

---

# Engenharia de Software

## Análise de Requisitos – Parte 1

**Luís Morgado**

Instituto Superior de Engenharia de Lisboa  
Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

---

# Processo de Desenvolvimento

---

- **Análise**

- Conhecer o problema (*domínio do problema*)
  - Identificar, analisar e compreender o problema
- Definir a solução (*domínio da solução*)
  - Identificar e especificar os requisitos da solução

- **Concepção**

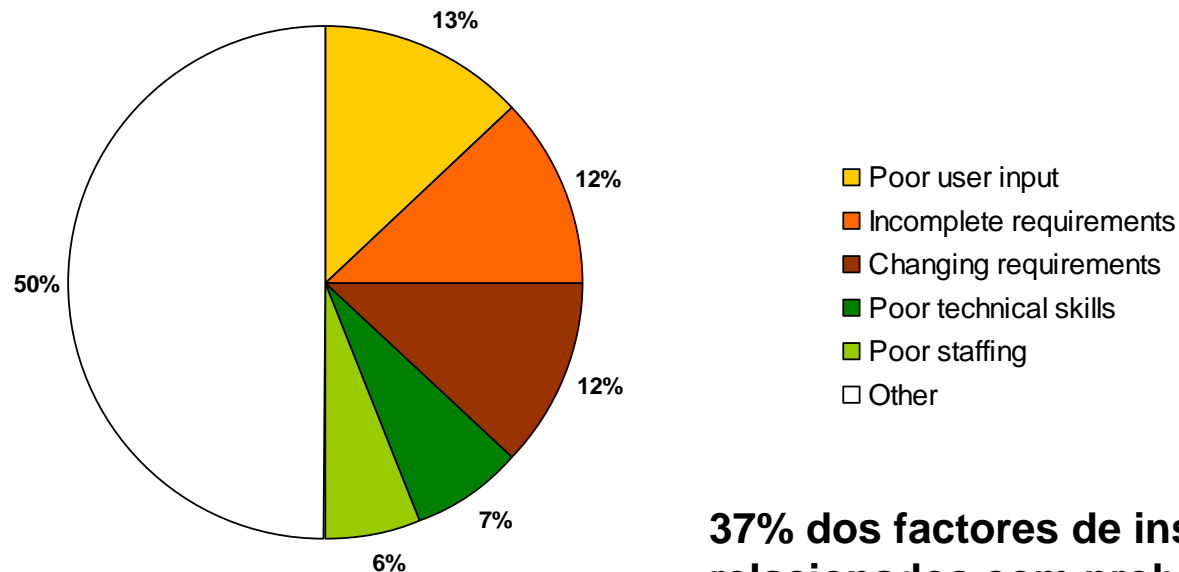
- **Construção**

- **Verificação**

# O Problema dos Requisitos

**“The hardest single part of building a software system is deciding precisely what to build.”**

Frederick Brooks, *"No Silver Bullet: Essence and Accidents of Software Engineering"*



**37% dos factores de insucesso estão relacionados com problemas de definição e gestão de requisitos!**

[Standish Group, 1994]

# Requisitos

---

- **O conceito:**
  - Características ou capacidades do sistema necessárias à concretização dos objectivos para os quais é concebido
- **A prática:**
  - Necessidades, desejos, gostos, preferências, ..., dos utilizadores, dos clientes, ...
  - **Ambiguidade!**

# O Problema dos Requisitos

## Requisitos / Indivíduos



Ambíguos  
Inconsistentes  
Incompletos

Criativos  
Adaptáveis  
Versáteis

**Desejos, necessidades,  
preferências, ...**

## Organização / Sistema

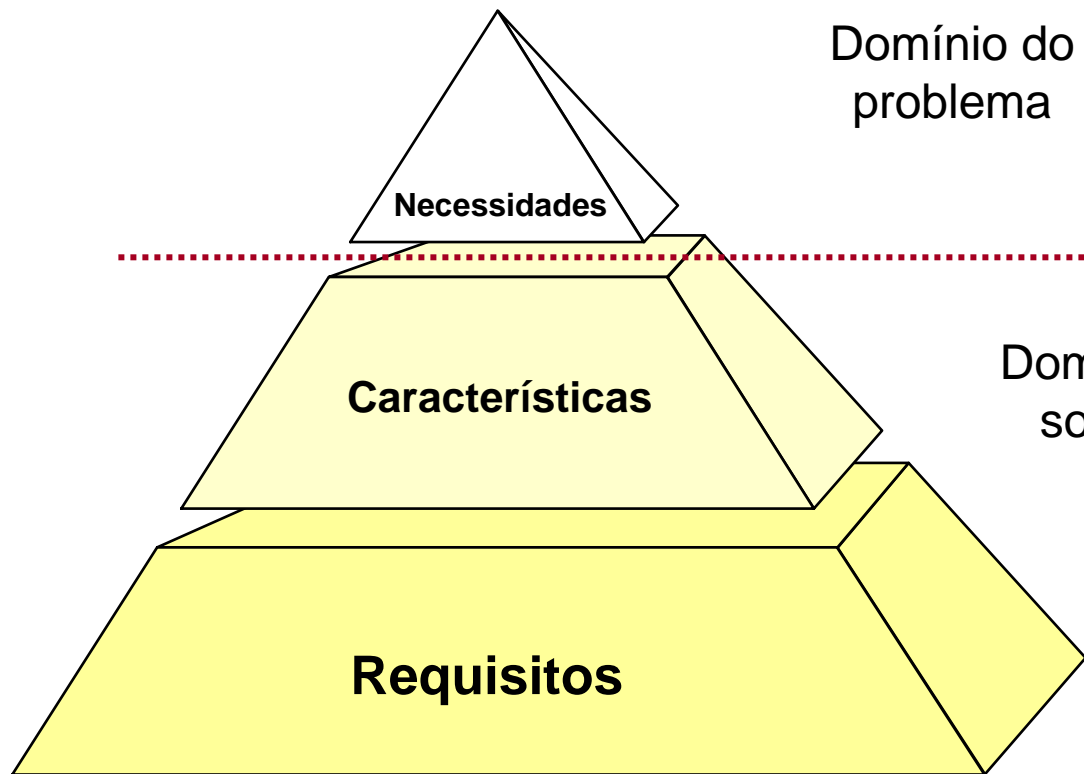


Precisos  
Consistentes  
Completo

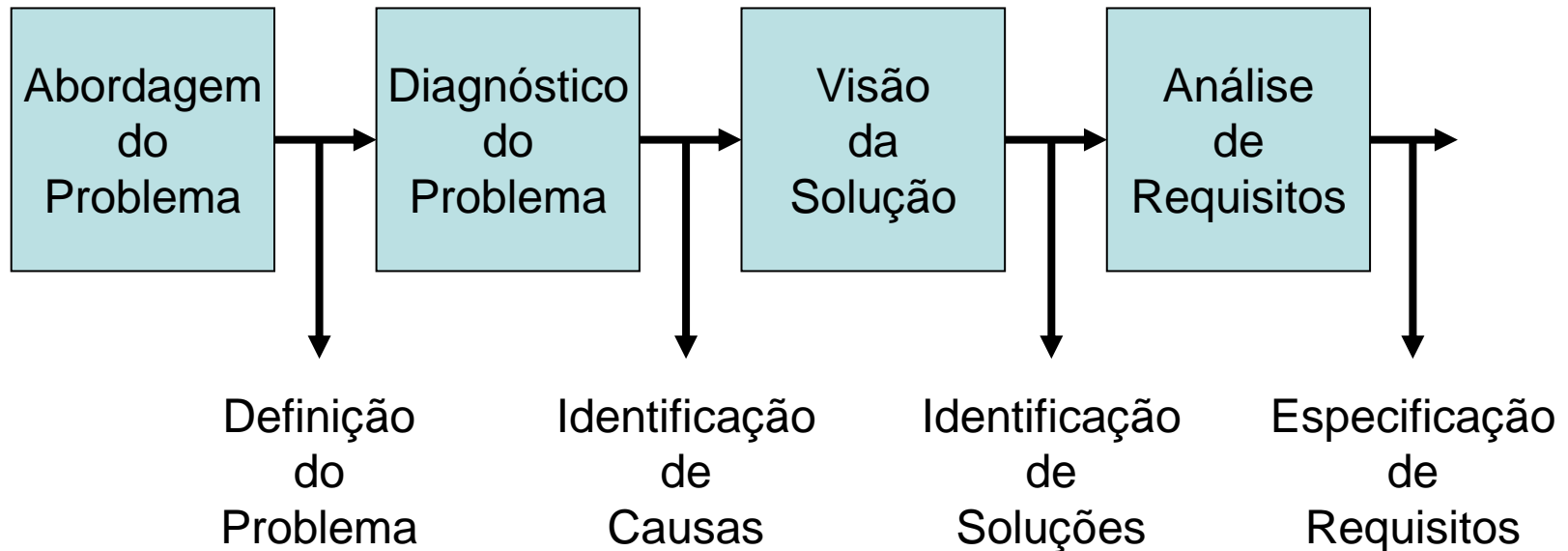
Não criativos  
Não adaptáveis  
Rígidos

**Sistemas  
implementados**

# O Problema dos Requisitos



# O Processo de Análise



# Desenvolvimento de Software

---

- **Análise do problema**
  - Estudo do contexto alvo e de eventuais sistemas existentes para obter informação mais detalhada acerca dos problemas que estão na origem do projecto
  - Normalmente são encontrados novos problemas
    - Questão importante:
      - Os benefícios de um sistema para resolver estes problemas excedem os custos desenvolvimento do sistema?
  - Participantes
    - Proprietários do sistema
    - Gestores de projecto
    - Analistas de sistemas
    - **Utilizadores** ← **Princípio 1: Envolver os utilizadores**



# Análise do Problema

---

- **Definição de âmbito:**
  - Qual o ambiente em que se enquadra o problema?
  - Quem está envolvido?
  - Que meios existem ou foram tentados para abordar o problema?

# Quem está Envolvido?

---

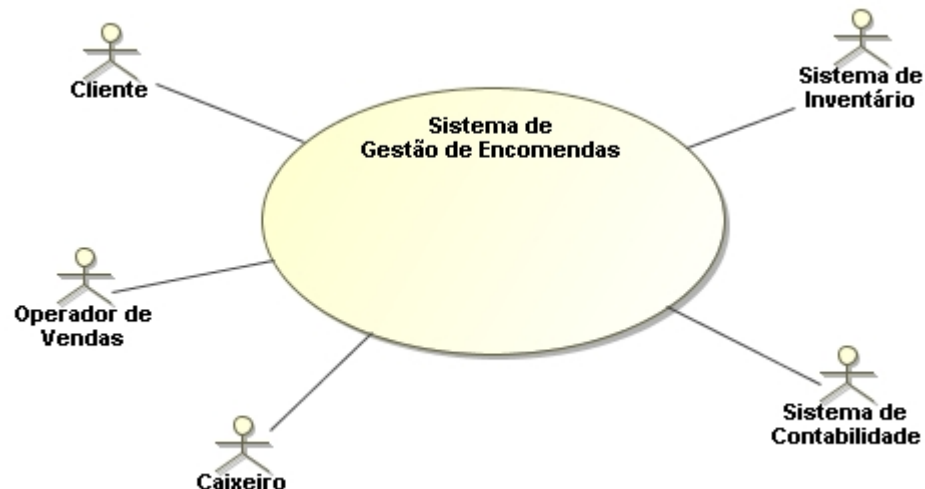
- Quem irá fornecer, utilizar ou remover informação do sistema?
- Quem vai operar o sistema?
- Quem vai manter o sistema?
- Que outras entidades interagem com o sistema?

↓  
**Actor**

# Definição do Âmbito da Solução

- O que **faz parte** da solução
- O que **não faz parte** da solução
- Entidades que interagem com o sistema
  - Actores
    - Algo ou alguém exterior ao sistema que interage com o sistema

## Diagrama de contexto



# Documento de Visão

---

- Descreve o sistema em termos gerais
  - Contexto operacional e organizacional
  - Partes envolvidas (cliente, utilizadores, ...)
  - Características do sistema
- Define de forma geral o problema e a solução

# Manter um glossário do projecto

---

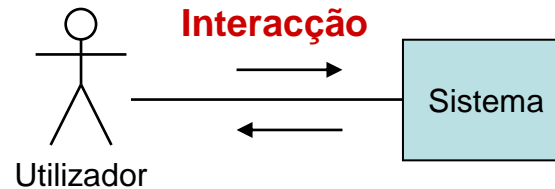
- **Uma das maiores de dificuldade no desenvolvimento de software são as falhas de comunicação!**
  - Em particular a comunicação cliente-fornecedor (quem utiliza - quem desenvolve);
  - Cada interlocutor utiliza vocabulário próprio.
- **Uma das práticas essenciais é a definição precisa dos termos utilizados:**
  - Definir uma linguagem comum! → **Glossário**

# Análise de Requisitos

**Visão** .....→ **Requisitos** .....→ Arquitectura .....→ Sistema

- Gerais
- Detalhados → Especificação das características e funcionalidades que o sistema deve apresentar

**Perspectiva externa**



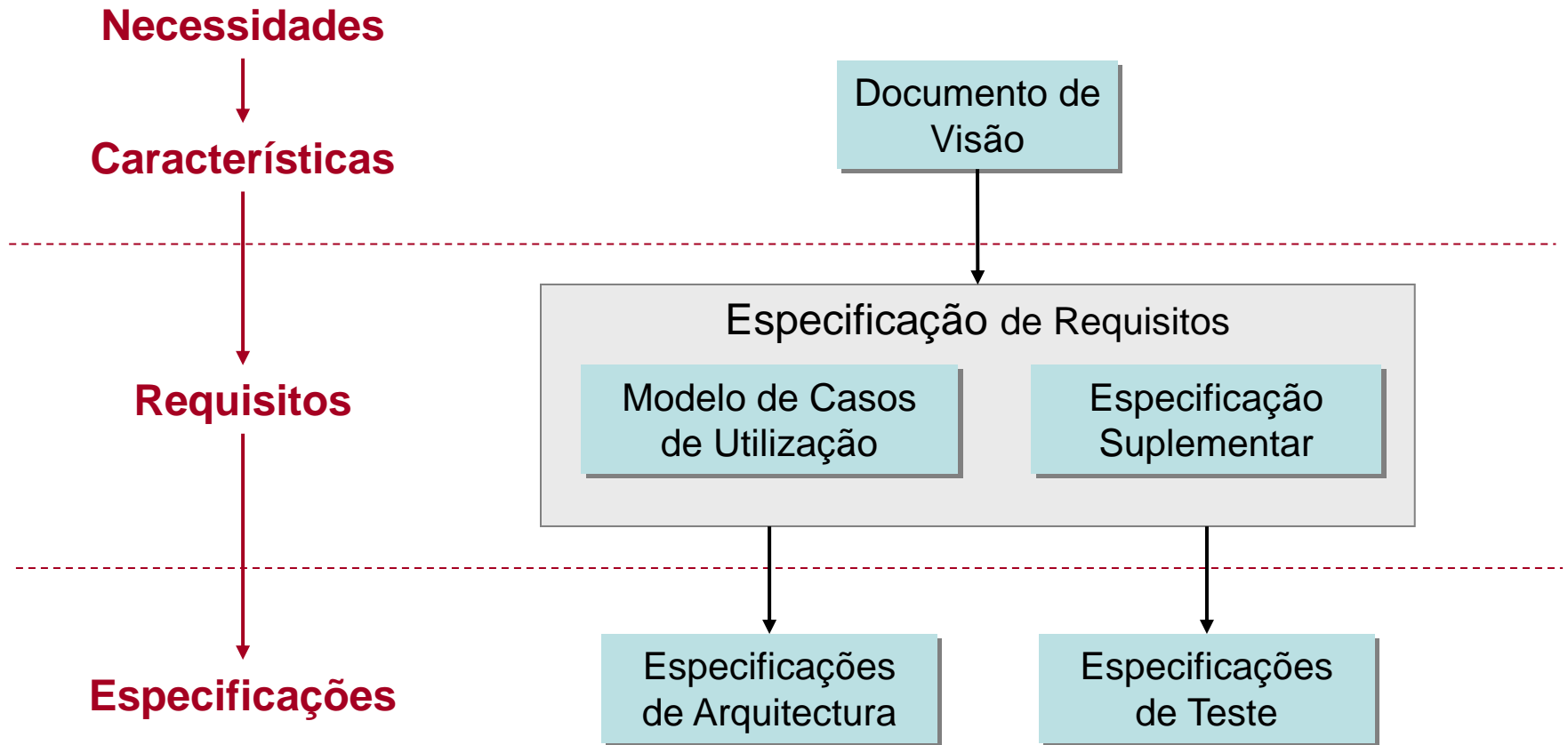
**Casos de Utilização**

- Cenários de utilização do sistema
- Identificação e caracterização do sistema numa perspectiva externa

**Suporte UML:**

- Diagramas de Casos de Utilização

# Análise de Requisitos



# Análise de Requisitos

---

- Identificação de requisitos
- Especificação de requisitos
- Verificação de requisitos
- Gestão de requisitos



# Identificação de requisitos

---

- Técnicas de identificação de requisitos
  - Entrevistas
  - Questionários
  - Sessões de identificação de requisitos (*requirement workshops*)
  - Histórias de utilização
  - Casos de utilização
  - Simulação de papéis de utilização (*role playing*)
  - Protótipos

# Identificação de Requisitos

---

- **Nível do domínio** (*Business level*):
  - Nível mais geral e abrangente
  - Compreensão das metas e objectivos do projecto
  - Definição de uma visão do problema e da solução
- **Nível do utilizador** (*User level*):
  - Requisitos de utilizadores
  - Casos de utilização
- **Nível técnico** (*Technical level*):
  - Requisitos funcionais e não funcionais
  - Especificação detalhada

# Tipos de Requisitos

---

- **Funcionais:**

- Descrevem a interacção entre o sistema e o seu ambiente;
- Descrevem como o sistema se deve comportar dado um determinado estímulo.

- **Não funcionais:**

- Descrevem restrições a que o sistema deve obedecer.

# Funcionalidades e Atributos

- **Funcionalidades de um sistema**

- Representam o que o sistema deve fazer
- Para verificar que determinada característica é de facto uma função do sistema, essa característica deve fazer sentido na seguinte frase:

***“O sistema tem que fazer...”***

- "O sistema tem que fazer *actualização de existências após empréstimo*"
- "O sistema tem que fazer *registo de reserva de títulos não disponíveis*"
- "O sistema tem que fazer *notificações por e-mail*"

- **Atributos de um sistema**

- Representam características não-funcionais do sistema
- Essas características não fazem sentido na frase anterior:
  - "O sistema tem que fazer *simples de utilizar*" !
  - "O sistema tem que fazer *independente do sistema operativo*" !

# Atributos

---

- **Os atributos do sistema são características ou dimensões do sistema:**
  - Não são funções!
- **Exemplos de tipos de atributos de um sistema:**
  - Facilidade de utilização
  - Interface de utilização
  - Plataformas
  - Tolerância a falhas
  - Tempo de resposta

# Atributos:

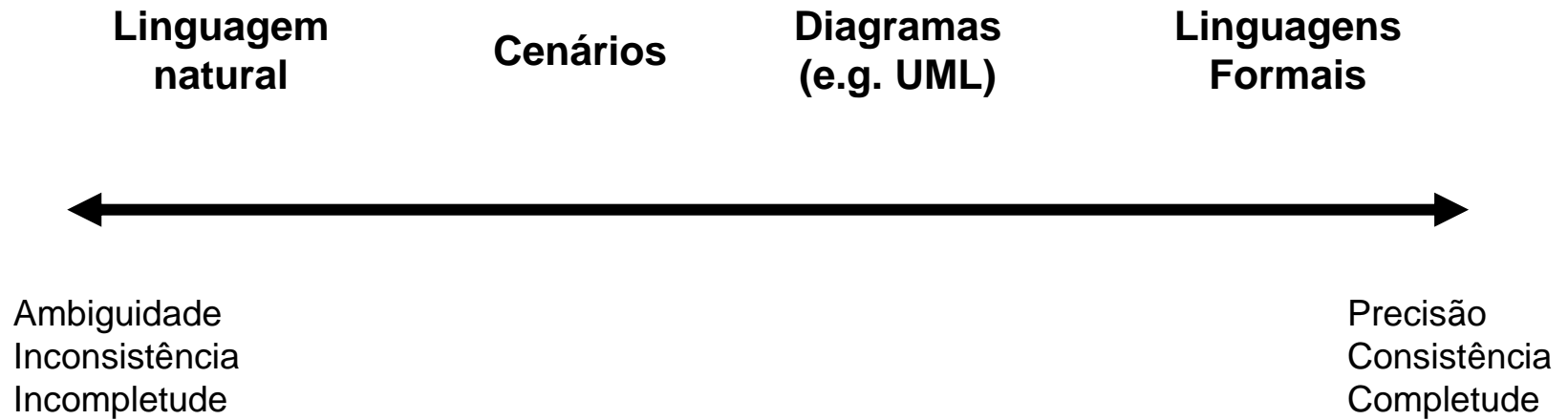
## Domínios de Variação

---

- Podem variar num conjunto de ***valores de detalhe***
  - Normalmente valores discretos e simbólicos
  - Exemplo:
    - Modo de operação  $\in \{ \text{hora, data, cronómetro} \}$
- Podem ter ***restrições de fronteira***:
  - Normalmente um intervalo de valores numéricos
  - Exemplo:
    - Tempo de resposta  $< 5$  segundos

# Especificação de Requisitos

## Tipo de especificação



# Especificação de Requisitos

## Exemplos

---

### **Requisitos expressos em linguagem natural:**

“O sistema deve permitir a emissão de relatórios acerca dos resultados do último trimestre. No que se refere ao acesso ao sistema, este deve limitar o acesso dos utilizadores às funções do sistema consoante o seu nível de acesso. Para além disso, todos os dados do sistema devem ser armazenados com base no sistema de gestão de base de dados já existente.”



# **Especificação de Requisitos**

## **Exemplos**

---

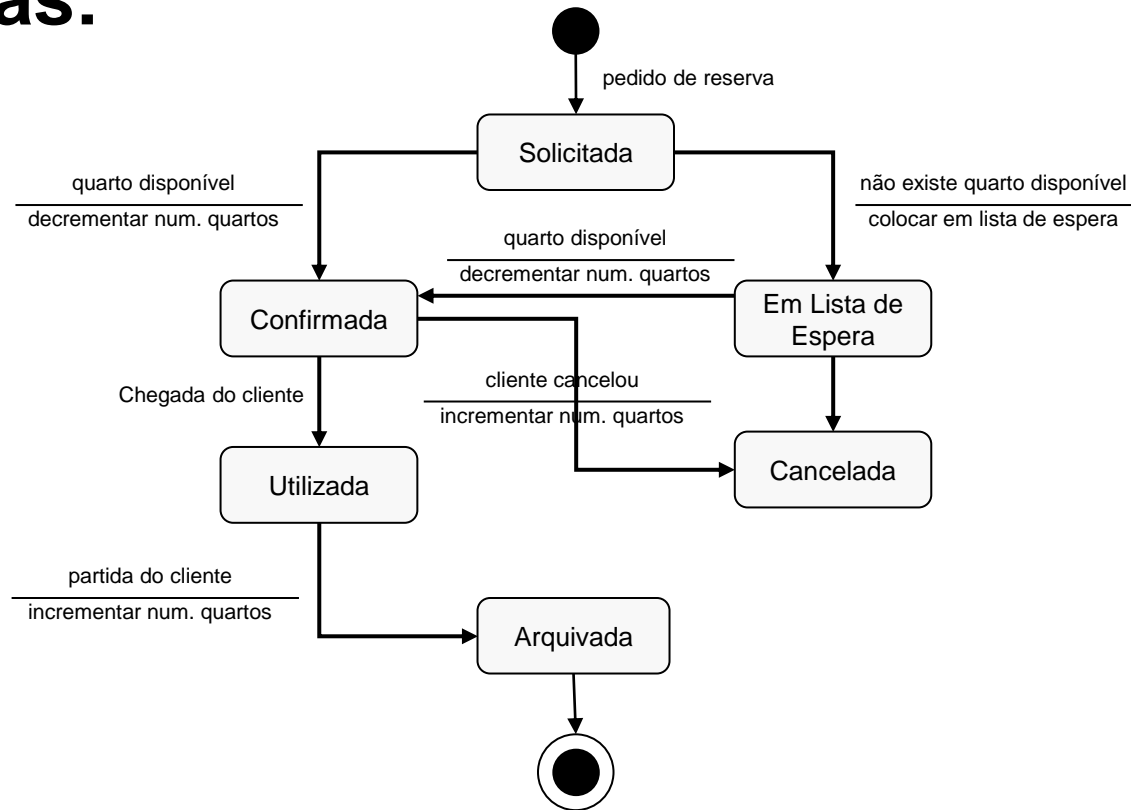
### **Requisitos expressos em linguagem natural, mas estruturados:**

- R1: O sistema deve permitir a emissão de relatórios acerca dos resultados do último trimestre.
- R2: O sistema deve limitar o acesso dos utilizadores às funções do sistema consoante o seu nível de acesso.
- R3: Todos os dados do sistema devem ser armazenados com base no sistema de gestão de base de dados já existente.

# Especificação de Requisitos

## Exemplos

### Diagramas:



Reservas de Hotel

# Especificação de Requisitos

## Exemplos

---

### **Histórias de utilização (Cenários):**

“O cliente chega à caixa com vários produtos para comprar. O funcionário utiliza o sistema de gestão de vendas para registar cada produto vendido. O sistema apresenta o valor de cada produto e o total num ecrã visível pelo cliente. O cliente realiza o pagamento, o qual é registado no sistema. O sistema actualiza o registo de mercadorias. O cliente recebe um recibo emitido pelo sistema e parte com os produtos adquiridos.”

# Identificação de Requisitos

- Os requisitos estão inerentemente ligados a **objectivos**:
    - Dos clientes;
    - Dos utilizadores.
- } Quem afecta (**Actores**)



- **Funcionalidades do sistema**
- **Atributos do sistema**

# Definição de Objectivos

## Exemplo

---

- **Metas globais:**

“Neste projecto serão automatizados os processo de empréstimo e reserva de livros e revistas. O sistema desenvolvido será utilizado numa biblioteca.”

- **Objectivos genéricos:**

“Pretende-se que este sistema contribua para:

- responder "assim que possível" a quem solicita títulos não disponíveis;
- obter informação que suporte as decisões de aquisição de cópias;
- garantir aos utilizadores respostas imediatas e correctas.”

- **Objectivos concretos:** Detalhe iterativo.

# Especificação de Requisitos

---

- Principal objectivo:
  - **Estabelecer** um acordo entre a equipa de desenvolvimento e o cliente acerca das **características que o sistema deve apresentar**
- Serve de **base para estimativas de esforço** de desenvolvimento do sistema
- Serve de **base ao planeamento** do projecto
- Serve de **base à concepção, verificação e teste** do sistema
- Uma especificação de requisitos descreve **o que o sistema deve fazer** e **não como fazer**

# Gestão de Requisitos

---

- Os requisitos devem ser organizados de acordo com critérios relevantes para o desenvolvimento do projecto
  - Prioridade
    - Crítico, importante, útil
  - Esforço
    - Alto, médio, baixo
  - Risco
    - Alto, médio, baixo
  - Estabilidade
    - Fechado, aberto

# Medidas de Qualidade de Requisitos

---

- Uma especificação de requisitos deve ser:
  - Correcta
  - Não ambígua
  - Completa
  - Consistente
  - Verificável
  - Modificável
  - Rastreável
  - Compreensível



# Desenvolvimento de Software

---

- Análise de requisitos
  - **Uma das fases mais importantes** no desenvolvimento de um sistema
  - Identificar, especificar e atribuir prioridades aos requisitos de “negócio”
    - **O que se pretende** que o sistema faça
    - **Que propriedades** se pretende que o sistema tenha
  - Definir o que o sistema deve fazer (**não como fazer!**)
  - **Erros e omissões** na análise de requisitos resultará na **insatisfação** dos utilizadores finais
  - Os **objectivos** da solução, identificados na análise do problema, servem de **guia** na identificação de requisitos

# Desenvolvimento de Software

---

- Análise de requisitos (cont.)
  - Participantes
    - Gestores de projecto
    - Analistas de sistemas
    - Utilizadores
  - O **envolvimento** próximo dos **utilizadores** é muito importante nesta fase do projecto no sentido de identificar **necessidades** e **prioridades**
  - Informação recolhida de **diferentes formas**
    - Reuniões, entrevistas, questionários, análise de relatórios de problemas, ...
  - Descrição essencialmente textual, complementada com esboços gráficos (interfaces de utilização, relatórios, ...)

# Bibliografia

---

[Brooks, 1975]

Frederick Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, 1975.

[Leffingwell & Widrig , 1999]

Dean Leffingwell, Don Widrig, *Managing Software Requirements*, Addison Wesley, 1999.

[Pressman, 2003]

Roger Pressman, *Software Engineering: a Practitioner's Approach*, McGraw-Hill, 2003.