

## Aula 09 - Diagrama de Atividade

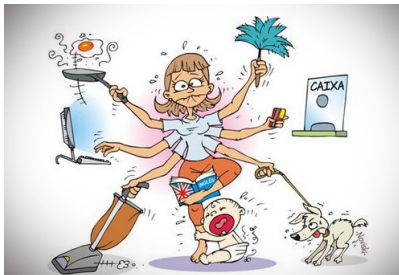
Rafaela Priscila Cruz Moreira / Samara Soares Leal

**Curso:** Ciência da Computação / Sistemas de Informação

**Disciplina:** Requisitos, Análise e Projeto Orientado a Objetos

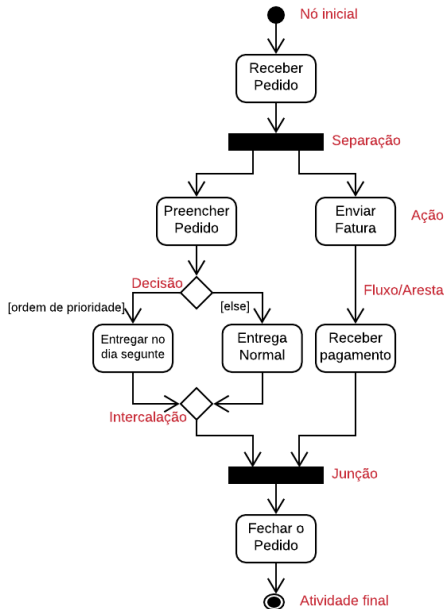
# Diagrama de Atividades

- Modelo gráfico para representar o **comportamento** do sistema de software.
- Técnica usada para descrever lógica de métodos, fluxo de trabalho ou cenários de casos de uso.
- Diagrama de Atividades X Fluxograma:
  - Ambos descrevem o fluxo de negócio.
  - **Principal diferença:** Diagramas de atividade suportam comportamento paralelo.



- Podem ser utilizados para:
  - **Modelar o Fluxo de Trabalho** (nível especificação):
    - \*. Modelar e detalhar as regras de negócios do sistema, considerando o relacionamento entre vários sistemas e atores.
    - \*. Modelar os fluxos de um caso de uso.
  - **Modelar uma Operação** (nível implementação/projeto):
    - \*. Modelam o funcionamento lógico de métodos (aspectos do código-fonte).

# Diagrama de Atividades - Exemplo 1



## Componentes:

- **Nó inicial:** Descreve o início do diagrama
- **Ações:** Representam um passo do fluxo de trabalho do diagrama.  
**Exemplo:** Receber Pedido, Preencher pedido, etc.
- **Separação:** Possui um fluxo de entrada e vários fluxos concorrentes de saída.  
**Exemplo:** Preencher Pedido, Enviar Fatura e as ações subsequentes ocorrem em paralelo, ou seja, a sequência entre elas é irrelevante.
- **Junção:** O fluxo de saída é executado somente quando todos os fluxos de entrada chegarem à junção.  
**Exemplo:** O pedido só é fechado quando entregue e pago.

## Componentes:

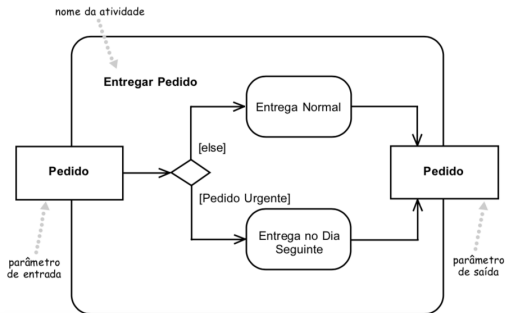
- **Decisão:** Possui um único fluxo de entrada e vários de saída. Cada fluxo de saída tem uma sentinela: uma condição booleana colocada entre colchetes.  
**Exemplo:** Depois que um pedido é preenchido, há uma decisão: Se for um pedido urgente, será uma "Entrega no Dia Seguinte"; caso contrário, uma "Entrega Normal".
- **Intercalação:** Marca o final de um comportamento condicional iniciado por uma decisão.

## Componentes

- **Sub-atividades:**

Ações podem ser decompostas em sub-atividades.

**Exemplo:** Pode-se pegar a lógica de entrega do exemplo e defini-la como sua própria atividade.

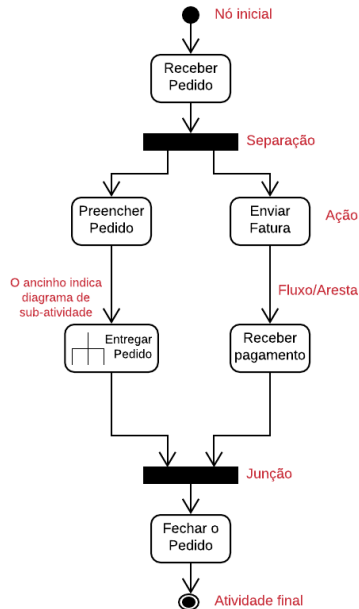


# Diagrama de Atividades - Exemplo 1

## Componentes

- **Sub-atividades:** Ações podem ser decompostas em sub-atividades.

Continuação... Então, pode-se chamá-la como uma ação:

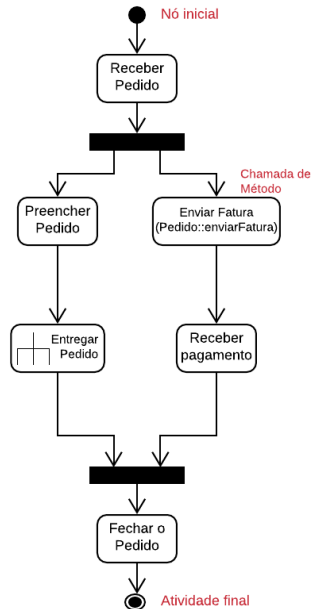




# Diagrama de Atividades - Exemplo 1

## Componentes

- **Métodos:** As ações também podem ser implementadas como métodos nas classes.
  - \* **Sintaxe:**  
*nome-da-classe::nome-do-método.*
  - \* Pode-se escrever um fragmento de código no símbolo de ação, se o comportamento executado não for uma única chamada de método.

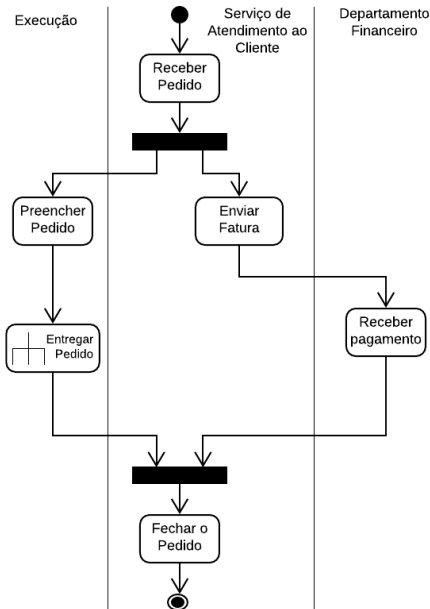


# Diagrama de Atividades - Exemplo 1

## Componentes

- **Partições:** Pode-se dividir um diagrama de atividades em partições (raias de piscina) para mostrar quem faz o que (os responsáveis pelas ações).

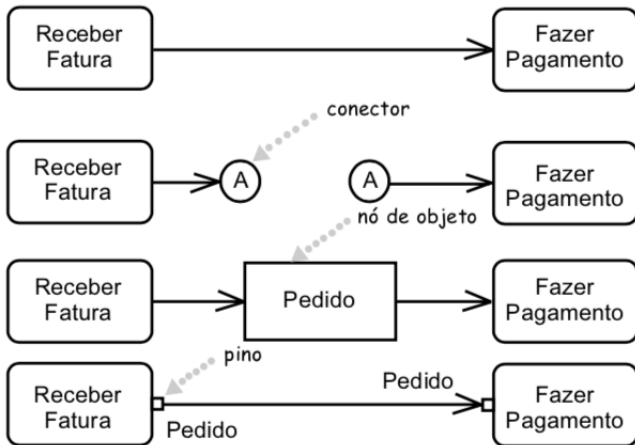
**Exemplo:** As ações envolvidas no processamento do pedido podem ser separadas entre vários departamentos.



# Diagrama de Atividades - Exemplo 1

## Componentes

- **Fluxos e Arestas:** Descreve as conexões entre ações.



## Componentes

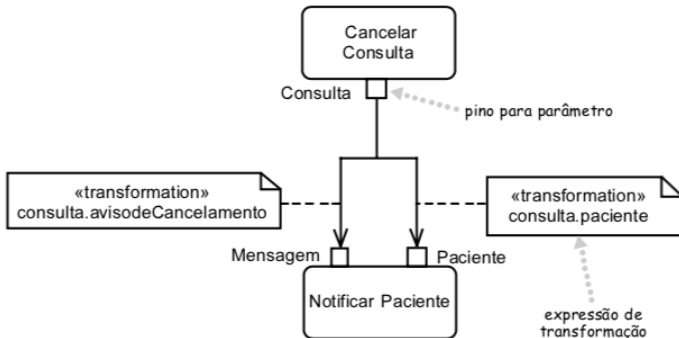
- **Fluxos e Arestas:** Descreve as conexões entre ações. Tipos:
  1. **Setas:** O sentido da seta indica o sentido do fluxo.
  2. **Conectores:** Rótulos que determinam a continuação do fluxo.
  3. **Objetos:** Objeto que está sendo passado entre as ações.
  4. **Pinos:** Representam parâmetros entrando ou saindo de uma ação
    - Quando um parâmetro de saída de uma ação não corresponde ao parâmetro de entrada em outra ação, usa-se uma **transformação** para indicar a modificação.
    - A transformação deve ser simples, como comando `get` sobre o pino de saída da ação anterior. modificação

## Componentes

- **Fluxos e Arestas:** Descreve as conexões entre ações. Tipos:

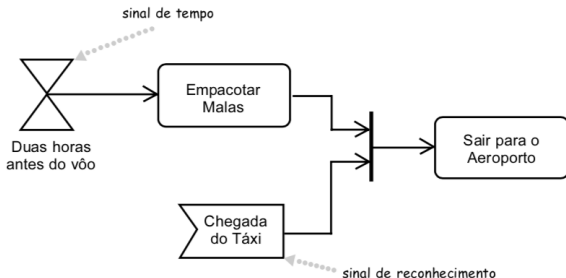
4. **Pinos:** Representam parâmetros entrando ou saindo de uma ação

- Exemplo:



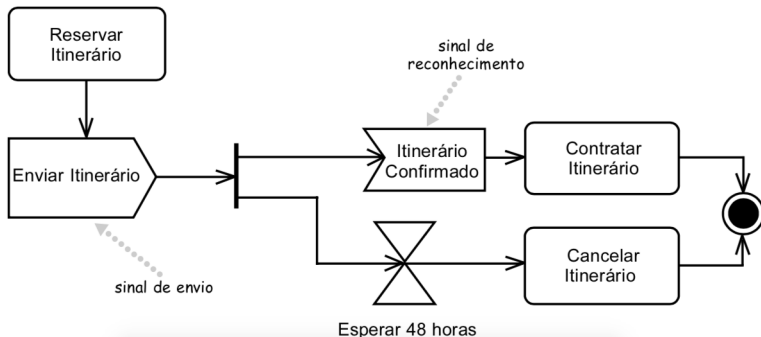
## Outros Componentes...

- **Sinais de tempo:** ocorrem devido à passagem do tempo.
- **Sinais de Reconhecimento:** indica que a atividade recebe um evento de um processo externo
- **Exemplo:**
  - 2 horas antes do voo sair, pode-se começar a fazer as malas.
  - Mesmo com as malas prontas, ainda não é possível sair enquanto o táxi não chegar.
  - Se o táxi chegar antes das malas estarem prontas, ele precisará esperar, antes de sair.



## Outros Componentes...

- Assim como se pode aceitar sinais, também é possível enviá-los.
- Útil quando é necessário enviar uma mensagem e depois esperar pela resposta, antes de continuar.
  - Exemplo:** Na figura abaixo, dois fluxos estão em uma disputa: o primeiro a chegar ao estado final vencerá e terminará o outro fluxo.



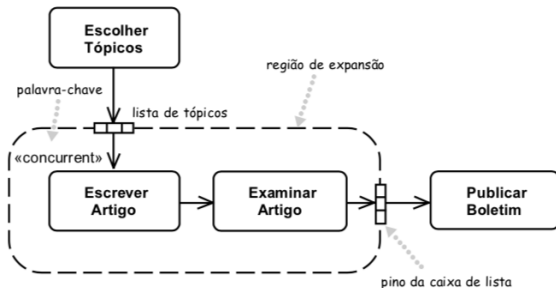
## Outros Componentes...

- **Tokens:** Contadores usados para controlar o fluxo do diagrama.
  - **Exemplo:** O nó inicial cria um token, o qual é passado para a próxima ação, que é executada e depois passa o token para a seguinte.



## Outros Componentes...

- **Região de Expansão:** área do diagrama onde as ações ocorrem uma vez para cada item de uma coleção.
  - **Exemplo:** "Escolher Tópicos" gera uma lista como saída. Cada elemento da lista será entrada para "Escrever Artigo". Cada ação "Examinar Artigo" gera um artigo, adicionado à lista de saída da região, gerando um único token para a lista, que é passado para "Publicar Boletim".



## Outros Componentes...

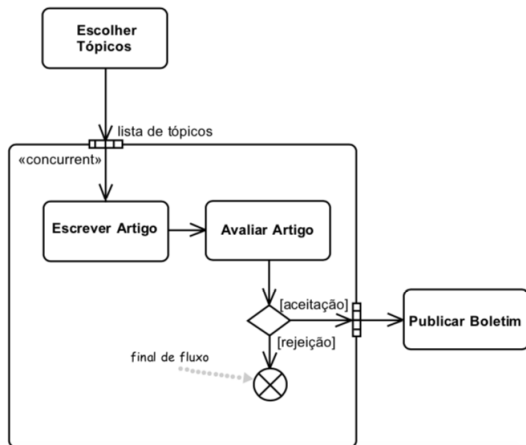
- **Região de Expansão:**

- A região de expansão recebe uma lista e pode ter como saída outra lista de mesmo tamanho ou **não**.
- Quando a lista sai com menos elementos, diz-se que a região de expansão age como um filtro.
- Para que as iterações da região seja executadas em paralelo, deve-se usar um estereótipo *concurrent*.

## Outros Componentes...

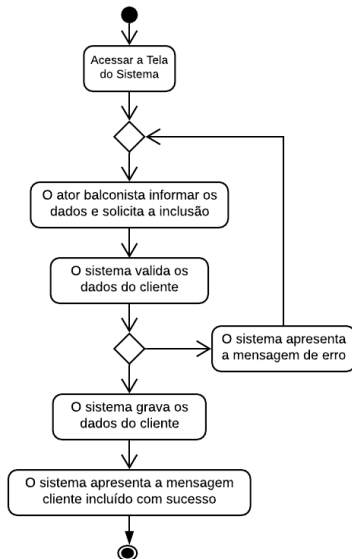
- **Final de Fluxo:**

- Indica o término de um fluxo em particular, sem terminar a atividade inteira .
- Usado com as regiões de expansão para reduzir o número de itens na lista de saída, transformando essa região em um filtro.



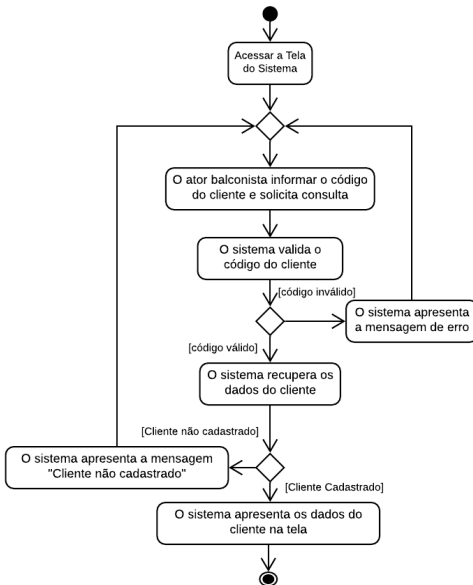
# Mais exemplos - Incluir Clientes

Diagrama de atividades para o cenário do caso de uso **Incluir Cliente**:



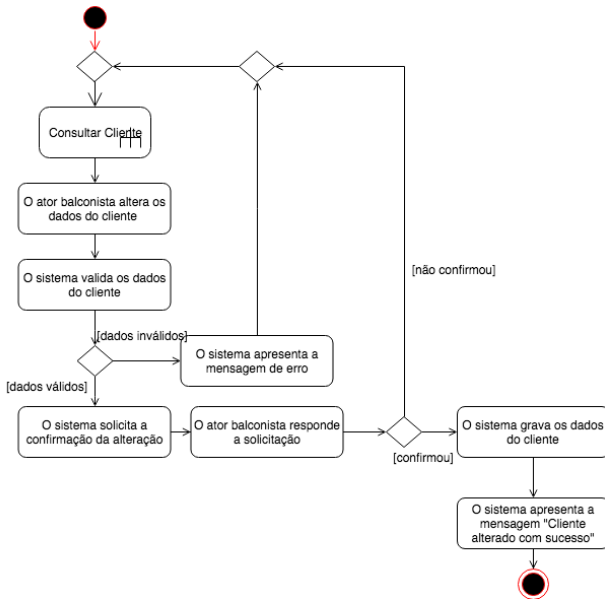
# Mais exemplos - Consultar Clientes

Diagrama de atividades para o cenário do caso de uso **Consultar Cliente**:



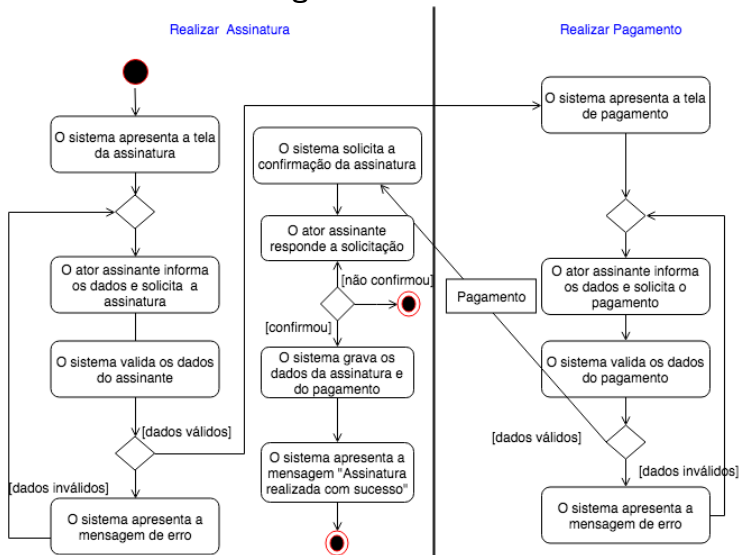
# Mais exemplos - Alterar Clientes

Diagrama de atividades para o caso de uso **Alterar Cliente**:



# Mais exemplos - Realizar Assinatura e Informar Pagamento

Diagrama de atividades para os cenários dos casos de uso **Realizar Assinatura** e **Realizar Pagamento**:



# Diagrama de Atividades - Quando utilizar?



- Para usufruir das vantagens das separações e junções na escrita de algoritmos **paralelos** para programas concorrentes.

**Maior qualidade dos diagramas de atividade:** suportam o comportamento paralelo.

- Para modelar processo de trabalho
- Para modelar um método
- Para modelar um cenário de caso de uso.





-  LARMAN, C. Utilizando UML e Padrões. Bookman, 2002. ISBN 9788577800476. Disponível em: <<https://books.google.com.br/books?id=hzl2tmT8QkUC>>.
-  SOMMERVILLE, I. Engenharia de software. PEARSON BRASIL, 2011. ISBN 9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>>.