

Programação Orientada a Objetos – Aula 15

Prof. Dr. Eduardo Takeo Ueda
eduardo.tueda@sp.senac.br

Diagrama de Sequência

- Procura determinar a sequência de eventos que ocorre em um determinado processo
- Identifica quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem
- Baseia-se no caso de uso, havendo normalmente um diagrama de sequência para cada caso de uso

Diagrama de Sequência

- Pode ser utilizado em dois momentos:
 - Fase de análise – identificar eventos e operações do Sistema
 - Auxiliado por diagramas de caso de uso
 - Fase de projeto – descrever como objetos colaboram para realizar uma funcionalidade do Sistema
 - Auxiliado por diagramas de classes

Diagrama de Sequência na Análise

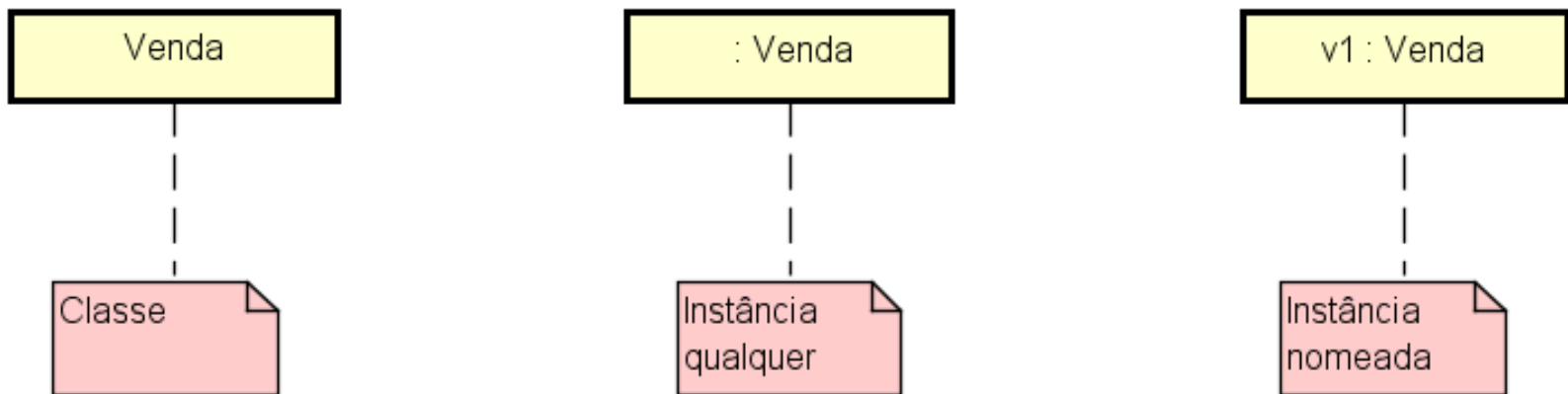
- Ilustra eventos partindo de atores e estimulando o sistema
- São elaborados para casos de uso mais importantes e situações cruciais
- Mostram eventos gerados por atores externos, sua ordem, além de eventos envolvendo outros sistemas

Diagrama de Sequência no Projeto

- Na fase de projeto, são utilizados para mostrar como os objetos colaboram trocando mensagens
- É útil para achar possíveis alterações ou correções no diagrama de classes do sistema
- Lembre-se que é menos custoso corrigir na fase de projeto do que implementação

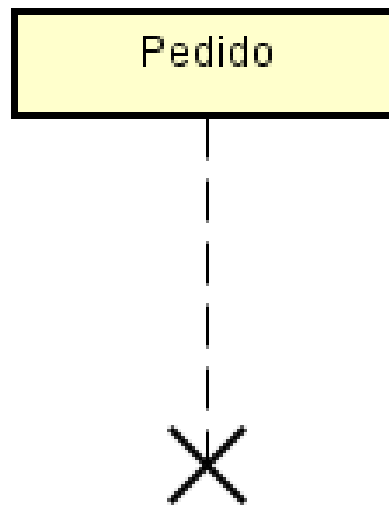
Objeto

- Um objeto pode existir desde o início do processo ou ser criado durante a execução do mesmo



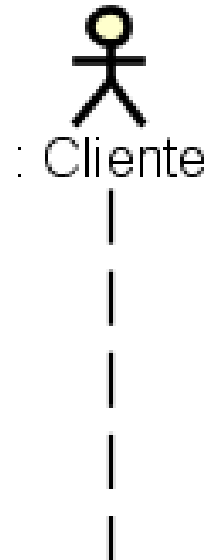
Linha de vida

- Linha tracejada projetada para baixo
- Representa ciclo de vida de um objeto durante uma interação
- É interrompida com um **X** quando o objeto é destruído



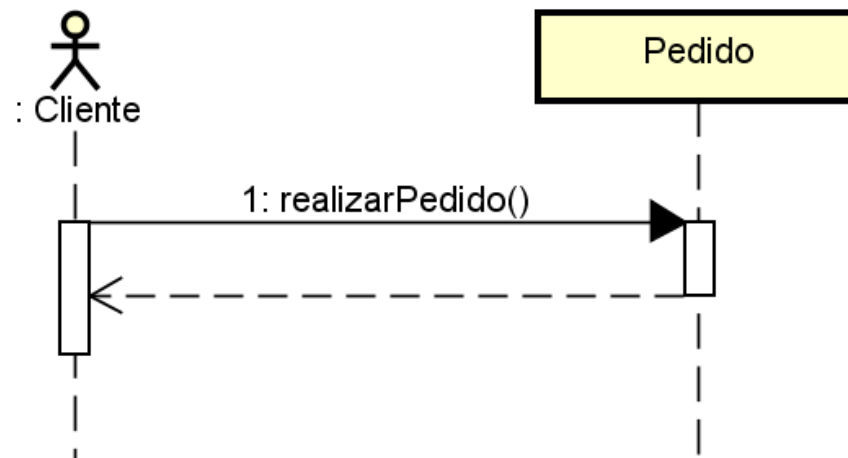
Atores

- São os mesmos usados nos casos de uso, porém contendo uma linha de vida



Mensagens

- É representada por uma linha seta dirigida horizontal entre as linhas de vida de dois objetos
- As linhas são acompanhadas por um rótulo que contém o nome da mensagem e, opcionalmente, os parâmetros da mesma



Mensagens Síncronas e Assíncronas

■ Mensagens síncronas

- O emissor fica parada esperando uma resposta
- Representada por uma linha contínua com seta preenchida

■ Mensagens de retorno

- Ocorre apenas após uma mensagem síncrona
- Representada por uma linha pontilhada com seta aberta

Mensagens Síncronas e Assíncronas

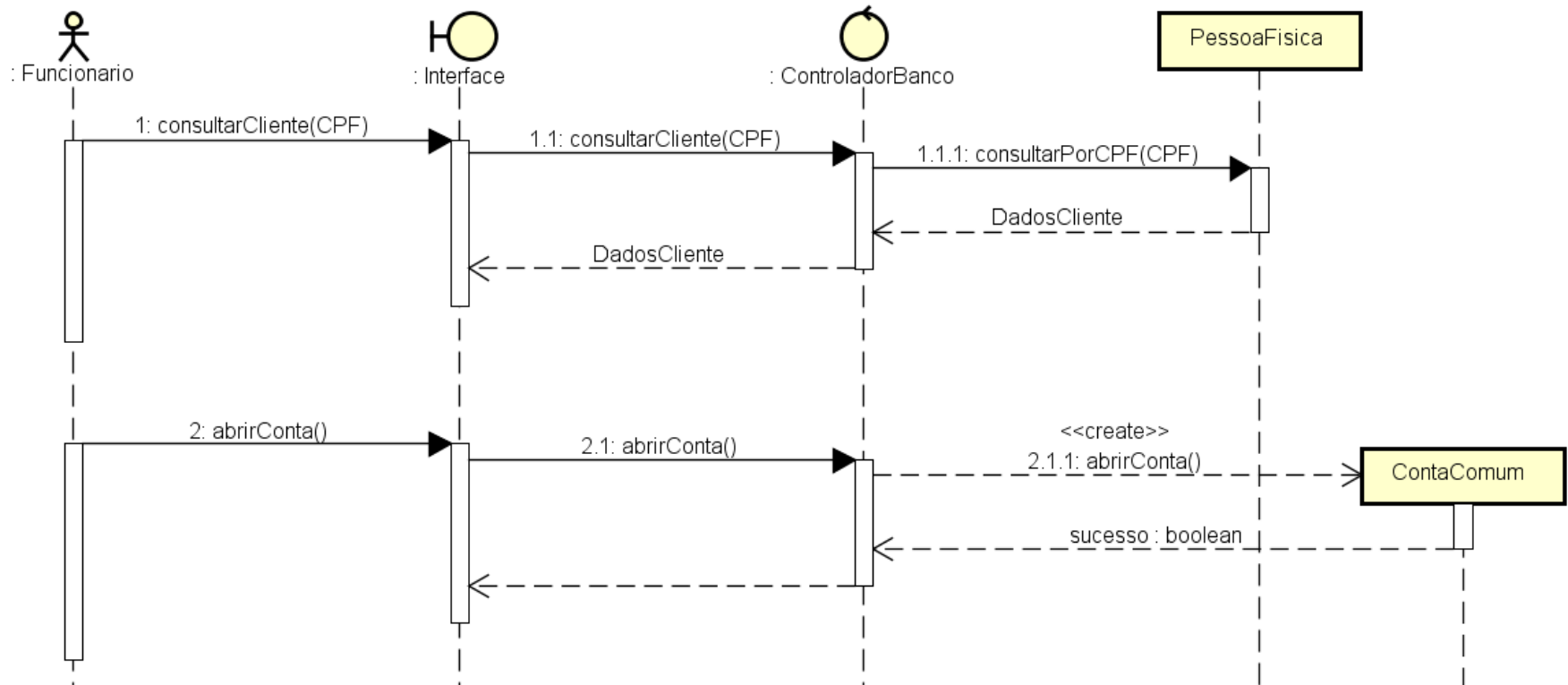
■ Mensagens assíncronas

- O emissor não fica parado à espera de uma resposta
- Representada por uma linha contínua com seta aberta

■ Mensagens de instanciação

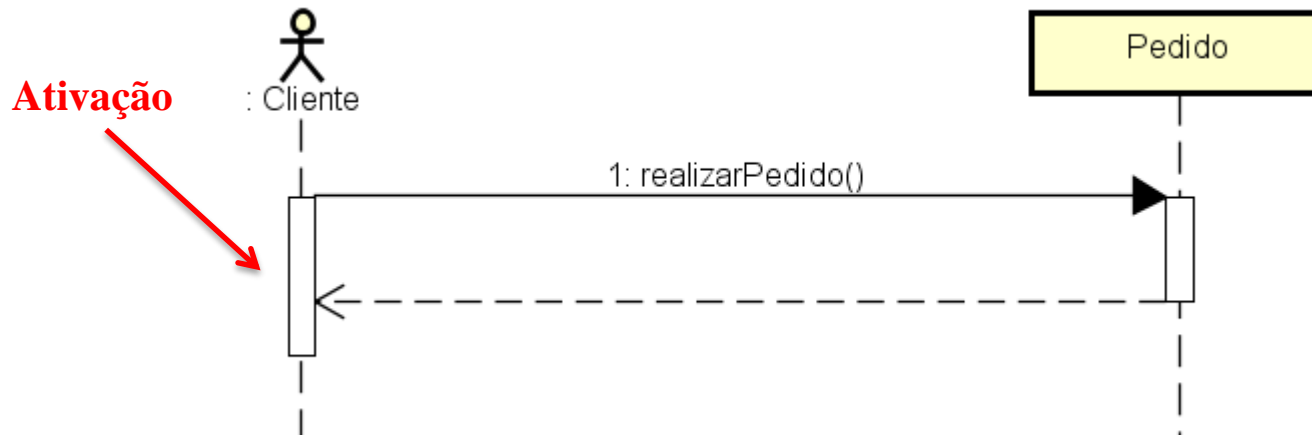
- Cria um novo objeto
- Representada por uma linha pontilhada com seta aberta

Exemplo: Abertura de Conta



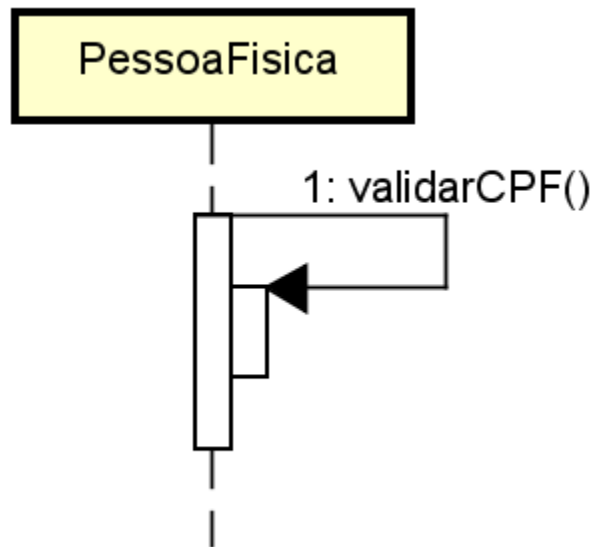
Foco de controle (ou ativação)

- Indica o período de tempo em que determinado objeto está participando ativamente do processo
- Determina a janela de tempo que um objeto está executando diretamente uma ação



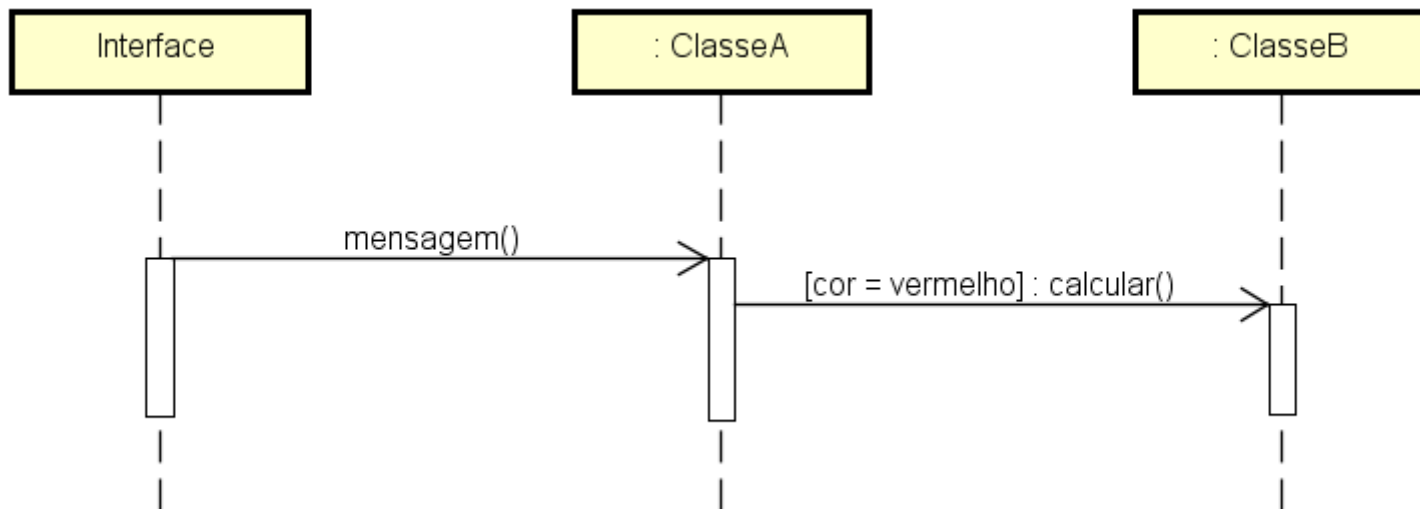
Auto chamada

- Também conhecido como auto delegação ou ou chamada reflexiva
- São mensagens que um objeto envia para si mesmo



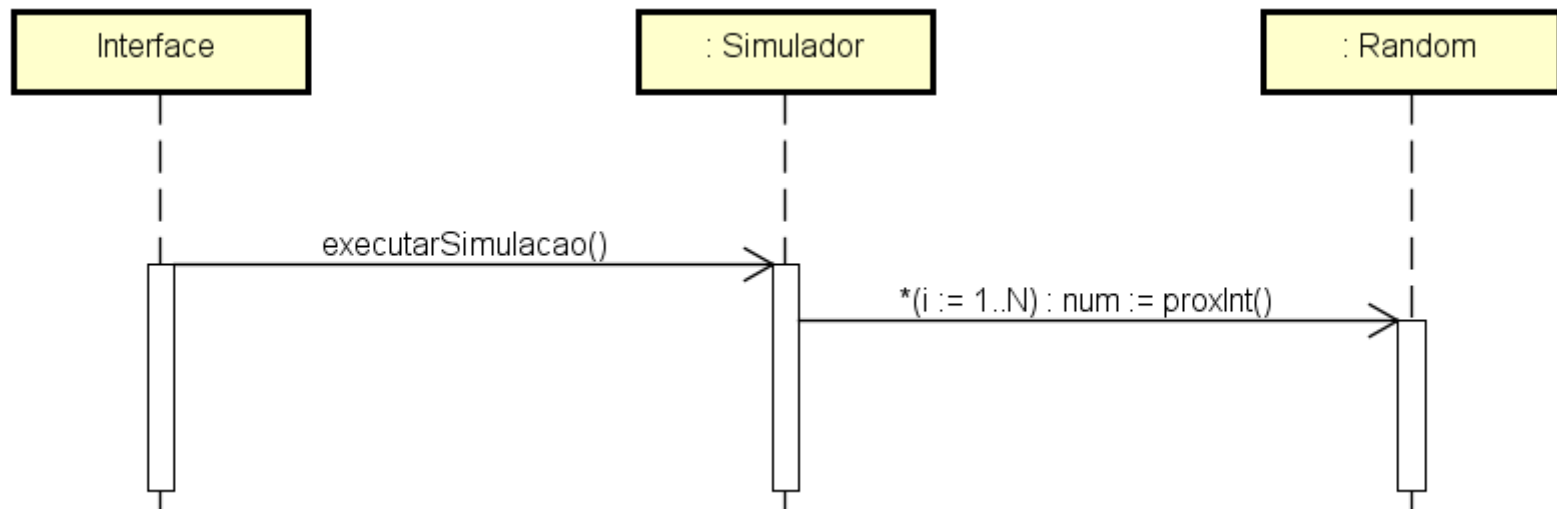
Condições

- Condiciona o envio de uma mensagem
- A mensagem sob uma condição só é enviada se a condição for satisfeita
- Representado de modo textual entre colchetes



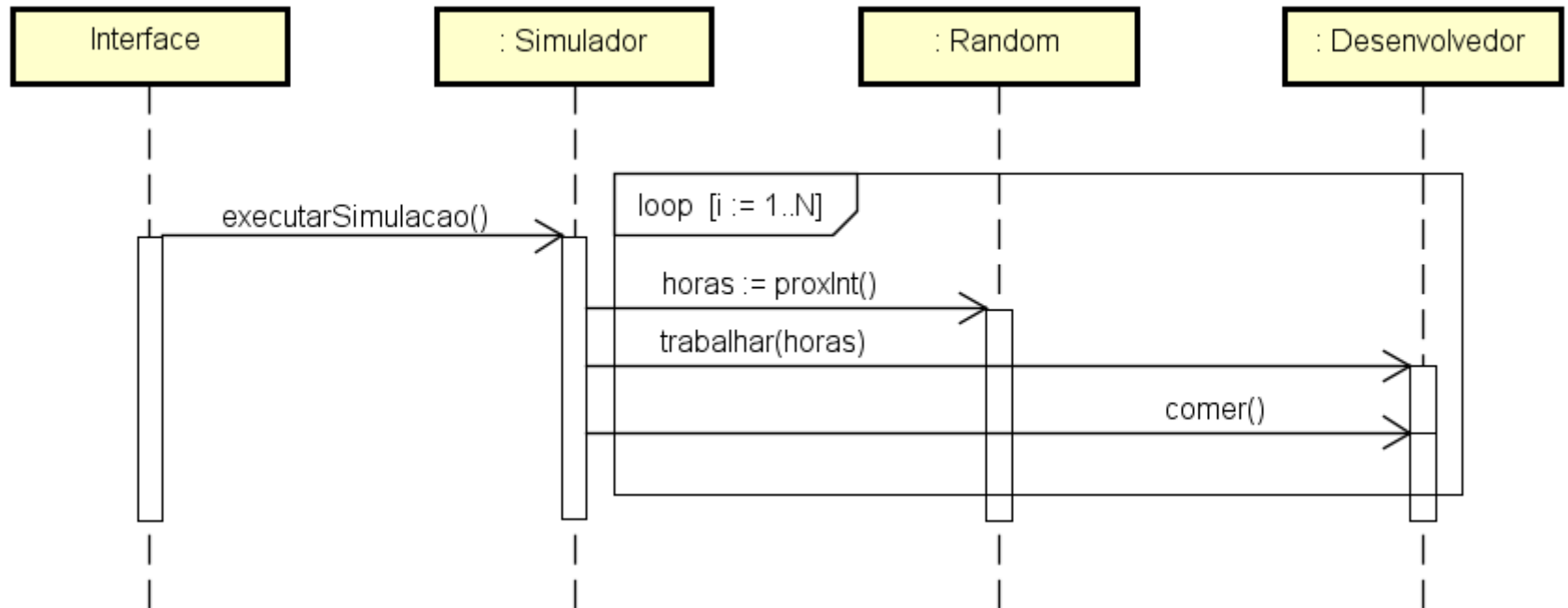
Iteração para uma Mensagem Única

- Quando uma mensagem é enviada repetidas vezes, indica-se com um asterisco seguido da condição da repetição
- A condição pode ser uma expressão lógica ou a especificação de um laço controlado



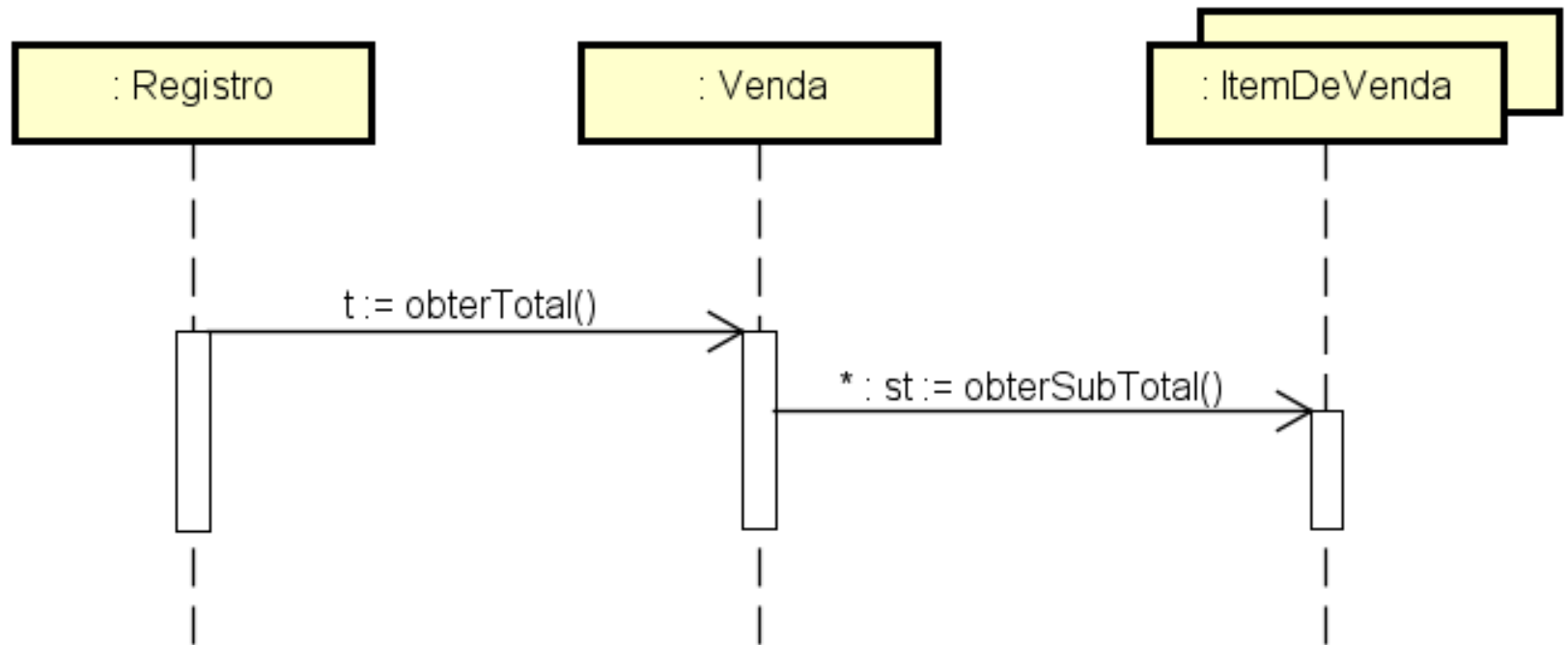
Iteração de uma Série de Mensagens

- Cerca-se o conjunto de mensagens que se repete com um retângulo e adiciona-se a condição da repetição



Iteração sobre uma Coleção

- Uma mensagem enviada para cada elemento de uma coleção de objetos



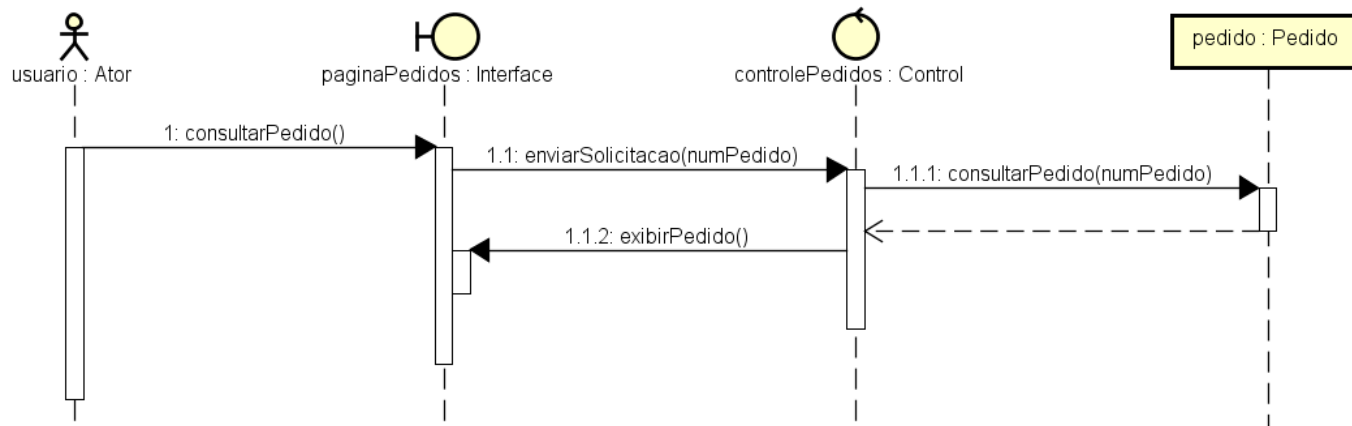
Estereótipos

■ <<boundary>>

- indica uma classe fronteira que comunica atores externos e o sistema
- muitas vezes é associado a própria interface e geralmente interage com classes <<control>>

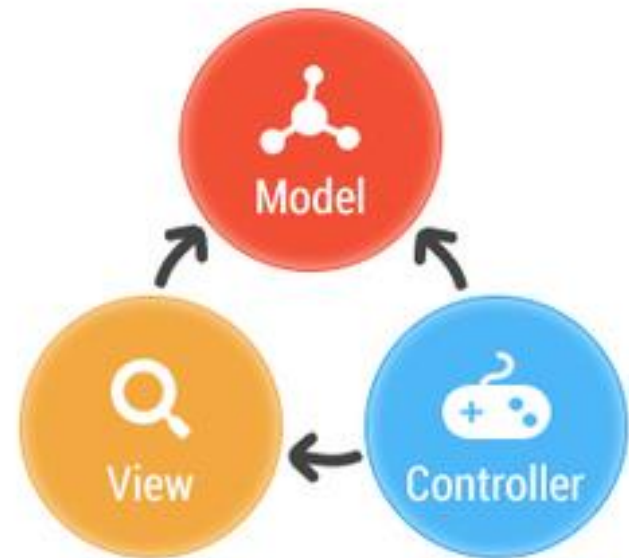
■ <<control>>

- intermediária entre a classe <<boundary>> e as outras do sistema
- responsável por interpretar eventos ocorridos sobre objetos <<boundary>>

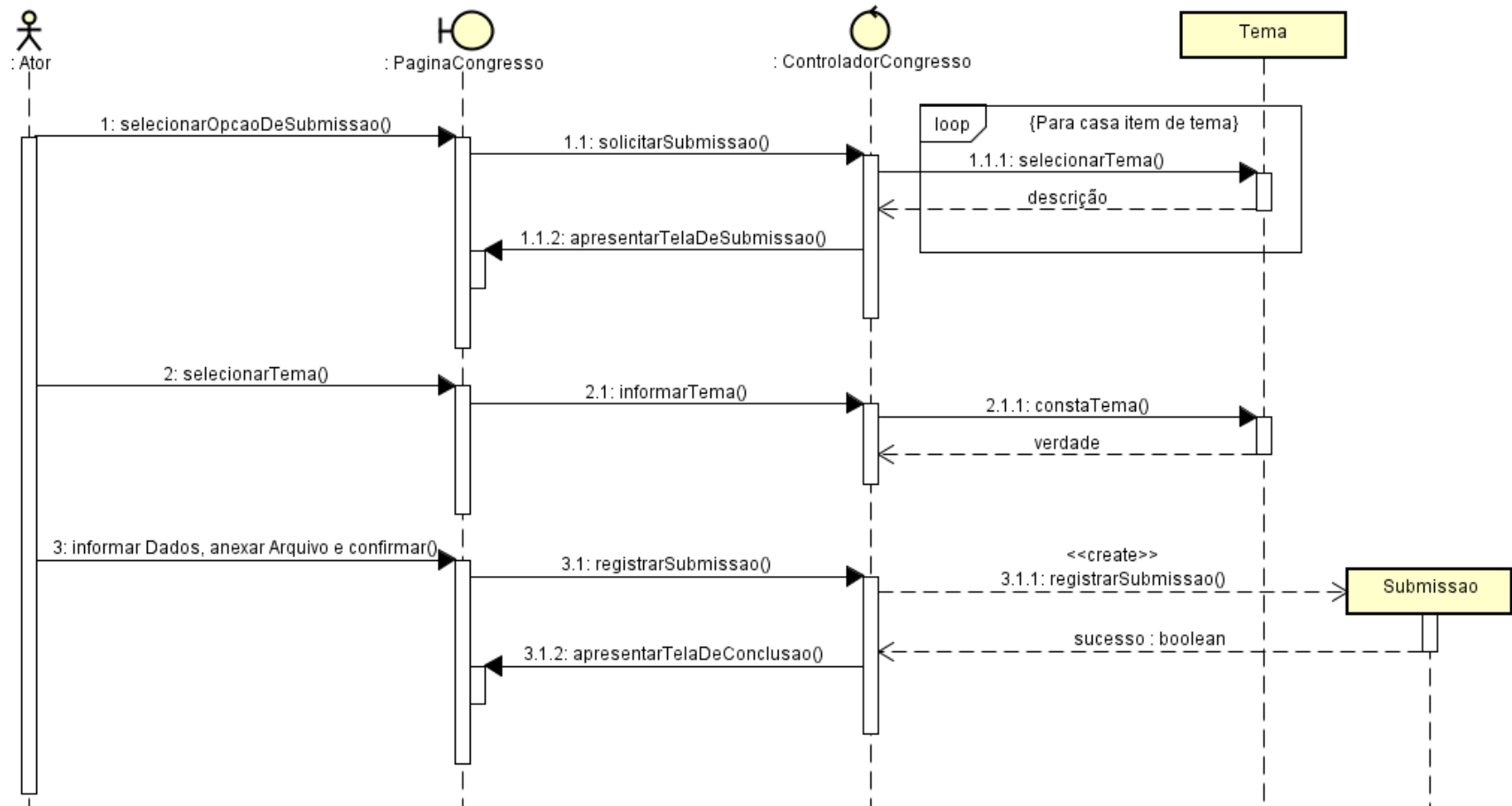


Padrão MVC (Model-View-Control)

- **Modelo** – Modelagem de dados (diagrama de classes, por exemplo)
- **Visão** – Apresentação da aplicação (interface com usuário)
- **Controle** – Processamento da aplicação (controla visão e modelo)



Exemplo: Submissão em Congresso



Categoria	Diagrama	Descrição
Diagramas dinâmicos	Visão geral	Representa o Modelo de Processos de Negócio de um sistema
	Casos de uso	Expressam a funcionalidade de um sistema
	Atividades	Representam o fluxo de atividades dos processos de negócio
	Sequência	Define a ordem e a troca das mensagens entre objetos
	Comunicação	Representa o diagrama anterior de colaboração
	Máquina de estados	Representa as ações ocorridas em resposta ao recebimento de eventos
	Temporal	Mostra mudança de estado de um objeto
Diagramas estruturais	Classes	Apresenta elementos conectados por relacionamentos
	Objetos	Apresenta objetos e valores de dados
	Componentes	Mostra dependências entre componentes de software
	Pacotes	Usado para organizar elementos de modelos e mostrar dependências entre eles
	Implantação	Mostra a arquitetura do sistema em tempo de execução, as plataformas de software, etc.
	Estrutura composta	Usado para mostrar a composição de uma estrutura complexa.