



Workflow de Análise e Projeto

Diagrama de Sequência

Workflows do Processo Unificado



- Entrar em acordo sobre o contexto do sistema
- Elicitar e especificar Requisitos

Requisitos

- Escopo do sistema
- Lista de funções
- Diagrama de casos de uso
- Especificação de casos de uso
- Modelo conceitual
- Diagrama de atividades

- Foco distancia do cliente
- Foco em o que os desenvolvedores precisam para construir o sistema correto
- Estruturar os requisitos de uma forma que facilite o entendimento técnico
- Pré-projeto

Análise

- Classes de Análise
- Realização de casos de uso de análise
 - Diagrama de sequência
- Diagrama de classes Final

Projeto

Codificação

Teste

Requisitos versus Análise e Projeto

<u>Modelo de Casos de Uso</u>	<u>Modelo de Análise</u>
Descrito utilizando a linguagem do cliente.	Descrito utilizando a linguagem do desenvolvedor.
Visão externa do sistema.	Visão interna do sistema.
Estruturado em casos de usos – oferece estrutura para a visão externa.	Estruturado em classes estereotipadas e pacotes – oferece estrutura a visão interna.
Utilizado como um contrato entre clientes e desenvolvedores sobre o que o sistema deve ou não fazer.	Utilizado pelos desenvolvedores para entender como o sistema deve ser concebido (projetado e implementado).
Pode conter redundâncias, inconsistências, etc., entre requisitos.	Não deve conter redundâncias, inconsistências, etc., entre requisitos.
Captura as funcionalidades do sistema, incluindo funcionalidades significantes do ponto de vista de arquitetura.	Esboça como concretizar os casos de uso no sistema, incluindo funcionalidades significantes do ponto de vista de arquitetura – funciona como uma prévia do projeto.
Define os casos de uso que serão futuramente analisados no modelo de análise.	Define a concretização dos casos de uso, cada um representando a análise do modelo de caso de uso.

Workflow de Análise

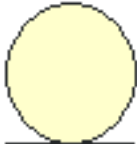
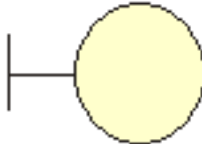




- Descobrir as classes de análise.
- Realização dos casos de uso: descobrir como concretizar os casos de uso através das classes

Classes de Análise

- As classes de análise são óbvias no contexto do domínio do problema. São mais conceituais e de granularidade maior do que as classes de projeto e implementação.
- Definem responsabilidades – dificilmente incluem operações.
- Definem atributos, em um nível mais alto nível. Os atributos são conceituais e reconhecíveis no domínio do problema. Podem se tornar classes no projeto e implementação.
- São inicialmente obtidas a partir do modelo de objetos de negócios.

Classes de Análise

- São modeladas em termos de estereótipos, são eles:
 - Limite ou interface
 - Controle, controladora ou gerenciadora
 - Entidade

Entidade	Limite/Interface	Controle
 Reserva	 Interface de Reserva	 Gerenciador de Reserva
 Reserva	 Interface de Reserva	 Gerenciador de Reserva
<<entity>> Reserva	<<boundary>> Interface de Reserva	<<control>> Gerenciador de Reserva

Classes de Análise – interface/limite/boundary

- Classes de interface com o usuário – classes que realizam a interface entre o sistema e humanos.
- Classes de interface com sistemas – classes que realizam a interface com outros sistemas.
- Classes de interface com dispositivos – classes que realizam a interface com dispositivos externos como sensores.

Classes de Análise – controle/control

- São classes coordenadoras que representam um comportamento do sistema.
- Essas classes são encontradas analisando-se o comportamento do sistema a partir dos casos de uso.
- São usualmente chamadas de controlador ou gerenciador. Ex. Gerenciador de pedidos, Escalonador e Gerenciador de processos.

Classes de Análise – entidade/entity

- Modelam informações persistentes que comporão o banco de dados do sistema. Ex. cliente, livro e conta corrente.
- São inicialmente encontradas na concepção do Modelo de Objetos de negócio.
- São classes que:
 - Cruzam vários casos de uso;
 - São manipuladas por classes de controle;
 - Oferecem e recebem informações das classes limite por meio das classes de controle;
 - Representam elementos chave gerenciados pelo sistema;
 - São geralmente persistentes.

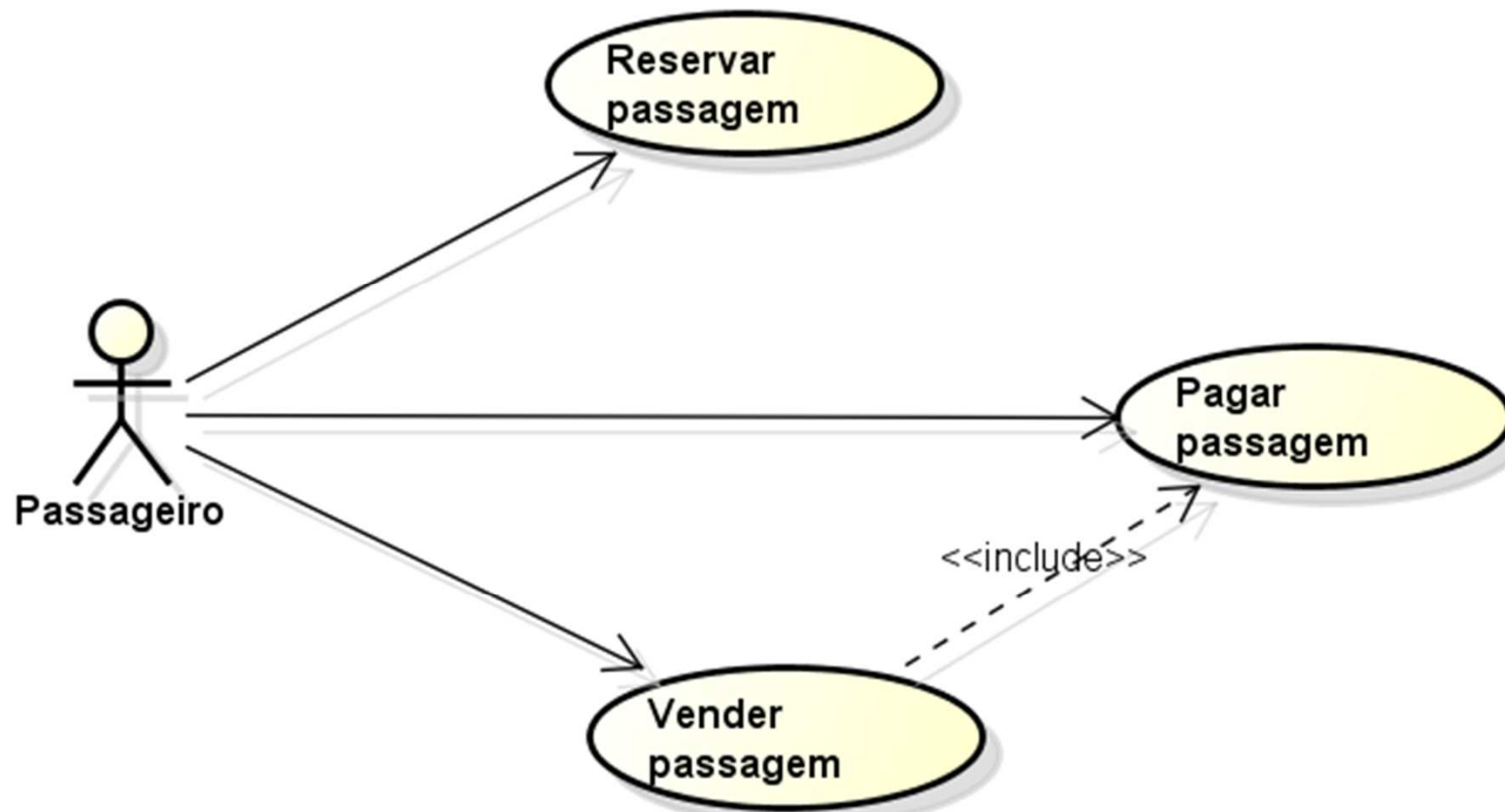
Dicas para criar classes de análise bem formadas (Jim Arlow & Ila Neustadt)

- As classes têm de três a 5 responsabilidades (operações).
- Classes não são solitárias – classes colaboram entre si.
- Fique atento
 - para a existência de muitas classes pequenas ou para classes muito grandes – encontre o balanço certo.
 - Classes onipotentes – pretendem fazer tudo.

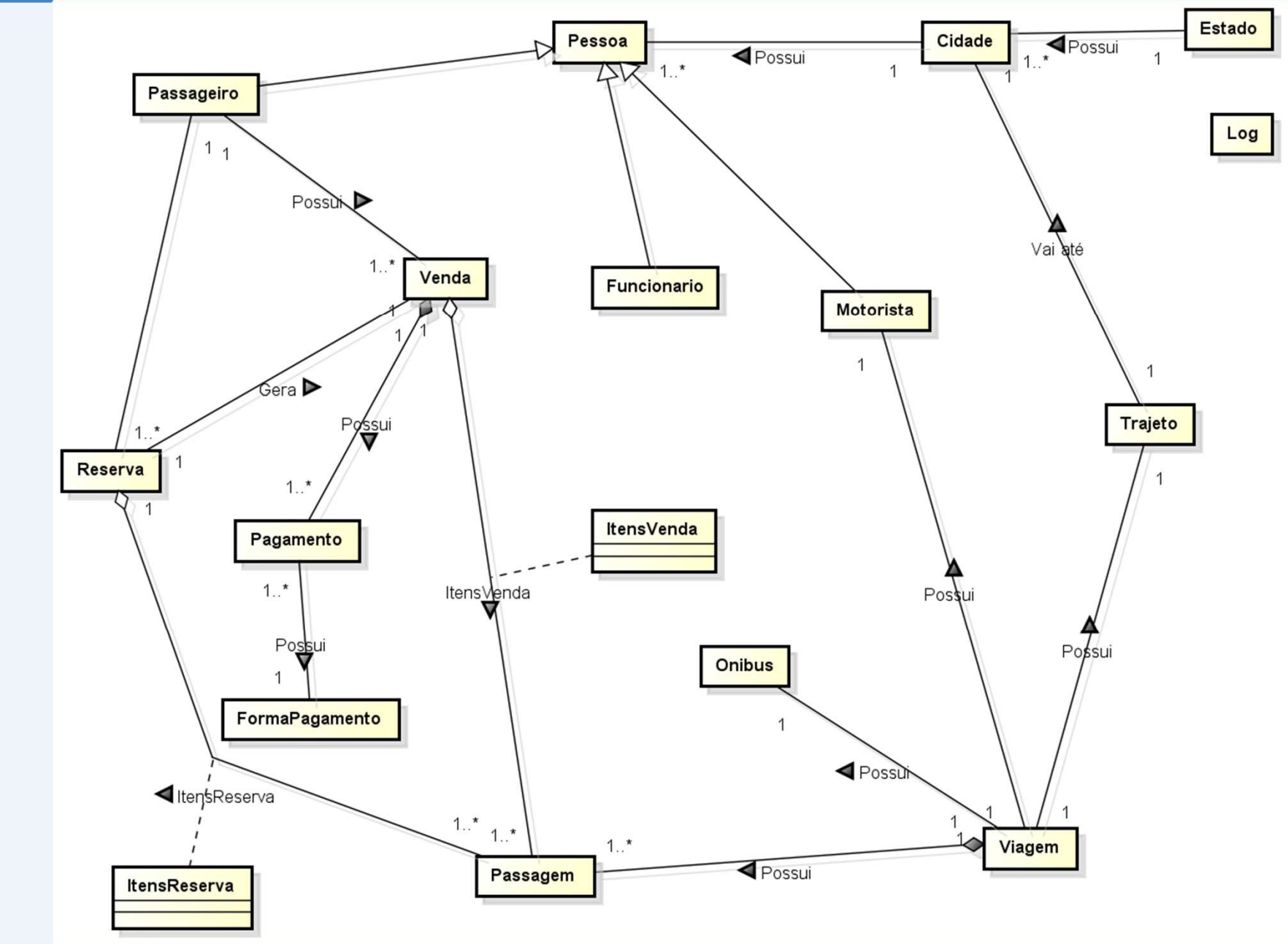
Realização dos casos de uso

- Identificar as classes de análise necessárias para executar o fluxo de eventos do caso de uso.
- Cada caso de uso deve ser analisado para representar a comunicação necessária entre as classes de análise para realizar a ação pretendida pelo caso de uso.

Realização dos casos de uso



Realização dos casos de uso



Realização dos casos de uso

- Descrição do comportamento do sistema por intermédio de diagramas de interação
 - Diagrama de comunicação
 - Antigo Diagrama de Colaboração
 - Diagrama de sequência
- Deve possuir rastreabilidade com a descrição dos casos de uso
- Demonstra o comportamento do sistema e as classes que participam para a realização deste comportamento

Diagrama de sequência

- Diagrama de seqüência → possui duas características que o diferencia dos diagrama de comunicação
 - Existe uma linha de vida do objeto
 - Existe foco no controle e seqüência das mensagens trocadas entre os objetos (tempo)
- Três tipos de mensagens
 - Simples: não descreve detalhes da mensagem por não ser importante no momento (não utilizado para fase de projeto)
 - Síncrona: possui uma mensagem de retorno/espera.
 - Assíncrona: serve para envio de mensagem sem nenhuma espera/retorno
 - Mensagem de retorno: retorno de uma mensagem síncrona
- Mensagens tornam-se os métodos das classes



Assíncrona



Síncrona

Diagrama de sequência

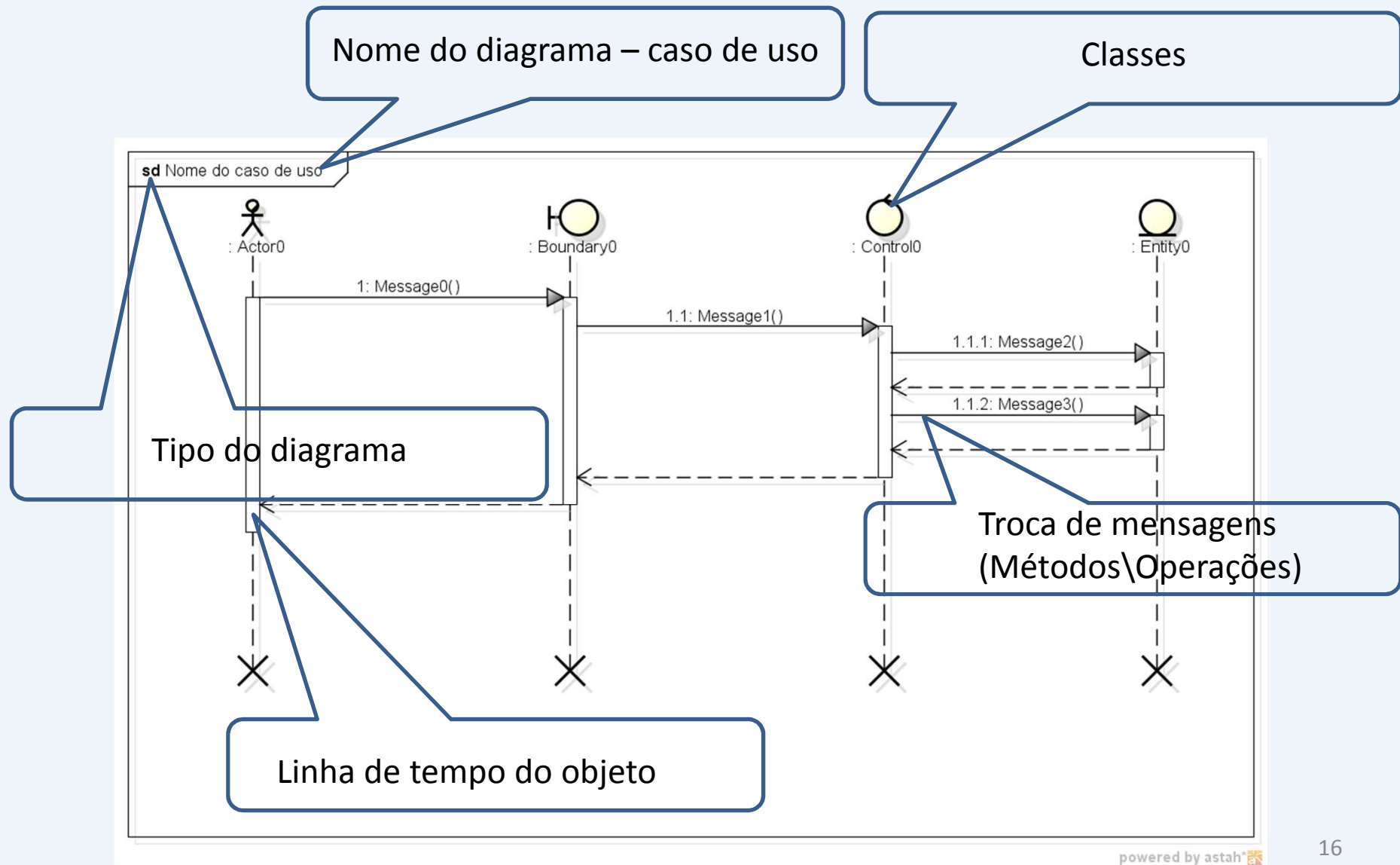


Diagrama de sequência

- Uma mensagem pode ter:
 - Tipo de retorno
 - Parâmetros
- Sintaxe:
 - VR = Mensagem (Par1: Tipo; Par2: Tipo;): TR
 - Onde:
 - VR -> Variável que recebe o retorno
 - Mensagem -> Nome da operação ou método
 - Par1 e Par2 -> parâmetros
 - TR -> Tipo de retorno
- Exemplo
 - ListaIngresso = ListarIngressos(Show: integer):List<Ingressos>

```
Class Ingresso
{
  //atributos

  Public List<Ingressos> ListarIngressos (S:Integer)
  {
    //implementação da lista
  }

  //demais métodos e construtores
}
```

Diagrama de sequência

- Capacidade de representar lógica procedimental no diagrama (lógica de programação – fluxos de controle)
- Utiliza-se de fragmento combinado
 - sequência de mensagens (interações) combinadas em um quadro
 - Utiliza-se de um operador para definir o tipo do fragmento combinado (loop, if ... etc)

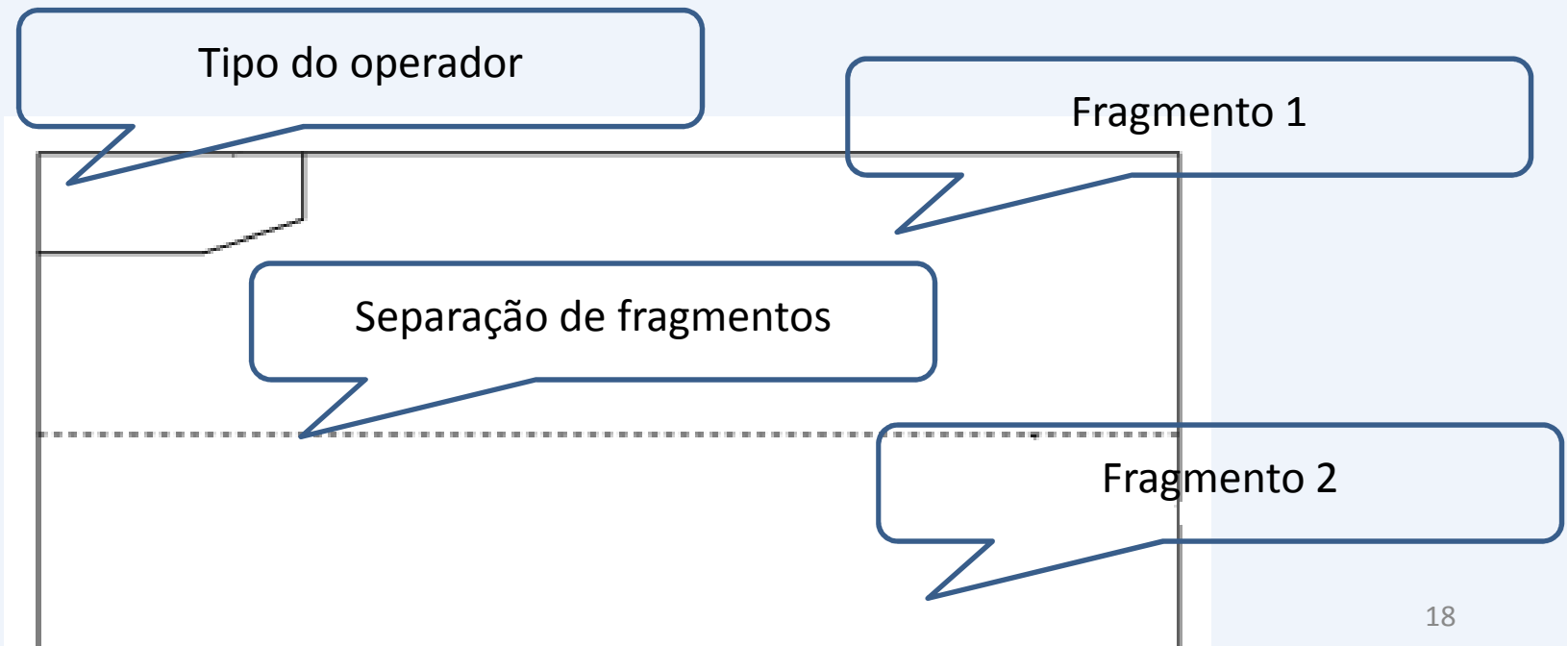


Diagrama de sequência

- O fragmento combinado pode conter
 - Operandos de interação ou fragmentos
 - Partes dos fragmentos que é executada sobre uma condição de guarda (condição específica)
 - Condições de guarda (restrições)
 - Expressão condicional associada a um operador de interação
 - $X > 10$
 - $X \leq 100$
 - Pode ser escrita em português
 - Operadores de interação comumente usados
 - Loop
 - Representa uma estrutura de repetição no código
 - Alt
 - Representa uma estrutura de seleção composta por mais de um fragmento
 - » If com else
 - » Switch, case
 - Opt
 - Representa uma estrutura de seleção composta por apenas um fragmento
 - » If sem else

Diagrama de sequência

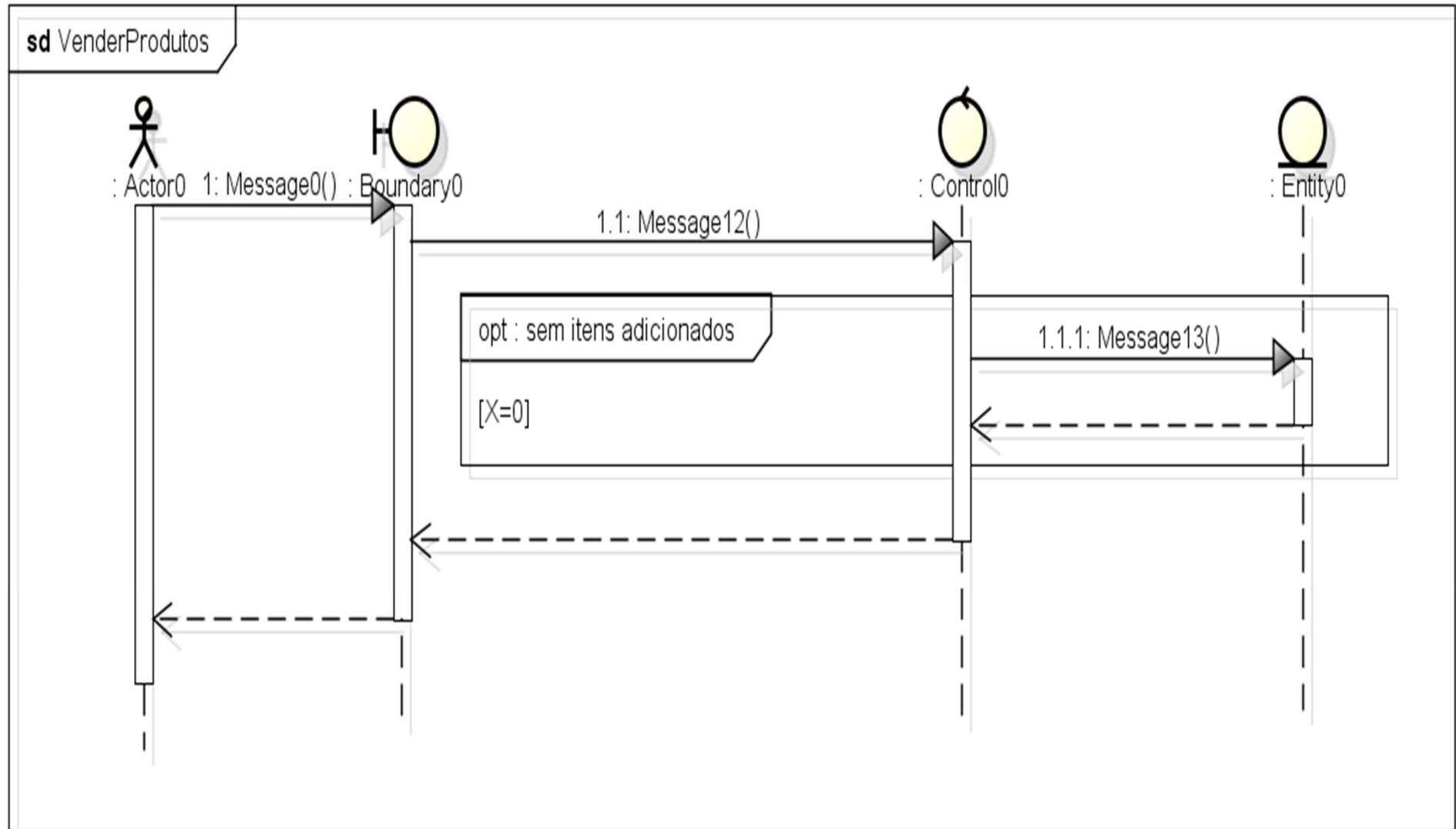


Diagrama de sequência

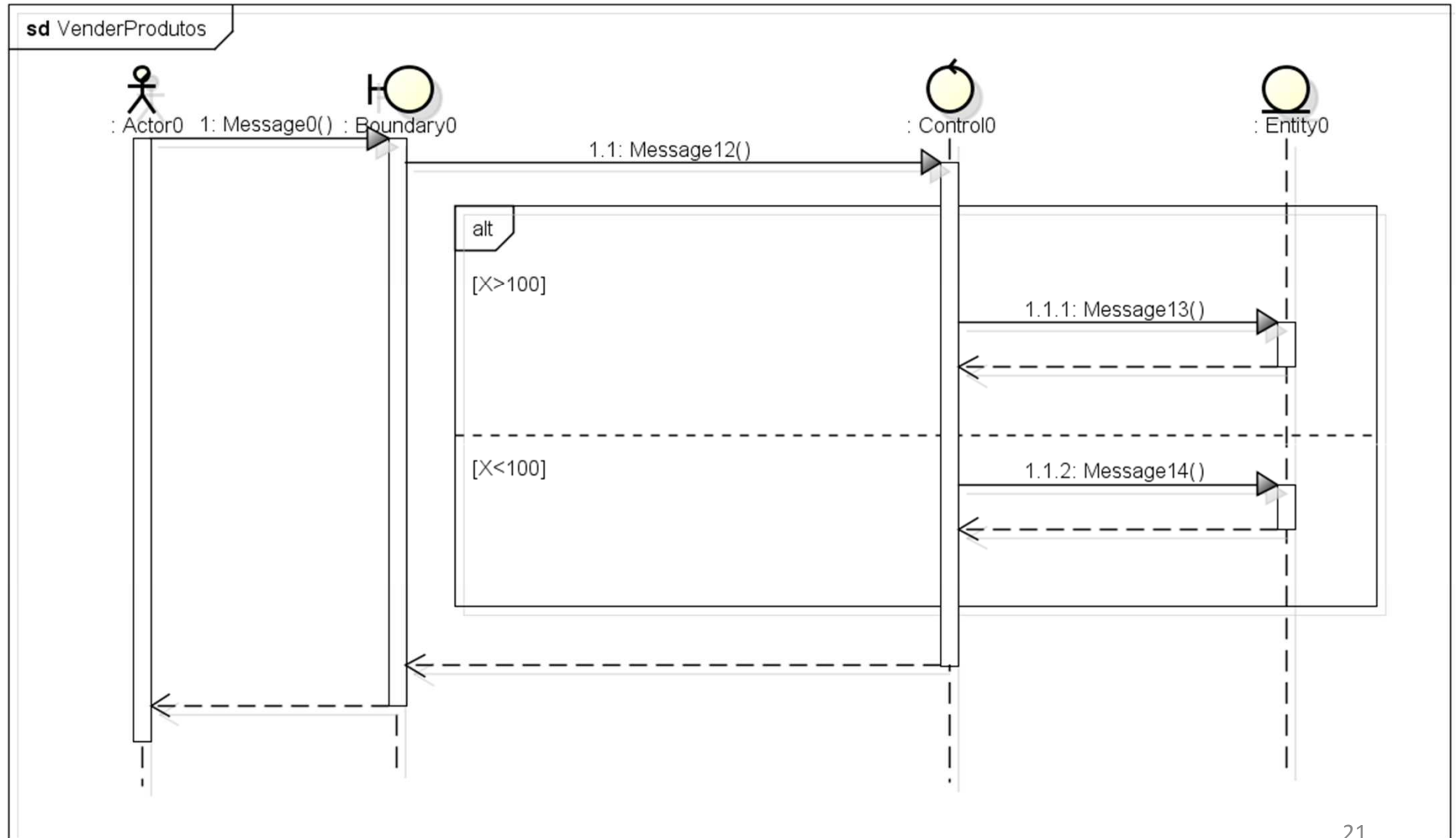
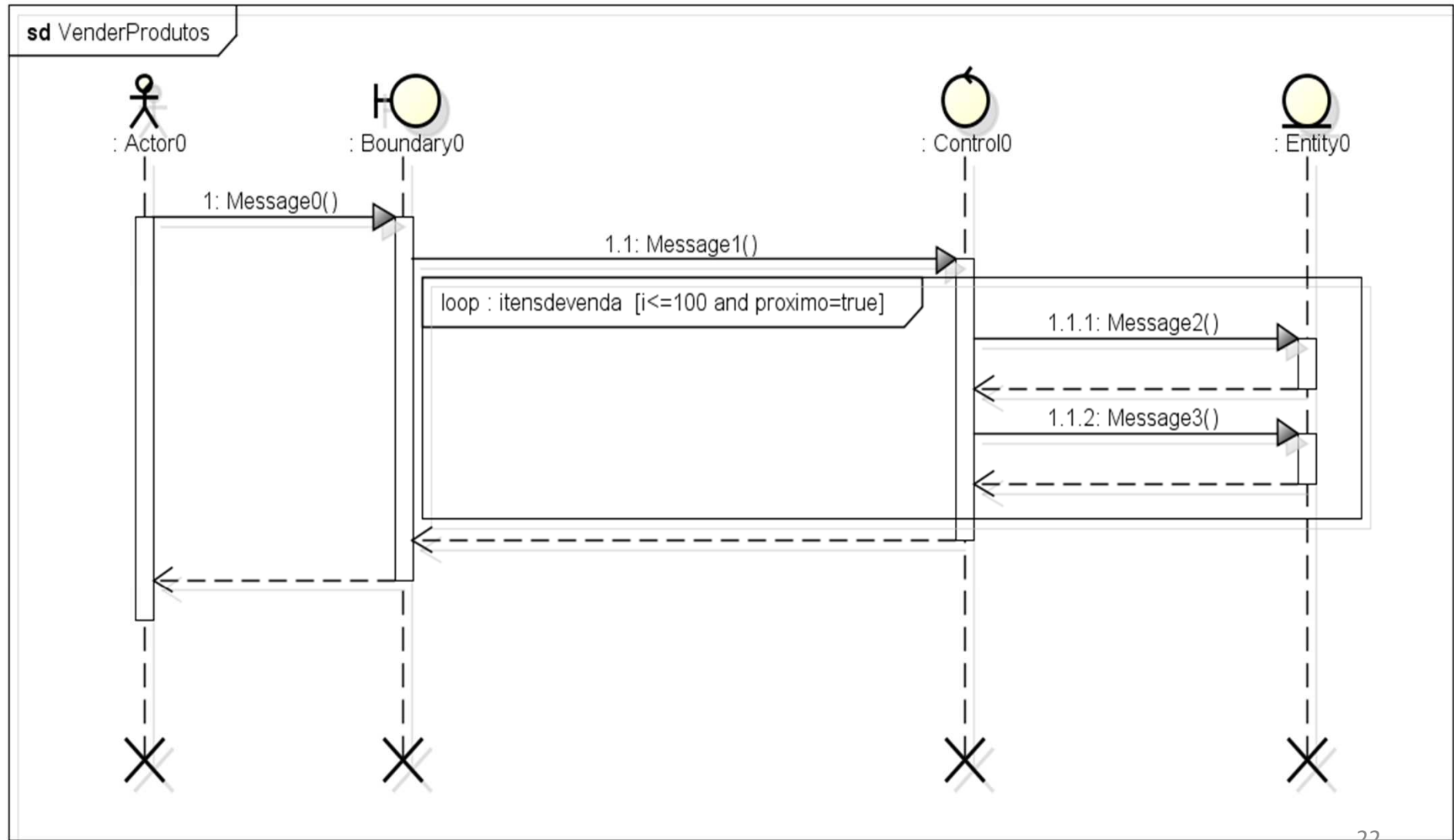


Diagrama de sequência



Exemplo para Reserva de Passagens.

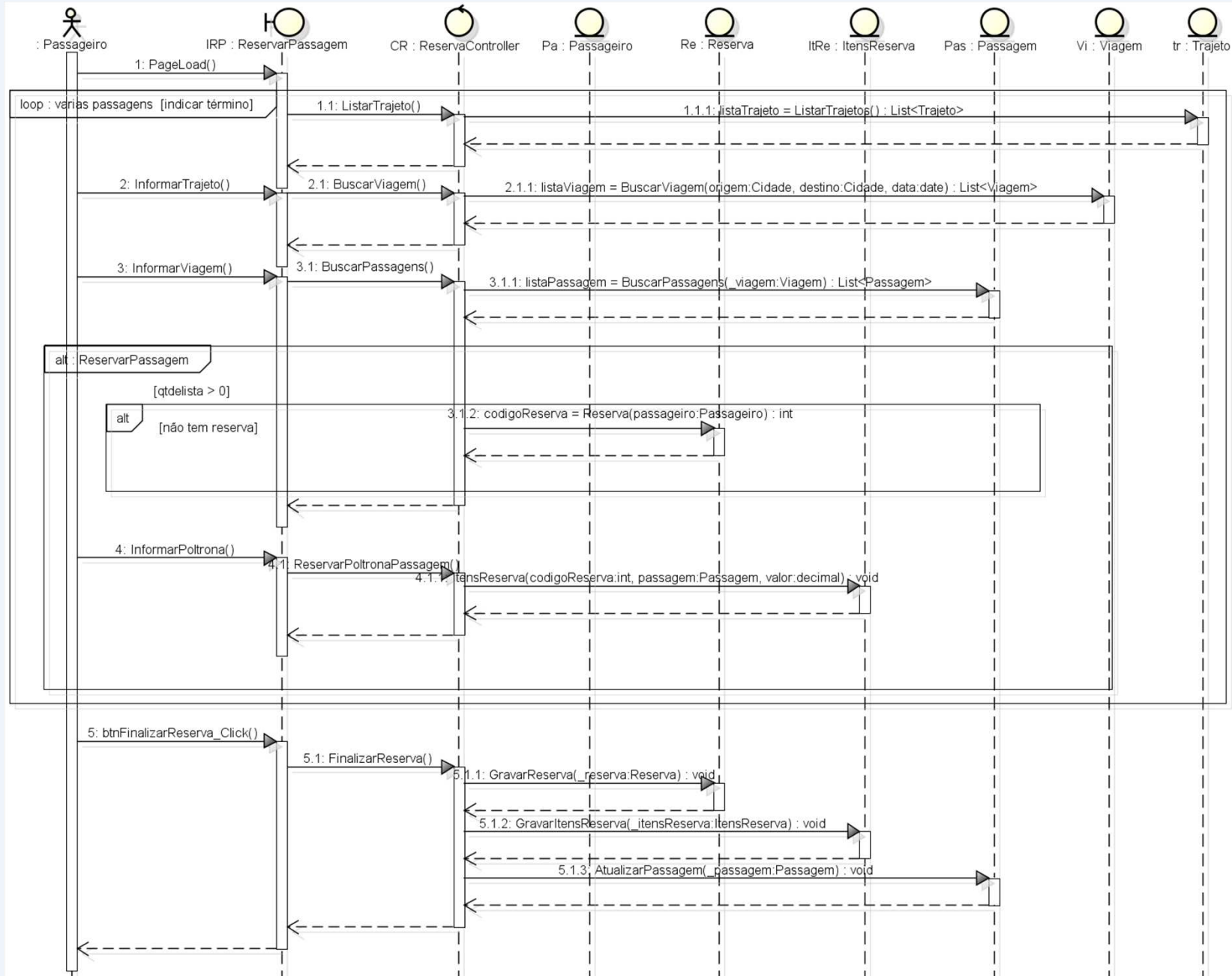


Diagrama de sequência

- Após o término dos diagramas de sequência é definido:
 - Métodos para as classes de análise (entidades, controladoras e interfaces)
 - Definido passagem de parâmetros para os métodos
 - Definidos tipos de retorno para os métodos
 - Representado estruturas de seleção e repetição
- Com este resultados
 - Modelo conceitual inicia evolução para o diagrama de classes do sistema

Workflows do Processo Unificado



- Entrar em acordo sobre o contexto do sistema
- Elicitar e especificar Requisitos

Requisitos

- Escopo do sistema
- Lista de funções
- Diagrama de casos de uso
- Especificação de casos de uso
- Modelo conceitual
- Diagrama de atividades

- Foco distancia do cliente
- Foco em o que os desenvolvedores precisam para construir o sistema correto
- Estruturar os requisitos de uma forma que facilite o entendimento técnico
- Pré-projeto

Análise

- Classes de Análise
- Realização de casos de uso de análise
 - Diagrama de sequência
- Diagrama de classes Final

Projeto

Codificação

Teste