ANALISE E PROJETO DE SISTEMAS

Cleviton Monteiro (cleviton@gmail.com)

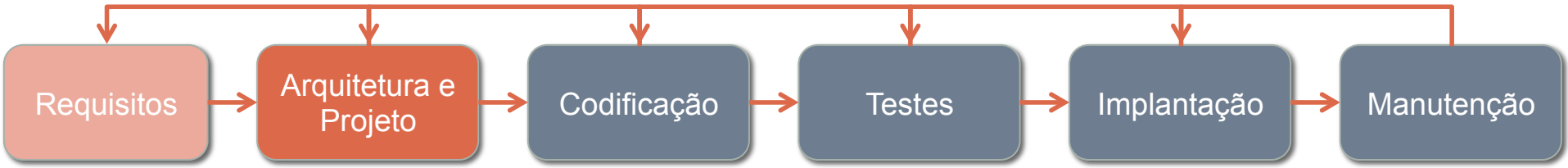
Roteiro

- Ciclo de vida
- Motivação
- Analise X Projeto
- Processo de análise e projeto: Papeis e artefatos
- Analise de casos de uso
- Diagrama de classes

Nota 4

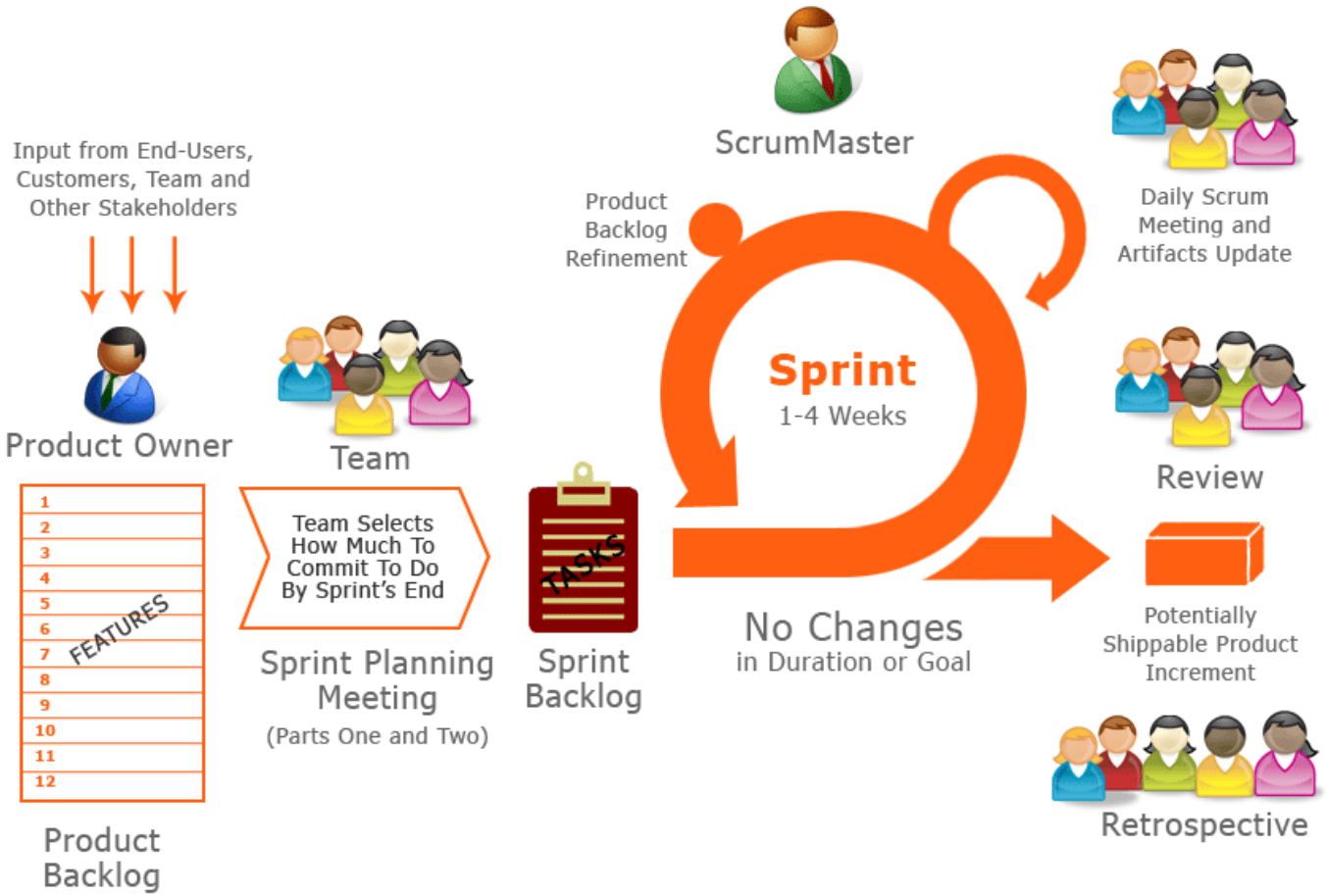
- Documento de Arquitetura e Projeto
  - Documento de arquitetura
  - Diagrama de classes

Ciclo de vida



Estamos saindo da fase de requisitos e entrando na Arquitetura e Projeto

Ciclo de vida – e no SCRUM?



Fonte: http://www.n-axis.in/

Ciclo de vida – e no SCRUM?

Backlog

- Primeiras Sprints
  - Inclusão de atividades de definição da Arquitetura e Projeto

Por que analisar e projetar?

Por que não começar logo pela implementação?

1. Padronizar!

Exemplos de problemas:

Desenvolvedor 1	Desenvolvedor 2
Classe Client	Classe Customer
Dividiu em camadas	Acessou o BD direto da view
Utilizou Hibernate	Não utilizou framework de persistência

Por que analisar e projetar?

Por que não começar logo pela implementação?

2. Minimizar redundâncias (repetição de código)

- Espalhamento de regras

Exemplo de redundância:

- Classes Atendente, Cliente, Usuário, Administrador
  - Todos possuem atributos semelhantes (nome, cpf, telefone, etc) e métodos semelhantes
- Como evitar: usar Herança

Por que analisar e projetar?

Por que não começar logo pela implementação?

3. Atender a requisitos não-funcionais

Exemplo:

> “O sistema deve guardar em um arquivo todos os erros que acontecerem em tempo de execução” (log de erros)

Qual será a abordagem escolhida para isso?

- Framework já existente de log
- Implementação individual em cada tratamento de erro
- Implementação de um subsistema de log

Por que analisar e projetar?

Por que não começar logo pela implementação?

4. Aumentar a produtividade

5. Aumentar a qualidade

6. Reduzir o custo de manutenção

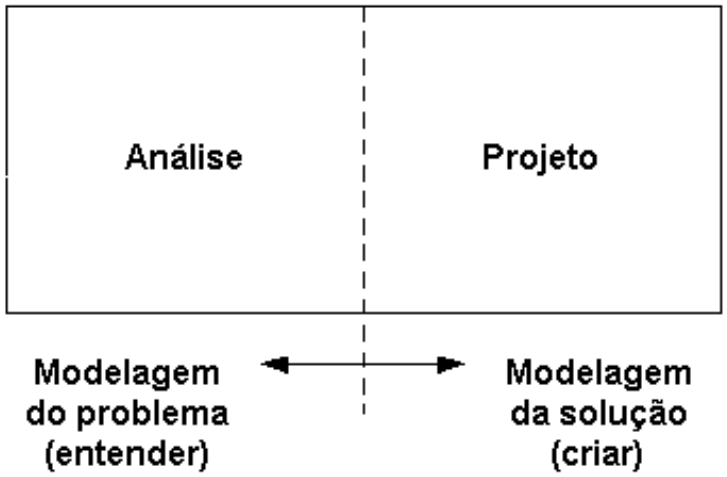
Analise X Projeto

Análise

- Modela o problema
- Entender o domínio
- Investigação

Projeto

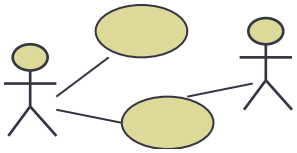
- Modela a solução
- Atividade de criação
- Proposta de resolução



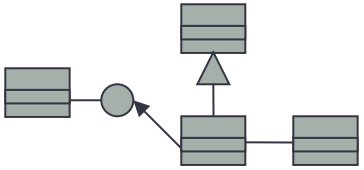
Analise

- Design Thinking Canvas
- Casos de uso
- Estórias dos usuários
- Prototipação
- Critérios de aceitação
  - Casos de testes

Projeto



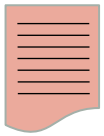
Modelo de
Casos de Uso



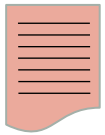
Modelo de
Análise e Projeto

Projeto
(projetar)

Documento de
Arquitetura

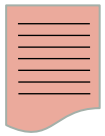


Glossário



Requisitos

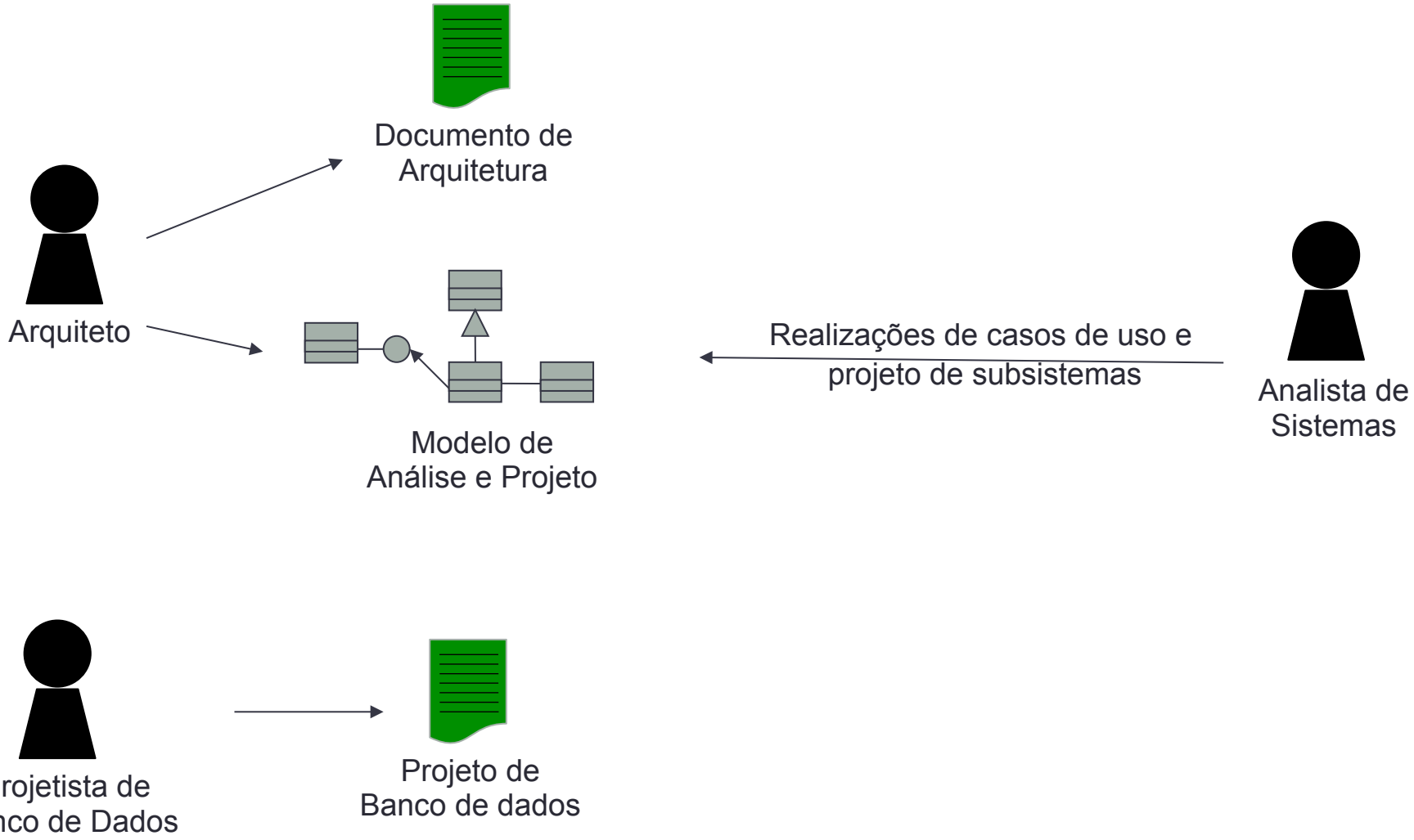
Estórias



Protótipos

Projeto de
Banco de dados

Papeis e artefatos



Artefatos do projeto

- Modelo de análise e projeto é o principal objetivo deste fluxo de atividades

- O projeto de banco
  - Contém o mapeamento do modelo OO para o relacional
  - Especifica tabelas, índices, *triggers*, *procedures,* etc.

- O documento da arquitetura
  - Usado para descrever em detalhes uma determinada arquitetura

Papeis: Arquiteto

- Lidera e coordena as atividades técnicas
- Constrói os artefatos do projeto
- Define a estrutura das visões arquiteturais
  - Decompõe o sistema em visões
  - Agrupa os elementos de projeto em:
    - Subsistemas
    - Pacotes
    - Módulos
  - Define as interfaces
  - Identifica unidades de concorrência
- Tem uma visão larga e superficial do sistema

Papeis: Analista de Sistemas

- Faz a realização dos casos de uso de forma consistente com a arquitetura

- Deve conhecer:
  - A tecnologia a ser usada no desenvolvimento do sistema
  - As técnicas de modelagem de casos de uso
  - Os requisitos do sistema
  - As técnicas de análise e projeto orientado a objetos
  - A linguagem UML

Papeis: Projetista de banco de dados

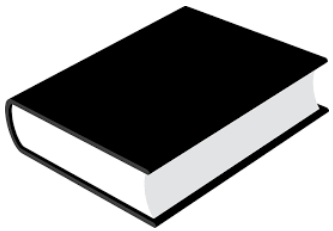
- Define a estrutura de dados da aplicação, como tabelas, índices, visões, triggers, etc.
- Mapeamento OO em Modelo ER
- Deve possuir um conhecimento sólido em análise e projeto orientado a objetos e banco de dados

Resumindo

Projeto Orientado a Objetos

- Objetivos:
  - Transformar os requisitos em um projeto (inicialmente abstrato) do sistema
  - Achar e descrever objetos (ou conceitos) no domínio do problema
  - Desenvolver uma arquitetura robusta para o sistema

- Exemplo (sistema de biblioteca)



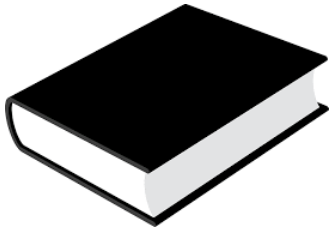
Livro

Cliente

Empréstimo

Resumindo

- Exemplo (sistema de biblioteca)



Livro

Cliente

Aluguel

Livro

```
Public Class Livro {
…
}
```

Cliente

AluguelLivro

Projeto orientado a objetos

Atributos de qualidade

- Controle da complexidade
  - Ex: Abstração
- Facilitar as mudanças
  - Ex: Baixo acoplamento facilita as mudanças através do isolamento
- Aumentar o reuso
  - Ex: Herança

Como fazer a

transição

da **análise**

para o **projeto**?

ANALISAR CASOS DE USO

Realização de casos de uso

- Descreve como o caso de uso é realizado, associando o caso de uso com classes e outros elementos de projeto
- Distribui responsabilidades entre as classes, assim como atributos e associações
- Em UML, uma realização de caso de uso pode ser representada através de um conjunto de diagramas:
  - diagrama de classe
  - diagrama de sequência
  - diagrama de colaboração

Passo a Passo

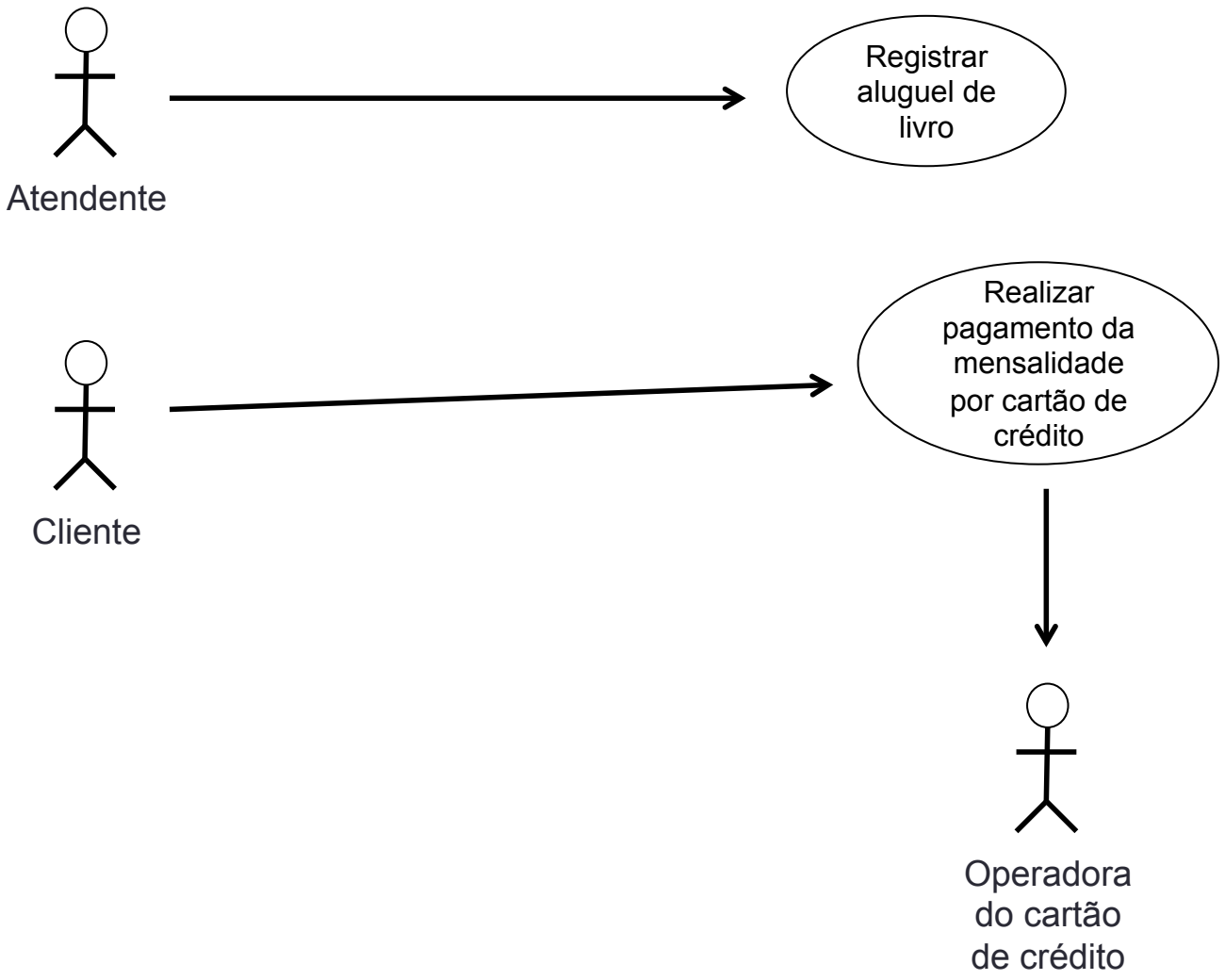
- Para cada caso de uso:
  1. Encontrar classes de análise
  2. Identificar persistência

- Para cada classe
  3. Distribuir comportamento entre as classes
  4. Descrever responsabilidades
  5. Descrever atributos e associações

Passo 1: Encontrar classes de análise

- Distribuir o comportamento dentre as seguintes classes de análise (estereótipos)
  - Fronteira
  - Controle
  - Entidade
- Esses estereótipos ajudam a encontrar as classes

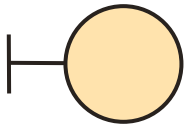
Casos de uso da biblioteca do concurseiro

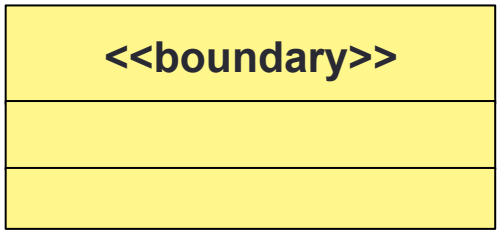


Classes de fronteira

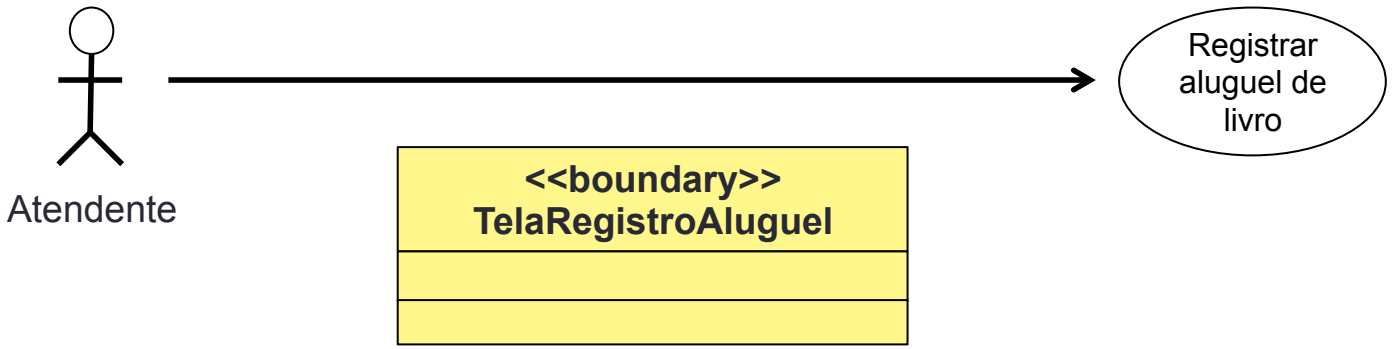
- Isolam o sistema de mudanças no ambiente externo
- Atores devem se comunicar apenas com classes de fronteira
- Exemplos de classes fronteira
  - GUI
  - Interface com outros sistemas
  - Interface com dispositivos

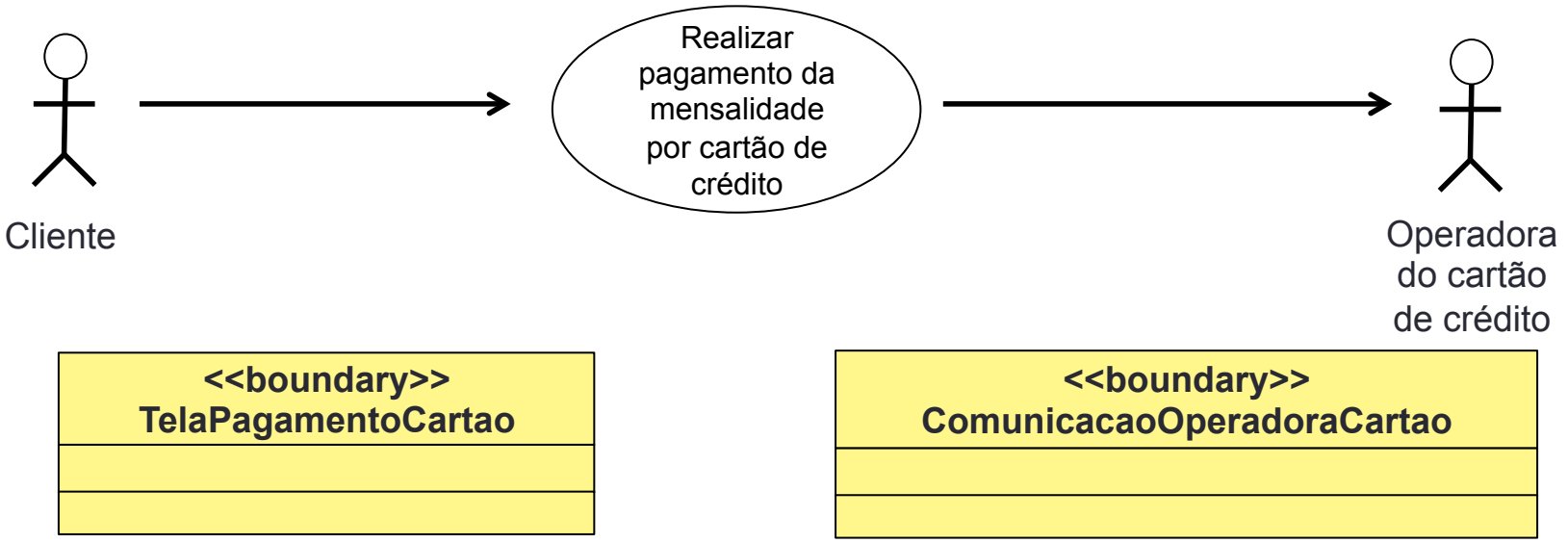
Notações em UML





Passo 1.1: identificar classes de fronteira

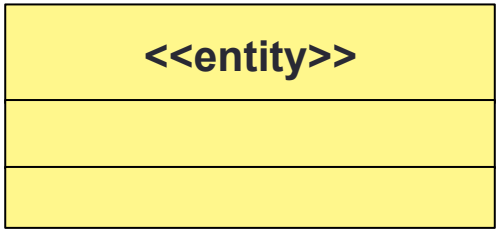




Classes de Entidade

- Abstrações e conceitos-chave dos casos de uso
- Armazenam e controlam informações no sistema
- Onde procurar:
  - Glossário
  - Documento de requisitos / Regras de negócios
  - Casos de uso
  - Estórias dos usuários / Critérios de aceitação / protótipos

Notações em UML



Caso de uso: Registrar aluguel de livro

Este caso de uso é responsável por registrar o aluguel de um livro.

...

Fluxo principal:

1. O atendente informa o CPF do cliente.
2. O sistema verifica se o cliente está cadastrado. [FE01]
3. O sistema exibe o nome do cliente.
4. O atendente informa o código do livro.
5. O sistema verifica se o código é válido. [FE02]
6. O sistema exibe o nome do livro.
7. O atendente escolhe a opção “Salvar”.
8. O sistema armazena o registro do aluguel.

Fluxo de exceção

FE01: Cliente não cadastrado

1. Caso o cpf não estiver associado a um cliente cadastrado, o sistema exibe a mensagem “Cliente não cadastrado” e o fluxo retorna ao passo 1.

...

Passo 1.2: identificar classes de entidades

- Busque na descrição do caso de uso os substantivos que devem representar entidades:
  - Identifique substantivos no fluxo de eventos
  - Remova candidatos redundantes
  - Remova atores que interagem com o sistema, mas não fazem parte da modelagem
  - Não utilize substantivos que representam atributos

Caso de uso: Registrar aluguel de livro

Este caso de uso é responsável por registrar o aluguel de um livro.

...

Fluxo principal:

1. O atendente informa o CPF do cliente.
2. O sistema verifica se o cliente está cadastrado. [FE01]
3. O sistema exibe o nome do cliente.
4. O atendente informa o código do livro.
5. O sistema verifica se o código é válido.
6. O sistema exibe o nome do livro.
7. O atendente escolhe a opção “Salvar”.
8. O sistema armazena o registro do aluguel.

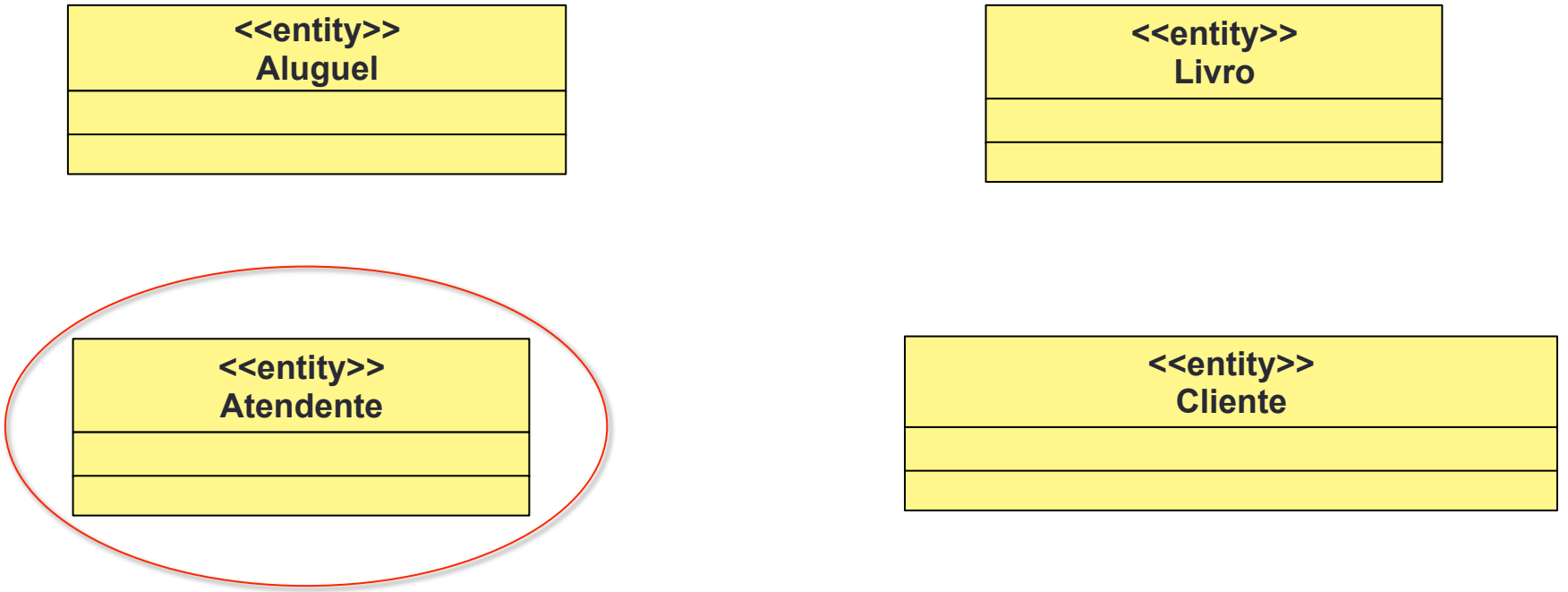
Fluxo de exceção

FE01: Cliente não cadastrado

1. O sistema exibe a mensagem “Cliente não cadastrado” e o fluxo retorna ao passo 1.

...

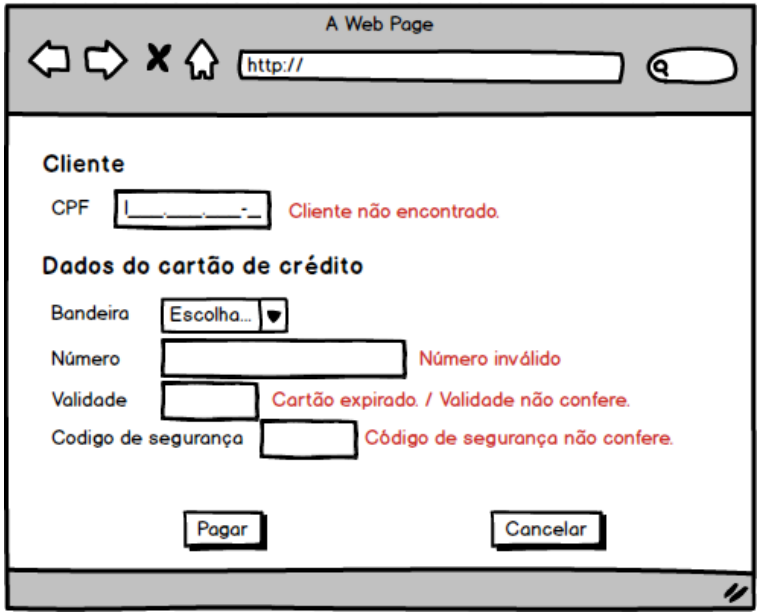
Passo 1.2: identificar classes de entidades



Em um diagrama de UC mais completo poderia ser uma especialização da entidade usuário

Estória, protótipo e critério de aceitação

Como um cliente, desejo realizar o pagamento da mensalidade por cartão de crédito assim não enfrento fila nem preciso portar dinheiro.

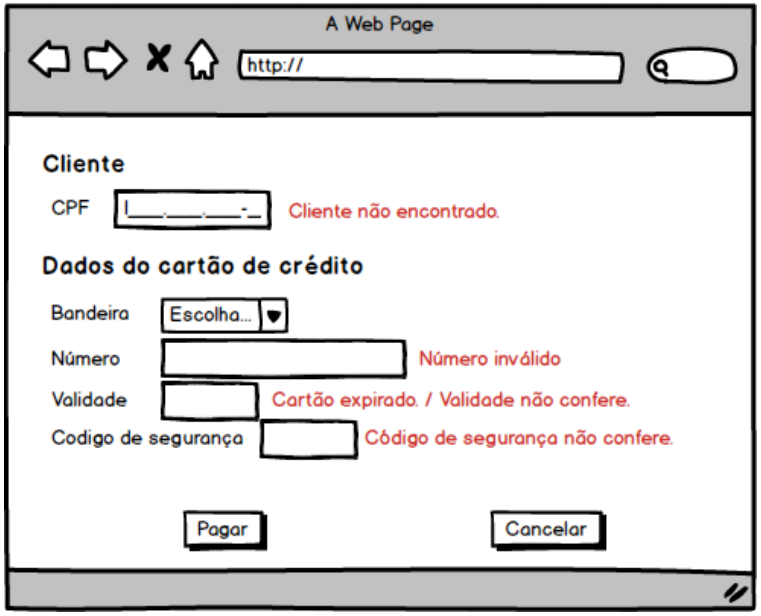


Critérios de aceitação:

- O sistema só deve aceitar CPF válido.
- O sistema deve exigir que o CPF seja de um cliente cadastrado.
- O sistema deve verificar com a operadora de cartão de crédito se os dados do cartão são válidos.
- O sistema deve verificar com a operadora de cartão de crédito se a transação pode ser realizada, em caso positivo, deve registrar o pagamento. Caso contrário, a mensagem “Operação não autorizada” deve ser exibido.

Estória, protótipo e critério de aceitação

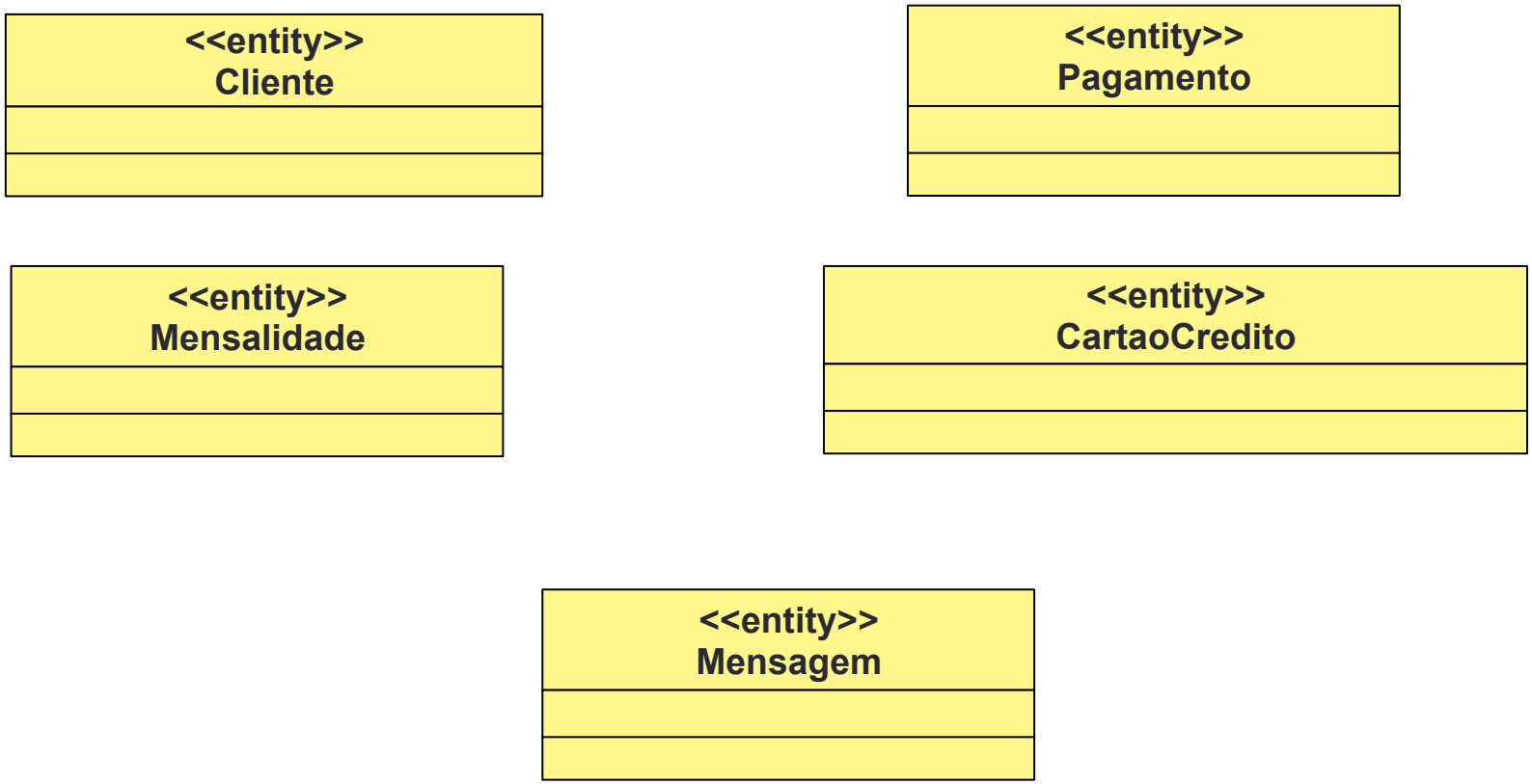
Como um cliente, desejo realizar o pagamento da mensalidade por cartão de crédito assim não enfrento fila nem preciso portar dinheiro.



Critérios de aceitação:

- O sistema só deve aceitar CPF válido.
- O sistema deve exigir que o CPF seja de um cliente cadastrado.
- O sistema deve verificar com a operadora de cartão de crédito se os dados do cartão são válidos.
- O sistema deve verificar com a operadora de cartão de crédito se a transação pode ser realizada, em caso positivo, deve registrar o pagamento. Caso contrário, a mensagem “Operação não autorizada” deve ser exibida.

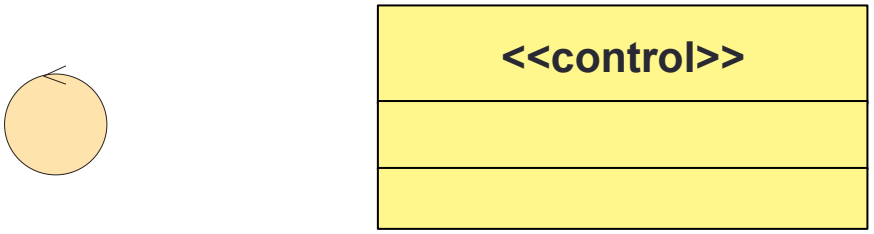
Passo 1.2: identificar classes de entidades



Classes de Controle

- Coordenam o comportamento (lógica de controle) do caso de uso
- Interface entre fronteira e entidade
- Permitem separação entre o uso da entidade (específico do sistema) do comportamento inerente à entidade

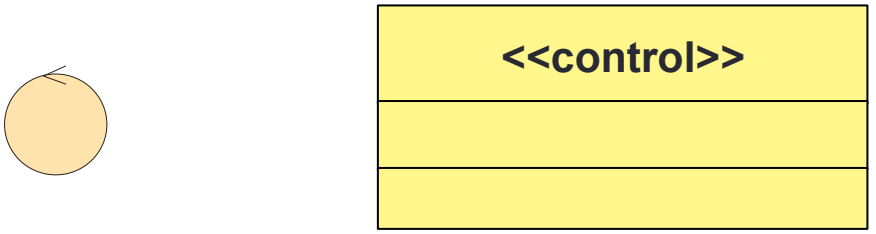
Notações em UML



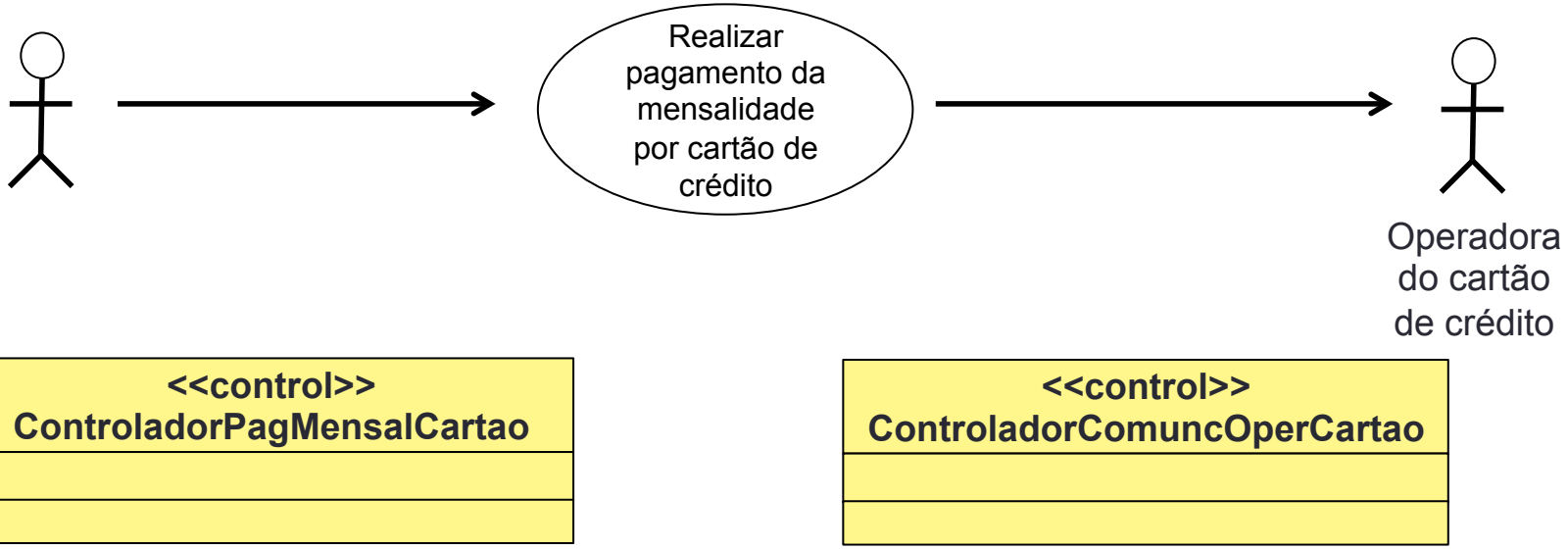
Passo 1.3: identificar classes de controle

- Usualmente, uma classe de controle por caso de uso
- Interface entre fronteira e entidade
- Normalmente, usa-se:
  - mais de uma classe de controle para comportamentos complexos
  - nenhuma para comportamentos simples, como manipulação de armazenamento

Notações em UML



Passo 1.3: identificar classes de controle



Passo a Passo

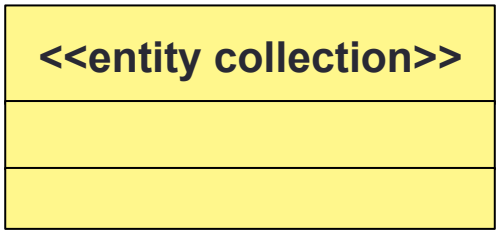
- Para cada caso de uso:
  1. Encontrar classes de análise
  2. Identificar persistência

- Para cada classe
  3. Distribuir comportamento entre as classes
  4. Descrever responsabilidades
  5. Descrever atributos e associações

Passo 2: identificar persistência

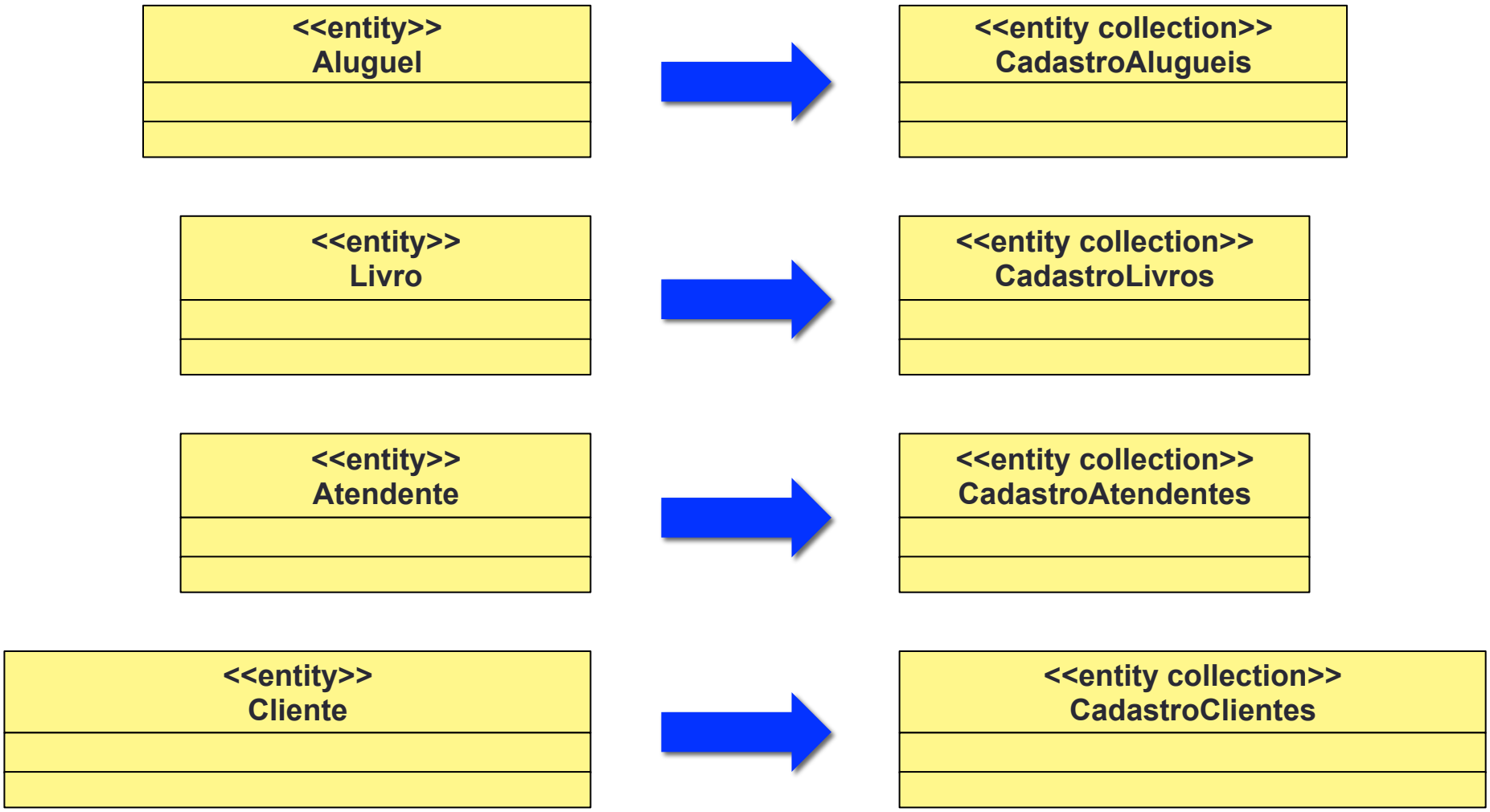
- Identificar quais classes de análise deverão ser persistentes (armazenadas)
- Criar, para cada classe persistente, uma classe de cadastro com estereótipo <<entity collection>>

Notações em UML



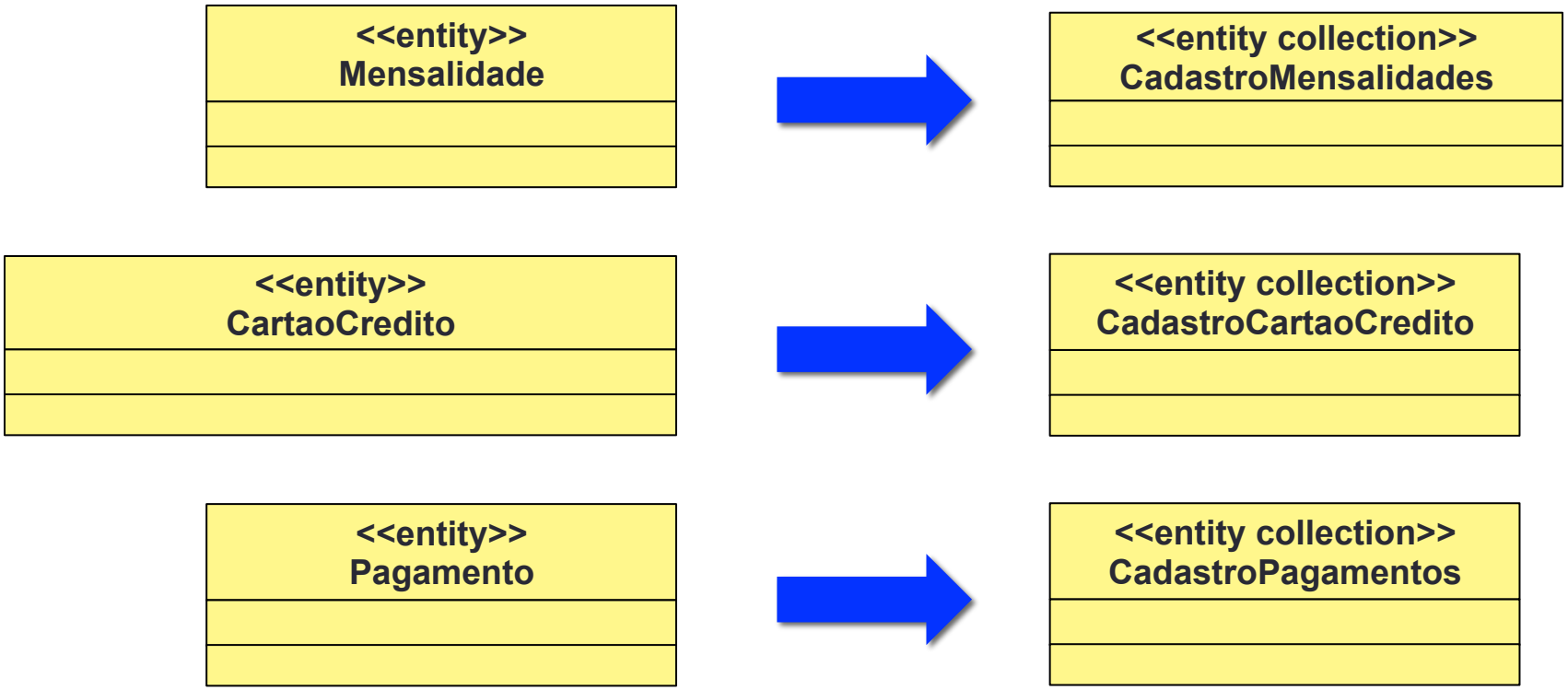
Passo 2: identificar persistência

Registrar aluguel de livro



Passo 2: identificar persistência

Realizar pagamento da mensalidade por cartão de crédito



Note que: CadastroCliente já está no caso de uso anterior e Mensagem não será persistida, então não será preciso uma classe de cadastro para ela.

Passo a Passo

- Para cada caso de uso:
  1. Encontrar classes de análise
  2. Identificar persistência

- Para cada classe
  3. Distribuir comportamento entre as classes
  4. Descrever responsabilidades
  5. Descrever atributos e associações

Passo 3: distribuir comportamento entre as classes

Para cada fluxo de eventos (fluxos principal, secundários, erros)

- Alocar responsabilidades do caso de uso às classes de análise
  - Serviços que os objetos da classe devem prover para os outros objetos
- Modelar interações entre as classes através dos diagramas de interação (sequência e colaboração)

Passo 3: alocando responsabilidades

Guidelines

- Classes de fronteira
  - Possuem comportamentos de comunicação/interação com atores
  - Ex: registrarAluguel, registrarPagamentoMensalidade
- Classes de entidade
  - Possuem comportamentos que envolvem informações encapsuladas nas entidades
  - Ex: getNome, getCPF, setLivro, debitar
- Classes de controle
  - Possuem comportamentos relacionados à logica de negócios do caso de uso (regras de negócio)
  - Ex: registrarPagamentoMensalidade, registrarAluguel, enviarPagamento

Passo a Passo

- Para cada caso de uso:
  1. Encontrar classes de análise
  2. Identificar persistência

- Para cada classe
  3. Distribuir comportamento entre as classes
  4. Descrever responsabilidades
  5. Descrever atributos e associações

Passo 4: alocar responsabilidades

As mensagens trocadas nos diagramas de interação viram responsabilidades das classes receptoras (fornecedoras)

OU

Analisar o caso de uso (ou estórias, protótipos, critérios de aceitação, regras de negócio) em busca das responsabilidades

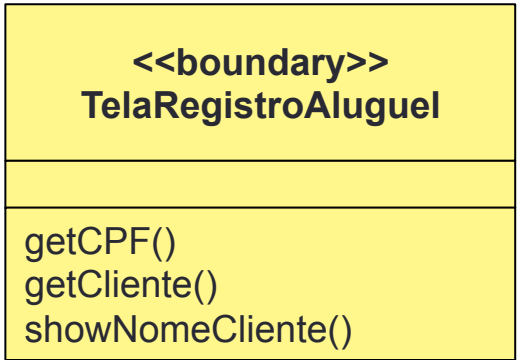
Caso de uso: Registrar aluguel de livro

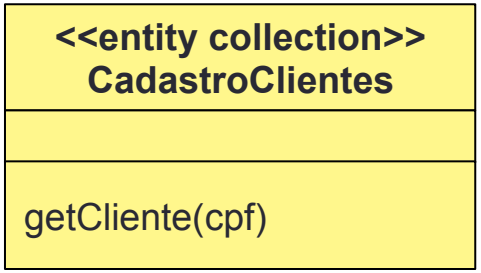
Este caso de uso é responsável por registrar o aluguel de um livro.

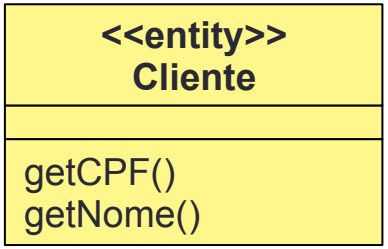
...

Fluxo principal:

1. O atendente informa o CPF do cliente. [Interação com ator – classe fronteira]
2. O sistema verifica se o cliente está cadastrado. [FE01] [Interação com ator – classe fronteira / Informação persistida– classe cadastro e entidade]
3. O sistema exibe o nome do cliente. [Interação com ator – classe fronteira / Atributo – classe entidade]

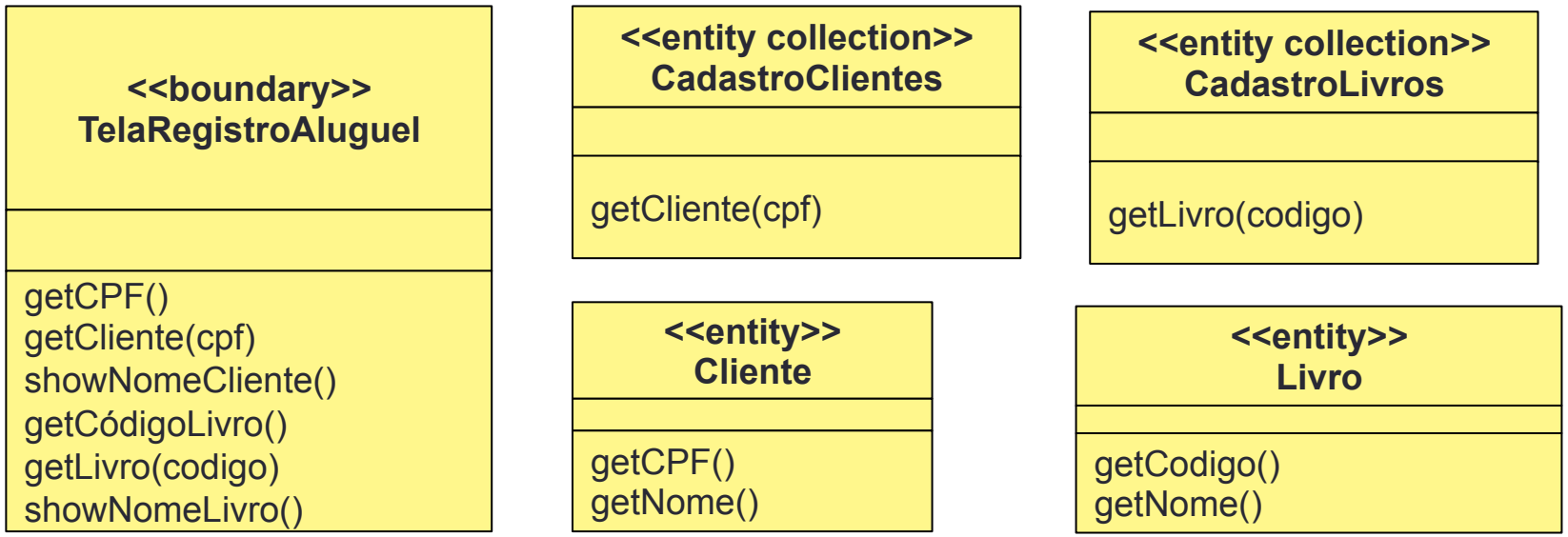






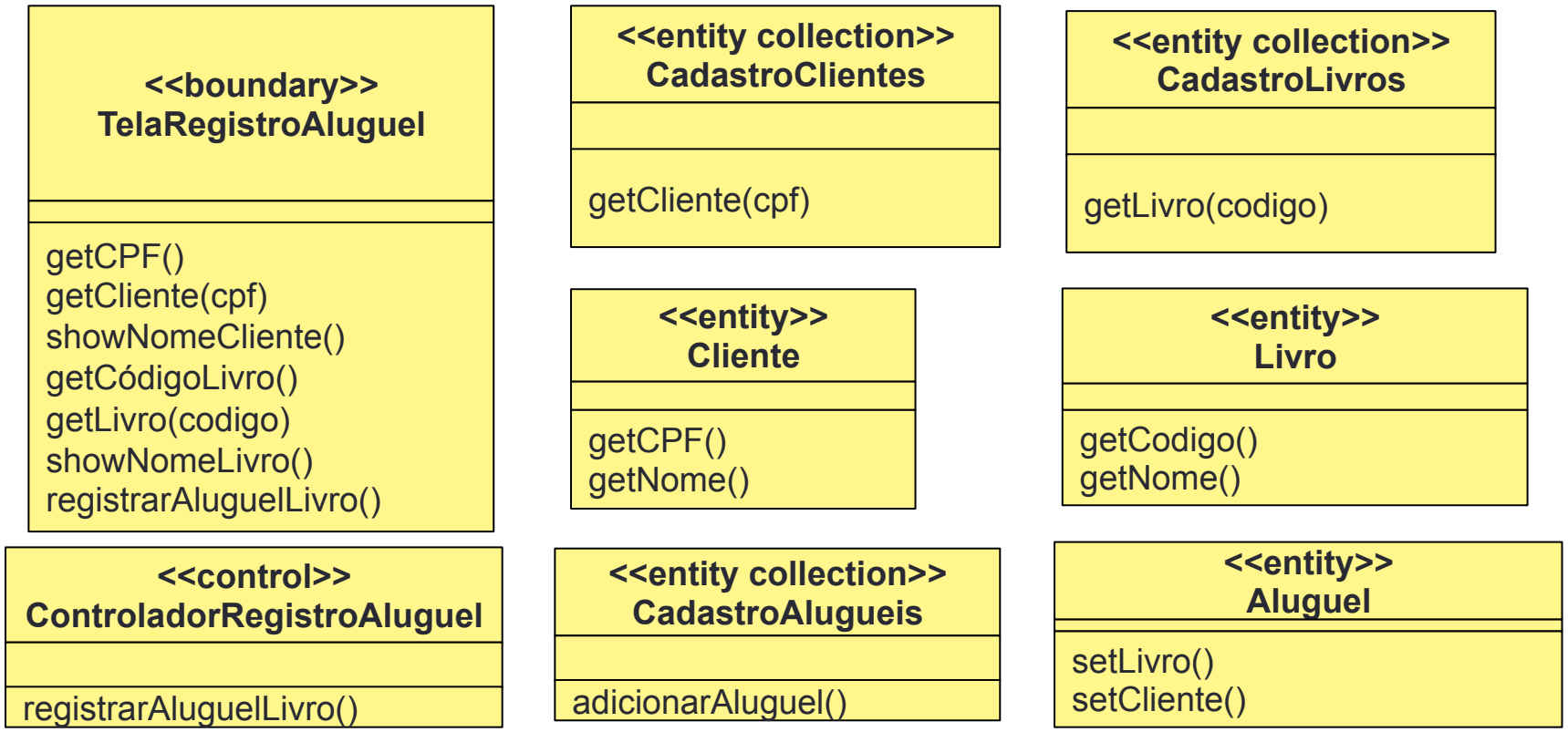
Caso de uso: Registrar aluguel de livro

4. O atendente informa o código do livro. [Interação com ator – classe fronteira]
5. O sistema verifica se o código é válido. [Interação com ator – classe fronteira / Informação persistida – classe cadastro e entidade]
6. O sistema exibe o nome do livro. [Interação com ator – classe fronteira / Informação persistida – classe entidade]



Caso de uso: Registrar aluguel de livro

7. O atendente escolhe a opção “Salvar”. [Interação com ator – classe fronteira]
8. O sistema armazena o registro do aluguel. [Regra de negócio – classe de controle / Informação persistida – cadastro e entidade]



Passo a Passo

- Para cada caso de uso:
  1. Encontrar classes de análise
  2. Identificar persistência

- Para cada classe
  3. Distribuir comportamento entre as classes
  4. Descrever responsabilidades
  5. Descrever atributos e associações

Passo 5: descrever atributos e associações

Detalhar mais as classes

- Definir atributos
- Estabelecer associações necessárias entre as classes

Passo 5: descrever atributos e associações

Diagramas de colaboração: links entre objetos indicam a necessidade de relacionamento entre as respectivas classes

OU

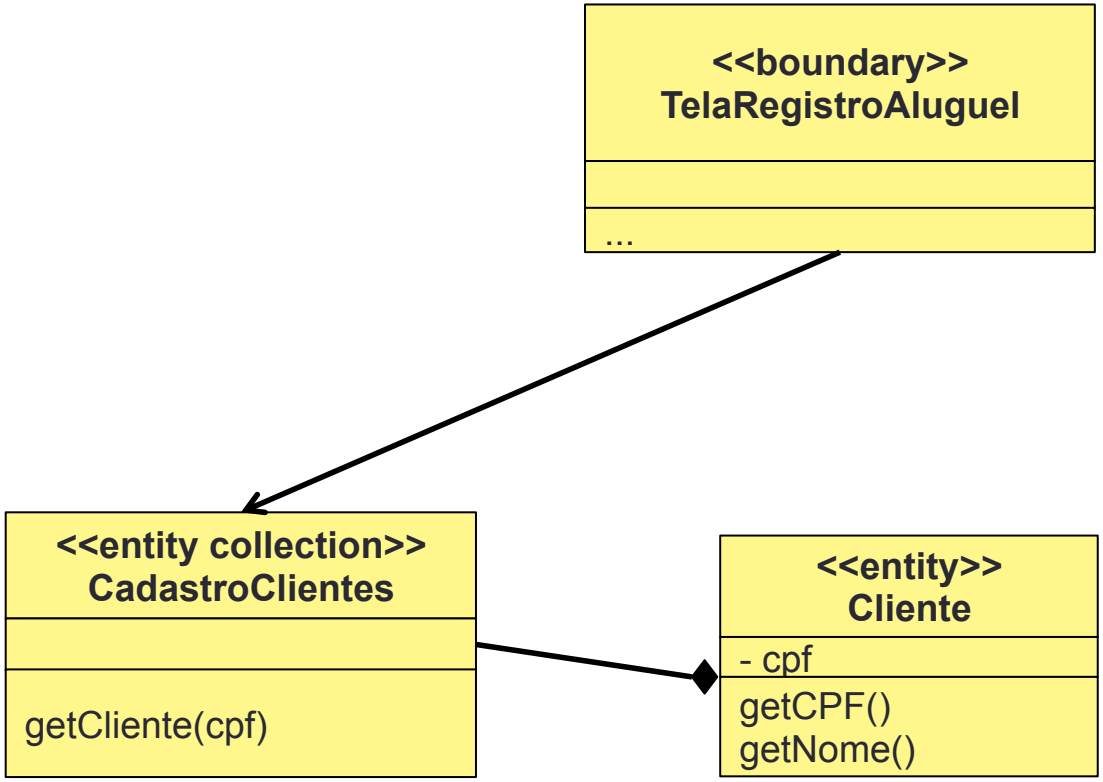
Fluxos do caso de uso (ou descrição das estórias, critérios de aceitação, regras de negócio):

- Identificar quais classes são responsáveis por atender a cada passo e como elas se relacionam para atende-lo
- Quais atributos são necessários

Passo 5: descrever atributos e associações

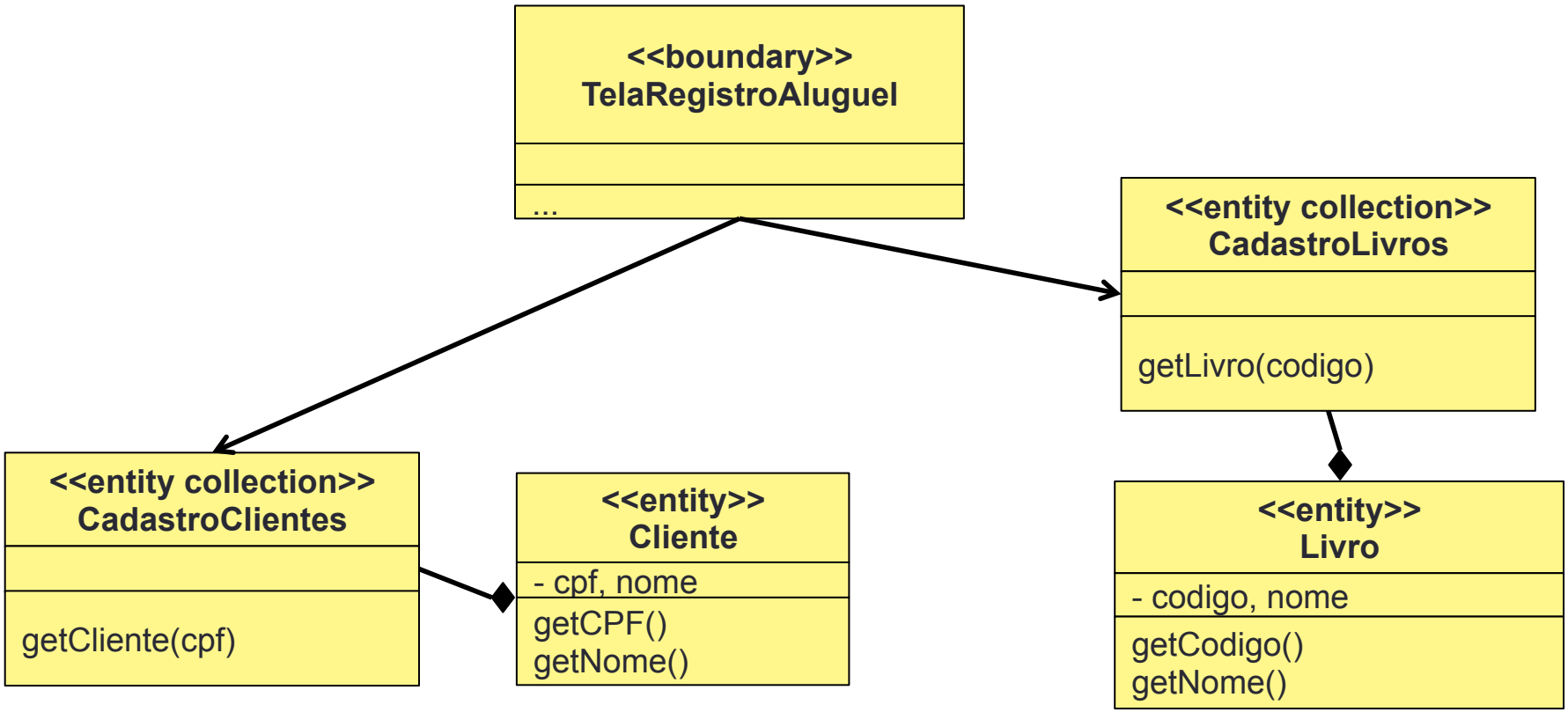
Fluxo principal:

1. O atendente informa o CPF do cliente. [atributo]
2. O sistema verifica se o cliente está cadastrado. [FE01] [fronteira checa com o cadastro / ou controlador]



Passo 5: descrever atributos e associações

3. O sistema exibe o nome do cliente. [atributo].
4. O atendente informa o código do livro. [atributo]
5. O sistema verifica se o código é válido. [fronteira checa com o cadastro / controlador]



Passo 5: descrever atributos e associações

7. O sistema armazena o registro do aluguel. [fronteira solicita à classe de controle, que por sua vez utiliza o cadastro de aluguel]