

# Projeto de Bloco

## Engenharia Disciplinada de Softwares

Graduação em Engenharia de Software - 2020

# Apresentações

Visão Geral do Projeto de Bloco

# Armênio Cardoso

- Iniciei minha carreira profissional em **1986** e desde **1990** procuro conciliar o trabalho em Desenvolvimento de Software com o de Professor.
- Participei em diversos projetos, atuando com modelagem, arquitetura e programação. Fui Desenvolvedor Pascal, Clipper, Visual Basic e C/C++.
- Trabalhei como Tradutor / Revisor de livros técnicos e Professor em várias instituições.
- Em **2002** vim para o Infnet onde dou aulas nos cursos de Graduação, Pós-Graduação e de Extensão.
- Desde **2012** trabalho como Engenheiro de Software em uma empresa da área de Medicina Diagnóstica.



# AVA - Ambiente Virtual de Aprendizado (Moodle)

- Linha Mestra para o acompanhamento do Projeto de Bloco.
- Cronograma:
  - **Pontualidade:** acompanhar o mais próximo possível as aulas.
  - **Competências:** adquirir as competências propostas através do **estudo cuidadoso** dos materiais e **prática dos exercícios**.
  - As competências serão apresentadas ao longo do tempo em uma ordem bem planejada.

# AVA - Ambiente Virtual de Aprendizado (Moodle)

- Bibliografia: será necessário se preparar para as aulas.
- Documento de Visão do Projeto de Bloco.
- Etapas:
  - Roteiros de Estudo com datas de início e fim.
  - Competências.
  - Slides das Aulas.
- Avaliações:
  - Regras gerais para todos os TPs e AT.
  - TPs = Construção do Projeto de Bloco = AT.

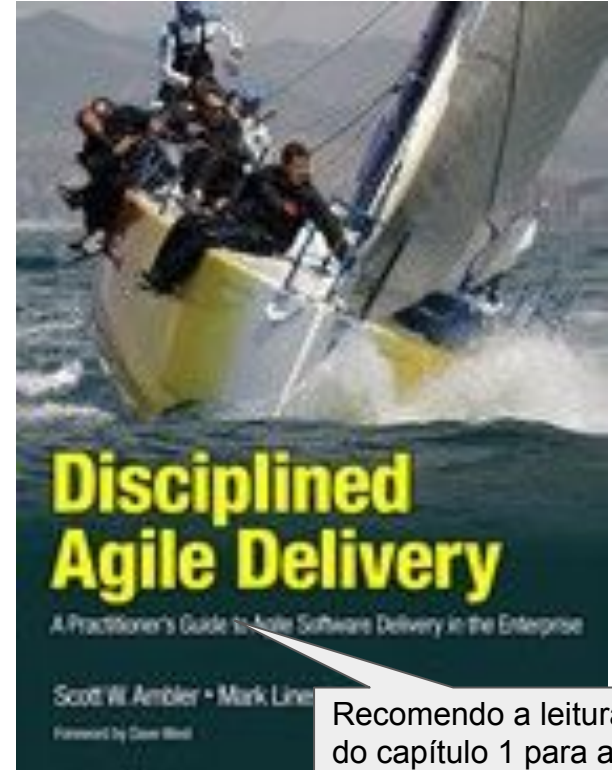
# Bibliografia

Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise

Scott Ambler / Mark Lines

IBM Press

ISBN: 9780132810098



Recomendo a leitura do capítulo 1 para a próxima aula.

# Etapa 1 Aula 1

Conceber um Projeto de Software

# Requisitos



# O Que São Requisitos?

- Os requisitos expressam as características e restrições do produto de software, do ponto de vista de **satisfação das necessidades do usuário**.
- Independem da tecnologia empregada na construção da solução.
- **É a parte mais crítica e propensa a erros no desenvolvimento de software.**

# O Que São Requisitos?

- Definição de Requisito:
  - Uma condição ou capacidade com a qual o sistema deve estar em conformidade.
  - Uma especificação do que deve ser implementado ou uma restrição de algum tipo do sistema.

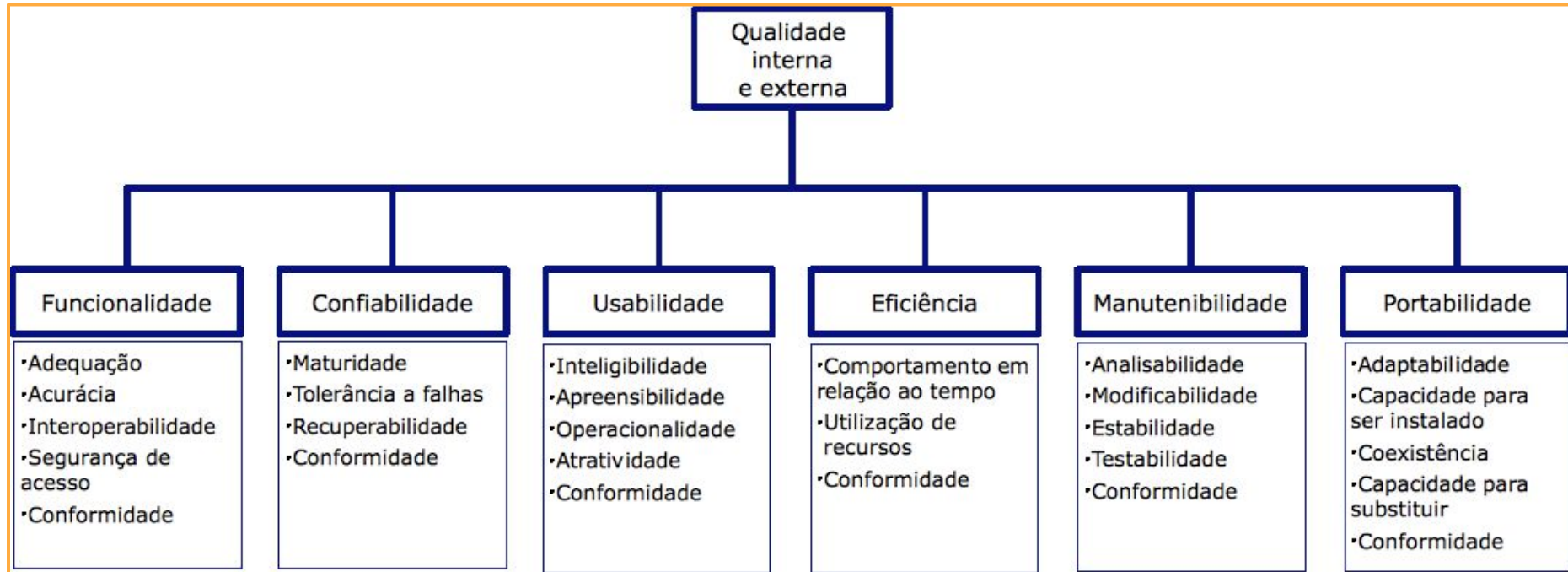
# Requisitos Funcionais

- Definem o **comportamento do sistema** a partir da perspectiva do usuário.
- Que tarefas o sistema deverá ter para atender às necessidades dos usuários?
- Uma dica para definir um **Requisito Funcional** é utilizar a seguinte frase:
  - **O sistema deve...**
  - Depois, coloca-se o verbo e o complemento, como, por exemplo, ... **realizar o cadastramento de funcionários.**

# Requisitos Não Funcionais

- Definem **características qualitativas** do sistema, recursos que o sistema precisa ter para cumprir os seus objetivos.
  - É necessário fazer um inventário dos **Requisitos Não Funcionais** nas entrevistas com os interessados, relacionando tudo o que é necessário para o sistema.
  - Na Arquitetura, os **Requisitos Não Funcionais** vão influenciar nas decisões de como o sistema deverá ser construído.

# Requisitos Não Funcionais



# Requisitos Não Funcionais

- **Exemplos:**

- A base de dados deve ser protegida para acesso apenas por usuários autorizados.
- O tempo de resposta do sistema não deve ultrapassar 30 segundos.
- O software deve rodar no sistema Linux.
- O tempo de desenvolvimento não deve ultrapassar seis meses.

# O Bom Requisito

- **Completo** – Eu leio e entendo. Não preciso de figuras para explicar.
- **Correto** – Só o usuário pode dizer se o requisito está correto. Não são papéis nem analistas, apenas o usuário diante do software.
- **Viável** – Existem dois tipos de viabilidade: técnica e financeira. A viabilidade técnica é olhar o requisito e verificar se é possível fazer. Existe a viabilidade técnica, mas o cliente está disposto a pagar?

# O Bom Requisito

- **Necessário** – Às vezes, o usuário se apegua a uma ideia que sabemos não ser necessária ao funcionamento do sistema.
- **Priorizado** – Toda vez que for entrevistado um novo usuário, os requisitos levantados entram na fila de acordo com o seu grau de prioridade.
- **Não ambíguo** – Tem que ser rápido demais. Como medir? É preciso ter algo absoluto ou uma referência. Por exemplo: essa tela tem que ser inicializada em cinco (5) segundos.
- **Verificável** – É necessário que o requisito possa ser testado; caso contrário, não pode estar na fila.



# Ficha de Requisitos

- Em uma reunião de Levantamento de Requisitos, é necessário obter o máximo de informações possíveis e registrar os requisitos.
- Uma técnica de registro de requisitos é criar uma “ficha” para cada um deles.

ID	DESCRIÇÃO
Prioridade	

# Ficha de Requisitos

- A prioridade pode ser classificada como:
  - **Essencial** – é o requisito sem o qual o sistema não entra em funcionamento. Tem que ser implementado.
  - **Importante** – é um requisito que permite ao sistema entrar em funcionamento, mas não de forma satisfatória.
  - **Desejável** – são os requisitos que não comprometem a funcionalidade básica do sistema. O sistema funciona de forma satisfatória sem eles. Estes requisitos podem ser deixados para versões posteriores do sistema.

# Uso da Ficha

- Como exemplo, vamos estudar como seria o Registro de Requisitos de um sistema de Imobiliária que foi feito por um Analista da nossa equipe.
- Requisitos Não Funcionais:

ID	DESCRIÇÃO
[NF001]	O sistema deverá ter uma interface amigável ao usuário iniciante sem tornar-se cansativa aos usuários mais experientes.
Prioridade	Importante

ID	DESCRIÇÃO
[NF002]	O sistema deverá funcionar em qualquer sistema operacional que utilize um navegador web e que siga as recomendações da W3C.
Prioridade	Essencial.

# Uso da Ficha

- Requisitos Funcionais:

<b>ID [RF001]</b>	
<b>NOME</b>	Manter Corretor
<b>DESCRIÇÃO</b>	O usuário entrará com os dados do corretor de forma a cadastrá-lo no sistema. Poderá também alterar, excluir e selecionar os dados.
<b>Prioridade</b>	Essencial

<b>ID [RF002]</b>	
<b>NOME</b>	Manter Cliente
<b>DESCRIÇÃO</b>	O usuário entrará com os dados do cliente de forma a cadastrá-lo no sistema. Poderá também alterar, excluir e selecionar os dados.
<b>Prioridade</b>	Essencial

<b>ID [RF003]</b>	
<b>NOME</b>	Manter FAC – Ficha de Atendimento ao Cliente
<b>DESCRIÇÃO</b>	O usuário entrará com os dados da FAC de forma a realizar o cadastro no sistema. Poderá também alterar, excluir e selecionar os dados.
<b>Prioridade</b>	Essencial

# Uso da Ficha


- Requisitos Funcionais:

<b>ID [RF004]</b>	
<b>NOME</b>	Manter Construtora
<b>DESCRIÇÃO</b>	O usuário entrará com os dados da construtora de forma a cadastrá-la no sistema. Poderá também alterar, excluir e selecionar os dados.
<b>Prioridade</b>	Essencial

<b>ID [RF005]</b>	
<b>NOME</b>	Manter Empreendimento
<b>DESCRIÇÃO</b>	O usuário entrará com os dados do empreendimento de forma a cadastrá-lo no sistema. Poderá também alterar, excluir e selecionar os dados.
<b>Prioridade</b>	Essencial

# Matriz de Requisitos

- Uma outra forma de registrar os requisitos é através de uma tabela em um documento.

 Cliente NOME DA UNIDADE/SETOR	MR – MATRIZ DE REQUISITOS NOME DO PROJETO		Vr. N.	Revisão Dd/MM/yyy	
	Usuário Gestor NOME USUARIO GESTOR		Elaborador NOME ELABORADOR		

APROVAÇÕES	
Usuário Gestor:	____/____/____
Gerente:	____/____/____
Coordenador Desenvolvimento:	____/____/____

REQUISITOS					
Nº	Tipo	Descrição	Caso de Uso	Co	Artefato
		Descrever o requisito	Identificar o Caso de Uso		Identificar o componente de software .DLL, Wservice, etc...

# Matriz de Requisitos

- A Matriz tem como vantagem o agrupamento de todas as informações em um único documento.
- Na Matriz, informamos o N° do requisito, o tipo NF para não funcional e RF para funcional, a descrição, o caso de uso e os artefatos correspondentes.

# Casos de Uso



# Fundamentos

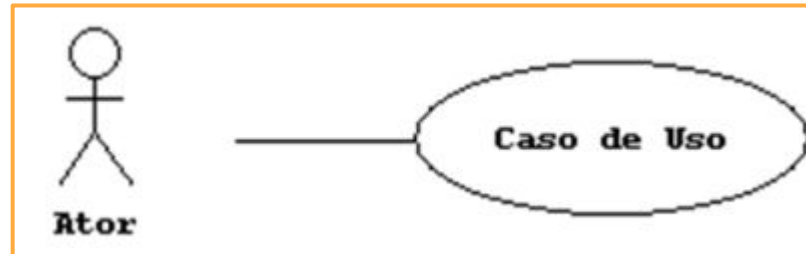
- Um caso de uso descreve o comportamento do sistema do **ponto de vista do usuário**, fornecendo uma descrição funcional.
- Este diagrama descreve interações do sistema com o exterior (atores).
- Novas funcionalidades são agregadas ao contexto com a inclusão de novos elementos no diagrama.

# Fundamentos

- Objetivos do Diagrama de Casos de Uso:
  - Descrever os requisitos funcionais do sistema – “O Que” o sistema deve fazer.
  - Descrever claramente as responsabilidades que devem ser cumpridas pelo sistema.

# Componentes

- Em um diagrama de caso de uso, aparecem:
  - **Ator**: gerador de estímulos no sistema.
  - **Processo**: vai absorver e processar os estímulos (eventos) do ator. É o caso de uso propriamente dito.

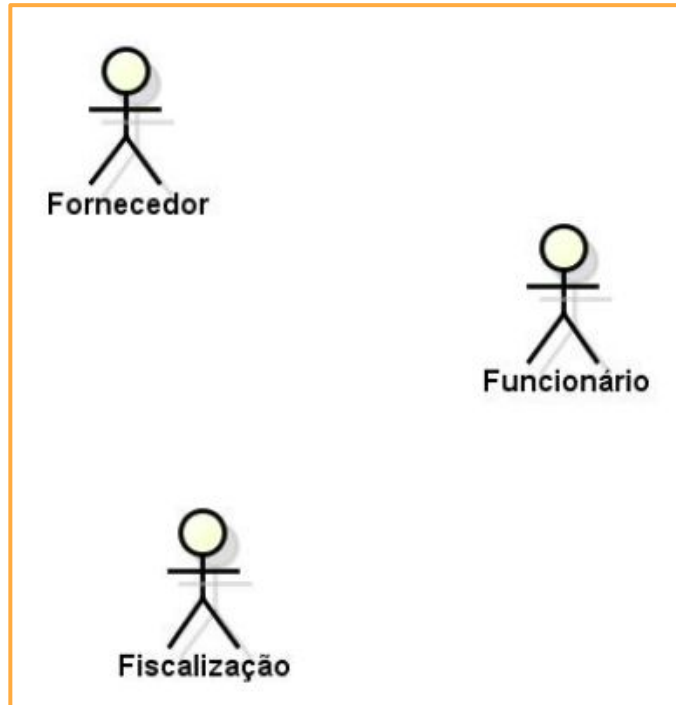


# Conceito de Ator

- Ator é um agente que interage com o sistema.
- Um ator é uma classe, não uma instância.
- Representa uma regra, um papel e não um usuário individual do sistema.
- O nome do ator deve refletir o seu papel.
- Exemplos de atores: operador, cliente, gerente, atendente, consultor, sistema financeiro, sistema contábil, **tempo**.

# Conceito de Ator

- Um Ator é uma classe com um ícone padrão.



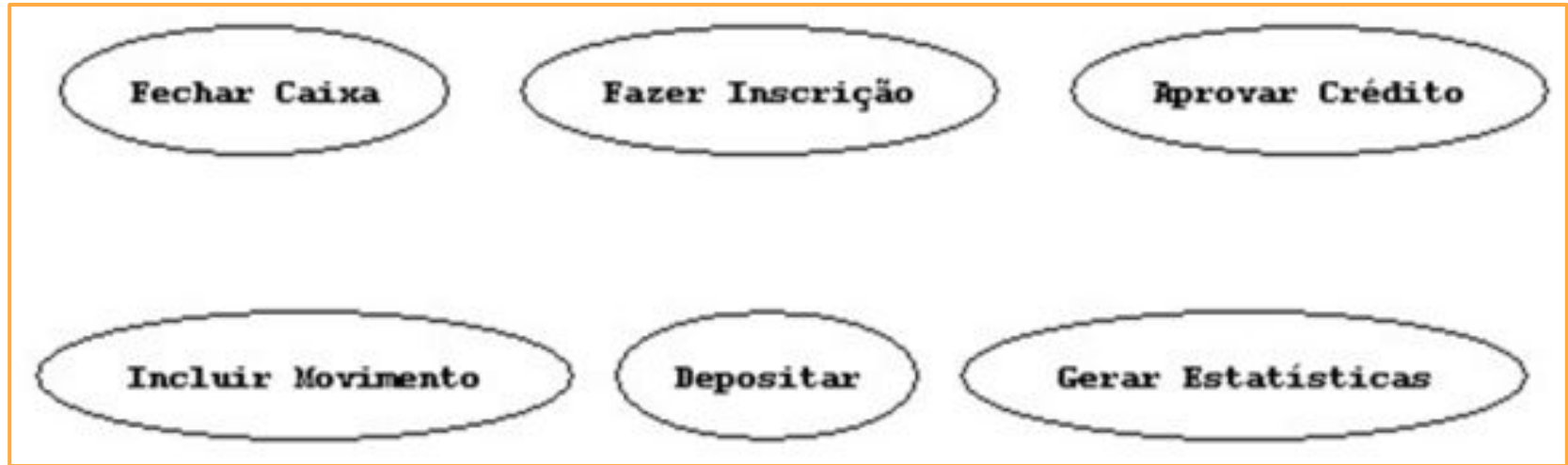
# Conceito de Ator

- Para facilitar a identificação dos atores, faça as seguintes perguntas:
  - Quem irá usar as principais funcionalidades do sistema?
  - Quem irá administrar e manter o sistema?
  - O sistema irá interagir com outros sistemas?
  - O sistema dispõe de algum mecanismo automático ou de algum processo que deva ocorrer em um dado momento?

# Conceito de Caso de Uso

- É uma Sequência de ações que um sistema desempenha para produzir um **resultado observável** por um ator específico.
- O nome do Caso de Uso deve ser uma frase indicando a ação realizada.
- Um caso de uso é um conjunto de passos, alternativas e o tratamento das suas possíveis exceções.

# Conceito de Caso de Uso

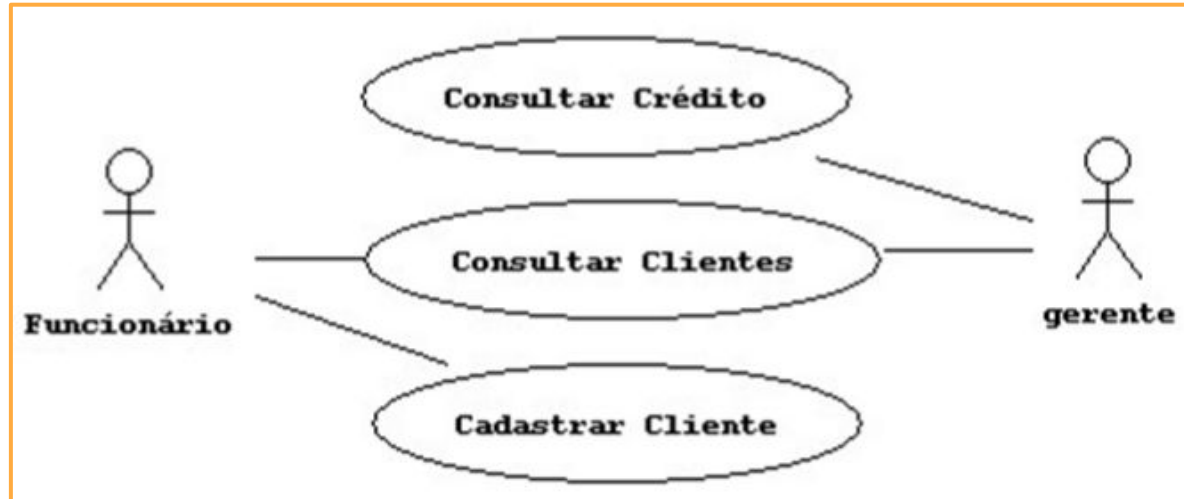


**Caso de Uso é o propósito do Ator no sistema**



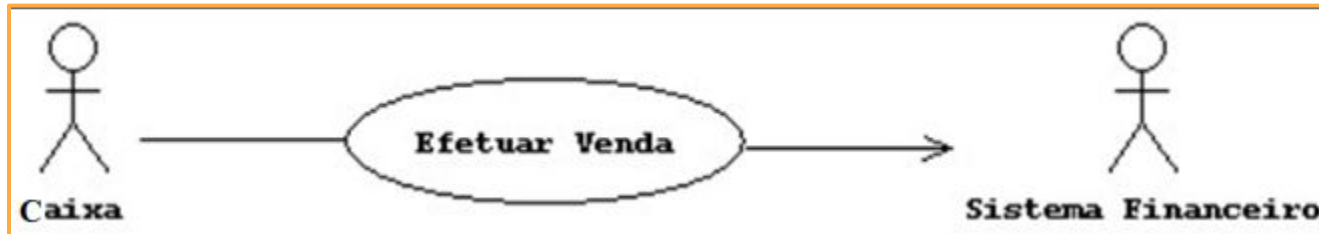
# Conceito de Caso de Uso

- É sempre inicializado por um ator e devolve uma resposta.
- Os casos de uso são conectados aos atores com associações de comunicação. A associação é bidirecional.
- Um caso de uso tem início, meio e fim.



# Conceito de Caso de Uso

- Um ator se comunica com o sistema enviando e recebendo mensagens para um Caso de Uso.
- Mensagens = Estímulos = Eventos.
- Existem dois tipos de atores:
  - **Ativos**, iniciam algum Caso de Uso.
  - **Passivos**, recebem mensagens de um Caso de Uso (em geral são sistemas).

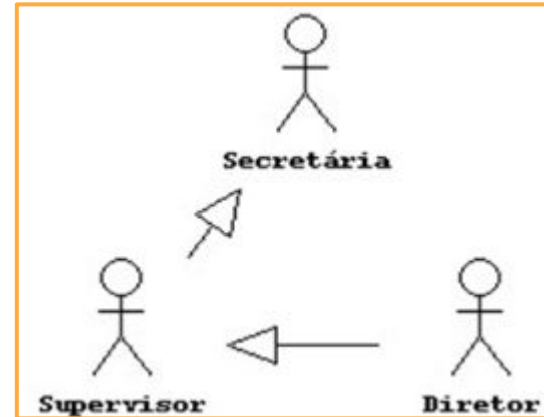
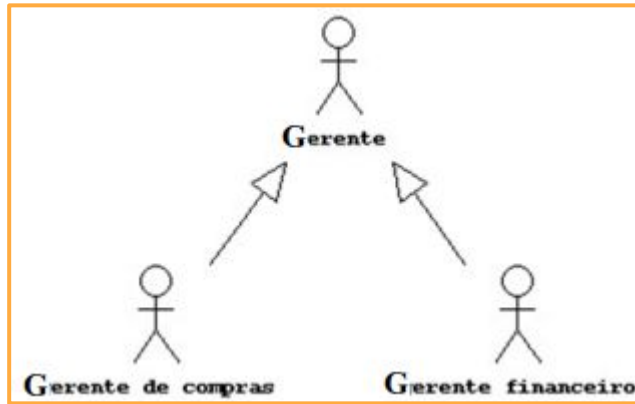


# Conceito de Caso de Uso

- Os relacionamentos indicam a existência de comunicação entre atores e casos de uso.
- Um caso de uso pode estar associado a mais de um ator.
- A comunicação será representada como uma ligação sem direção, em geral.
- Quando a iniciativa parte do caso de uso, a comunicação deve ser direcionada por uma seta.

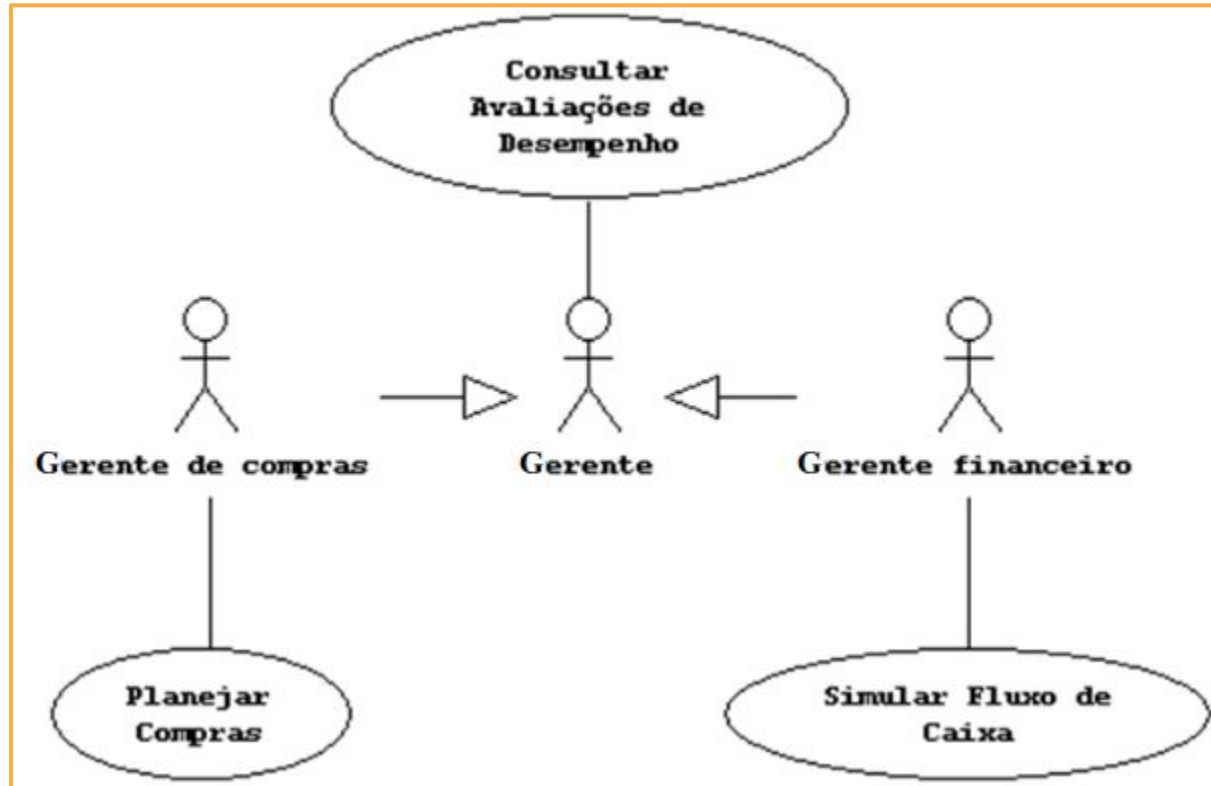
# Generalização

- Os diagramas de casos de uso podem ser simplificados por meio da herança entre atores.



**Não confundir com herança de classes**

# Generalização

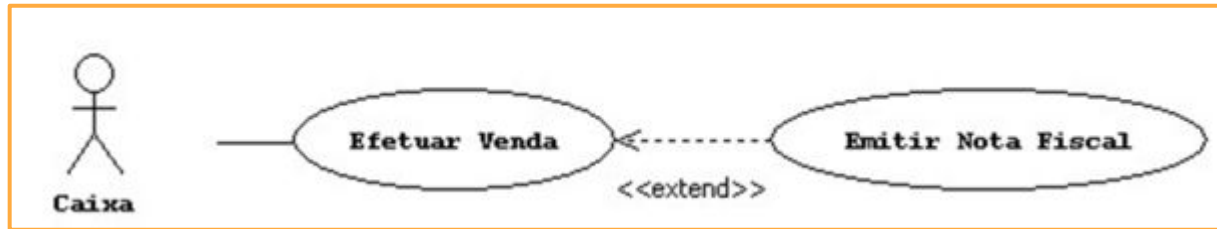


# Casos de Uso Secundários

- Casos de uso secundários são utilizados para facilitar a descrição de **funcionalidades mais complexas**.
- Simplificam o comportamento dos casos de uso primários através dos mecanismos de Extensão e Inclusão.
- Devem ser usados com muito cuidado para não decompor o diagrama sem necessidade (Caso de Uso = Macrofuncionalidades).

# Extensão

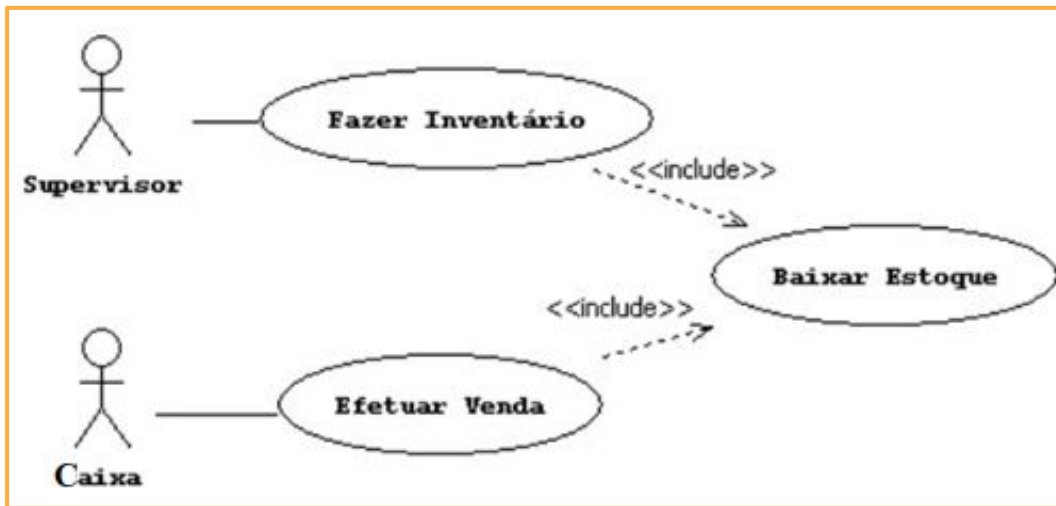
- Essa notação pode ser usada para representar fluxos complexos opcionais.
- Dependem do Caso de Uso principal (veja a seta pontilhada).



**A emissão da nota fiscal pode ser feita manualmente**

# Inclusão

- Essa notação pode ser usada para representar subfluxos complexos e comuns a vários casos de uso.
- O Caso de Uso principal depende do secundário para ser executado.

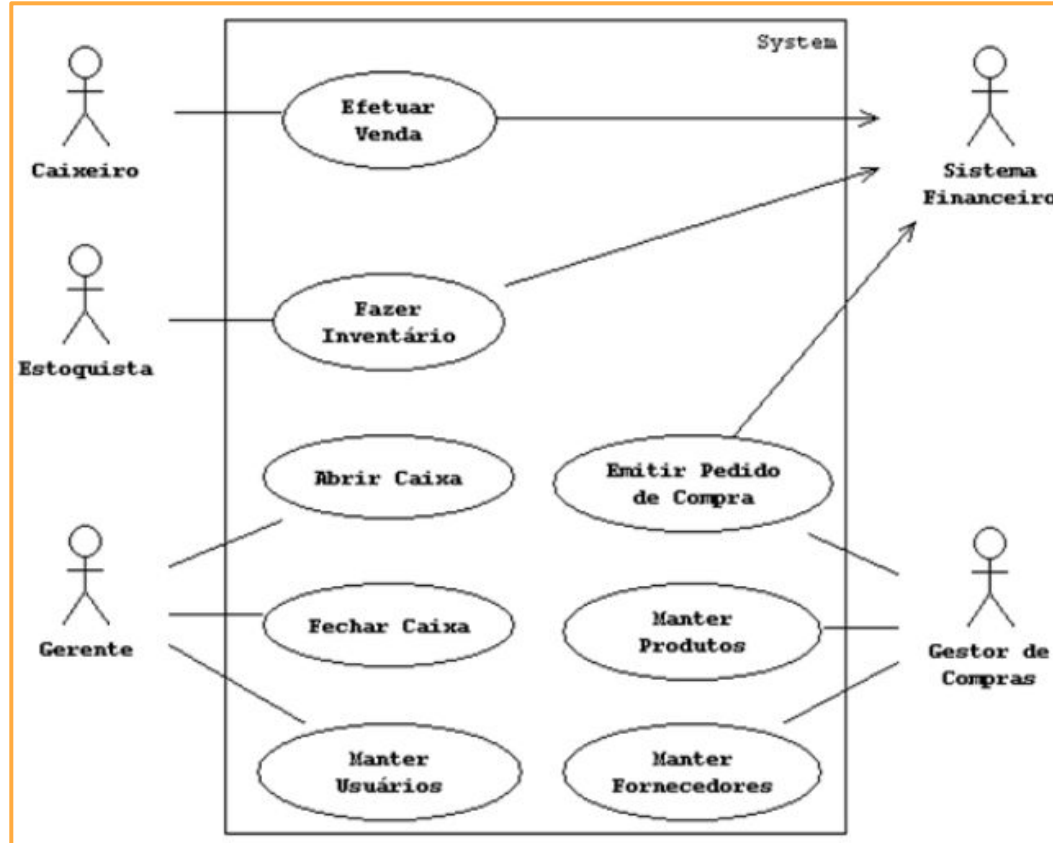




# Roteiro para o Desenho

- Uma vez identificados os atores, as seguintes perguntas auxiliarão na identificação dos Casos de Uso:
  - O que (quais funções) o ator necessita do sistema?
  - O ator necessita criar, modificar, excluir, ler ou armazenar informações no sistema?
  - O trabalho do ator poderia ser simplificado ou mais eficiente através de novas funções do sistema? Quais seriam essas funções?
  - Quais as entradas e saídas, assim como a origem e o destino, que o sistema requer?

# Desenho do Diagrama



## Tarefas para Próxima Aula - 30/07/2020

1. Estudar o capítulo 1 do livro-texto do Projeto de Bloco.
2. Ler o Documento de Visão do Projeto de Bloco. Anotar as dúvidas.
3. Ler as Regras para Construção de TPs e AT.
4. Ler o enunciado do TP1 a ser entregue no dia 03/08/2020.