicado pela **letra A**, sinaliza uma **repetição**.
- No caso, a mensagem **informarSenha** será **enviada** no **mínimo uma vez** e no **máximo três vezes**, conforme **indicado** no **operador** de **controle**.

# Diagrama de Sequência

## ESTRUTURA DE REPETIÇÃO



Exemplo de estrutura de repetição em diagrama de sequência

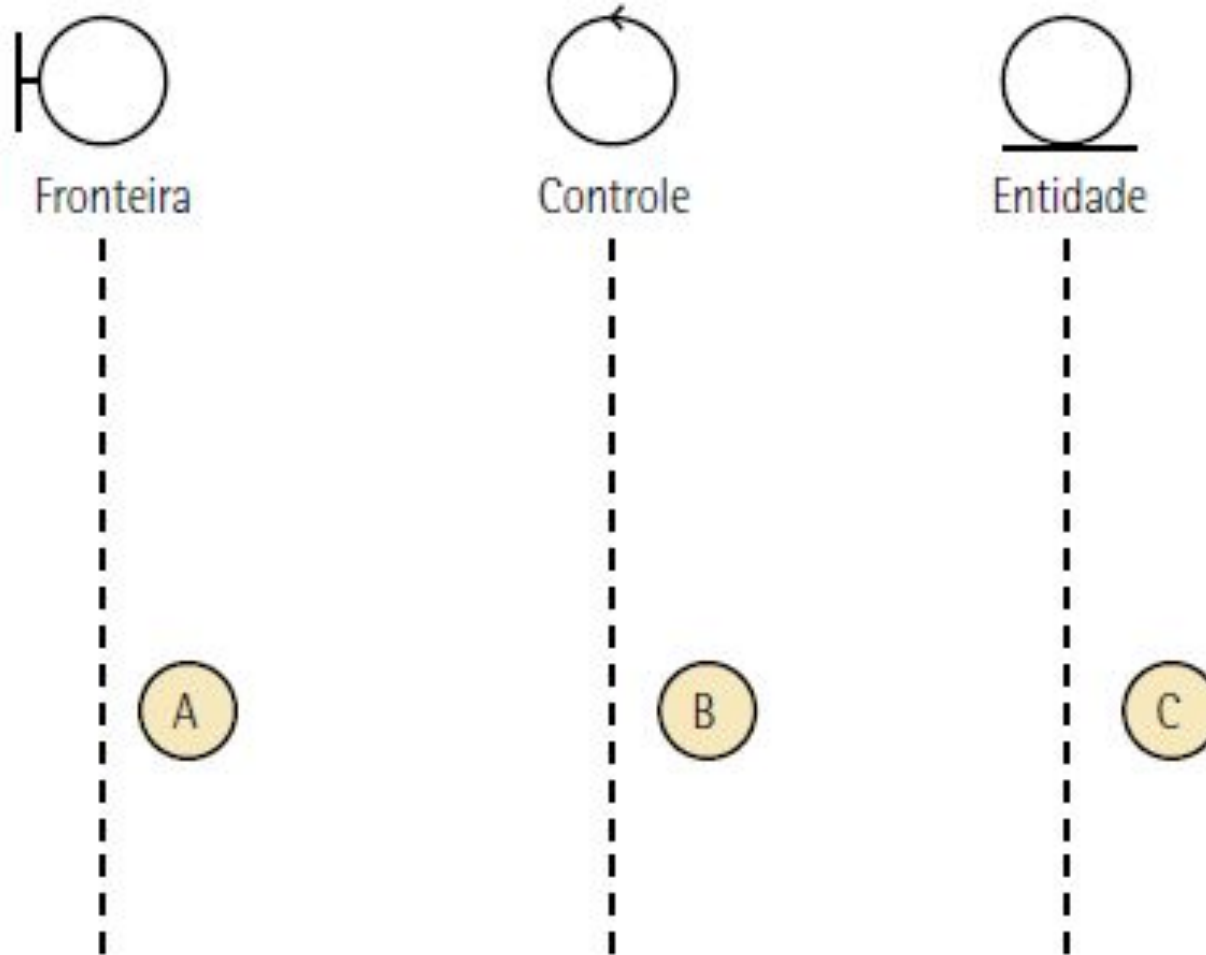
# Diagrama de Sequência

## ESTEREÓTIPOS DE CLASSES DE ANÁLISE

- As **classes** podem ser **classificadas e organizadas** de **acordo** com a **responsabilidade** que possuem **dentro** do **modelo**:
  - **ENTIDADE** (geralmente dados que vão para o banco de dados)
  - **CONTROLE** (fazem o “meio de campo”)
  - **FRONTEIRA** (geralmente nas interações com o usuário).
- No **diagrama de sequência** podemos **utilizar** os **estereótipos** que **auxiliam** na **interpretação** das **responsabilidades** dos **objetos**, como vemos na figura a seguir

# Diagrama de Sequência

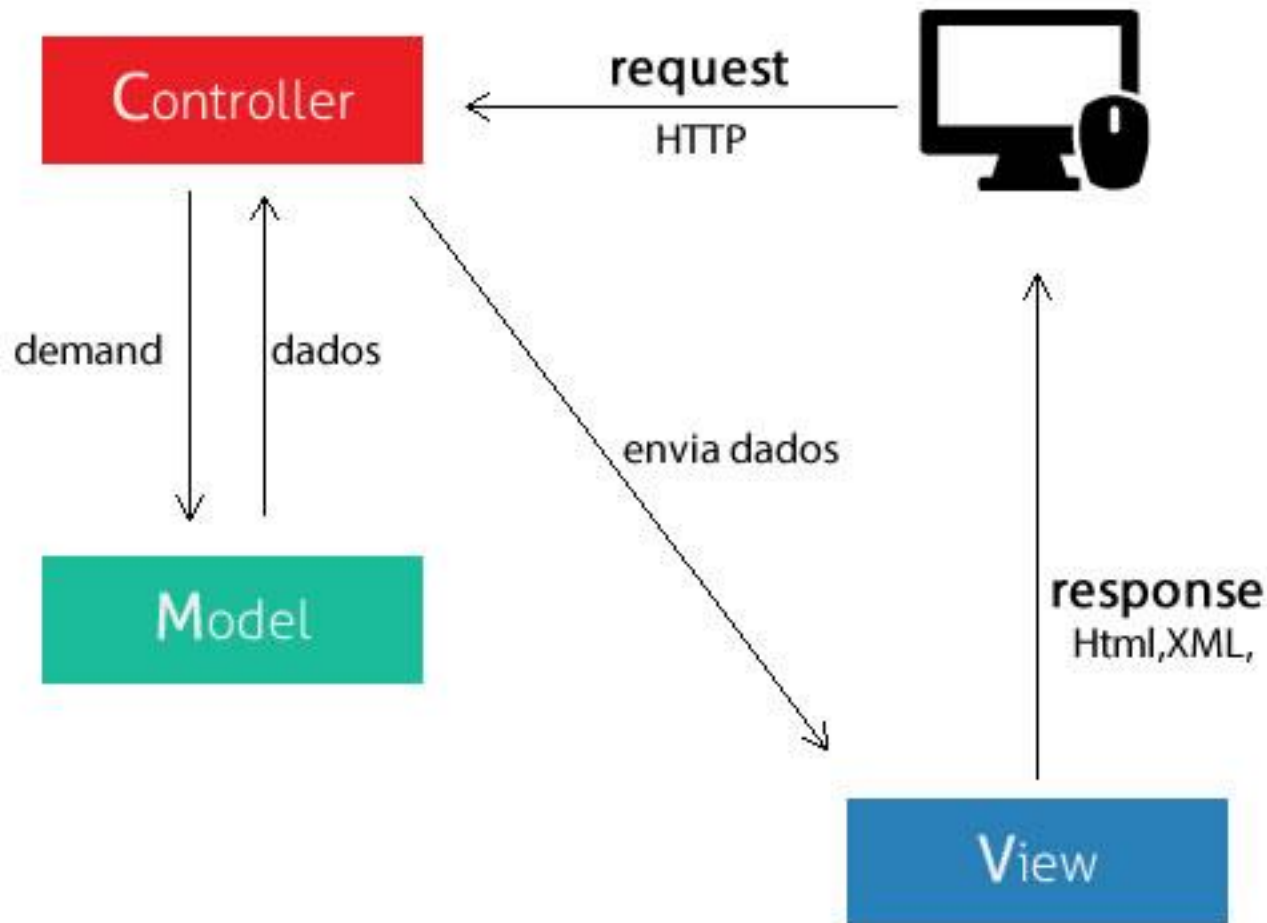
## ESTEREÓTIPOS DE CLASSES DE ANÁLISE



Estereótipos de classes de análise

# Diagrama de Sequência

## ESTEREÓTIPOS DE CLASSES DE ANÁLISE

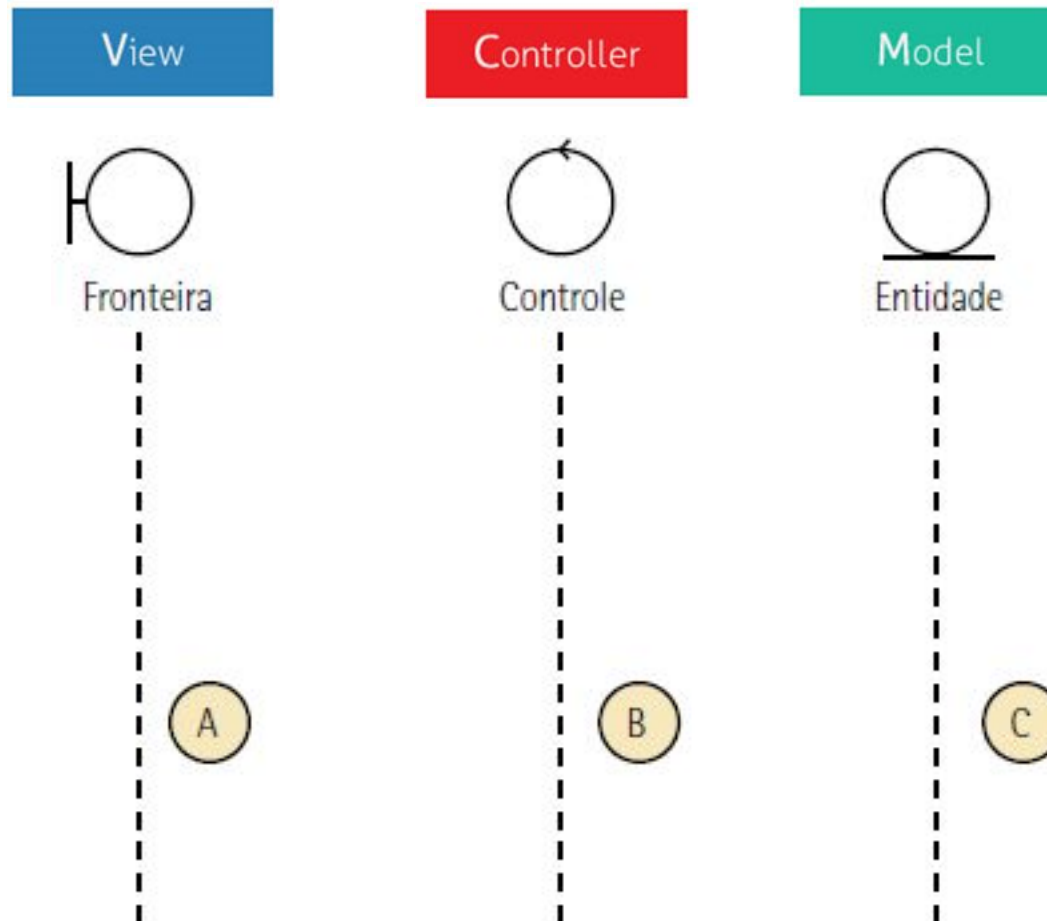


Exemplo de MVC



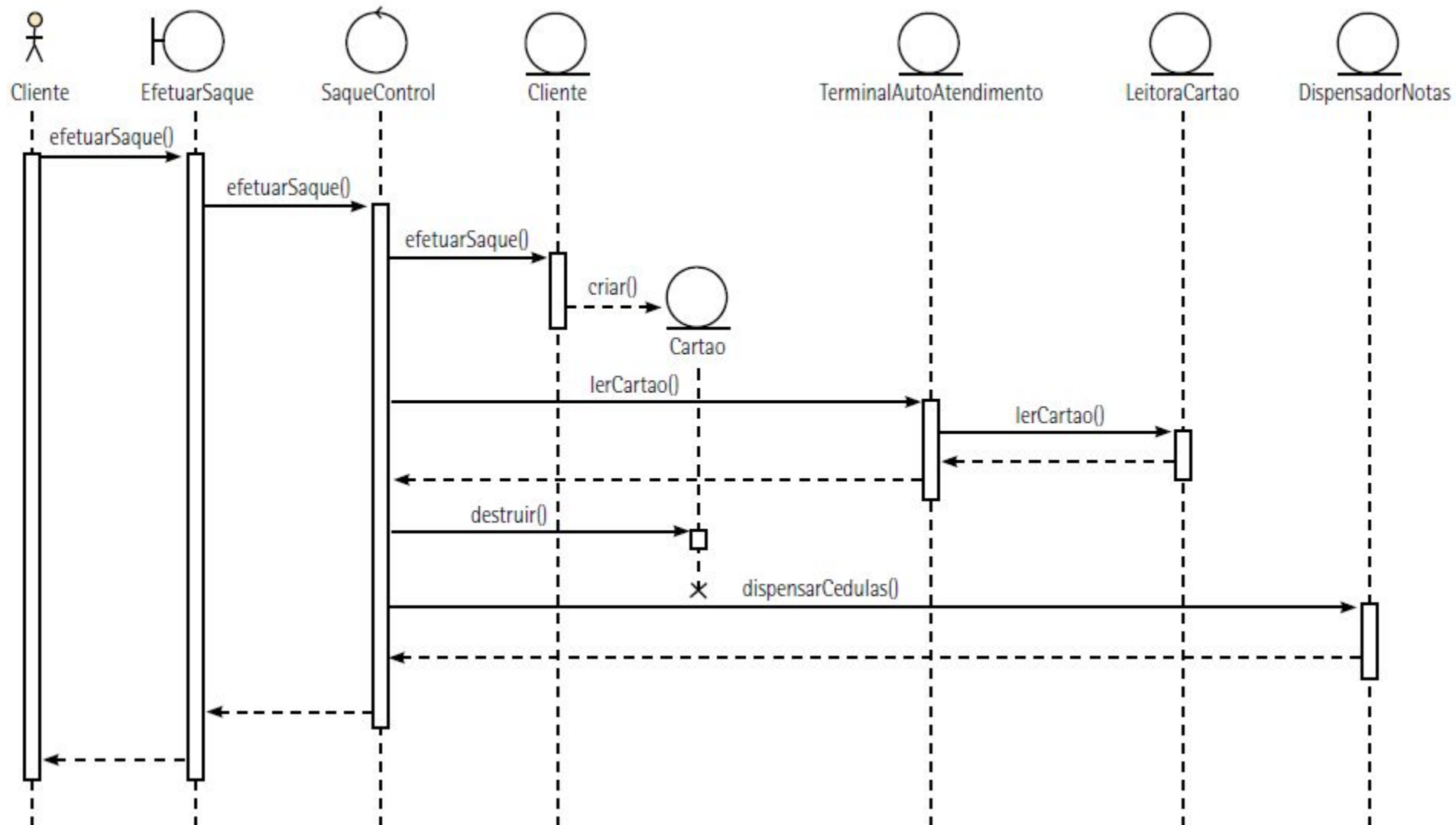
# Diagrama de Sequência

## ESTEREÓTIPOS DE CLASSES DE ANÁLISE



Estereótipos de classes de análise e analogia com MVC

# Diagrama de Sequência

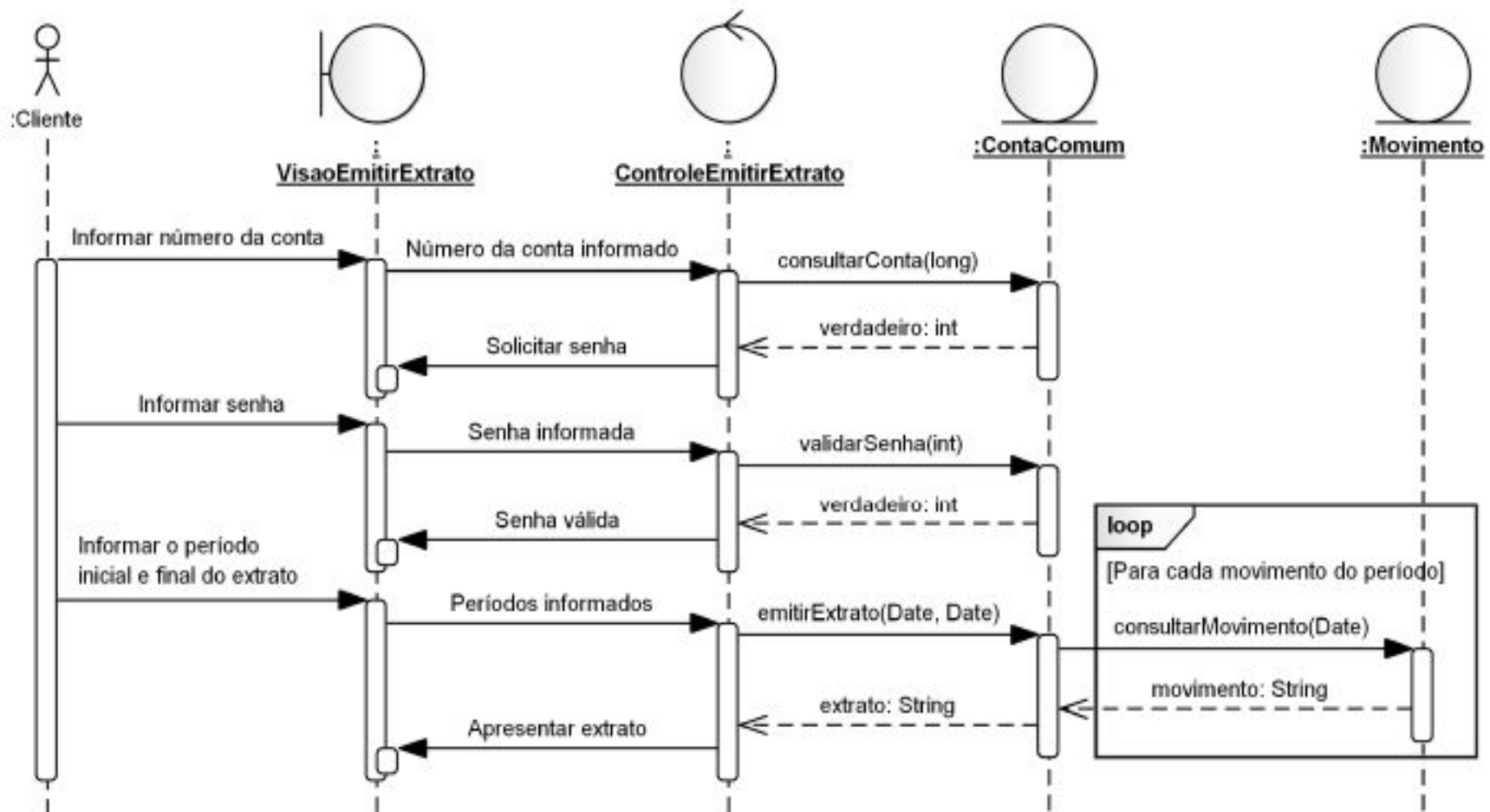


Exemplo de diagrama de sequência com estereótipos

# Diagrama de Sequência

## EXEMPLOS

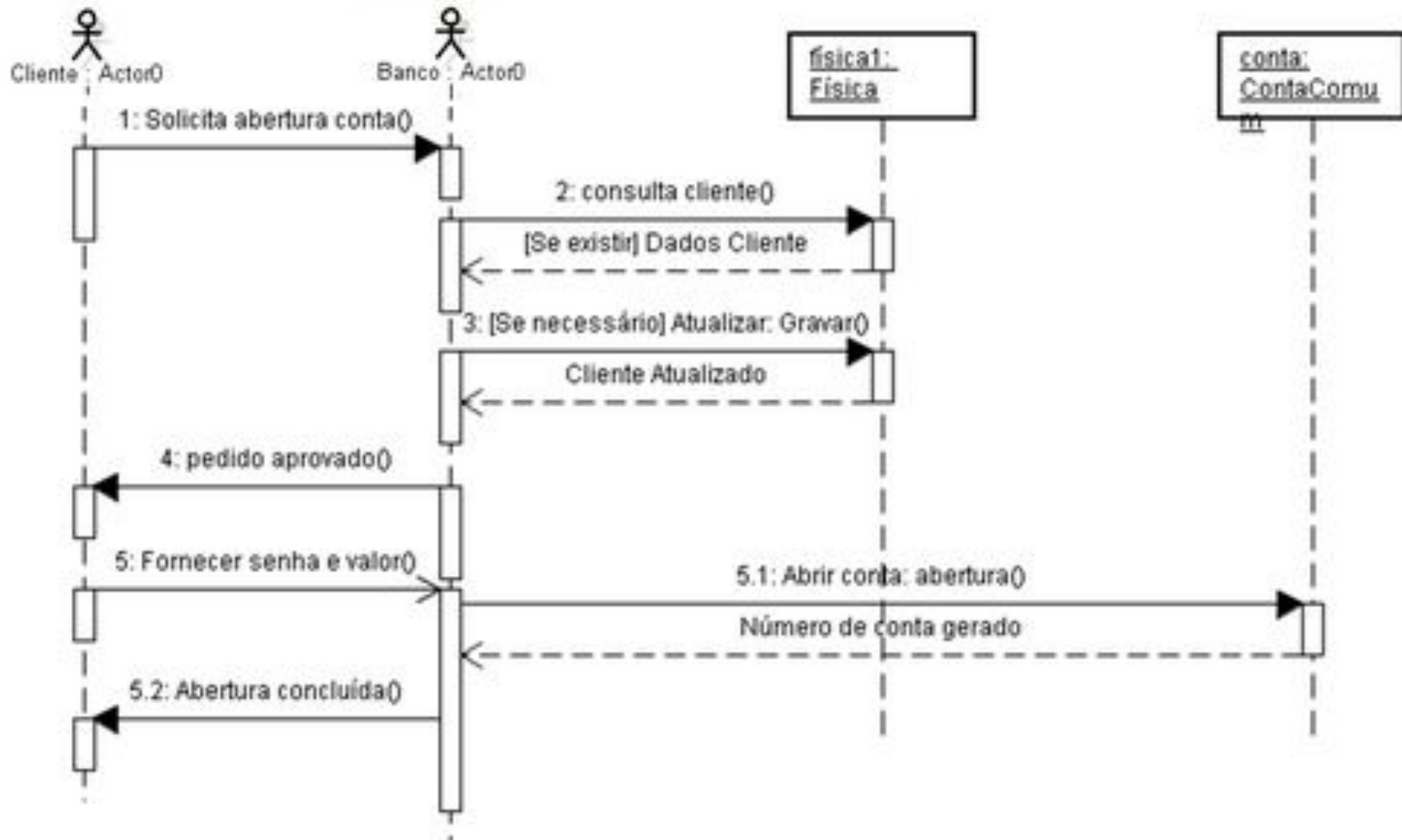
sd Emitir Extrato



Exemplo 1 de diagrama de sequência completo

# Diagrama de Sequência

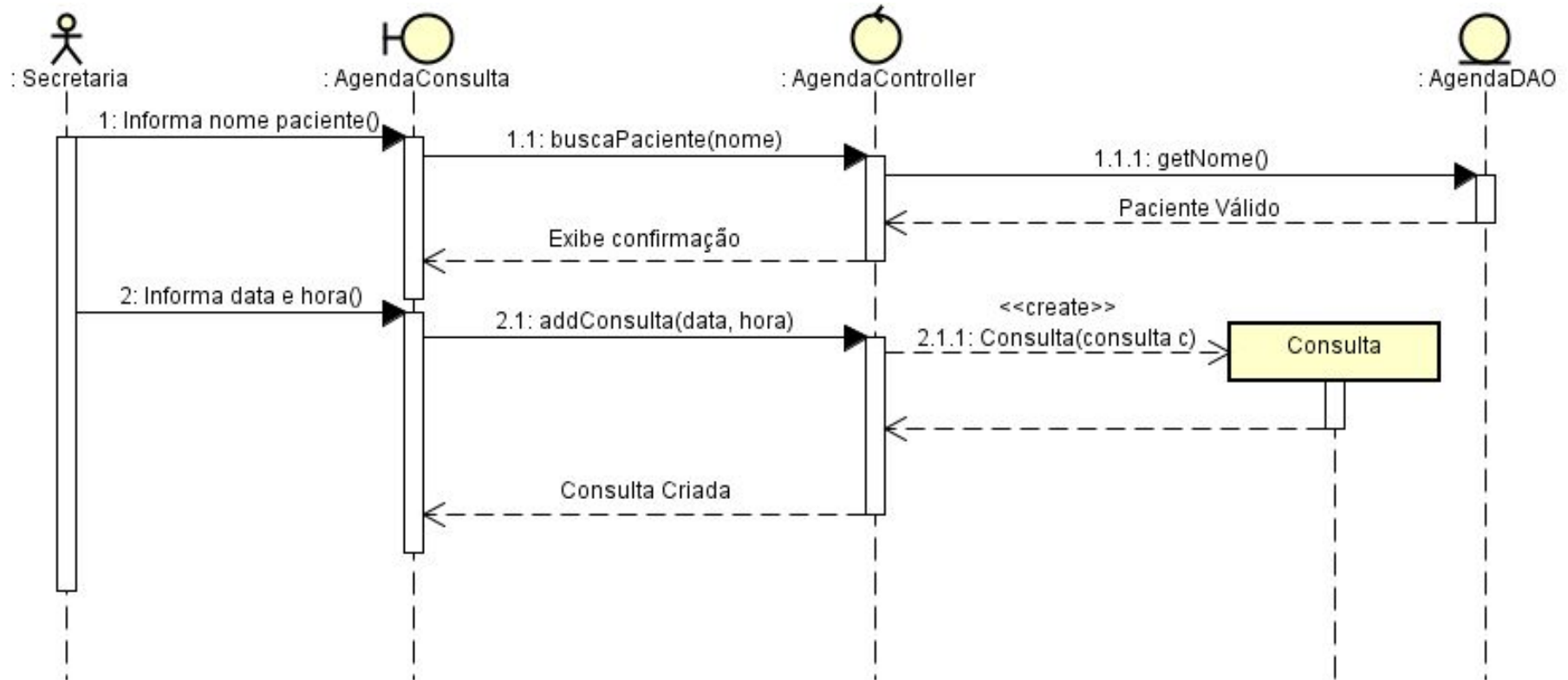
## EXEMPLOS



Exemplo 2 de diagrama de sequência completo

# Diagrama de Sequência

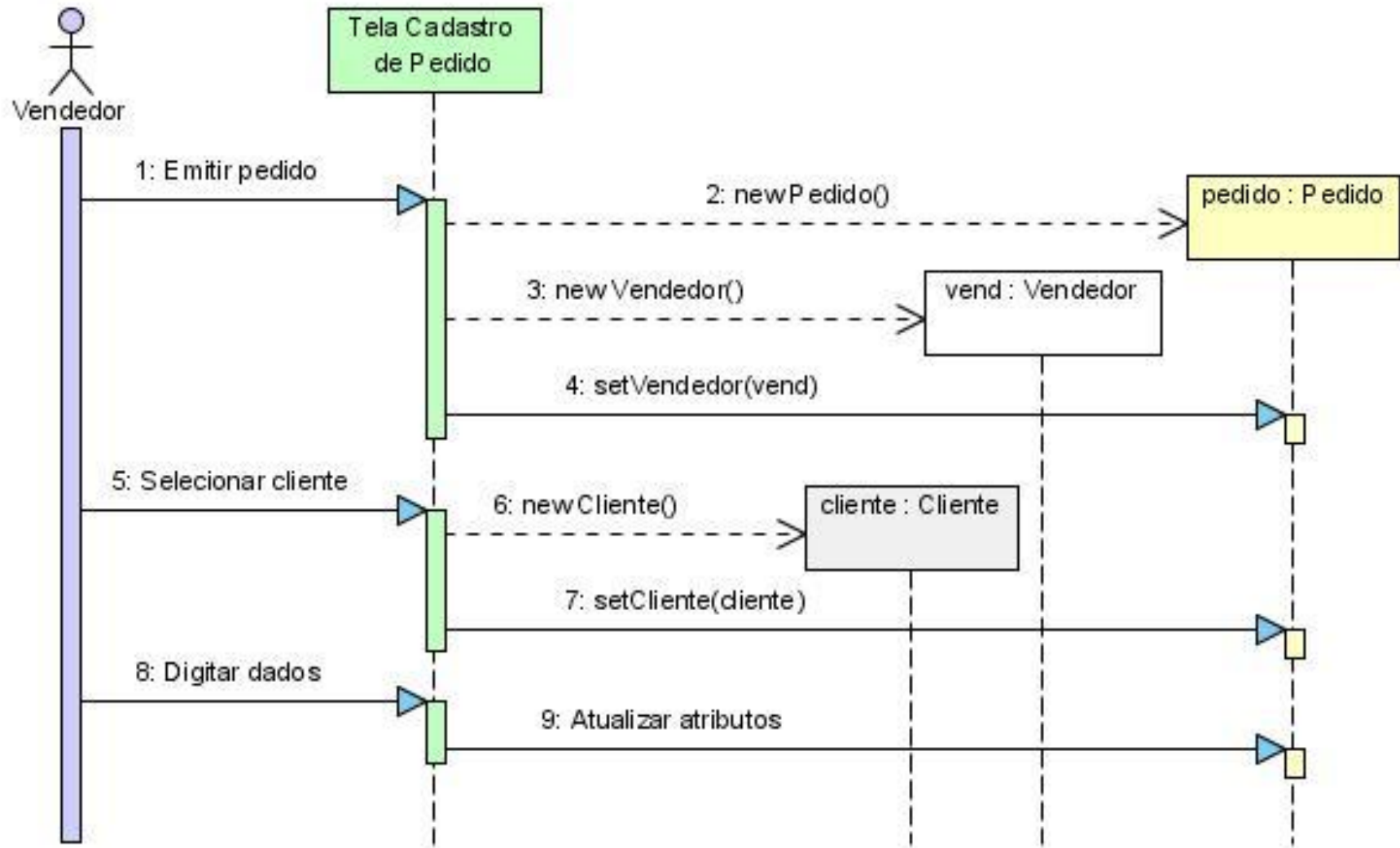
## EXEMPLOS



Exemplo 3 de diagrama de sequência completo

# Diagrama de Sequência

## EXEMPLOS

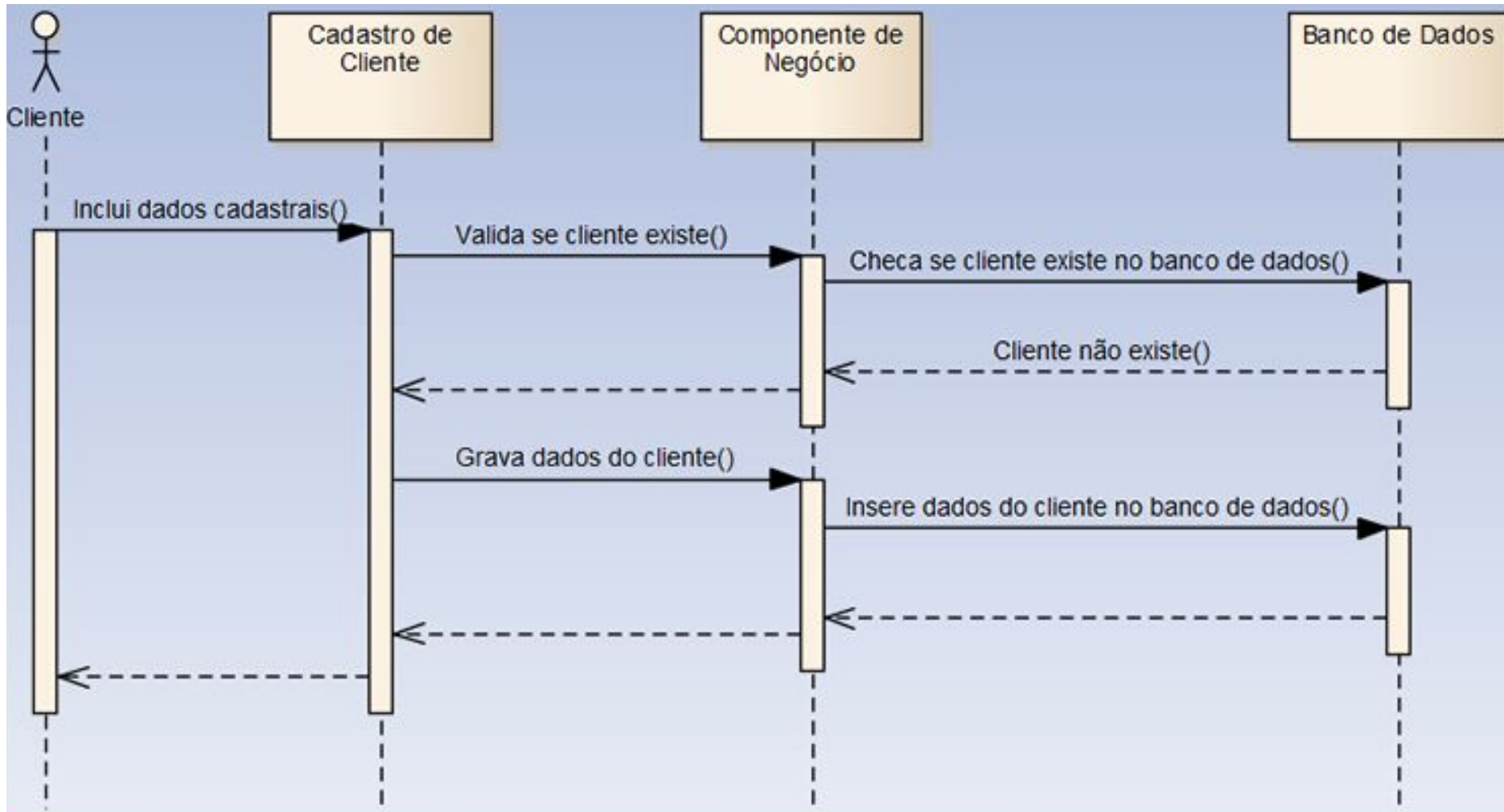


Exemplo 4 de diagrama de sequência completo

# Qual usar?

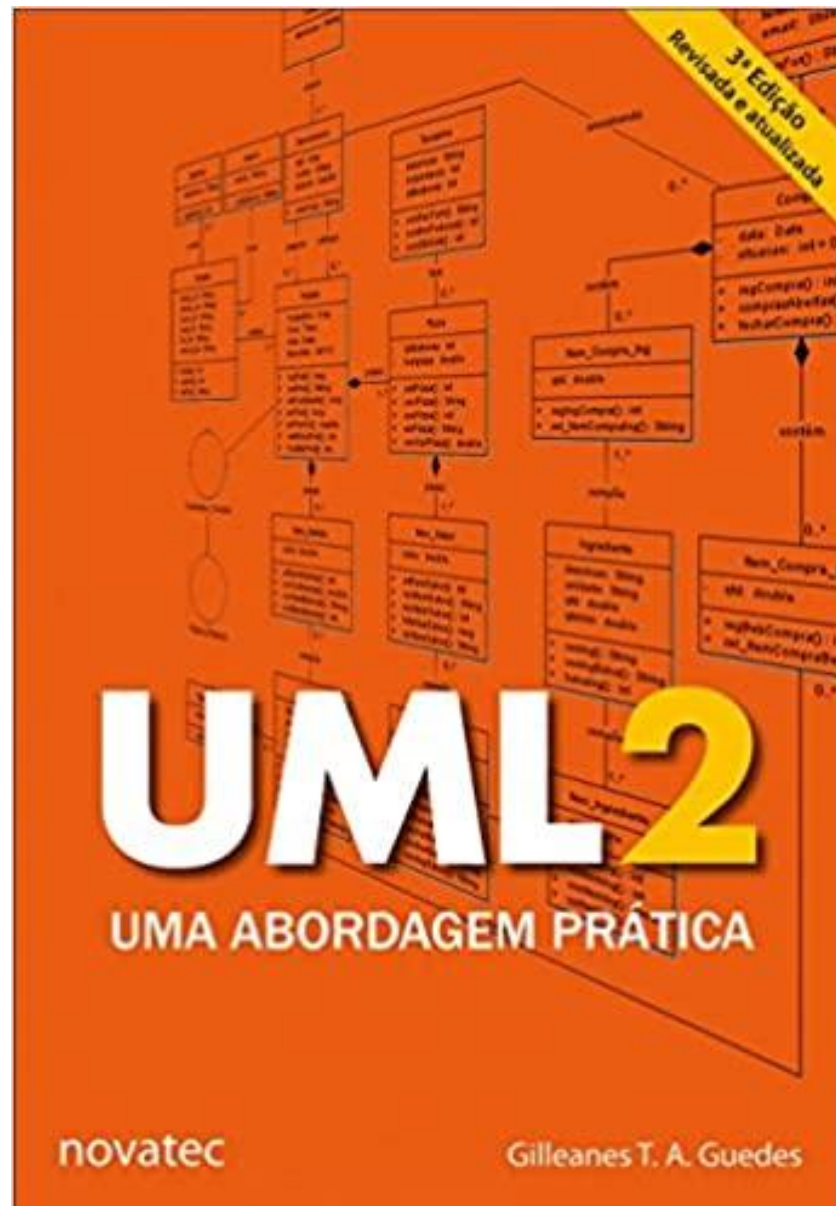
1. O usuário (ator) “Atendente” abre a página “Cadastro de Cliente”.
2. O usuário (ator) “Atendente” digita o nome do cliente e clica no comando “Pesquisar”.
3. A página “Cadastro de Cliente” aciona o método “CadastroClienteUI.PesquisarClientePorNome(...)” passando a expressão informada como termo de pesquisa.
4. A página “Cadastro de Cliente”, através do método “CadastroClienteUI.PesquisarClientePorNome(...)", instancia a classe de negócio e chama o método “ClienteNegocio.PesquisarClientePorNome(...)" passando a string com a expressão informada para pesquisa.
5. O método “ClienteNegocio.PesquisarClientePorNome(...)" instancia a classe de dados e chama o método “ClienteDados.PesquisarClientePorNome(...)".
6. Método “ClienteDados.PesquisarClientePorNome(...)" retorna uma lista com os registros encontrados na busca.
7. Método “ClienteNegocio.PesquisarClientePorNome(...)" recebe uma lista com os registros informados pelo método “ClienteDados.PesquisarClientePorNome(...)".
8. A página “Cadastro de Cliente”, através do método “CadastroClienteUI.PesquisarClientePorNome(...)" recebe uma lista com os registros informados pelo método “ClienteNegocio.PesquisarClientePorNome(...)".
9. A página “Cadastro de Cliente” exibe a lista de registros recebidos para o usuário chamador.

# Qual usar?



Exemplo de diagrama de sequência completo





Livro UML

# Atividade 8

O próximo passo agora será realizar a Atividade 6.

Nos vemos nela!



# Dúvidas!

**Profº. Me. Flávio Henrique Fernandes Volpon**  
**[flavio.volpon@docente.unip.br](mailto:flavio.volpon@docente.unip.br)**







**Obrigado!**

# Bibliografia

## I - BIBLIOGRAFIA BÁSICA

- PAULA FILHO, W. de P. **Engenharia de software: fundamentos, métodos e padrões**. 3.ed. Rio de Janeiro: LTC, 2012.
- PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed. AMGH, 2011.
- SOMMERVILLE, I. **Engenharia de Software**. 9.ed. São Paulo: Adison-Wesley, 2011.

## II - BIBLIOGRAFIA COMPLEMENTAR

- PRIKLADNICKI., Rafael, WILLI, Renato, and MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Bookman, 2014.
- COHN, M. **Desenvolvimento de Software com Scrum**. Bookman, 2011.
- SCHACH, S. R. **Engenharia de software: os paradigmas clássico e orientado a objetos**. 7.ed. São Paulo: McGraw-Hill, 2009.
- HIRAMA, K. **Engenharia de software: qualidade e produtividade com tecnologia**. Rio de janeiro, campus, 2011.
- WAZLAWICK, R. **Engenharia de software: conceitos e práticas**. Rio de janeiro, campus, 2009.