

Modelagem Temporal com UML

Diagrama de Atividades

Diagrama de Atividades

- “Tipo especial de diagrama de estados, onde são representados os **estados de uma atividade**, ao invés dos **estados de um objeto**.”
- Estados de Atividade:
 - Exemplo 1: passos de um algoritmo
 - Exemplo 2: etapas de um *workflow*

Diagrama de Atividades

- Um diagrama de atividade exibe os passos de uma computação.
 - Cada estado é um passo da computação, onde o sistema está realizando algo.
 - É orientado a fluxos de controle (ao contrário dos DTEs que são orientados a eventos).
- *Fluxogramas* estendidos...
 - Além de possuir toda a semântica existente em um fluxograma, permite representar ações concorrentes e sua sincronização.
- Elementos podem ser divididos em dois grupos: controle seqüencial e controle paralelo.

Diagrama de Atividades

- Notação básica

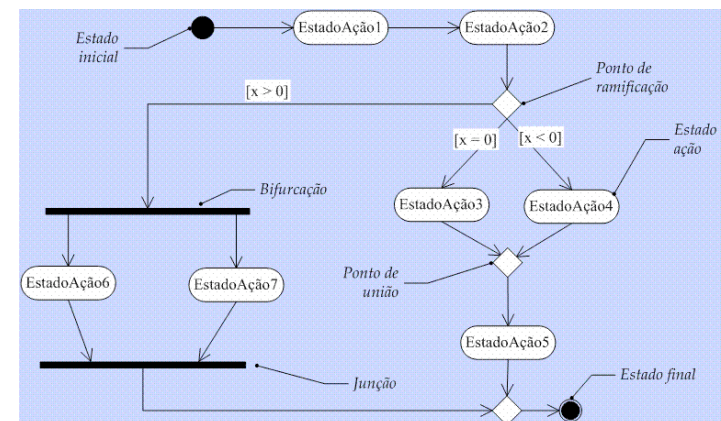


Diagrama de Atividades

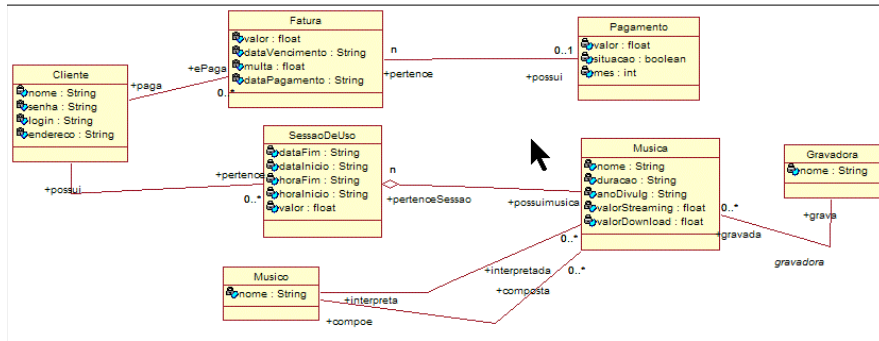


Diagrama de Atividades

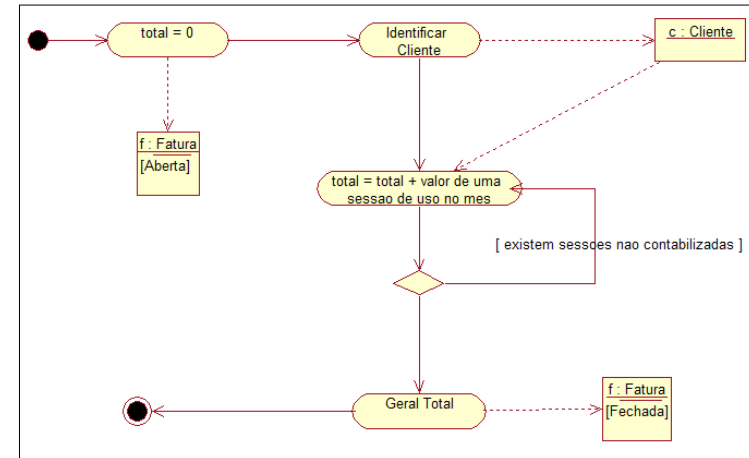
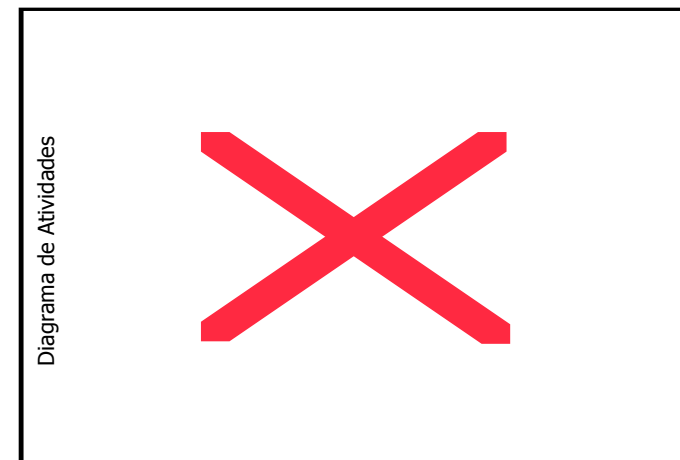


Diagrama de Atividades

■ Concorrência/Paralelismo

- Fluxos de controle paralelos: dois ou mais fluxos sendo executados simultaneamente.
- Uma **barra de bifurcação** recebe uma transição de entrada, e cria dois ou mais fluxos de controle paralelos.
 - cada fluxo é executado independentemente e em paralelo com os demais.
- Uma **barra de junção** recebe duas ou mais transições de entrada e une os fluxos de controle em um único fluxo.
 - Objetivo: sincronizar fluxos paralelos.
 - A transição de saída da barra de junção somente é disparada quando todas as transições de entrada tiverem sido disparadas.

Diagrama de Atividades



Modelagem Temporal com UML

■ Diagrama de Atividades: *Swimlanes*

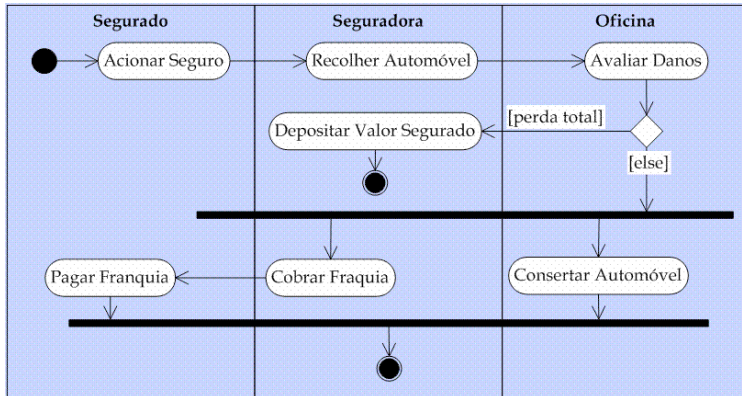
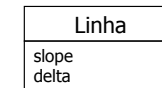


Diagrama de Atividades

■ Geração de código: exemplo (1)

□ Classe Linha: método interseção

- Um parâmetro de entrada (l: Linha) e um de retorno



Slope: inclinação da linha
Delta: deslocamento em relação à origem

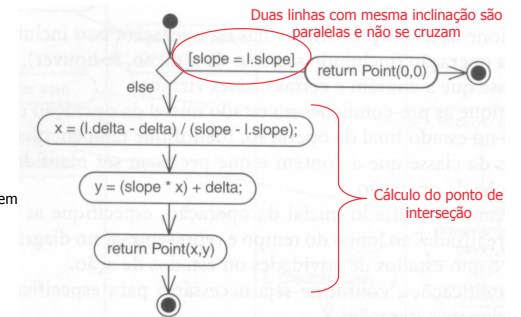
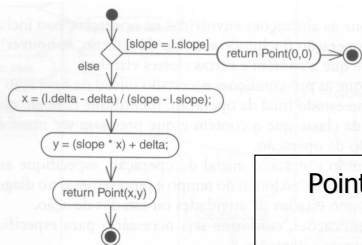


Diagrama de Atividades

■ Geração de código: exemplo (2)



```

Point Line:: intersection (l: Line) {
    if (slope == l.slope) return Point(0,0);
    int x = (l.delta - delta) / (slope - l.slope);
    int y = (slope * x) + delta;
    return Point(x,y);
}
    
```

Diagrama de Atividades

■ Diagrama de Atividades: Uso

- O diagrama de atividades é pouco utilizado na prática para modelagem de aspectos temporais de software
- É fortemente utilizado na modelagem de fluxos de trabalho de processos negócio
 - A própria descrição do Processo Unificado é fortemente baseada em diagramas de atividades

Modelagem Temporal com UML

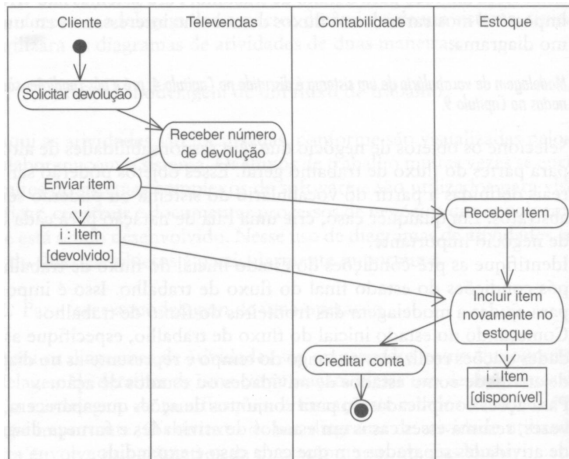


Figura 19.9: A modelagem de um fluxo de trabalho

Modelagem Temporal com UML

Diagramas de Interação:

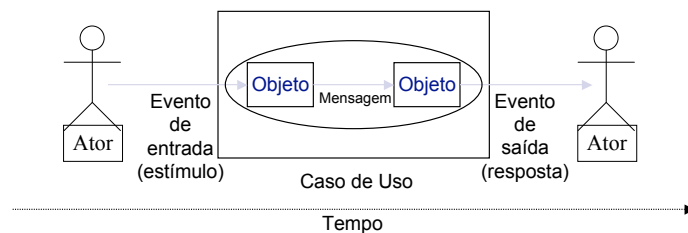
- Diagrama de Seqüência
- Diagrama de Colaboração

Modelagem Temporal com UML

■ Diagramas de Interação

□ Interações entre objetos

- Sequência de trocas de mensagem entre um conjunto de objetos para realizar um caso de uso.



Modelagem Temporal com UML

■ Diagrama de Interação

- Identifica os estados de um objeto em um caso de uso específico
- Aspectos temporais
 - Decisões
 - Ordem dos eventos
- Dois tipos
 - Diagrama de Seqüência
 - Diagrama de Colaboração

Diagrama de Seqüência

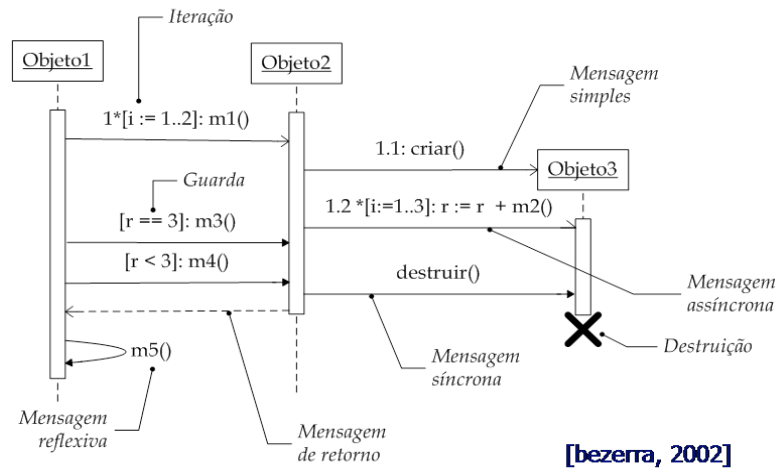


Diagrama de Seqüência

■ Características

- Há preocupação com ordem das ações
- Na Análise, erros e situações de tratamento de exceção **não** são considerados
- Geralmente envolve a interação do sistema com os usuários
- Cada mensagem é rotulada com
 - nome
 - argumentos
 - informações de controle
 - Condições de guarda

Diagrama de Seqüência

■ Características

- Principal objetivo: identificar quais mensagens devem ser implementadas pelas classes
 - Classe/Objeto Emissor da mensagem é um Cliente
 - Classe/Objeto Receptor da mensagem é um Servidor

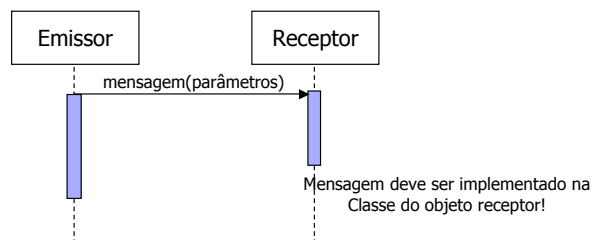


Diagrama de Seqüência

■ Características

- Tipos de Interação

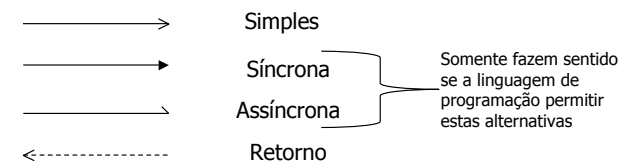


Diagrama de Seqüência

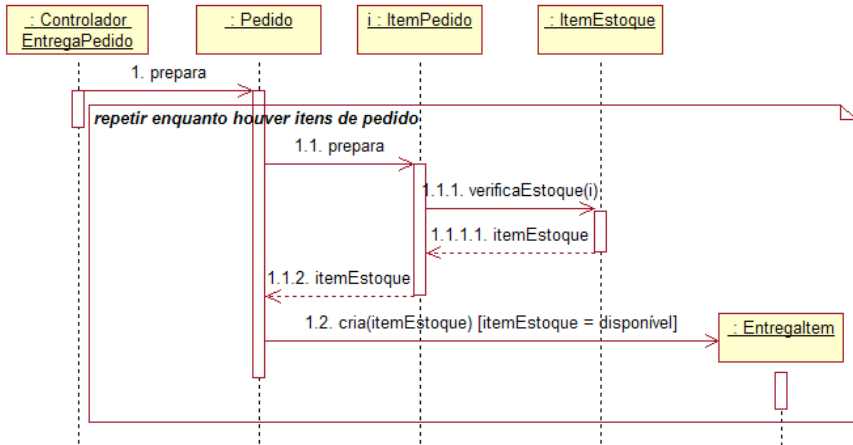


Diagrama de Seqüência

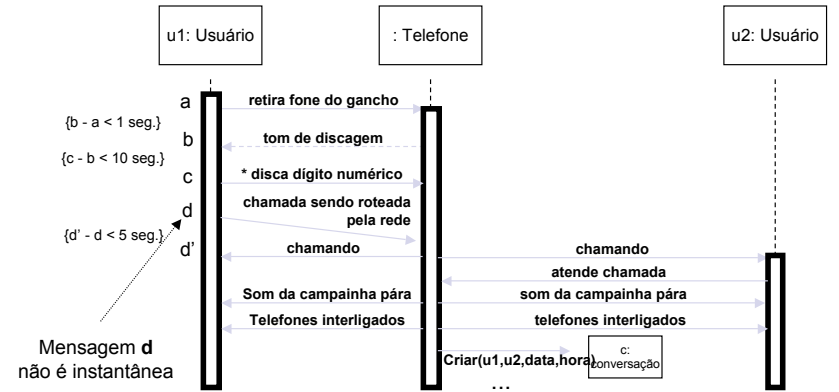


Diagrama de Seqüência

■ Correspondência entre código e diagrama

```

Reserve(o: Order)
{
    OrderPosition opos;
    Article article;
    int number;
    for (int i=1; i < o.getOrderNumber(); i++) {
        opos = o.giveOrderPosition(i);
        article = opos.giveArticle();
        number = opos.giveNumber();
        articleStock.reserve(article, number);
    }
}
    
```

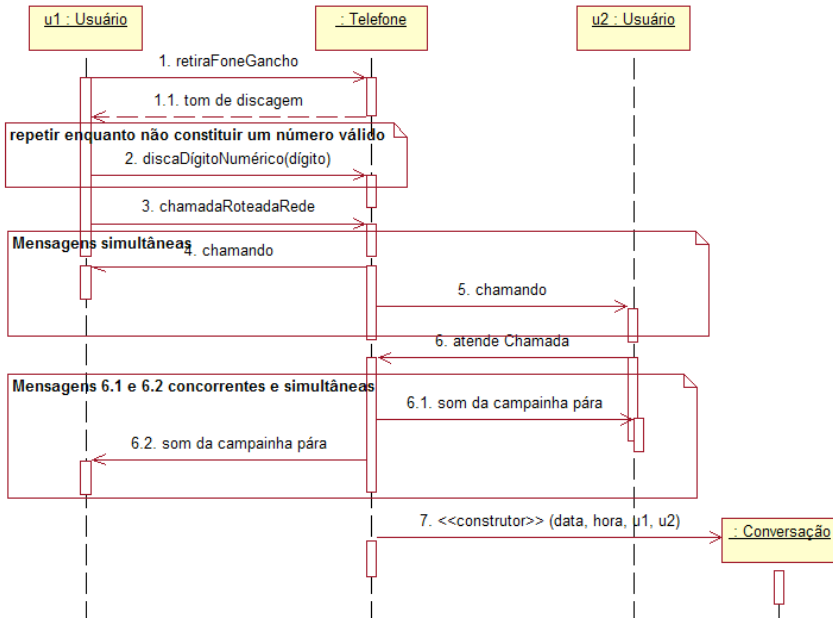


Diagrama de Seqüência

■ Sistema de Ponto de Vendas

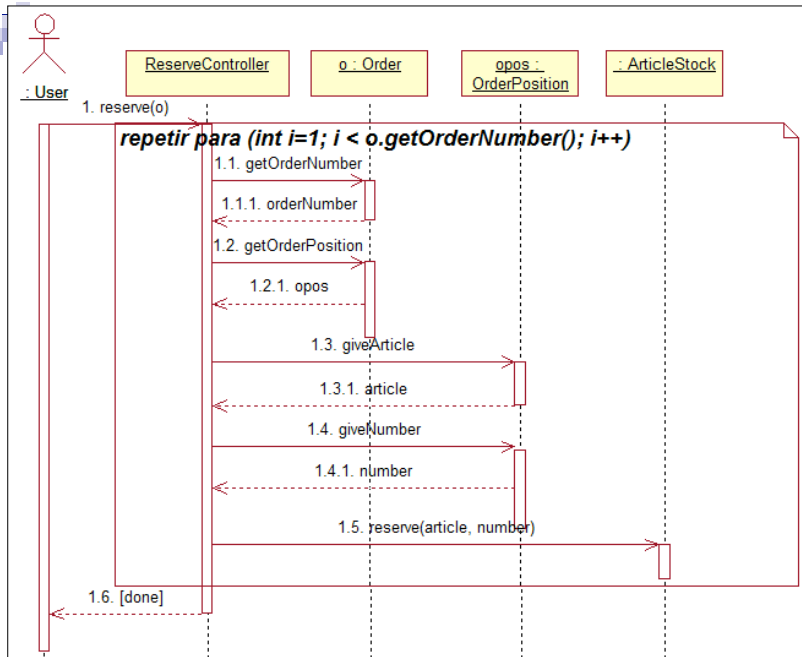
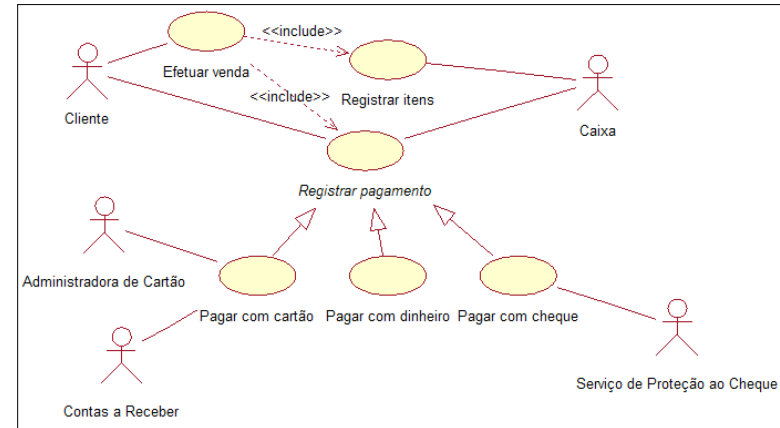
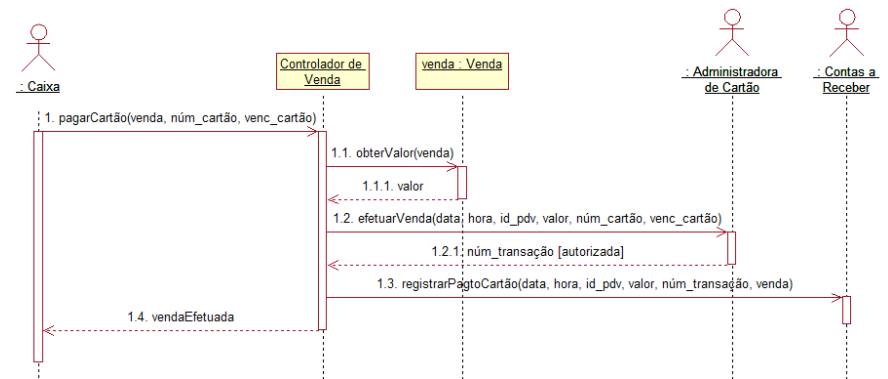


Diagrama de Seqüência

Pagar com cartão



Registrar Itens

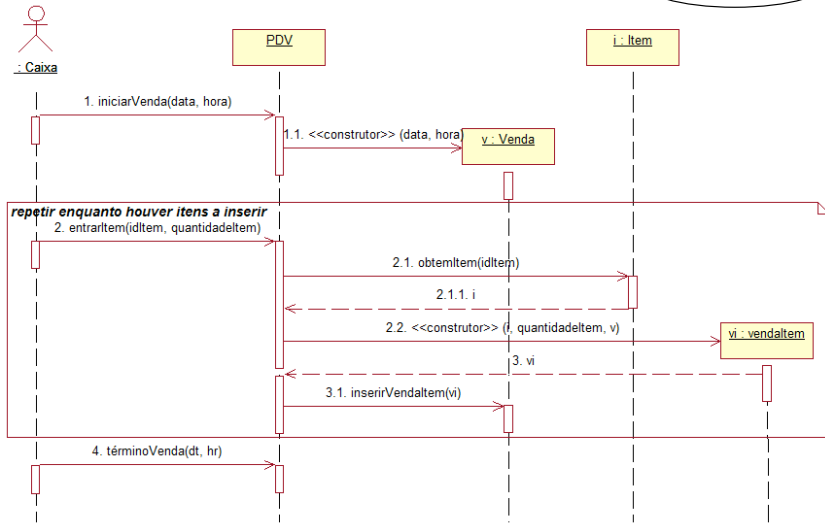


Diagrama de Colaboração

■ Características

- Apresenta a mesma informação do diagrama de sequência mas não representa o tempo como uma dimensão separada)
- Utilização: visão geral da interação de objetos
 - Condensa em um único diagrama toda a troca de mensagens entre um grupo de classes
 - Normalmente é gerado automaticamente pelas ferramentas CASE a partir dos diagramas de seqüência

Diagrama de Colaboração

■ Notação

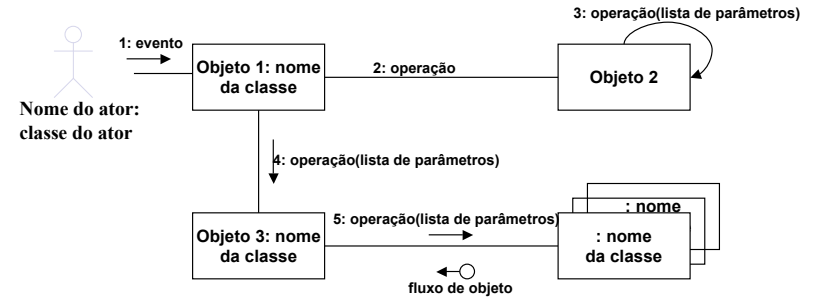


Diagrama de Colaboração

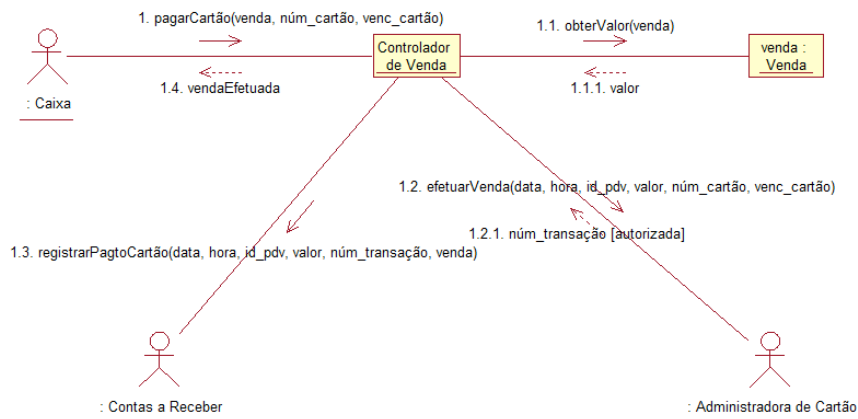


Diagrama de Colaboração

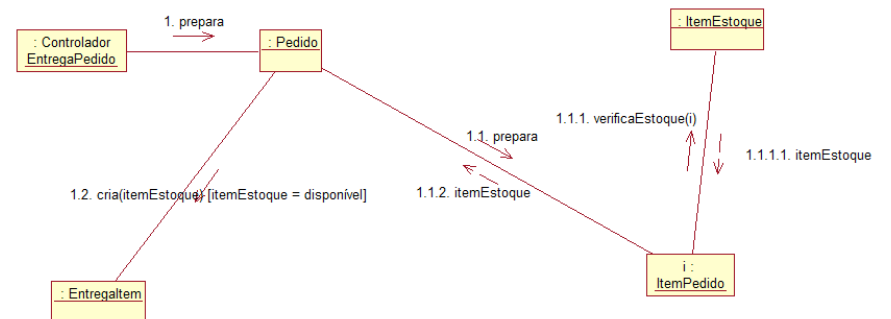
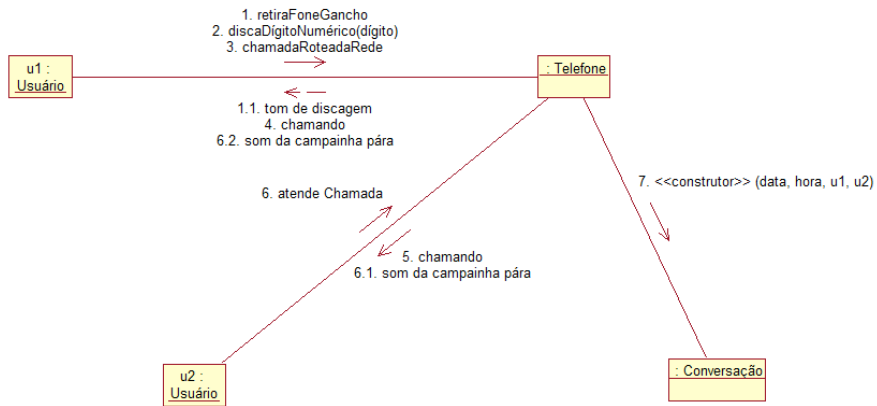


Diagrama de Colaboração



Integração dos Modelos

