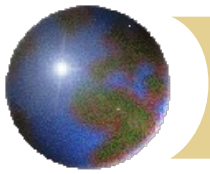




# *Importância da Análise de Sistemas*

Prof. Paulo Quaresma

Escola Secundária Monte de  
Caparica

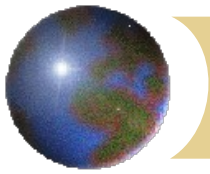


# *Analise de Sistemas*

✦ É o processo de:

- ✦ Analisar,
- ✦ Projetar,
- ✦ Implementar e avaliar sistemas voltados a fornecer informações que sirvam de apoio às operações e aos processos de tomada de decisão numa empresa.

✦ Concentra-se em **todos os elementos** do sistema, não apenas no software.



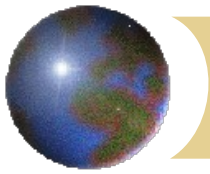
# *Participantes na Análise de Sistemas*

“Em virtude do tamanho e complexidade, o desenvolvimento de sistemas de software é um empreendimento realizado em equipa”

✚ Um analista de sistemas bem sucedido, além de conhecimento sobre as tecnologias, deve ter algumas aptidões interpessoais.

✚ Analistas de Sistemas;

- ❑ Utilizadores;
- ❑ Gerentes;
- ❑ Auditores / Avaliadores da Qualidade
- ❑ Programadores;
- ❑ Pessoal Operativo (Suporte, comunicação, hardware).

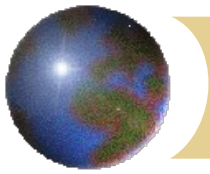


# *Analista de Sistemas*

- ✦ Deve ser capaz de lidar ao mesmo tempo com os utilizadores, programadores e a administração.
- ✦ Além de se preocupar com o desenvolvimento de software, o analista terá de considerar o hardware, as pessoas que o operam, as entradas de dados, a segurança, a auditoria - em resumo, todos os componentes do sistema.

## **✦ Algumas Funções Do Analista De Sistemas:**

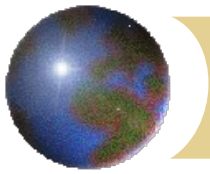
- ✦ 1) Planear, supervisionar e coordenar a análise e o levantamento dos serviços, identificando as suas principais características, estudando a viabilidade económica e técnica das soluções possíveis, propondo conseqüentemente alterações, visando a melhoria do desempenho.



# *Analista de Sistemas*

## ✚ **Algumas Funções do Analista De Sistemas:**

- ✚ 2) Definir e supervisionar a construção e implementação de novos sistemas.
- ✚ 3) Coordenar e orientar as revisões de projectos.
- ✚ 4) Supervisionar directamente a elaboração da documentação.
- ✚ 5) Definir os programas de teste ao sistema, que permitam a sua avaliação.



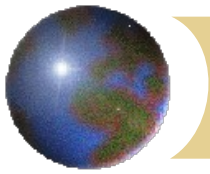
# *Utilizadores:*

✦ É a pessoa para quem o sistema é construído. O **contacto** entre o analista e os utilizadores deve ser constante.

✦ Deve estar informado de todas as fases do desenvolvimento. Deve-se manter **reuniões semanais ou quinzenais** para acompanhamento do projecto.

✦ Os **utilizadores são muito heterogéneos**, e apresentam características distintas. Podemos classificá-los como:

- ✦ **Por tipo de função;**
- ✦ **Por nível de experiência em processamento de dados.**

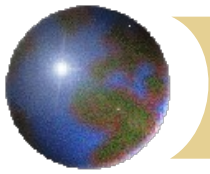


# *Utilizadores:*

## **Tipo de Função:**

### Utilizador operativo.

- ❖ Dentre todos os possíveis tipos de utilizadores, este é aquele que terá o **maior contacto** com o sistema. (ex.: secretarias, pessoal que processa pedidos, caixas, etc).
- ❖ Tem uma **visão local** do sistema. (Apenas as funções relacionadas directamente com a sua função).



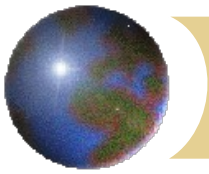
# *Utilizadores:*

## **Tipo de Função:**

### Utilizador supervisor

- ❑ São funcionários que chefiam um conjunto de funcionários operativos. São chefes ou gerentes.
- ❑ São as pessoas que definem o funcionamento do software. Não se preocupam com detalhes. Eles têm uma visão global, tem uma noção de todo o sistema.
- ❑ Normalmente agem como um intermediário entre o utilizador operativo e o analista de sistemas.



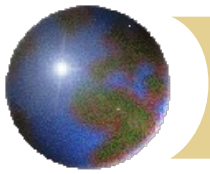


# *Utilizadores:*

## **Tipo de Função:**

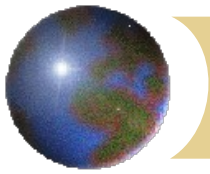
### Utilizador Executivo

- ❏ São as pessoas interessadas no funcionamento geral do sistema.
- ❏ Necessitam de informações refinadas do sistema para os poder auxiliar na tomada de decisões.
- ❏ É o responsável pelas finanças do projecto.



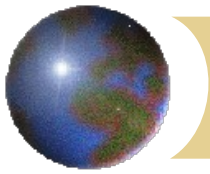
## *Gerentes:*

- ❖ A gerência é um grupo homogêneo de pessoas que pensam do mesmo modo.
- ❖ Gerentes utilizadores: Importam-se com os **relatórios internos e tendências de curto prazo.**
- ❖ Gerente Geral: Estes directores estão normalmente mais interessados nas **informações de planeamento estratégico e de apoio à decisão.**
- ❖ Gerente do projecto: analista de sistemas **responsável pelo projeto.**



# *Auditores / Avaliadores da Qualidade*

- ❑ Controlo de Qualidade.
- ❑ Participam da padronização do projecto.
- ❑ Estão presentes no final de todas a fases de um sistema.
- ❑ Verificam se a equipa de desenvolvimento está a utilizar a Engenharia de software de modo adequado. Não desenvolvem nada, apenas monitorizam o desenvolvimento.



# *Desenvolvimento de Software*

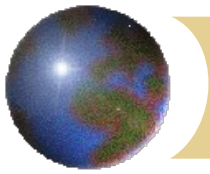
❖ Um conjunto estruturado de actividades necessárias para desenvolver um sistema de software, envolve várias fases:

- ❖ Especificação

- ❖ Projecto

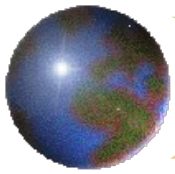
- ❖ Validação

- ❖ Evolução

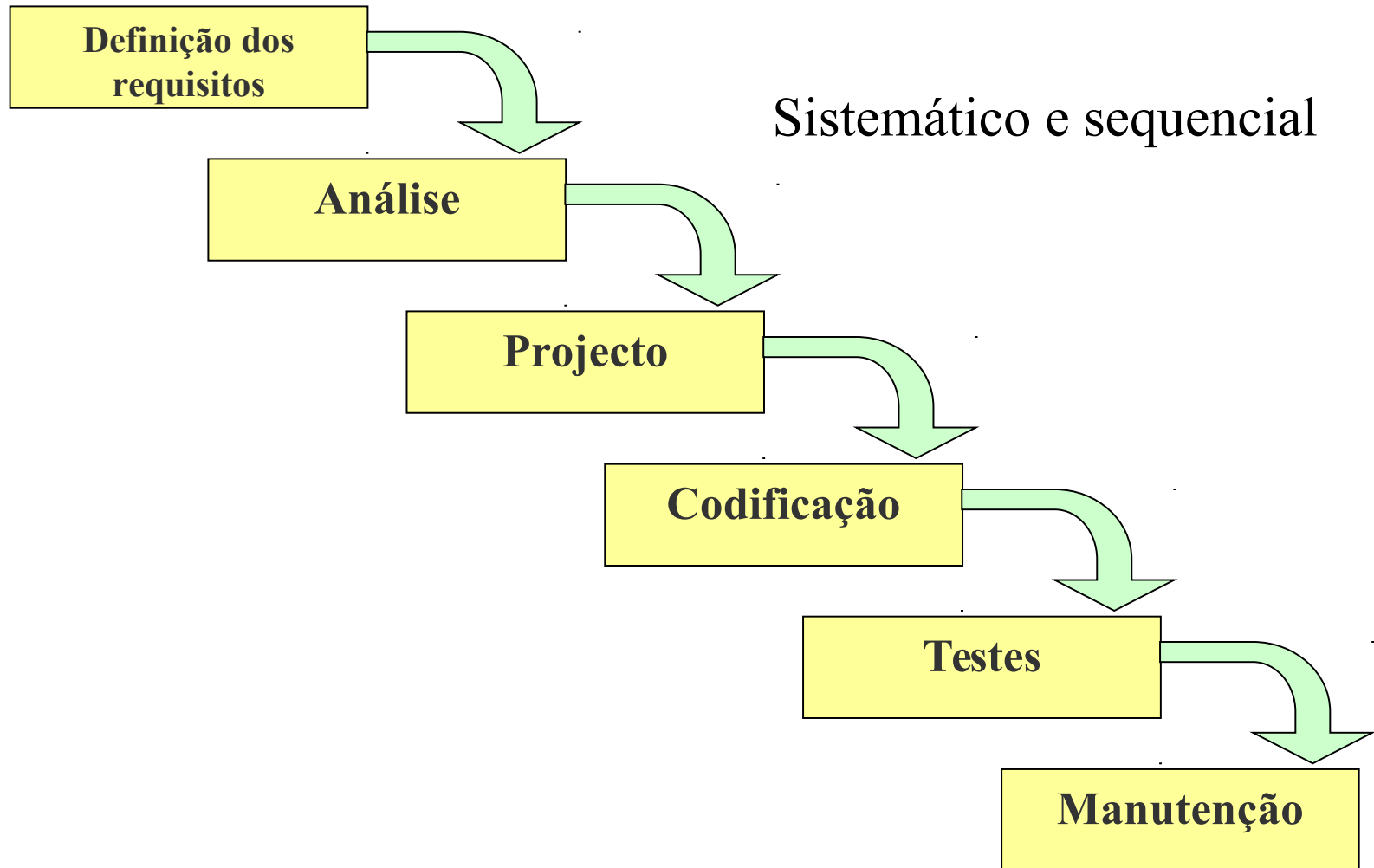


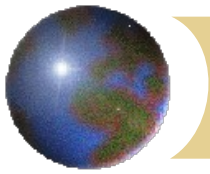
## *Modelo em Cascata*

- ✚ Um dos primeiros modelos (Royce, 1970).
- ✚ O desenvolvimento de um estágio deve terminar antes do próximo se iniciar.
- ✚ Simples, mas não reflecte, efectivamente, o modo como o código é desenvolvido.
- ✚ Derivado do mundo do hardware (linhas de montagem).



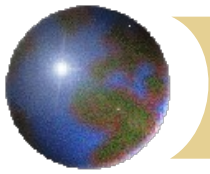
# *Modelo em Cascata*





## *Análise Estruturada (Tradicional)*

- ✦ A análise estruturada é uma abordagem sistemática para fazer a análise de um sistema de modo a produzir a uma especificação funcional.
- ✦ A especificação funcional define as funções e estruturas de dados que constituem o sistema.
- ✦ É necessário que o analista saiba comunicar com os clientes e garantir clareza de ideias.



# *Modelos de Análise*

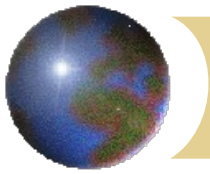
## ✦ Modelo essencial do sistema

- Indica o que o sistema deve fazer e que dados necessita para satisfazer os requisitos do utilizador. Define o sistema num ambiente ideal.

## ✦ Modelo de implementação

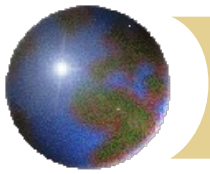
- Apresenta a solução de implementação do sistema ideal, definindo os limites de automatização, as interfaces do utilizador com o sistema e algumas características operacionais relevantes. Define o sistema num ambiente real.





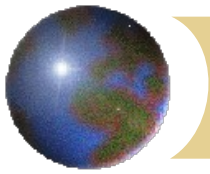
# *Modelo Ambiental*

- ✦ O modelo ambiental descreve o ambiente no qual o sistema se insere, ou seja, descreve o contexto do sistema
- ✦ Componentes do modelo ambiental:
  - ▣ Descrição dos objectivos
  - ▣ Diagrama de Contexto
  - ▣ Lista de Eventos



## *Descrição dos Objectivos*

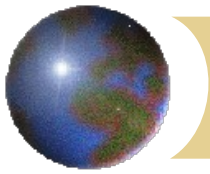
- ✦ É uma descrição textual concisa do propósito do sistema.
- ✦ É voltada para aqueles que não estão envolvidos directamente com o desenvolvimento do sistema
  - ✦ Ex.: Altos escalões
- ✦ É constituída por um ou mais parágrafos
- ✦ Exemplo:
  - ✦ “O propósito da Amazon Bookstore é gerir electronicamente todos os detalhes dos pedidos de livros, envio, cobrança”



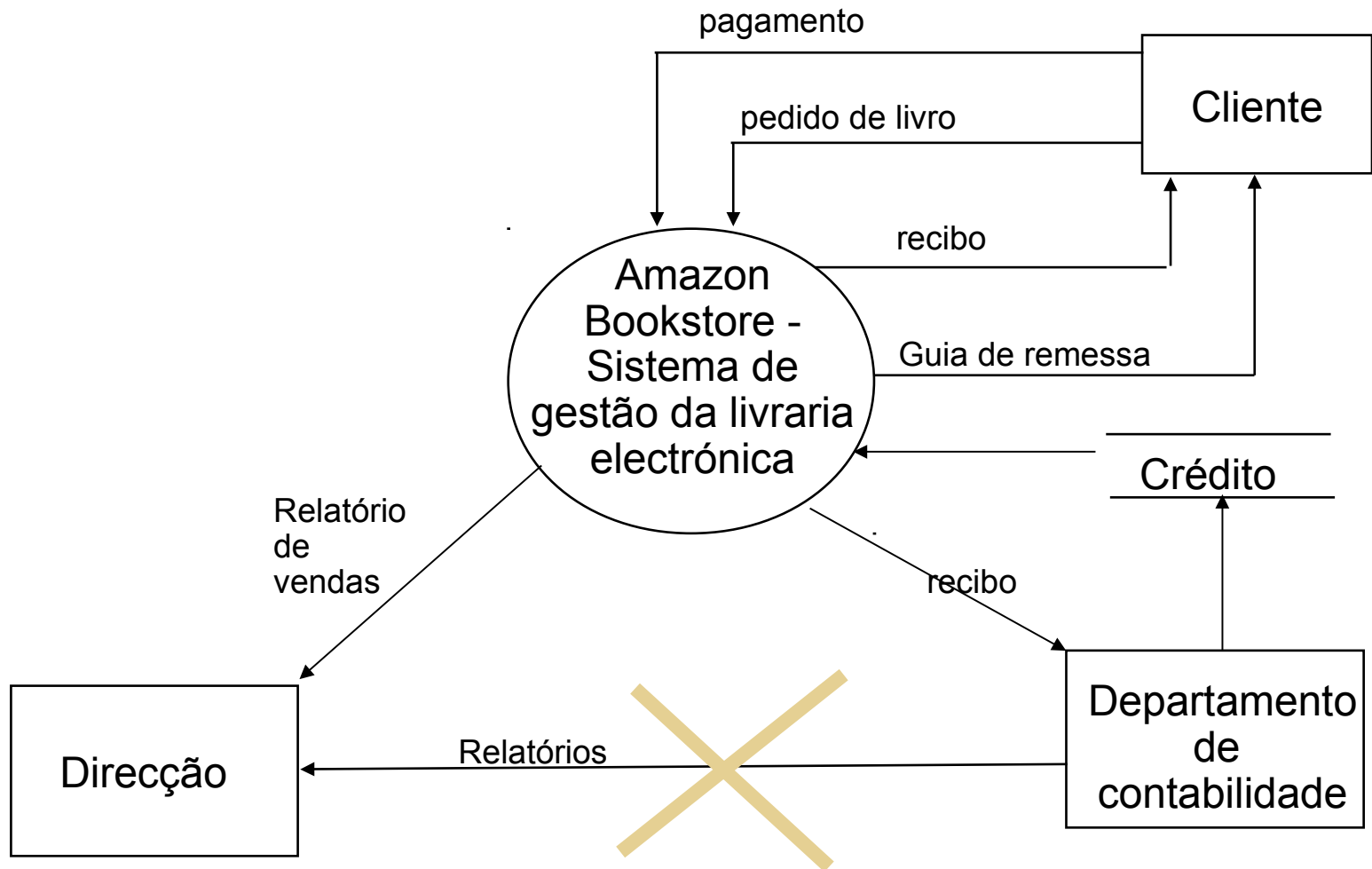
# *Diagrama de Contexto*

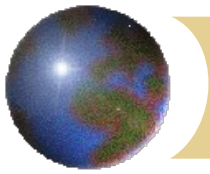
## ❖ Componentes:

- ❑ Entidades externas.
- ❑ Depósitos de dados compartilhados entre o sistema e o ambiente.
- ❑ A interface entre o sistema e o ambiente.
- ❑ Dados que o sistema recebe do ambiente.
- ❑ Dados que o sistema manda para o ambiente.
- ❑ Um processo que represente o sistema



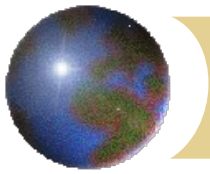
# Diagrama de Contexto (Exemplo)





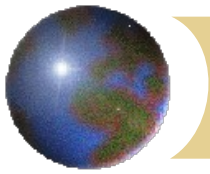
# *Lista de Eventos*

- ✦ Mostra todos os estímulos que ocorrem a a partir do ambiente e aos quais o sistema deve responder.
- ✦ Os eventos classificam-se em:
  - Orientados a fluxo: quando transportam dados;
  - Temporais: quando acontecem periodicamente; é como se o sistema tivesse um relógio interno. Ex.: Recibos devem ser gerados às 16:00 horas
  - Condicionais: quando acontecem devido à verificação de uma condição pelo sistema. Ex: Se o stock for baixo emitir nota de compra ao fornecedor
  - De controlo assíncrono, imprevisíveis: Ex.: interrupções



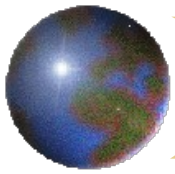
## *Construção de uma lista de eventos:*

- ✦ Identificação dos eventos do ponto de vista do ambiente.
- ✦ Examinar cada entidade externa e verificar os efeitos de suas acções no sistema.
- ✦ Incluir situações de excepção.
- ✦ A lista de eventos pode ser construída antes ou depois do diagrama de contexto, mas é preferível que seja construída antes.



## *Lista de Eventos vs. Diagrama de Contexto*

- ⊕ Eventos por fluxo:
  - ⊞ Originam pelo menos um fluxo de entrada no sistema
  - ⊞ Podem ou não originar fluxos de saída
- ⊕ Eventos temporais e condicionais
  - ⊞ Não originam fluxos de entrada no sistema
  - ⊞ Podem ou não originar fluxos de saída
- ⊕ Eventos de controlo
  - ⊞ Não originam fluxos de dados de entrada no sistema
  - ⊞ Podem ou não originar fluxos de saída



## *Exercício 1 (1)*

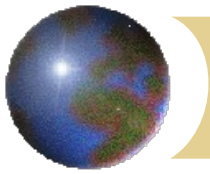
Uma empresa de transportes, cujo serviço é transportar mercadorias no país, pretende automatizar a sua área de gestão de veículos. O director da empresa descreve o sistema da forma que se segue.

Sempre que se compra um veículo regista-se no sistema (as suas características são dadas ao sistema pelo departamento de compras). O veículo só entra em serviço quando chega uma notificação de seguro da companhia seguradora. Esta notificação contém o número de apólice, tipo de seguro com a sua descrição, a identificação da seguradora (código, nome, morada e telefone), e ainda os dados referentes ao veículo. Toda esta informação deve ser inserida no sistema.

Para fazer os trajectos a empresa tem um conjunto de condutores que podem conduzir qualquer veículo da empresa. Os dados dos condutores (nome, morada e telefone) estão guardados num depósito de dados partilhado com o departamento de pessoal.

Sempre que um cliente pede um serviço, preenche uma notificação de serviço onde indica os seus dados pessoais e o





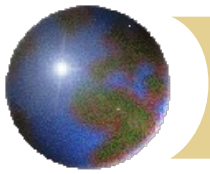
## *Exercício 1 (2)*

Sempre que um condutor termina uma entrega, faz uma notificação de fim de serviço e informa se ocorreu algum acidente. Em caso de acidente, o condutor faz uma descrição do ocorrido.

Só depois do regresso do condutor e caso não haja acidente é que o veículo fica disponível para outro serviço.

Para um melhor atendimento ao cliente, sempre que não se pode realizar um serviço, por falta de veículo ou por falta de condutor, coloca-se o pedido em lista de espera. Logo que seja possível dar resposta a este serviço, envia-se um aviso ao cliente.

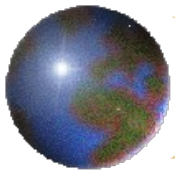
No final do ano, o sistema deve emitir um resumo dos acidentes por veículo onde, para além dos dados do veículo e do seguro, deve constar cada um dos trajectos feitos pelo veículo com o respectivo condutor e os acidentes registados nesse trajecto.



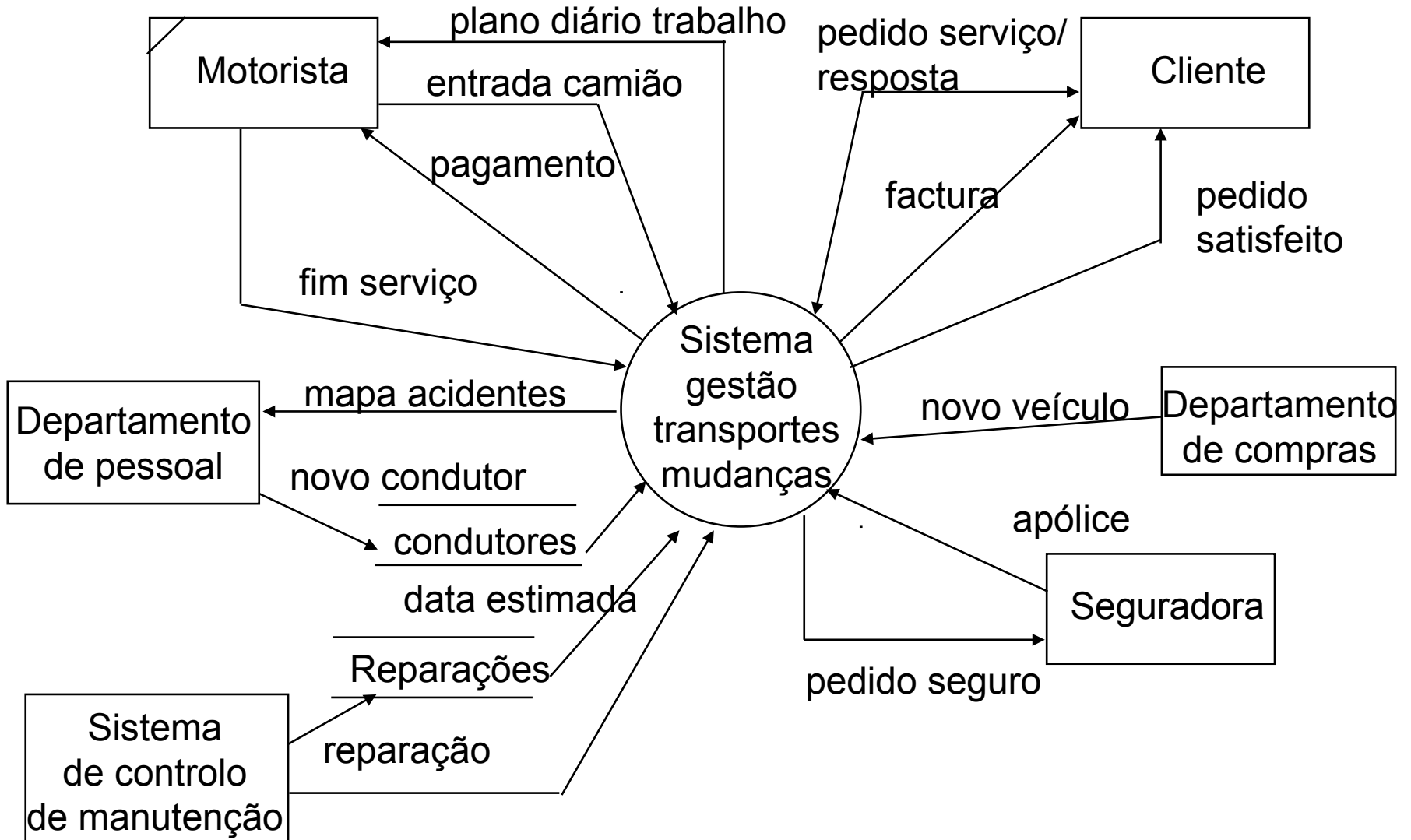
## *Exemplo :*

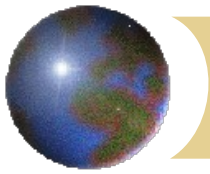
### Lista de eventos:

1. Diariamente emitir lista de serviço (temporal)
2. Departamento de compras regista novo veículo (fluxo)
3. Seguradora informa apólice (fluxo)
4. Cliente pede serviço (fluxo)
5. Condutor informa fim de serviço (fluxo)
6. Condutor dá descrição de acidente (fluxo)
7. ...



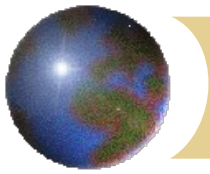
# Exemplo





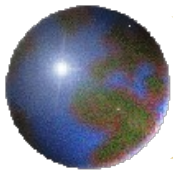
# *Modelo Comportamental*

- ❖ O modelo comportamental descreve as acções que o sistema deve realizar para responder da melhor forma aos eventos definidos no modelo ambiental.
- ❖ Técnicas utilizadas:
  - ❖ Diagrama de Fluxos de Dados (DFD)
  - ❖ Dicionário de Dados (DD)
  - ❖ Diagrama de Entidades e Associações (ou Relacionamentos) (DEA ou DER)
  - ❖ Especificação de Processos (EP) - (DESENHO)
  - ❖ Diagrama de Transição de Estados (DTE)



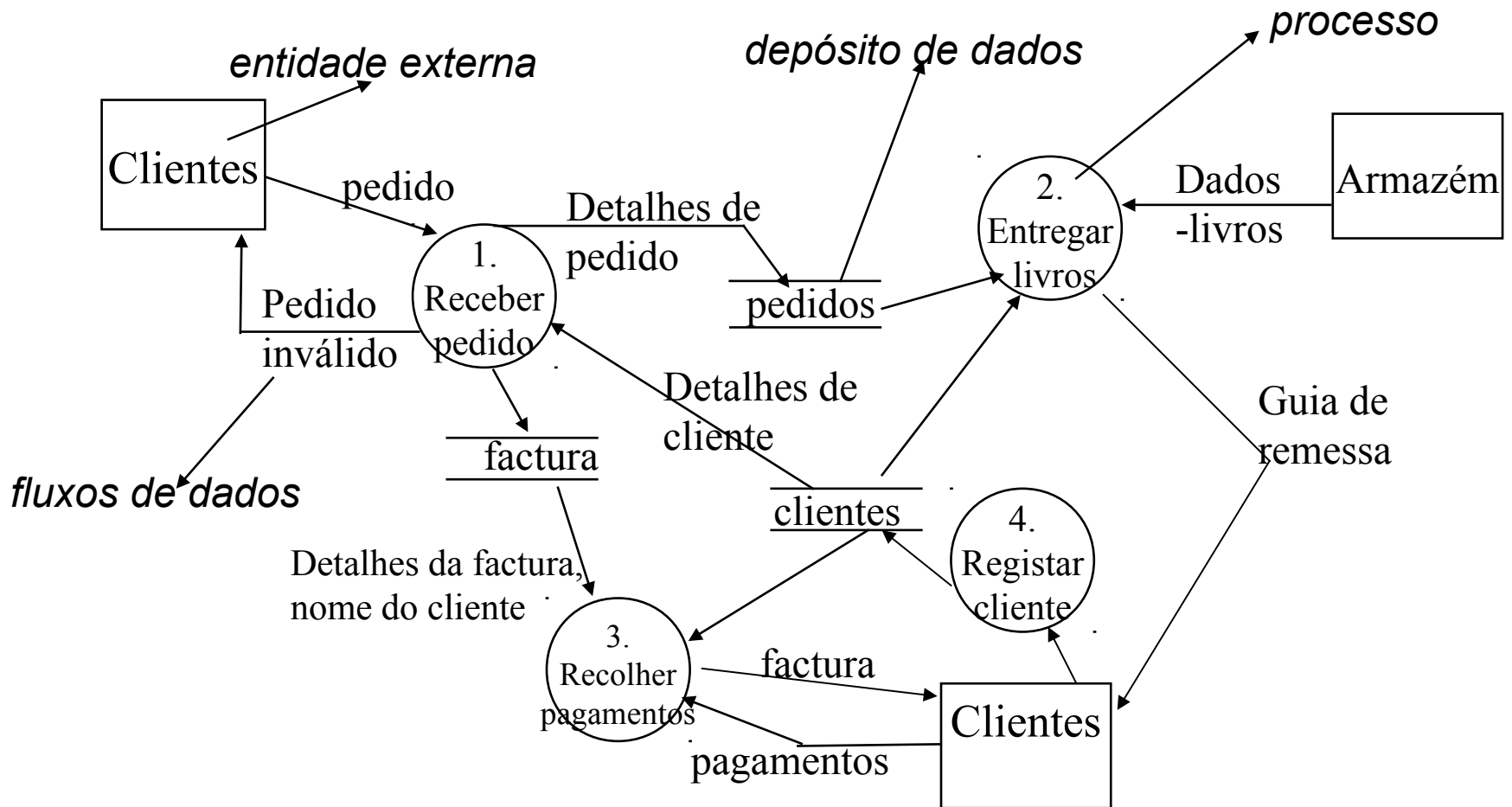
## *Diagrama de fluxo de dados (DFD)*

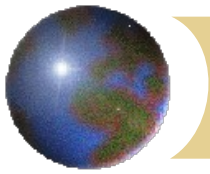
- ✦ Um DFD é um modelo que permite representar o sistema como uma rede de processos, salientando as funções que precisam ser implementadas e os fluxos de dados manipulados por estas funções.
- ✦ Um DFD contém:
  - Processos;
  - Fluxos de dados;
  - Depósitos de dados;
  - Entidades externas com que o sistema interage



# DFD

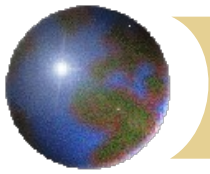
- ❖ O DFD é composto por processos, depósitos de dados, fluxos de dados e entidades externas.





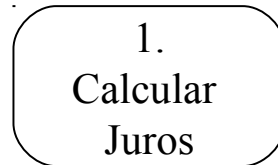
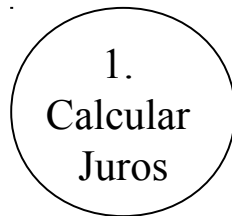
## *Regras para construir DFD's*

- ✦ Escolher nomes significativos para os processo, fluxos, entidades e depósitos de dados.
- ✦ Numerar os processos.
- ✦ Redesenhar os DFD's tantas vezes quantas forem necessárias.
- ✦ Evitar DFD's complexos
- ✦ Certificar-se de que o DFD é internamente consistente
  - ✦ Evitar processos que tenham entradas e nenhuma saída (e vice-versa)

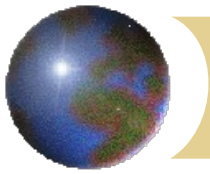


## *DFD: Processo*

- ✦ Função, transformação
- ✦ Transforma entradas em saídas
- ✦ Nome: numeração + verbo + objecto
- ✦ Exemplos:

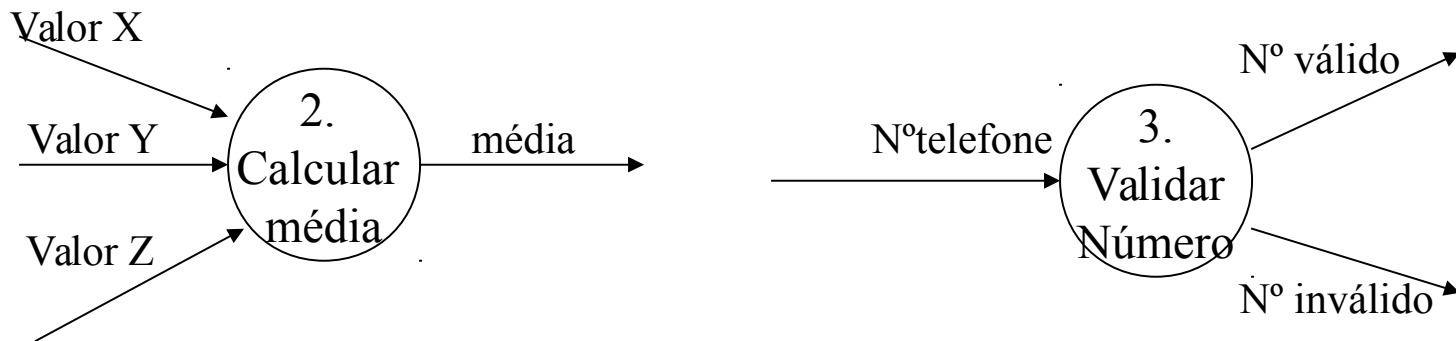


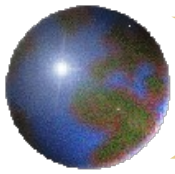




## *DFD: Fluxo (1)*

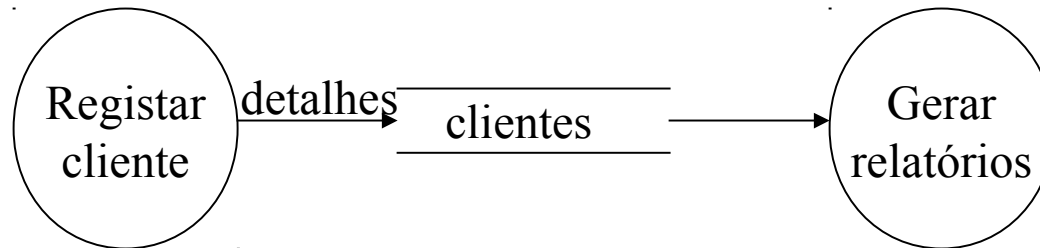
- ✚ Usado para descrever movimento de informação de um componente do sistema para outro
- ✚ Exemplos:





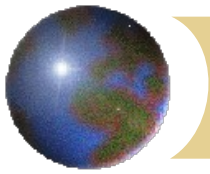
## *DFD: Depósito de Dados*

- Um fluxo para um depósito: actualizar, gravar, remover



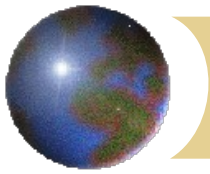
## *DFD: Entidades Externas*

- Estão fora do sistema
- O analista não está em posição de mudar o conteúdo das entidades
- Os fluxos conectando a entidade aos processos representam a interface entre o sistema e o ambiente



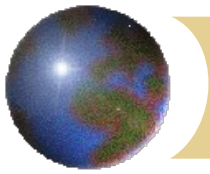
## ***Construção de DFD Preliminar***

1. Um processo -> um evento da lista de eventos.
2. O nome do processo deve estar de acordo com a resposta que o sistema deve dar ao evento.
3. Definir fluxos de dados de entrada e saída. O processo deve ser capaz de produzir a resposta correcta.
4. Usar os depósitos de dados necessários para guardar a informação que será depois usada por outros processos.
5. Verificar a coerência entre este DFD e o diagrama de contexto.



## *Dicionário de dados*

- ✦ Descreve o significado dos fluxos de dados e dos depósitos de dados.
- ✦ Descreve a composição de dados agregados.
  - ✦ Ex.: endereço
- ✦ Especificar os valores e unidades relevantes.
- ✦ Descrever os relacionamentos entre depósitos de dados (diagramas E/R).



## *Notação*

⊕ = é composto por

⊕ + e

⊕ () opcional

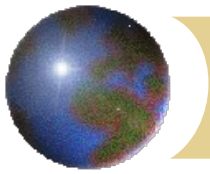
⊕ Lim\_inf{}Lim\_sup iteração

⊕ [] seleccionar uma das várias alternativas

⊕ \*\* comentário

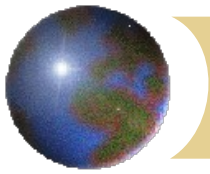
⊕ @ chave de um depósito

⊕ | separa alternativas quando se usa []



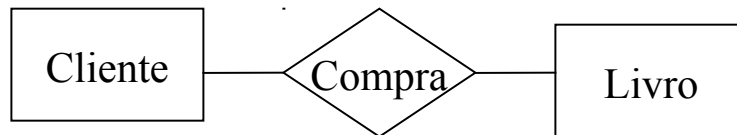
## *Dicionário de dados - Exemplos:*

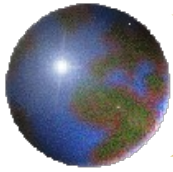
- ⊕ Nome = título + primeiro-nome + apelido
- ⊕ título = [Sr. | Sra.| Prof.| Dr.| Eng.]
- ⊕ primeiro-nome = 1{caracter-válido}
- ⊕ apelido = 1{caracter-válido}
- ⊕ caracter-válido = [A-Z | a-z | ' | - | ]
- ⊕ endereço = \* ainda não definido\*



## *Diagrama de Entidade-Associação (DEA)*

- ❖ Também chamado Diagrama de Entidades-Relacionamentos (DER)
- ❖ Utilizada para:
  - ❑ Interagir com o utilizador.
  - ❑ Definir depósitos de dados.
  - ❑ Focalizar relacionamentos entre depósitos de dados.





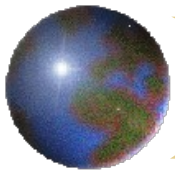
## Componentes de um DEA

- ❑ Tipo Objecto
- ❑ Relacionamento
- ❑ Indicador de tipo objecto associativo

## Tipo objecto

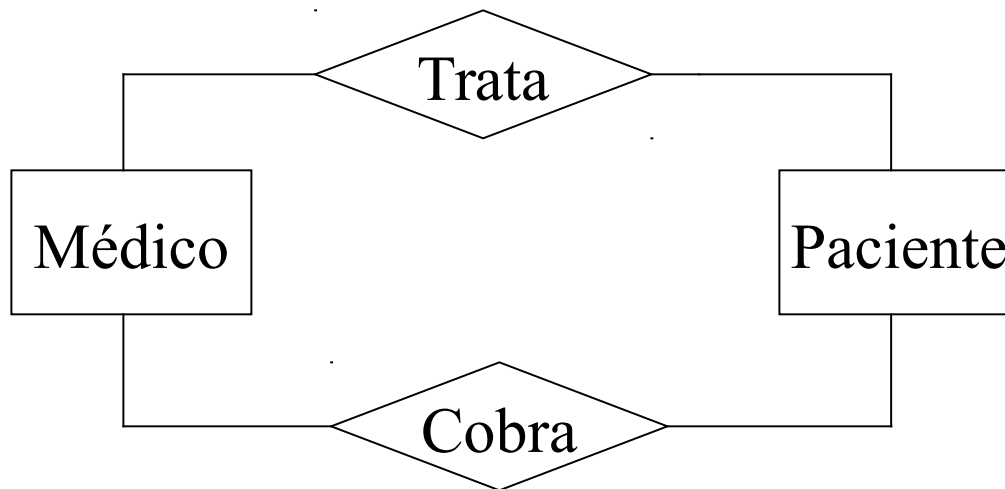
- ❑ Representado por uma caixa rectangular.
- ❑ Representa uma colecção ou conjunto de objectos do mundo real.
- ❑ Objectos são instâncias do tipo.
- ❑ Cada objecto desempenha um papel no sistema e este pode ser:
  - identificado unicamente
  - ser descrito por um ou mais elementos de dados

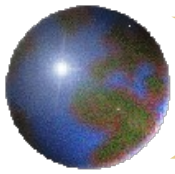




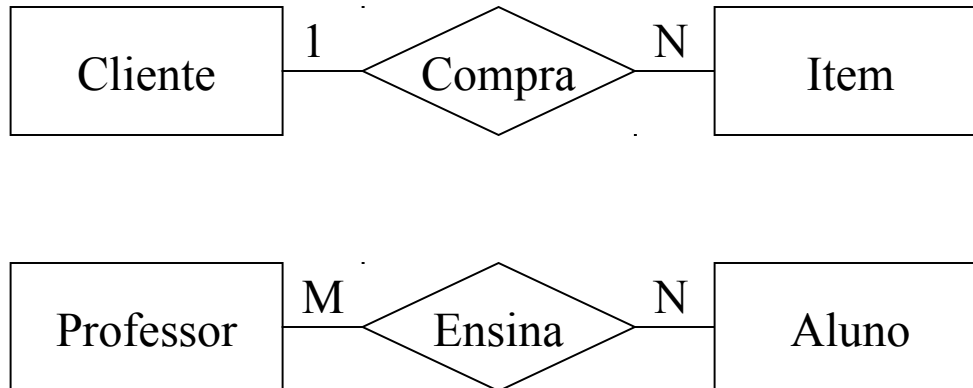
## ❖ Relacionamento

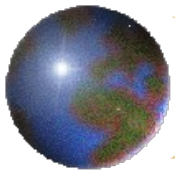
- ❖ Representado por um losango
- ❖ Um relacionamento representa um conjunto de conexões entre objectos
- ❖ Pode haver mais de um relacionamento entre dois objectos





- ✦ Cardinalidade: é o número de objectos de cada tipo objecto que participa no relacionamento
- ✦ Pode ser.
  - ✦ 1:1 ; 1:N ; N:1 ; M:N



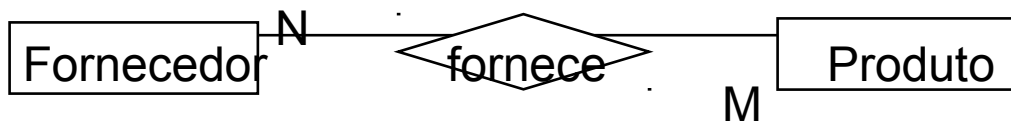


## Exemplo 1:



- Um Aluguer pertence a no máximo um Cliente
- Um Cliente pode ter vários alugueres

## Exemplo 2:



- Um Fornecedor fornece vários Produtos
- Um Produto pode ser fornecido por vários Fornecedores