

# ***Sistemas de Tempo-Real***

## **Aula 1**

# **Restrições temporais: origem e caracterização**

**Conceitos básicos de *tempo-real*  
Requisitos dos Sistemas de Tempo-Real**

Adaptado dos slides desenvolvidos pelo Prof. Doutor Luís Almeida para a  
disciplina “Sistemas de Tempo-Real”

# *Definições relacionadas com “Tempo-Real”*

Existe uma grande diversidade de definições relacionadas com **Tempo-Real**, os **sistemas** que lidam com Tempo-Real, os **serviços** que prestam e as **funcionalidades** que desempenham.

Em comum têm o facto de exprimirem a **dependência de um sistema** computacional face ao **tempo** tal como existe num determinado **processo físico**.

# ***Funcionalidades e Serviços de Tempo-Real***

## **Funcionalidade ou Serviço de Tempo-Real**

Que tem de ser desempenhada ou prestado dentro de **intervalos de tempo finitos impostos por um processo físico**

## **Sistema de Tempo-Real**

Aquele que desempenha pelo menos uma funcionalidade de tempo-real ou que presta pelo menos um serviço de tempo-real

(PDC, 1990)

# *Ciência de Tempo-Real*

## Ciência de Tempo-Real

Ramo da **Ciência da Computação** que estuda a introdução de **Tempo-Real** nos sistemas computacionais.

## Tempo-Real (propriedade de um sistema computacional)

Propriedade que caracteriza a capacidade de um sistema computacional de **estabelecer correspondências** entre diferentes **sistemas de medição e/ou contagem de tempo**.

(Motus and Rodd, 1994)

# Computação de Tempo-Real

Os **resultados das computações** devem ser

- Logicamente correctos
- Produzidos a tempo

(Stankovic, 1988)

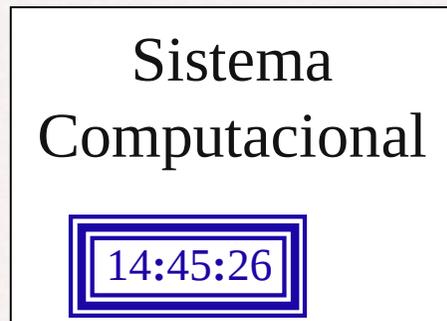
**Pontualidade**  
(timeliness)



**Correcção  
lógica**

# Sistema de Tempo-Real

## Sistema Controlador



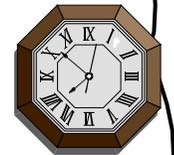
Operador

## Sistema Controlado

Actuadores

Sensores

Meio envolvente  
(processo físico)



$t^n_{out/phy}$

$t^n_{in/phy}$

Restrições temporais impostas pelo meio envolvente

$$\forall n, t^n_{out/phy} - t^n_{in/phy} < \delta$$

$$\forall n, t^{n+1}_{in/phy} - t^n_{in/phy} < T$$

# Noção de “Tempo-Real”

O **meio envolvente** com o qual o sistema computacional interage (**processo físico**) possui o seu próprio **ritmo de evolução**, i.e., a sua **própria dinâmica**.

Esse **ritmo** é **inerente** ao próprio processo físico e não pode (ou não deve, caso dos simuladores) ser controlado externamente. Designa-se por **Tempo-Real**.

# Noção de “Tempo-Real”

Assim, o meio envolvente **impõe** ao sistema **requisitos temporais** de acordo com o seu *tempo-real*, i.e., a sua dinâmica.

Para que o sistema computadorizado seja capaz de **interagir** com o seu meio envolvente, tem de actuar sobre ele **a tempo**, i.e., de acordo com o respectivo *tempo-real*.

# Noção de “Tempo-Real”

Notar que *tempo-real* não significa rapidez mas apenas um ritmo de evolução próprio de um certo processo físico (**termo relativo**)

Notar igualmente o **antagonismo** face a situações em que o **ritmo de evolução pode ser controlado** quer pelo operador quer pelo próprio sistema de controlo (e.g., o sistema de reservas de viagens de avião, sistemas bancários de controlo de contas). Nestes casos, perante um excesso de solicitações, o sistema atrasa a resposta a cada uma continuando o respectivo processamento ao ritmo que for possível (**best effort**). Leva à formação de filas de acesso...



# Noção de “Tempo-Real”

- Exemplo - **piloto de F1**  
(mas aplica-se a qualquer condutor ou a um robô...)



- O **controlo** da direcção tem de ser **preciso**, qualquer que seja a **velocidade** a que o carro circula
- todos os **eventos inesperados** (e.g. carros acidentados, pessoas na pista, poça de água, furo numa roda) que surjam enquanto o **carro circula a alta velocidade** têm que ser **tratados a essa velocidade**

A **velocidade do carro** determina o **tempo-real**

**Não é possível parar instantaneamente para pensar !!...**

# Noção de “Tempo-Real”

- Genericamente, quando um sistema de controlo ou monitorização consegue **acompanhar o estado** de um dado processo físico e, se necessário, **actuar a tempo sobre ele**, então trata-se de um **sistema de tempo-real**.
- Todos os seres vivos são *sistemas de tempo-real* relativamente aos seus habitats naturais, os quais determinam o respectivo tempo-real.
- Por outro lado, quando construímos **máquinas** (programáveis) para **interagir com processos físicos**, necessitamos de usar **técnicas de programação** e **infraestruturas de SW** que nos permitam ter **confiança na capacidade de actuação pontual**.

# Objectivo do estudo dos STR

- O principal **objectivo** do estudo dos Sistemas de Tempo-Real (STR) é o de **desenvolver técnicas de**
  - **projecto,**
  - **análise e**
  - **verificação**

que permitam **obter garantias** de que um dado sistema, que se pretende de tempo-real, tenha **comportamento temporal adequado** à **dinâmica** do sistema com o qual deve interagir.

# *Objectivo do estudo dos STR*

Relativamente às **actividades computacionais** dos STR os aspectos que normalmente mais interessa **caracterizar** são:

- O tempo de execução
- O tempo de resposta
- A regularidade de eventos periódicos

# Objectivo do estudo dos STR

Alguns **aspectos** particularmente importantes relativamente ao

- Tempo de execução

- Estrutura do código (linguagem, condicionais, ciclos)
- DMA, caches, pipeline
- Sistema operativo ou kernel (system calls)

- Tempo de resposta e regularidade

- Interrupções
- Multi-tasking
- Acesso a recursos partilhados (buses, disks, com ports,...)

# *Requisitos dos Sistemas de Tempo-Real*

Os requisitos tipicamente impostos aos Sistemas de Tempo-Real são de 3 tipos:

- **Funcionais**
- **Temporais**
- **de Dependabilidade**

# *Requisitos funcionais*

## **Recolha de dados**

- Amostragem das variáveis do sistema (**entidades de tempo-real**) quer do tipo contínuo quer discreto

## **Controlo Digital Directo (DDC)**

- Acesso directo do sistema controlador aos sensores e actuadores

## **Interacção com o operador (MMI)**

- Informação do estado do sistema, registo histórico, suporte à correcta operação do sistema

# Requisitos funcionais

## Recolha de dados

Internamente ao sistema controlador existem **imagens** locais (variáveis internas) das **entidades de tempo-real** do sistema.

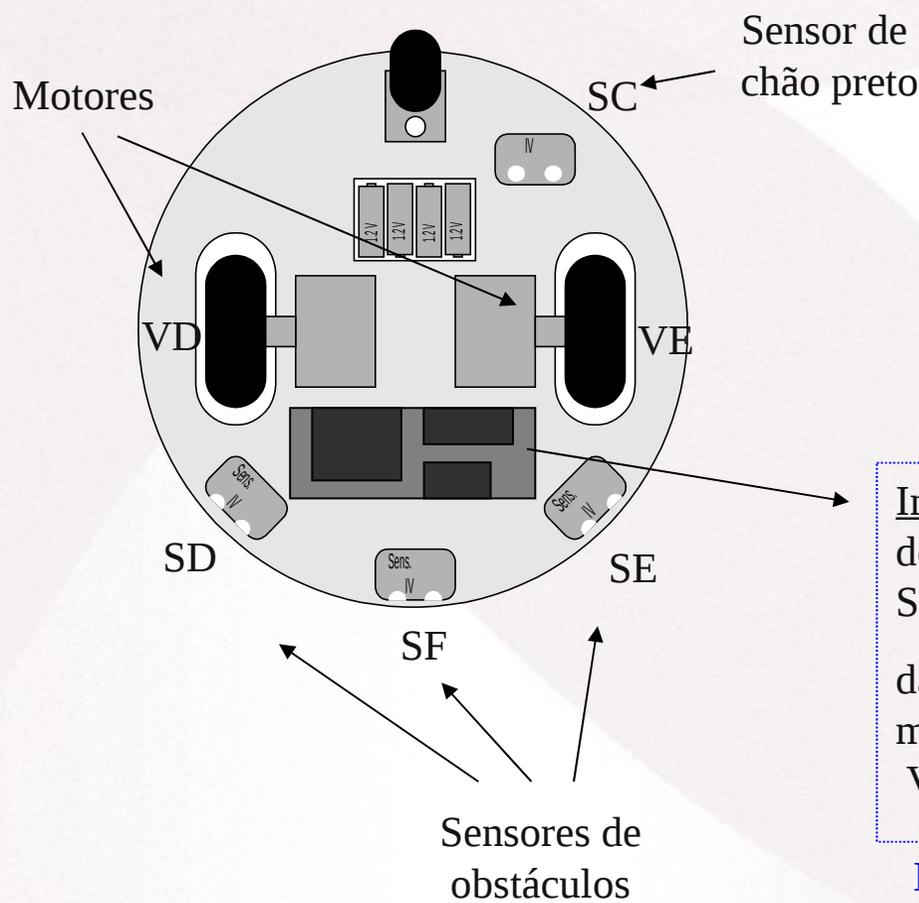
Cada imagem de uma entidade de tempo-real tem uma **validade temporal limitada** devido à dinâmica do processo físico.

O **conjunto das imagens** das entidades de tempo-real compõe a **base-de-dados de tempo-real**.

A **base-de-dados de tempo-real** tem que ser **actualizada** sempre que houver uma mudança de valor numa entidade de tempo-real.

# Requisitos funcionais

**Exemplo:**  
**Um pequeno robot móvel**



Entidades TR:  
sensores:  
SE, SF, SD e SC  
velocidades dos  
motores:  
VE e VD

Imagens internas:  
dos sensores:  
SE', SF', SD' e SC'  
das velocidades dos  
motores:  
VE' e VD'

**Base-de-dados TR**

# Requisitos temporais

Normalmente advêm da **dinâmica do processo físico** que se pretende controlar

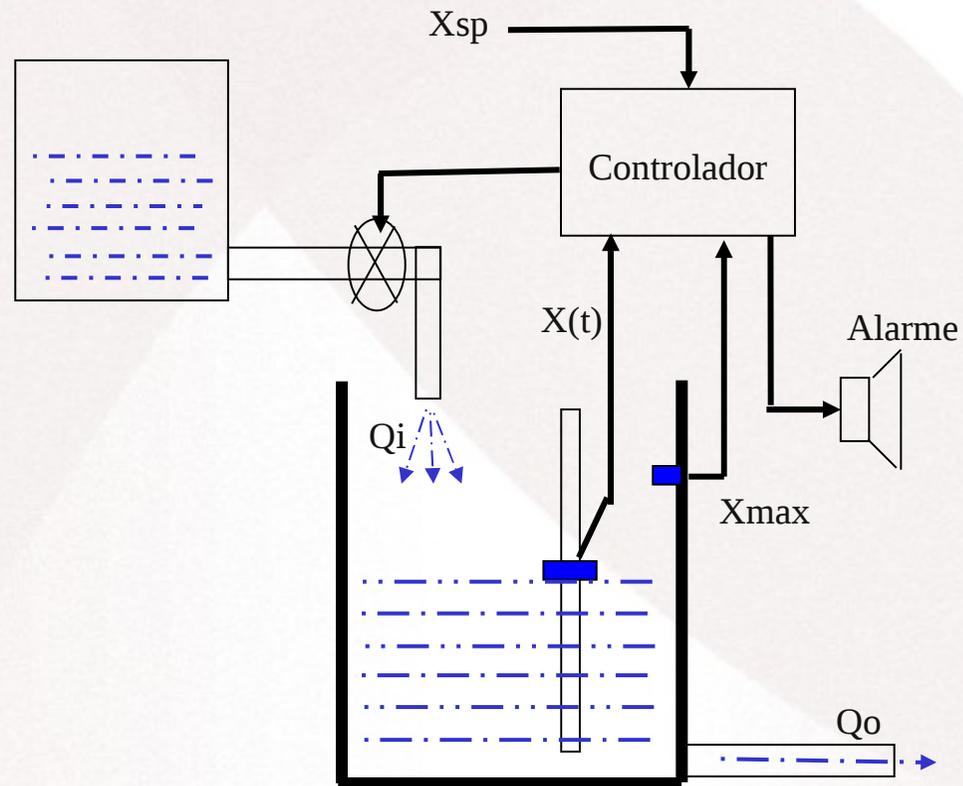
Impõem **restrições**:

- Aos **atrasos de observação** do estado do sistema
- Aos **atrasos de computação** dos novos valores de controlo (actuação)
- Às **variações dos atrasos** anteriores (***jitter***)

que têm de ser cumpridas em **todas as instâncias** (incluindo o pior caso) e **não apenas em termos médios**

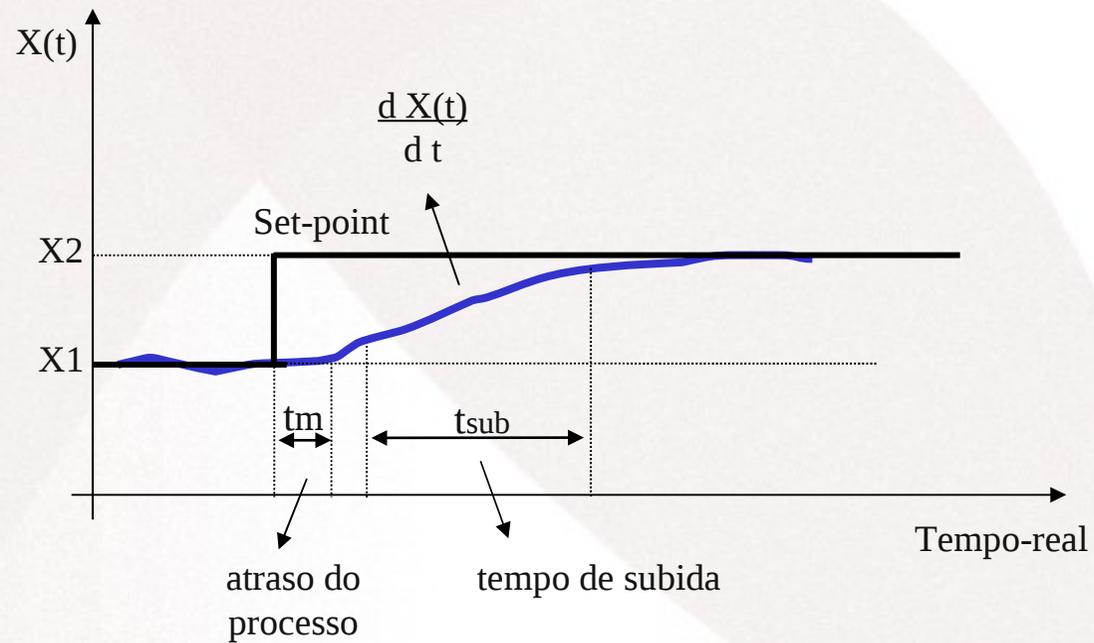
# Requisitos temporais

## Controlo de nível num depósito de líquido



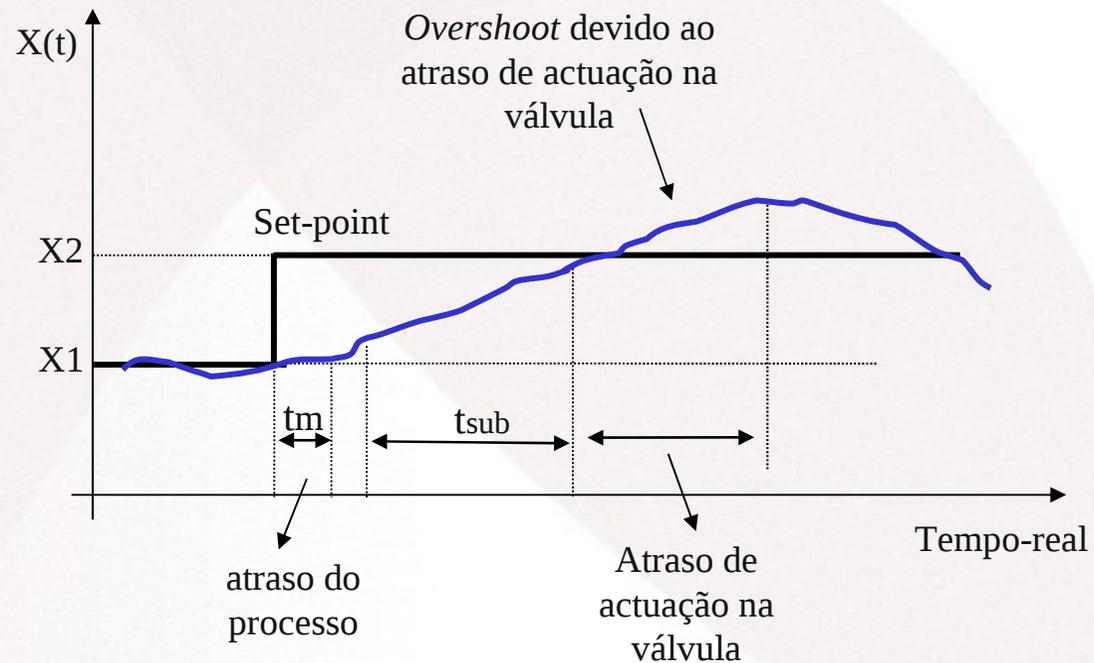
# Requisitos temporais

## Controlo de nível num depósito de líquido



# Requisitos temporais

## Atraso na actuação – degradação do controlo



# Requisitos temporais

## (Sistema de controlo)

Período de **amostragem** –  $T_s$  ( $\sim < 1/10 t_{sub}$  - controlo *quase-contínuo*)

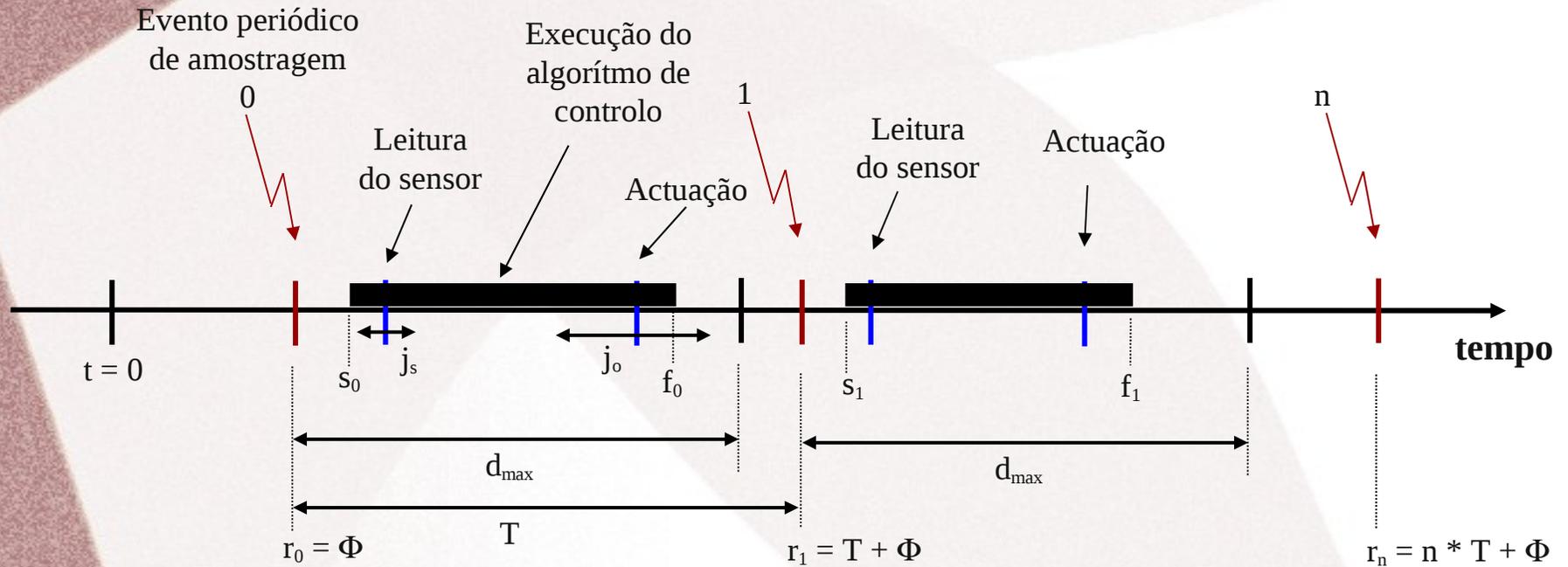
**Atraso** máximo na **resposta** da válvula –  $d_{max}$  ( $< T_s$ )  
(atraso de controlo – pode ser facilmente compensado)

**Variações** no atraso de leitura do nível (**jitter**) –  $j_{s,max}$  ( $\ll d_{max}$ )

**Variações** no atraso de actuação da válvula (**jitter**) –  $j_{o,max}$  ( $\ll d_{max}$ )  
(difíceis de compensar – degradação da qualidade do controlo)

**Atraso** máximo na **sinalização** de alarme –  $d_{al,max}$

# Requisitos temporais



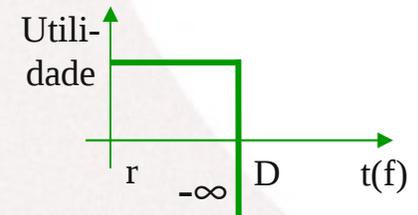
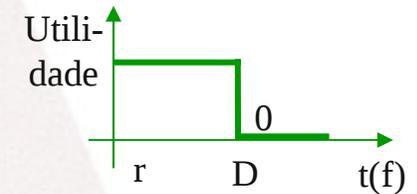
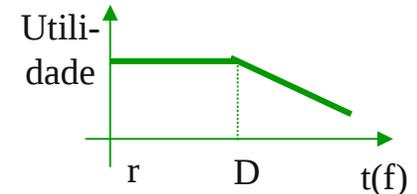
$r_n$  – activação  
 $s_n$  – início de execução  
 $f_n$  – fim de execução

# Requisitos temporais

## Classificação das restrições temporais:

(de acordo com a utilidade do resultado para a aplicação)

- **Suave (Soft)** – Restrição temporal em que o resultado que a ela está associado mantém **alguma utilidade** para a aplicação mesmo depois de um limite  $D$  embora haja uma degradação da qualidade de serviço.
- **Firme (Firm)** – Restrição temporal em que o resultado que a ela está associado **perde qualquer utilidade** para a aplicação depois de um limite  $D$ .
- **Rígida (Hard)** – Restrição temporal que, quando não cumprida, pode originar uma **falha catastrófica**.



# Requisitos temporais

## Classificação dos Sistemas de Tempo-Real:

(de acordo com o tipo das restrições temporais)

- **Soft Real-Time** – O sistema apenas apresenta restrições temporais do tipo **firm** ou **soft** (e.g., simuladores, sistemas multimedia)
- **Hard Real-Time** – O sistema apresenta pelo menos uma restrição temporal do tipo **hard**. São sistemas de **segurança crítica** (e.g. controlo de vôo de aviões, de mísseis, de centrais nucleares, de fábricas de produtos perigosos)

# Requisitos de Dependabilidade

Os **sistemas de tempo-real** são normalmente usados em **aplicações críticas** quer em termos de **segurança** quer **económicos** (e.g., centrais nucleares, controlo de tráfego ferroviário e aéreo, indústria de processo).

Daqui resulta um requisito de:

**Elevada Fiabilidade** – Em sistemas *hard real-time* são típicos requisitos de **fiabilidade ultra elevada** ( $\lambda < 10^{-9}$  falhas/hora – não se pode verificar experimentalmente!).

# Requisitos de Dependabilidade

## Aspectos importantes num sistema de **segurança crítica**:

- **Interfaces estáveis** entre os subsistemas críticos e os restantes por forma a evitar a **propagação de erros** de uns para os outros.
- **Cenários de pior caso** bem definidos. O sistema deve possuir os **recursos adequados** para fazer face aos cenários de pior caso sem necessidade de recurso a argumentos probabilísticos, i.e., deve fornecer **garantias de serviço** mesmo em tais cenários.
- **Arquitetura** composta por **subsistemas autónomos**, cujas propriedades podem ser verificadas independentemente uns dos outros (composabilidade).

# Resumo da Aula 1

# Resumo da Aula 1

- Noção de **tempo real** e de **sistema de tempo real**
- Antagonismo **tempo real** vs **best effort**
- Objectivo do estudo dos STR – obter **garantias** de **comportamento temporal adequado**
- Aspectos a considerar: **tempo de execução**, de **resposta**, e **regularidade** de eventos periódicos
- Requisitos dos STR: **funcionais**, **temporais** e de **dependabilidade**
- Noção de **base de dados de tempo real**
- Restrições **soft**, **firm** e **hard**, e **hard real time** vs **soft real time**
- Importância de ter em conta o **cenário de pior caso**