PCS5761 Especificação de Sistemas de Tempo Real

Prof. Dr. Jorge Rady de Almeida Jr.

Especificação de Requisitos de Sistema

Engenharia de Requisitos

Subdisciplina da Eng. de Software

- Estabelecer objetivos, funções e restrições do sistema
- Objetivo: criar especificações de requisitos completas, corretas e compreensíveis por projetistas e clientes

Processo de estabelecer serviços que o cliente requer/espera de um sistema e suas restrições de operação

Os requisitos representam a descrição desses serviços e suas restrições

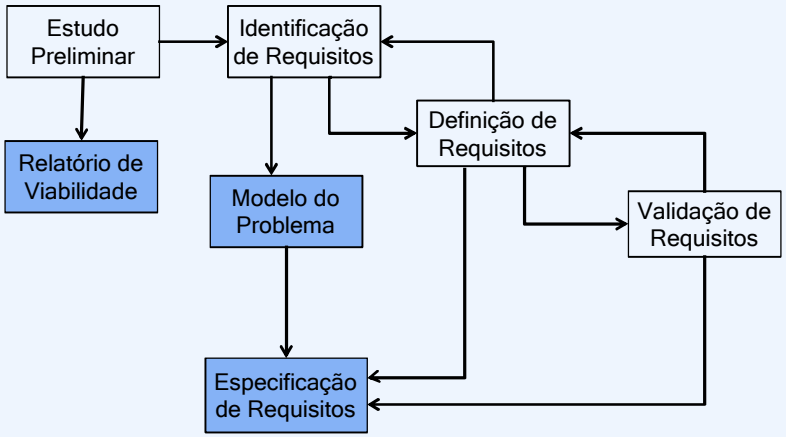
Engenharia de Requisitos

Requisitos

Desde uma descrição com alto nível de abstração até uma descrição matemática detalhada

Servem como base para uma contratação / ordem de serviço

Engenharia de Requisitos



Especificações X Projeto

Complementam-se

Engenharia de Requisitos

- Tipos de Requisitos
  - De Usuário: escritos em linguagem natural, complementado por alguns diagramas
  - De Sistema: documento estruturado com descrição detalhada das funções do sistema, contemplando o que deve ser implementado
  - Funcionais
  - Não Funcionais

Engenharia de Requisitos

Requisitos Funcionais: descrição de

- Entradas, saídas, condições de exceção, . . .
- Serviços/funções a serem ofertados pelo sistema
- Operações sobre as entradas
- Situações anormais
- O que o sistema não deve fazer
- Nível de detalhamento pode variar significativamente

Engenharia de Requisitos

Requisitos Não Funcionais

De produto: restrições nos serviços ofertados pelo sistema, temporizações usabilidade, desempenho, segurança

Organizacionais: processo de desenvolvimento, normas, características ambientais, operacionais e de desenvolvimento

Externos: interoperabilidade, questões regulatórias, éticas e contábeis

Engenharia de Requisitos

Requisitos Não Funcionais

- Podem gerar complementos nos requisitos funcionais
- Aplicam-se ao sistema completo

Engenharia de Requisitos

Métricas para Requisitos Não Funcionais

Propriedade	Medida
Velocidade	Transações processadas / segundo Tempos de resposta Tempo de refresh da tela
Tamanho	Mbytes Número de chips ROM
Facilidade de Uso	Tempo de treinamento Número de telas de auxílio
Confiabilidade	MTTF Disponibilidade Taxa de falhas
Robustez	Tempo de restart Porcentagem de eventos que causam falha

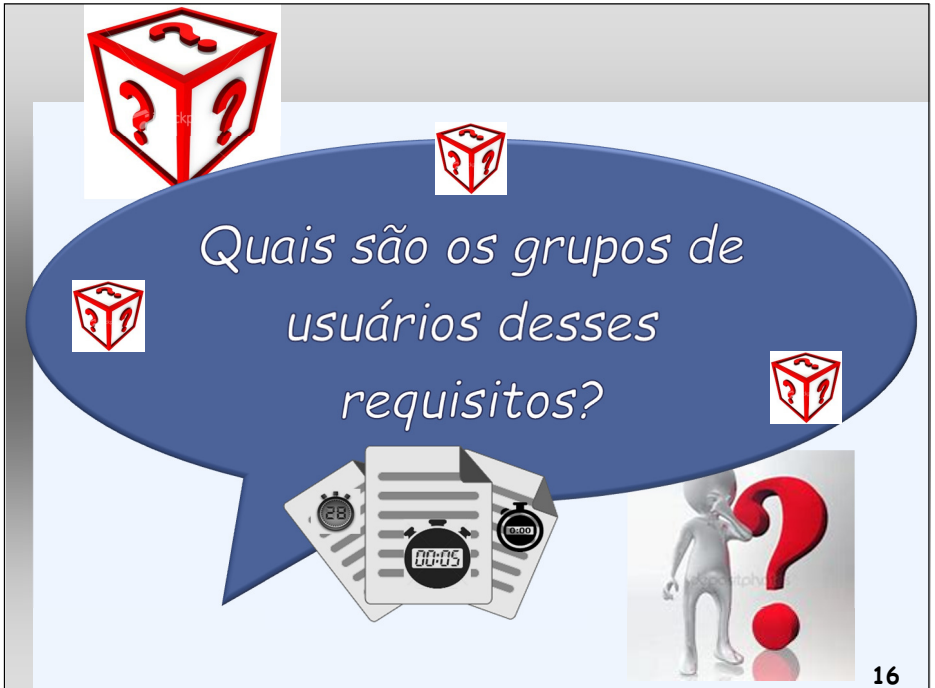
Engenharia de Requisitos

Interfaces externas: detalhamento

- Origem
- Faixa de validade
- Unidades de medida
- Relacionamentos entre elas
- Formatos dos dados
- Temporizações

Engenharia de Requisitos

Características importantes	Correção	Sem Ambiguidades
Completeza	Consistência / Sem Contradições	Classificação de Relevância
Verificável	Modificável	Auditável



Usuários do Documento de Requisitos

Usuário	Medida
Clientes	Especifica / confere / modifica os requisitos conforme suas necessidades;
Gerência	Utiliza para planejar orçamentos o processo de desenvolvimento
Engenheiros de Sistema	Utiliza para compreender qual é o sistema a ser desenvolvido
Engenheiros de Teste de Sistema	Utiliza para desenvolver planos de teste
Engenheiros de Manutenção de Sistema	Utiliza para entender o sistema e a relação entre as partes

Estrutura do Documento de Requisitos

Seção	Descrição
Prefácio	• Leitores esperados, histórico, justificativas da criação/nova versão
Introdução 	• Necessidade do sistema • Breve descrição das funções e da operação, inclusive em conjunto com outros sistemas • Como o sistema irá se enquadrar no quadro de negócios da empresa
Glossário	• Definição dos termos técnicos
Definição dos Requisitos do Usuário	• Descrição dos serviços a serem providos • Requisitos não funcionais • Especificação de normas e processos
Arquitetura do Sistema	• Descrição em alto nível da arquitetura a ser utilizada • Distribuição das funções nos módulos

Estrutura do Documento de Requisitos

Seção	Descrição
Especificação dos Requisitos do Sistema	• Requisitos funcionais e não funcionais em detalhes • Interfaces com outros sistemas
Modelos do Sistema	• Modelos gráficos contendo o relacionamento entre os módulos e com o ambiente
Evolução do Sistema	• Descrever premissas básicas empregadas • Antecipar possíveis alterações nos requisitos, bem como de evolução das técnicas de projeto
Anexos	• Maior detalhamento de partes da especificação
Índices	

Notações para Especificações

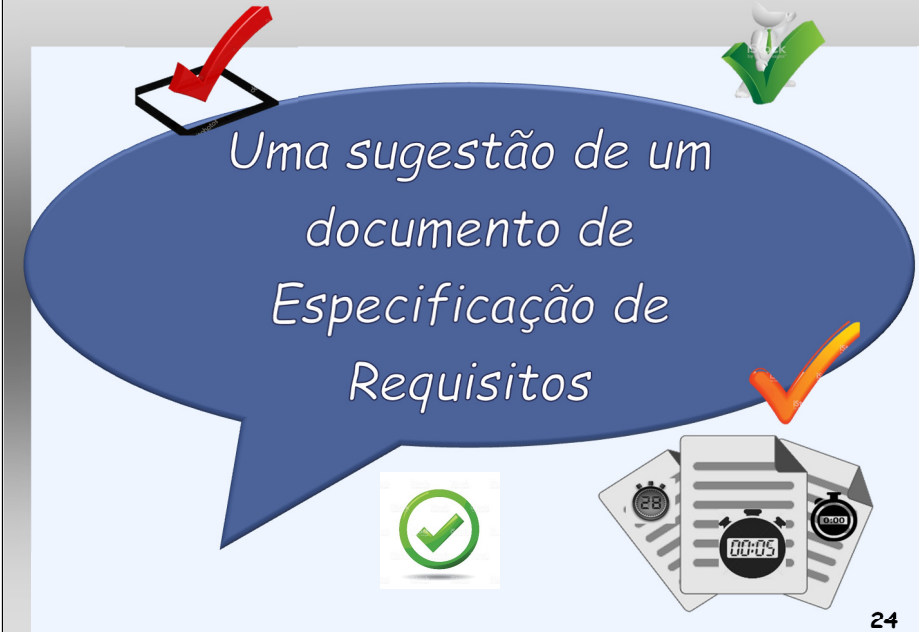
- Informal: não pode ser convertida para uma notação com regras rígidas
  - Linguagem natural
  - Linguagem natural estruturada
  - Notações gráficas
- SemiFormal:
  - Intermediária: possui regras mais fortes de representação
  - Exemplo: UML

Notações para Especificações

- Formal
  - Uso de uma "linguagem de especificação" / notação matemática
  - Pode ter sua correção/consistência demonstrada /provada
  - Maior dificuldade no uso

Especificações: Boas Práticas

- Usar "deve" para requisitos mandatórios
- Evitar linguagem técnica restrita
- Detalhar de acordo com a necessidade
- Não criar/utilizar "linguagem" proprietária



Sugestão de Documento

1. Prefácio
2. Introdução
3. Glossário
4. Descrição Geral
5. Formas de Utilização / Aplicação
6. Arquitetura do Sistema
7. Interfaces
8. Restrições

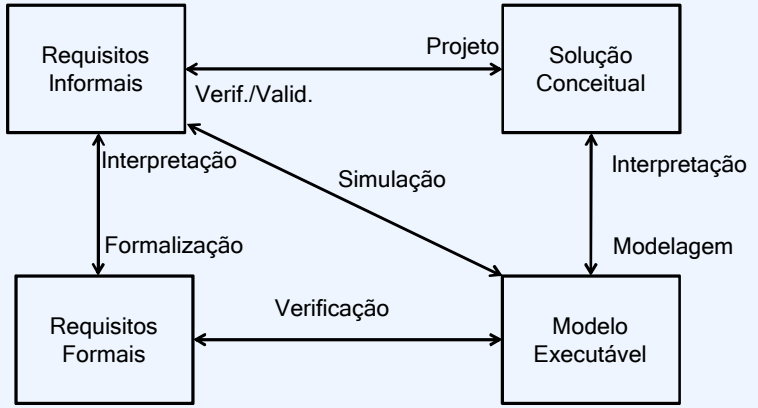
Sugestão de Documento

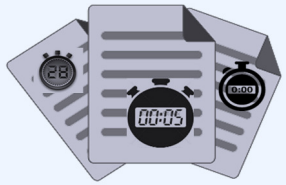
9. Requisitos Gerais: Funcionais e Não Funcionais
10. Requisitos Específicos: Funcionais e Não Funcionais
11. Requisitos Temporais
12. Requisitos de Usuários
13. Manual de Operação
14. Manual de Manutenção
15. Evolução do Sistema

Sugestão de Documento

16. Descrição do Hardware
    - Módulo H1
    - Módulo H2
    - . . . .
17. Descrição do Software
    - Módulo S1
    - Módulo S2
    - . . . .
18. Anexos

Engenharia de Requisitos





Análise de Requisitos de Sistemas de Tempo Real

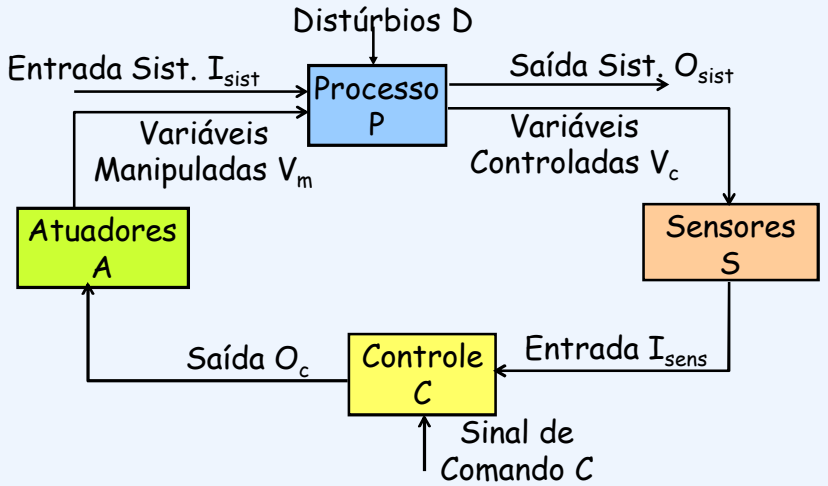
Referência

- Software Requirements Analysis for Real-Time Process Control
  - Matthew Jaffe, Nancy Leveson, Mats Heimdahl, Bonnie Melhart
  - IEEE Transactions on Software Engineering, vol. 17, no.3, march 1991
- Safeware System Safety and Computers, Nancy Leveson, Addison Wesley, 1995

Requisitos

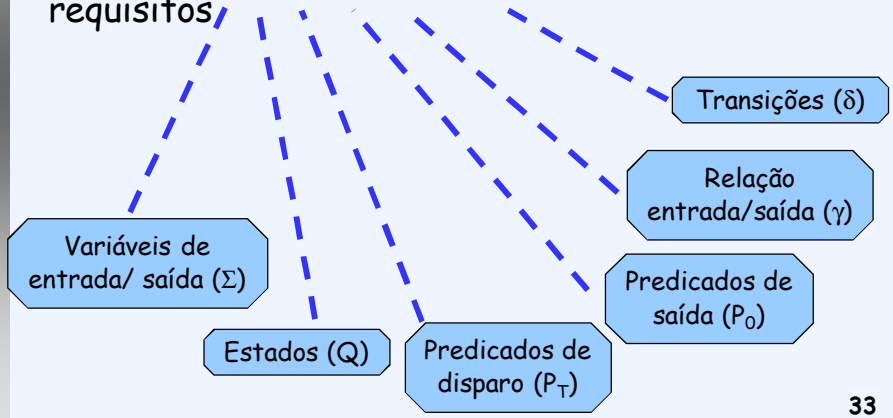
- Problemas/acidentes: deficiências em requisitos de software para tempo real
- Casos não especificados/não tratados
- Especificação de requisitos: distinguir entre comportamento de adequado de quaisquer outros
- Objetivo: obter procedimentos de análise para descoberta de falhas na especificação de requisitos
- Método: construção de um modelo

Modelo Básico

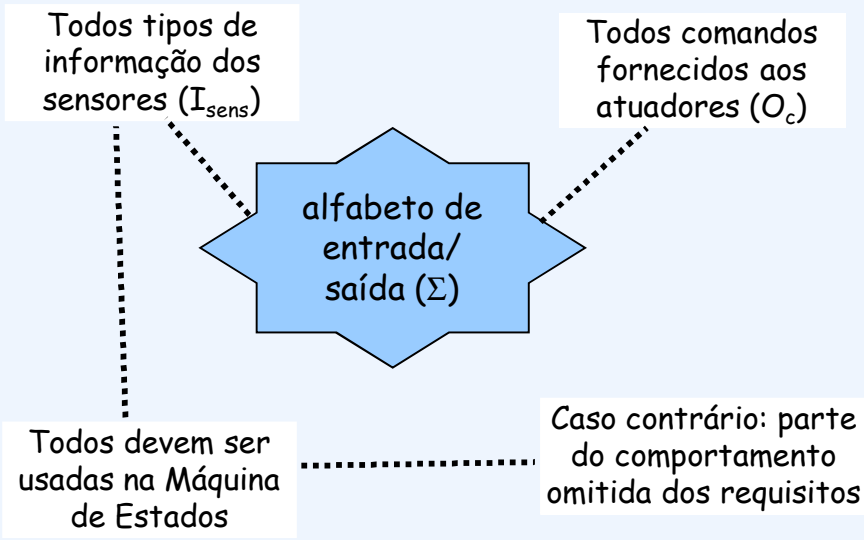


Máquina de Estados de Requisitos

- Análise semântica dos requisitos: critérios para examinar cada parte do *Máquina de Estados de Requisitos* - localizar falhas na especificação de requisitos



1.Variáveis de Entrada/Saída



1.Variáveis de Entrada/Saída

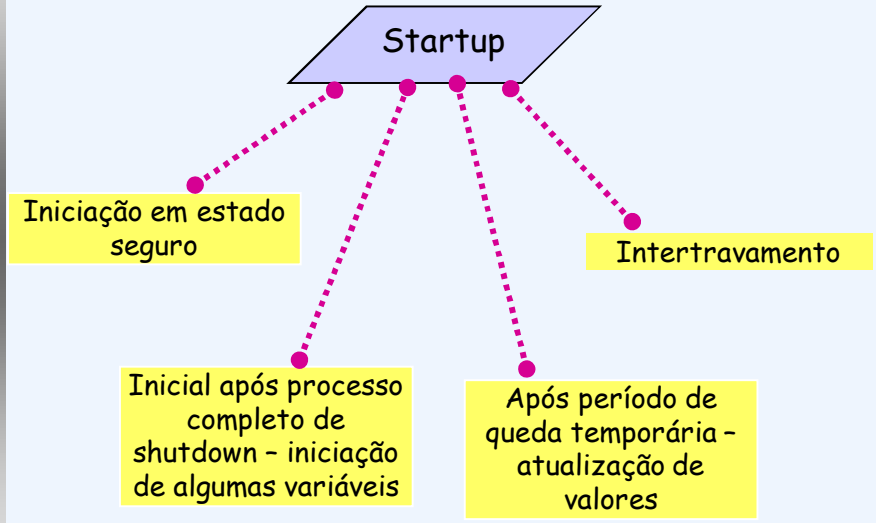
Critério 1.1: Toda informação de entrada (I_{sist}) deve ser usada na especificação de requisitos.

Critério 1.2: Valores legais de saída (O_{sist}) que nunca são produzidos devem ser verificados com relação a especificações incompletas.

2.Estados

- Requisitos específicos para estados de startup/ shutdown
  - Problemas: inadequação de transições entre processamento normal e queda parcial/total
- Minimizar tempo de permanência em estados com funcionalidade reduzida
- Resposta específica para chegada de entradas, em qualquer estado

2.Estados



2.Estados

Critério 2.1: O comportamento do software com relação a entradas recebidas antes do startup, após o shutdown ou quando o computador estiver desconectado do processo, deve ser especificado

3.Predicados de Disparo

- Sempre haver caminho que permita à Maquina de Estados sair de qualquer estado existente no modelo

Critério 3.1: Cada estado devem ter um comportamento ou transição definido para todas suas entradas possíveis

Critério 3.2: Pelo menos uma das condições de disparo das transições que causam a saída de um estado deve sempre ser verdadeira - se isto não ocorrer, há alguma condição esquecida

3.Predicados de Disparo

- Sempre haver caminho que permita à Máquina de Estados sair de qualquer estado existente no modelo

Critério 3.3: Todo estado deve possuir uma transição definida no caso de não ocorrerem entradas por um determinado período de tempo (timeout)

3.Predicados de Disparo

- Comportamento determinístico da Máquina de Estados: garantia de que transições de qualquer estado sejam disjuntas

Critério 3.4: O comportamento da Máquina de Estados deve ser determinístico (apenas uma das transições possíveis a partir de um estado deve ser permitida em um determinado instante)

3.Predicados de Disparo

- Suposição do Valor Essencial:

- Recepção de uma entrada com valor inesperado: ambiente tem comportamento não previsto

Critério 3.5: Requer-se uma verificação de valor para cada entrada I_{sens}

3.Predicados de Disparo

- Chegada de valores de entrada - incluir limite inferior/superior de tempo

Critério 3.6: Todas entradas I devem ser amarradas ao tempo, ou seja, um predicado

$$t_{low} \leq t\,(I_{sist}\uparrow) \leq t_{high} \quad \text{ou} \quad t_{high} \leq t\,(I_{sist}\downarrow) \leq t_{low}$$

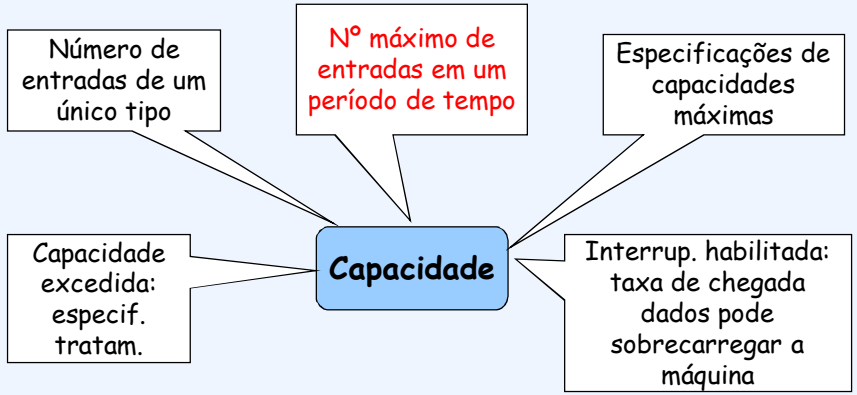
deve ser incluído em todas transições disparadas pela chegada de I_{sist}

3.Predicados de Disparo

- Limite superior no tempo que o sistema aguarda por uma entrada I_{sist}, antes de produzir uma saída O_{sist}

Critério 3.7: Um disparo envolvendo a não existência de uma entrada I durante um intervalo especificado deve ser completamente amarrado ao tempo $t > t_{start} + t_{dur}$ onde t_{start} e t_{dur} são calculados a partir de eventos observáveis ou expressos em tempo absoluto

3.Predicados de Disparo



3.Predicados de Disparo

Critério 3.8: Um evento que cause uma interrupção e que não seja controlado pelo sistema, requer suposições de capacidade sobre ele

Critério 3.9: Devem ser previstas taxas mínima e máxima de recepção de eventos de entrada

4.Predicados de Saída

- Especificação completa do comportamento da saída O_{sist}: especificação do valor e limites inferior / superior de tempo t ($O_{sist}\uparrow$)
- Capacidade do Ambiente
  - Taxa que atuadores aceitam/reagem a dados produzidos
  - Limitações da capacidade: físicas, comportamento do processo, considerações de segurança

Critério 4.1: Taxa de absorção pelo ambiente externo deve ser igual ou superior à taxa de eventos de entrada (sensores, controle, atuadores)

4.Predicados de Saída

- Envelhecimento dos Dados
  - Dados ficam obsoletos
  - Decisões de controle: dados do estado corrente.

Critério 4.2: Todas entradas utilizadas na especificação de saídas devem ser amarradas no tempo, considerando um fator de validade dos dados, D_v, ou o limite de "idade" para a entrada I_{sist} disparar a saída O_{sist}

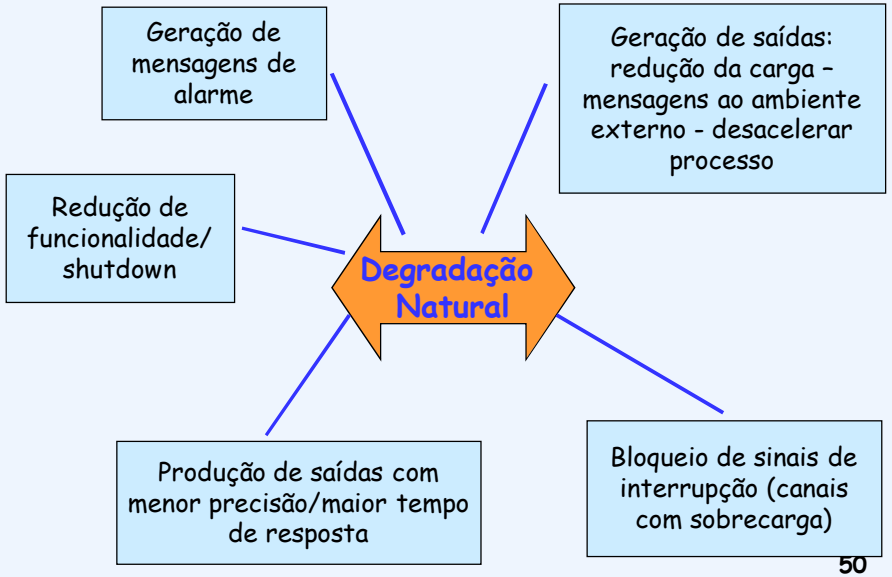
4.Predicados de Saída

- Latência

Critério 4.3: O fator de latência deve ser incluído na especificação, quando o limite superior do intervalo de não existência não é um evento diretamente observável.

Critério 4.4: Saídas safety-critical devem ser verificadas quanto a seus valores e tempo de geração

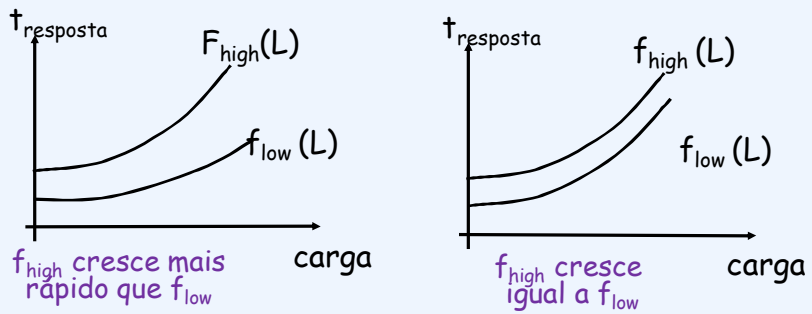
5.Relação Entrada/Saída

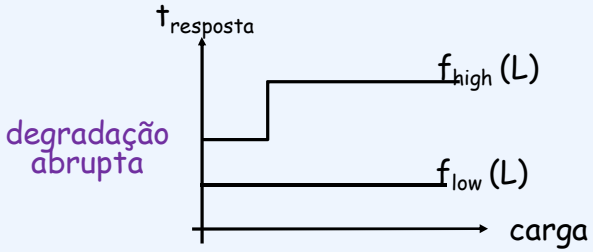


5.Relação Entrada/Saída

Critério 5.1: Se a resposta desejada a uma condição de sobrecarga for degradação no desempenho, a degradação deve ser natural ou suave, ou seja, $f_{low}(L) = f_{high}(L)$ (significa mesma taxa de variação ou crescimento) $f_{low}(L)$ e $f_{high}(L)$ definem limites no tempo de resposta (L = Load)

5.Relação Entrada/Saída





5.Relação Entrada/Saída

- Carga excedida: sistema → estado com desempenho degradado
  - Tomar alguma ação (alertar operador, desabilitar interfaces, gravar parâmetros críticos)
- Especif. condições para a volta ao estado normal
- Processo de histerese: sistema não ficar oscilando (normal/degradado)

Critério 5.2: Deve haver um atraso por histerese, bem como devem ser verificadas as condições apropriadas para o retorno ao processamento normal

6.Transições

- Alcançabilidade Básica: um estado q_m é alcançável a partir de um estado q_n, se existe um caminho entre q_n e q_m

Critério 6.1: Todos os estados devem ser alcançáveis a partir do estado inicial q_0

6.Transições

- Reversibilidade

Critério 6.2: Deve ser possível efetuar a reversão de qualquer comando existente no sistema

- Alcançabilidade de Estados Seguros

Critério 6.3: Não deve haver caminhos para estados inseguros

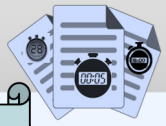
Critério 6.4: Todo caminho de um estado inseguro (se atingido) deve levar a um estado seguro

6.Transições

Critério 6.5: Se um estado seguro não puder ser alcançado a partir de um estado inseguro, todos os caminhos desse estado devem levar a estados de mínimo risco

Modelagem

Modelagem de STR

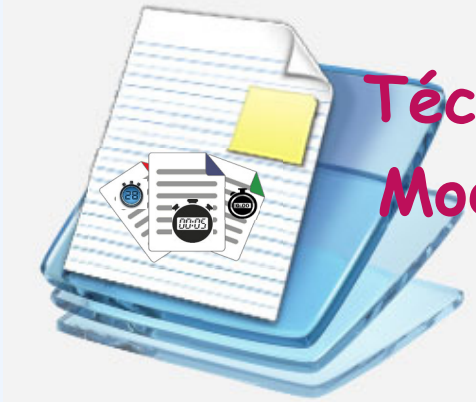


- Notação propriamente utilizada
  - Especificação de requisitos útil/compreensível
  - Base para todo o projeto/testes
- Prevenir ambigüidades, incompletezas e inconsistências
- Incentivo ao analista a pensar e escrever
  - Visão externa do produto
- Auxílio na organização da informação
- Adequação a cada aplicação particular

Modelagem de STR

- Não há técnica que resolva todos os problemas
- Devem ser usadas em conjunto
- Especificação do aspecto temporal
  - Mais importante em um STR
- Especificação: o que o sistema deve fazer
- Projeto: como o sistema deve executar suas atribuições

Técnicas de Modelagem



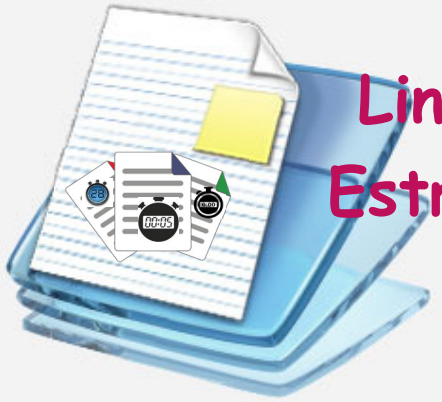
Técnicas de Modelagem

Linguagem Natural	Linguagem Estruturada	Pseudocódigo	Tabela de Decisão
Fluxograma	Diagrama Estrutural	Diagrama de Transição de Estados	Diagrama de Fluxo de Dados
Dicionário de Dados	Diagrama Entidade-Relacionamento	Autômatos	Redes de Petri
	Statecharts	Orientação a Objetos	

Linguagem Natural

Linguagem Natural

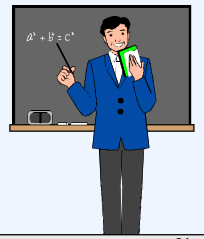
- Linguagens naturais – português, inglês, . . .
  - Necessárias para especificação de sistemas
  - Uso mais frequente: forma redundante de descrições em outros meios (ex. ressaltar pontos de maior importância)
  - Utilização isolada: descrições ambíguas e de leitura trabalhosa



Linguagem Estruturada

Linguagem Estruturada

- Subconjunto da linguagem natural
  - Objetivo - equilíbrio
    - Linguagens de programação (formais) e
    - Informalidade/legibilidade da linguagem natural

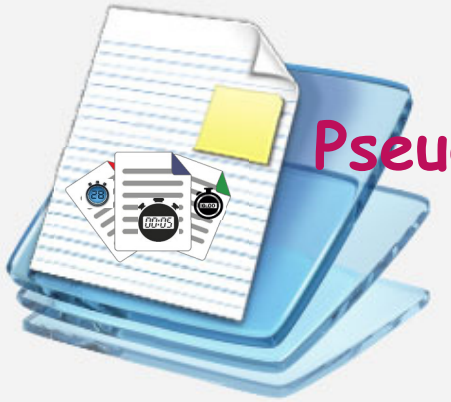


SE não houver trem nos próximo 5 circuitos de via ENTÃO, **em no máximo 2 s**

a) Libera código de velocidade máxima para o trem

b) Fecha todas as portas do trem

c) Emite aviso sonoro de partida



Pseudocódigo

Pseudocódigo

- Linguagem de Programação de Projeto ou Linguagem de Programação de Alto Nível
  - Alta abstração
  - Transformação direta em um programa
  - Não tem ligação com a estrutura das máquinas onde o sistema será executado

3.Pseudocódigo

- Exemplo: caixa eletrônico

```
begin
   Aceita cartão do cliente
   Exibe mensagem "Digite seu Código"
   Se Código incorreto
         begin
           Exibe (em até 1s) mensagem "Código Incorreto"
           Ejeta cartão
       end
      else
          begin
            Exibe (em até 1s) mensagem "Selecione Opção"
            Lê Tecla selecionada
            Se Tecla = "Conta Corrente"
            begin
            . . . . . .
end
```

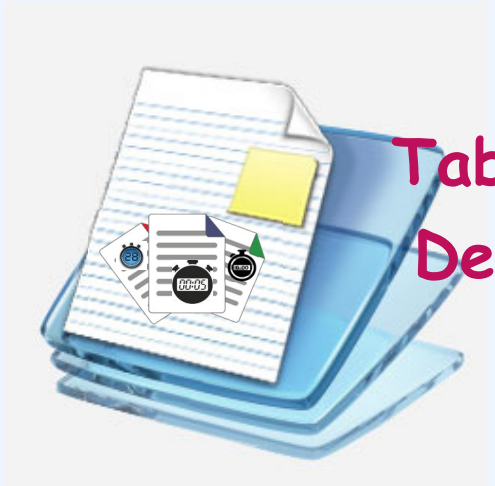


Tabela de Decisões

Tabela de Decisões

Aplicação

- Decisões baseadas em variáveis que podem assumir diferentes valores
- Processo produz saídas ou executa ações com base em decisões complexas
- Tabela de Decisões: relaciona todas entradas com todas ações relevantes
- Forma completa e sem omissões

4. Tabela de Decisões

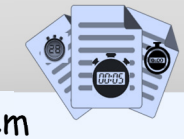
	CV1	CV2	CV3	CV4	CV5	CV6	CV7	CV8
Ocupação	N	S	N	N	N	N	N	S
Veloc. Máx. (Km/h)	0	44	44	30	30	30	0	0
Aceleração	N	S	S	N	N	N	N	N

Tempo máximo para chaveamento de código = 2s

Cada técnica permite a obtenção de aspectos distintos de um STR

Fluxograma

Fluxograma



- Forma gráfica mais antiga de modelagem
- Muito útil em sistemas simples
- Em sistemas multitarefa
  - Descrever cada tarefa separadamente
  - Interação entre tarefas: feita por outra representação

Fluxograma

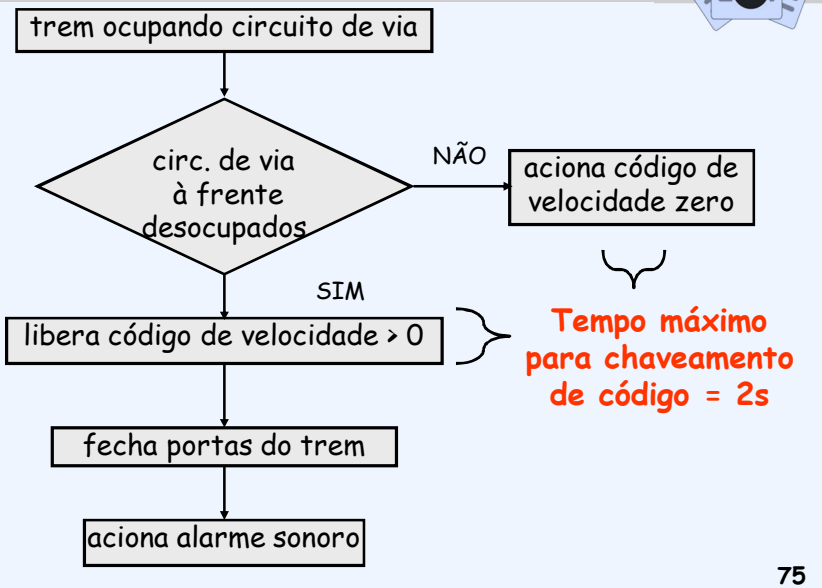
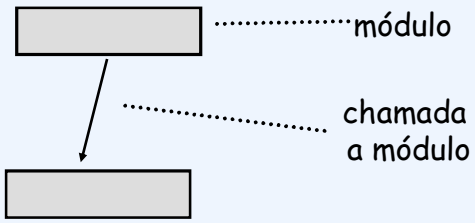


Diagrama Estrutural

Diagrama Estrutural

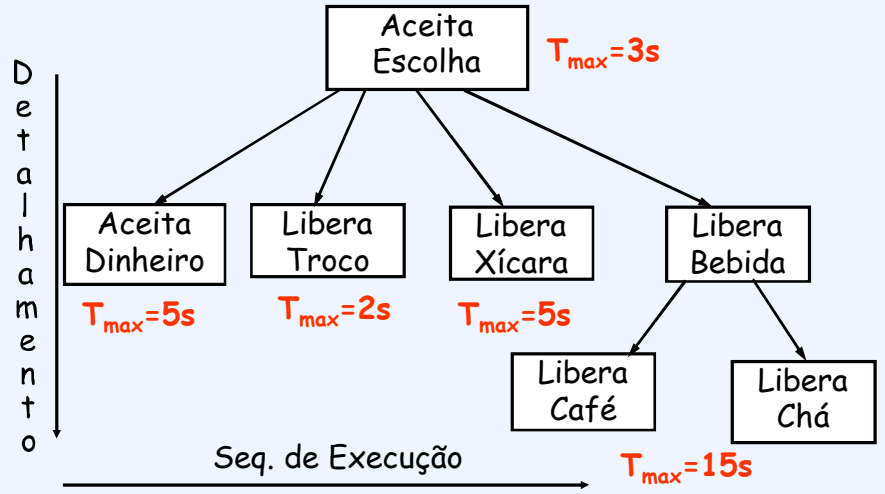
- Ou Árvore de Chamada
- Hierarquia de módulos de um sistema



- Detalhamento crescente → top-down

Diagrama Estrutural

Ex. máquina de vender café



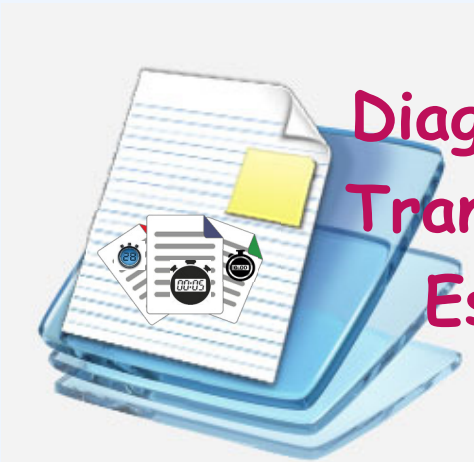


Diagrama de Transição de Estados

Diagrama de Transição de Estados

- Modelagem do comportamento em função de
  - Tempo
  - Seqüência de acesso aos dados
  - Seqüência de execução das funções

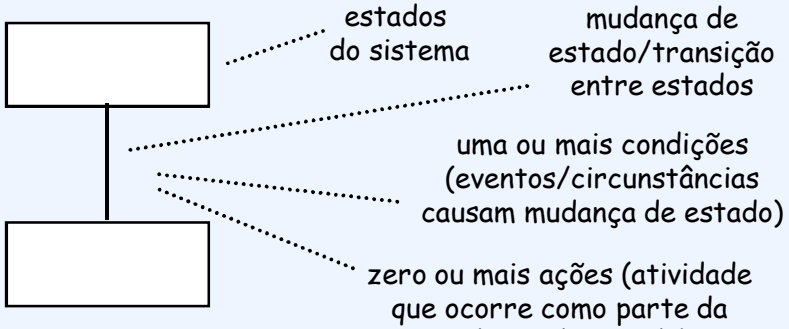


Diagrama de Transição de Estados

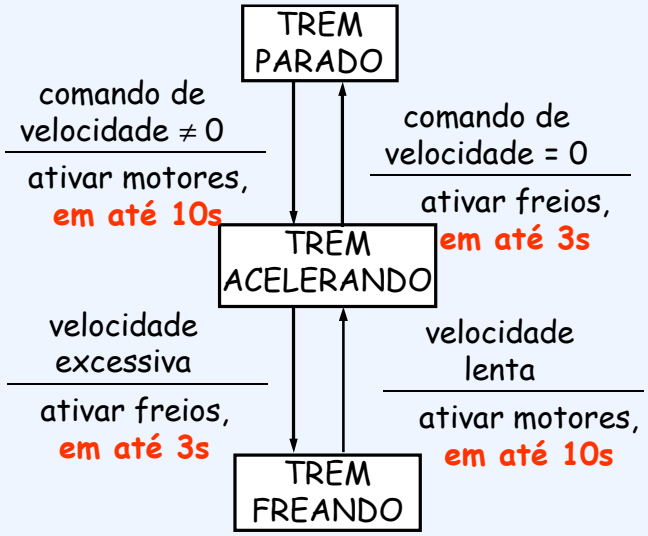


Diagrama de Fluxo de Dados

Diagrama de Fluxo de Dados

- Sistema representado como
  - Rede de processos funcionais
  - Interligados por transições e
  - Armazenamento de dados
- Transformação: dados se movem da entrada para a saída
- Particionado em níveis – detalhamentos sucessivos da funcionalidade do sistema

Diagrama de Fluxo de Dados

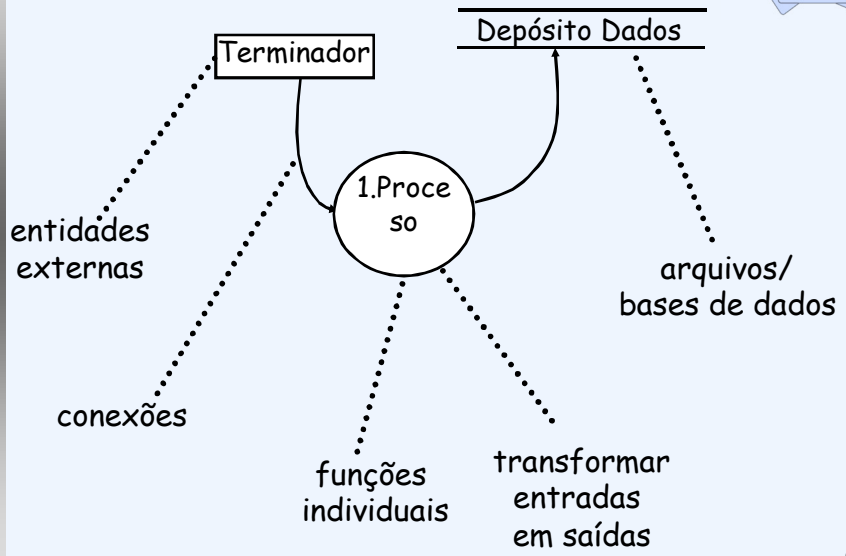


Diagrama de Fluxo de Dados

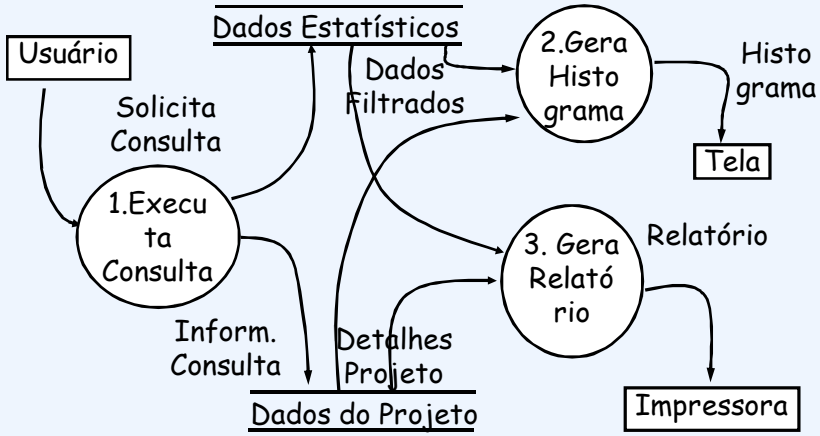
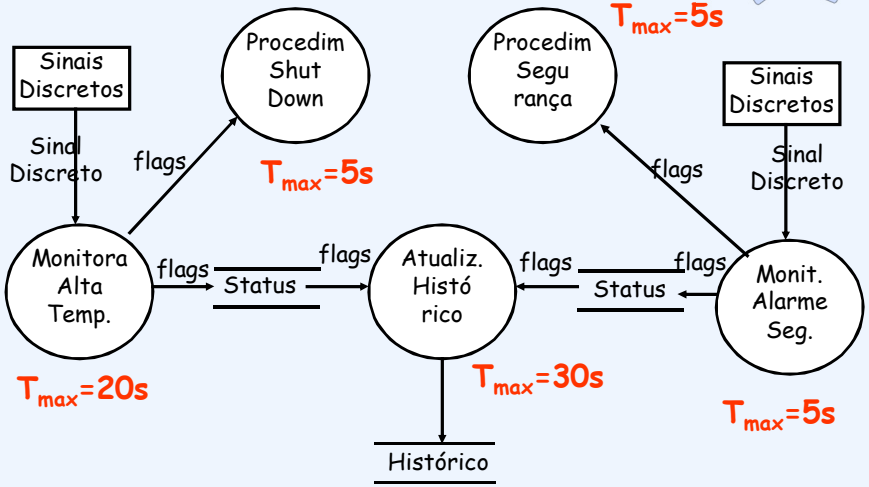
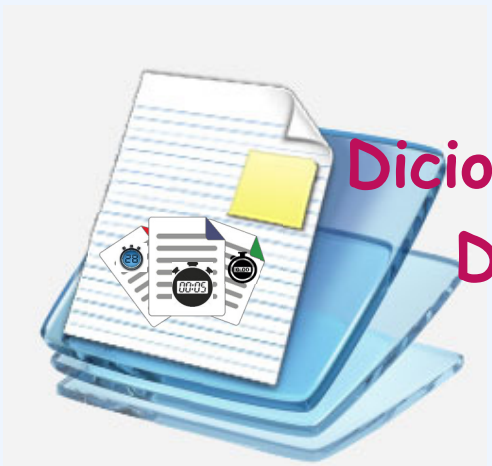


Diagrama de Fluxo de Dados





Dicionário de Dados

Dicionário de Dados

DFD — não fornece detalhes

visão geral principais componentes funcionais

Dicionário de Dados

definições precisas – todas entradas/ saídas, componentes de depósitos de dados

quais informações são transformadas e a como são transformadas

listagem organizada de todos elementos de dados

ferramenta de suporte textual de modelagem

Dicionário de Dados

companhia =	nome_cia + CGC_cia + IE_cia + endereço_cia
nome_cia =	{caractere válido}
endereço_cia =	{caractere válido}
CGC_cia =	{dígito válido}
IE_cia =	{dígito válido}
caractere válido =	[A-Z/a-z/'/-/ /]
dígito válido =	[1-0]

Diagrama Entidade-Relacionamento

Diagrama Entidade Relacionamento

- Objetivo: modelo de **Dados** do sistema
  - Universo formado por grupos de objetos (entidades) e por relacionamentos entre eles
- Entidade
  - Cada objeto distinguível de outros objetos
  - Atributos – propriedades que descrevem a entidade
- Relacionamentos
  - Associação entre entidades
  - Cardinalidade de mapeamento

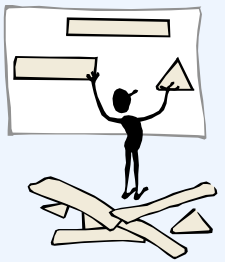
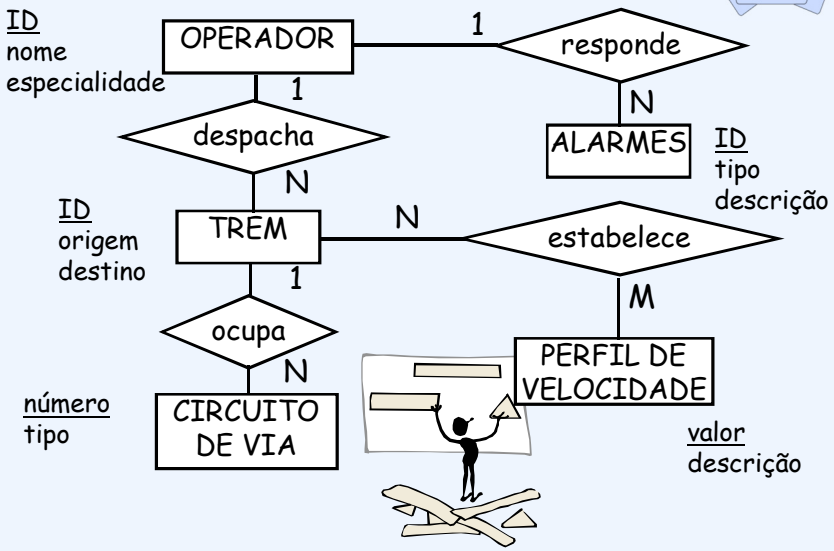


Diagrama Entidade Relacionamento



Pode haver sobreposição de aspectos cobertos entre as técnicas de modelagem

Autômatos

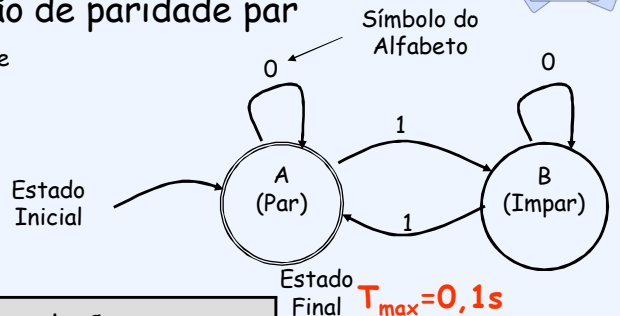
Autômatos

- Ou Máquinas de Estado Finito
- 3 formas de representação
  - Formulação matemática
  - Diagrama
  - Matriz
- Não apresenta ambigüidades
- Número de estados pode ser muito grande
- Técnicas para reduzir número de estados

Autômatos

- Ex. detecção de paridade par

Modelo de Moore



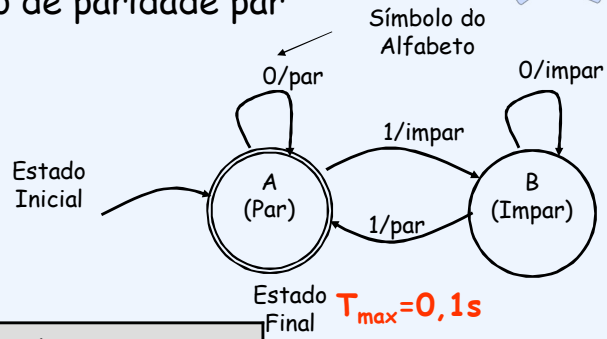
T_{max}=0,1s

Entrada	Estado Corrente	
	A	B
0	A	B
1	B	A

Autômatos

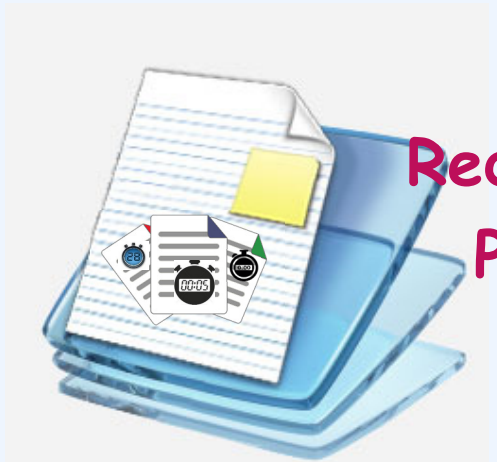
- Ex. detecção de paridade par

Modelo de Meali



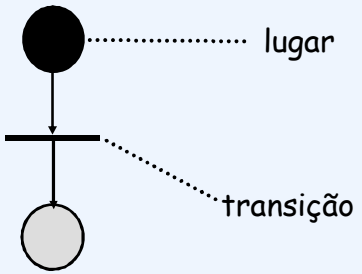
T_{max}=0,1s

Entrada	Estado Corrente	
	A	B
0	A/par	B/impar
1	B/impar	A/par



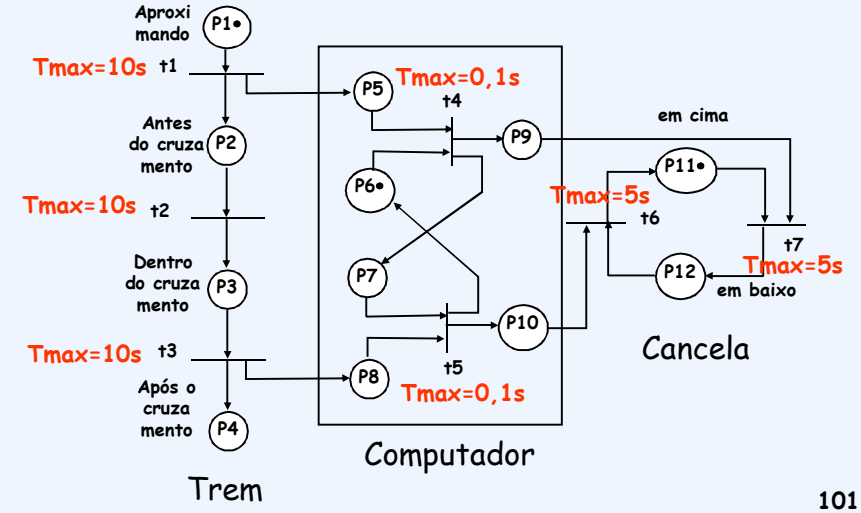
Redes de Petri

Redes de Petri



Redes de Petri

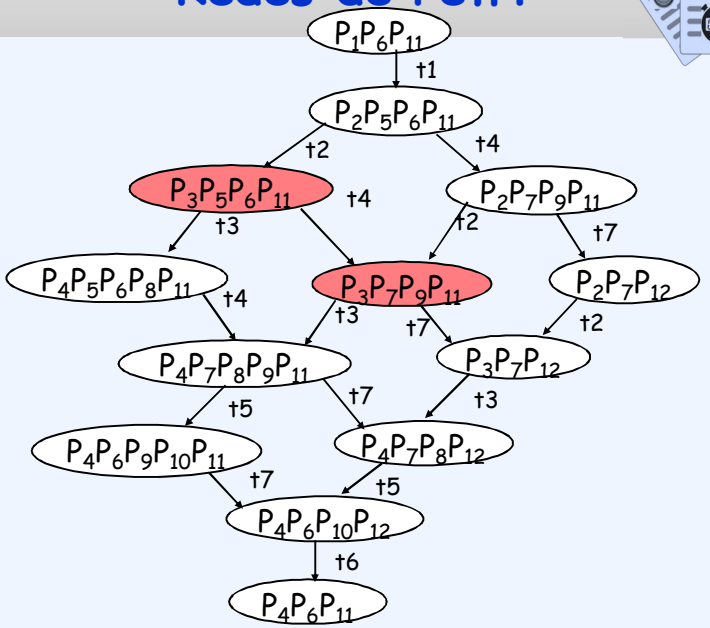
Ex.: sistema de controle de cancela: trem e sua linha, um computador de controle da cancela e a cancela propriamente dita



Redes de Petri

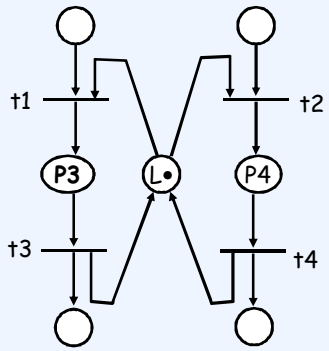
- Grafo de Alcançabilidade
  - Conjunto de lugares ocupados em todas possíveis execuções da Rede de Petri
    - Nós: estados (marcas)
    - Arcos: transições entre estados
  - Utilidade: identificação de condições perigosas

Redes de Petri

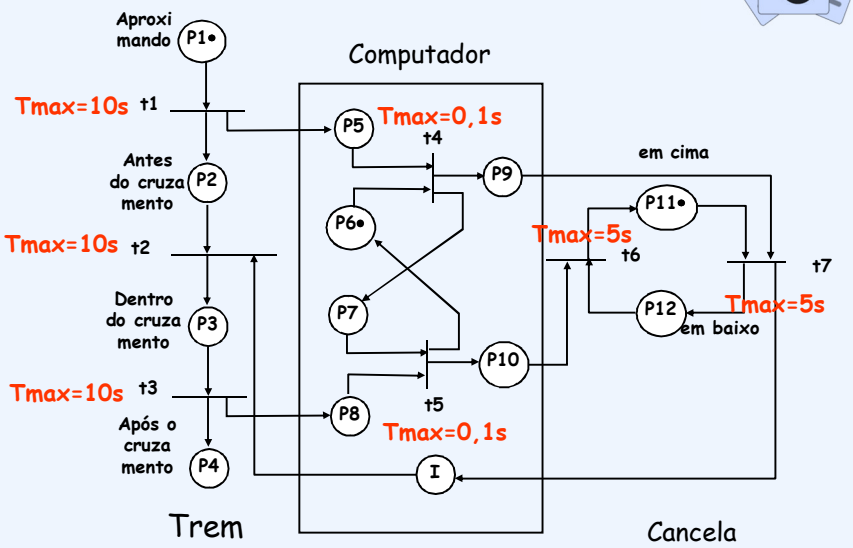


Redes de Petri

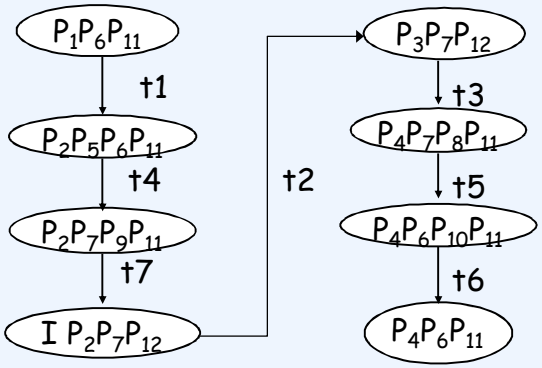
- Um evento não ocorre enquanto o caminho de execução não termina



Redes de Petri



Redes de Petri

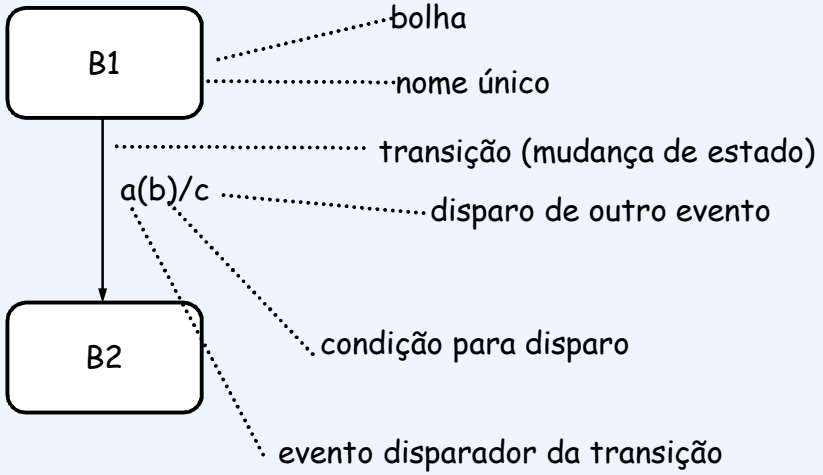


Statecharts

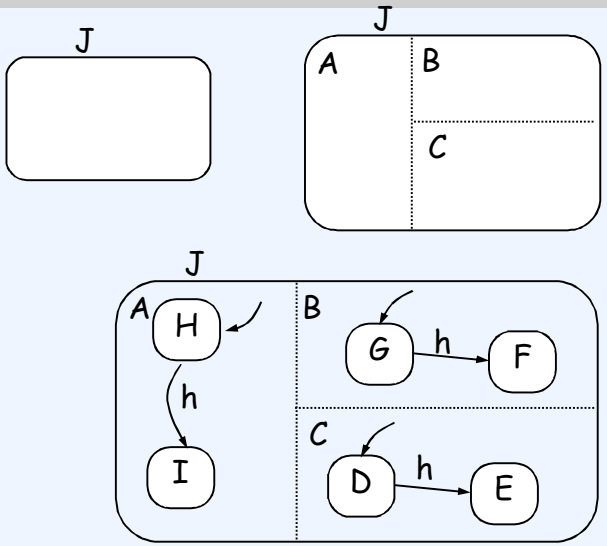
15. Statecharts

- Representa o comportamento dinâmico - comportamentos sequencial e concorrente
- Possui características das Máquinas de Estado Finito
  - Profundidade – detalhamento em diversos níveis
  - Ortogonalidade – representação de tarefas concorrentes – estados AND
  - Comunicação Broadcast (diversos processos ortogonais respondem ao mesmo evento)

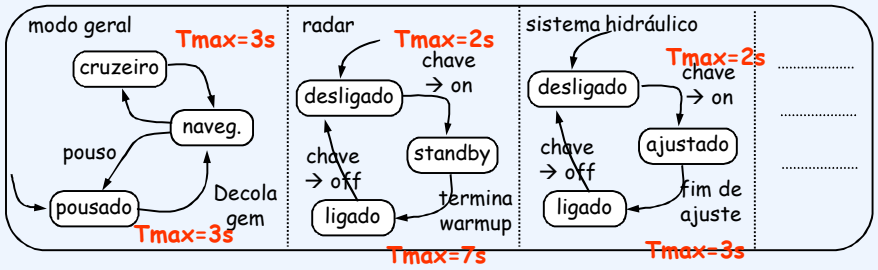
15.Statecharts



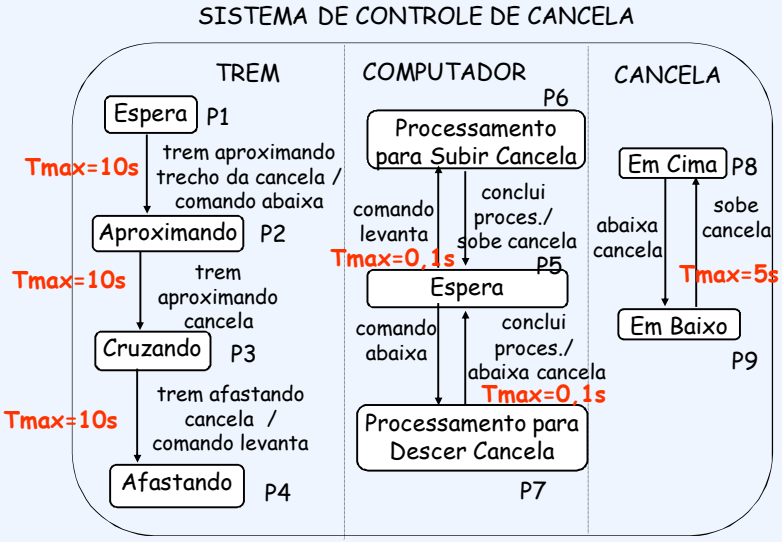
15.Statecharts



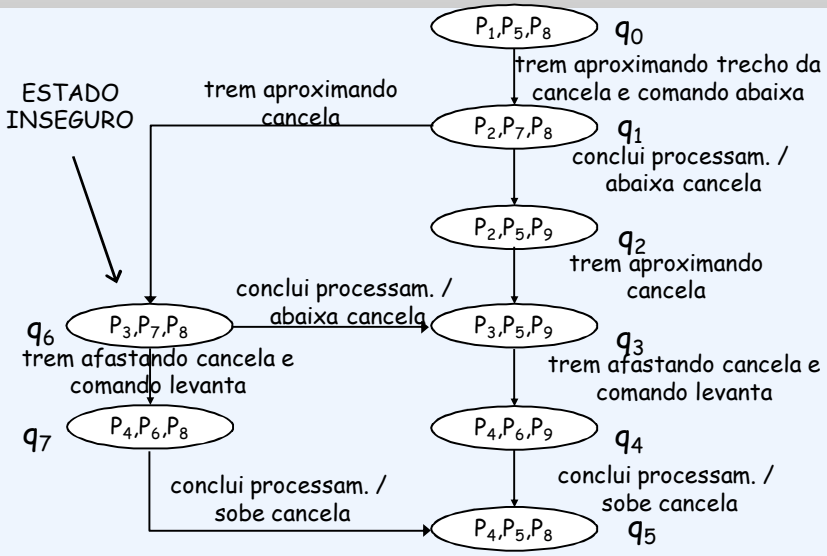
15.Statecharts



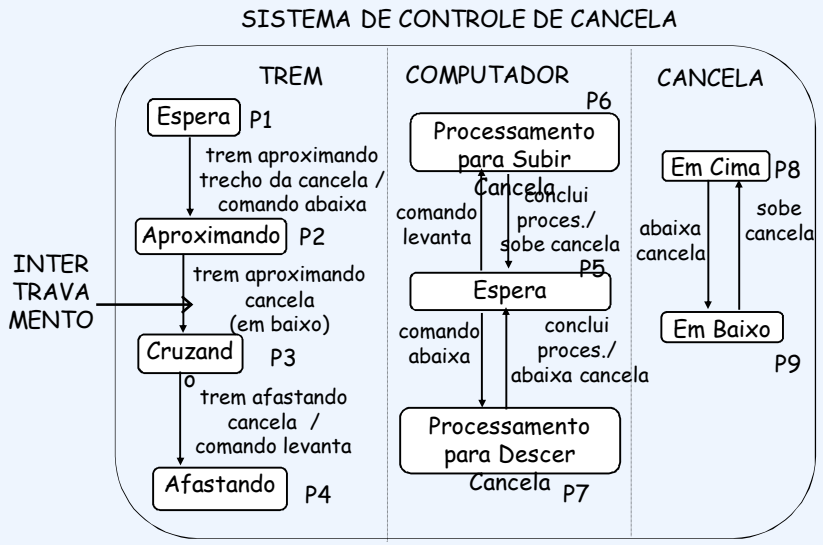
15.Statecharts



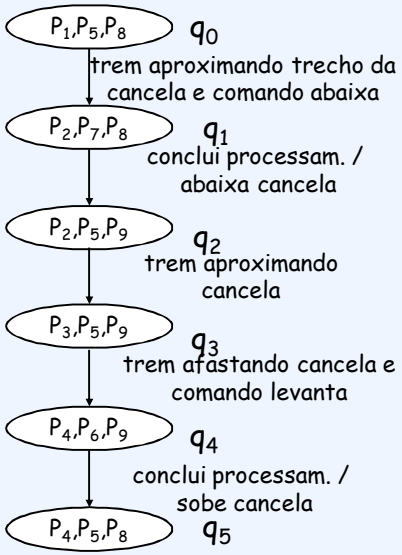
15.Statecharts

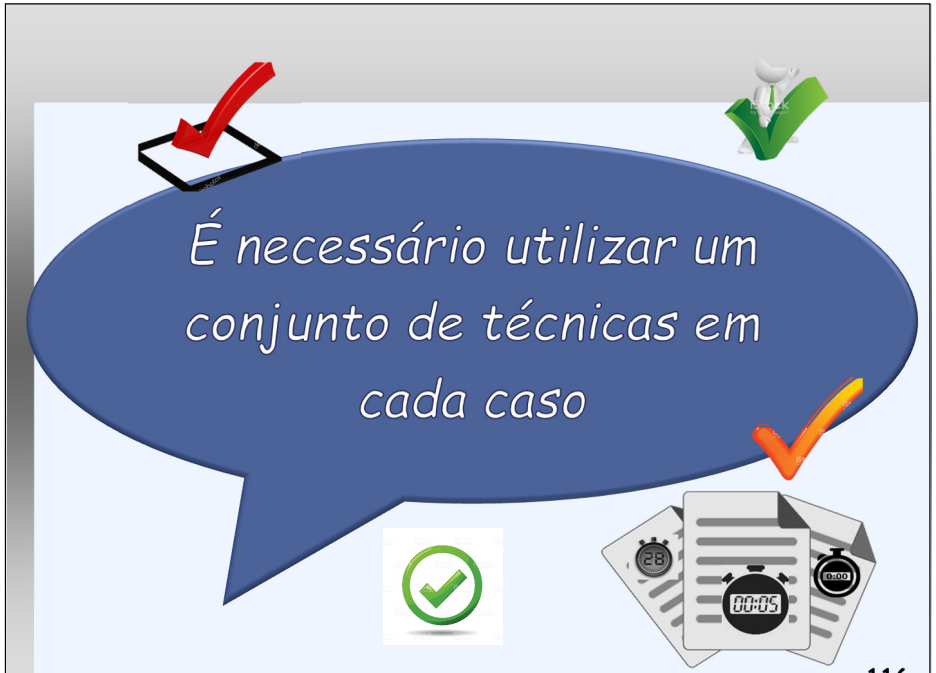


15.Statecharts



15.Statecharts

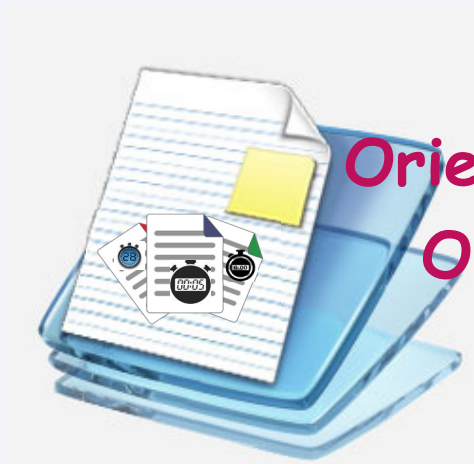








Há a necessidade de se contar com uma ferramenta automatizada



Orientação a Objetos

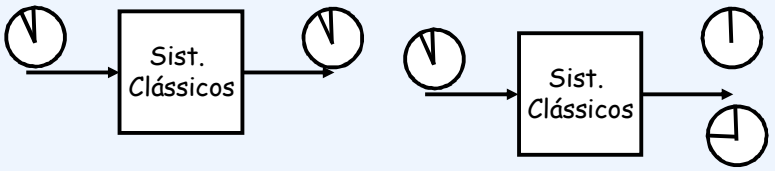
Orientação a Objetos

- Casos de Uso
- Diagrama de Classes
- Diagrama de Sequência
- . . . . . .

Controle Computadorizado

Caract. Controle Computadorizado

	Sistemas Clássicos	Sistemas Computacionais
Pequenas alterações nas entradas	pequenas alterações nas saídas	Sem alteração nas saídas
		Grandes alterações nas saídas



Caract. Controle Computadorizado

Sistemas Clássicos	Sistemas Computacionais
Espaço de estados contínuo	Espaço de estados discreto
Transições suaves	Transições abruptas
Conjunto de testes finito	Conjunto de testes muitíssimo grande
Permite interpolações	Interpolações não são adequadas
Modelagem matemática	Prototipação e testes
Comportamento predictível	Comportamento parcialmente predictível

Caract. Controle Computadorizado

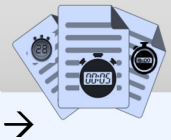
- Técnicas para evitar erros de software
  - Testes
    - Teste exaustivo: cobrir todos casos de teste → inviável
  - Diversidade de Projeto
    - Independência entre equipes de desenvolvimento
    - Não é provado
    - Técnica cara
  - Teste e diversidade de projeto não proporcionam resultados convincentes → **Métodos Formais**

Caract. Controle Computadorizado

- Prevenção de falhas no SW
  - Especificação rigorosa/formal de requisitos
  - Utilização de metodologias comprovadas de projeto
  - Documentação completa de desenvolvimento e de de implementação → Manutenção

Métodos Formais

Métodos Formais

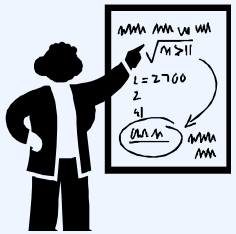


- Modelos matemáticos e lógicas formais → especificar/ verificar requisitos de projeto

Notação Matemática Simples

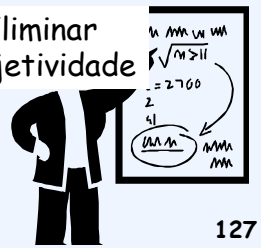
orçamento, planejamento, necessidades técnicas

Especificações Completamente Formais



Métodos Formais

- Erros ↓
- Inconsistências ↓
- Ambiguidades ↓
- Não dependência de extrapolação de testes
- Verificação da Completeza
- Ampla abrangência de casos
- Modelo Matemático → Propriedades Lógicas
- Eliminar subjetividade
- Seleção de sub-conjunto de Requisitos

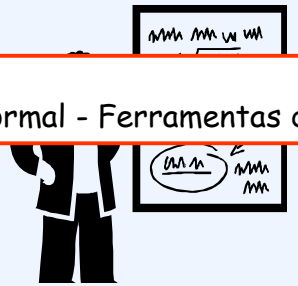


Métodos Formais

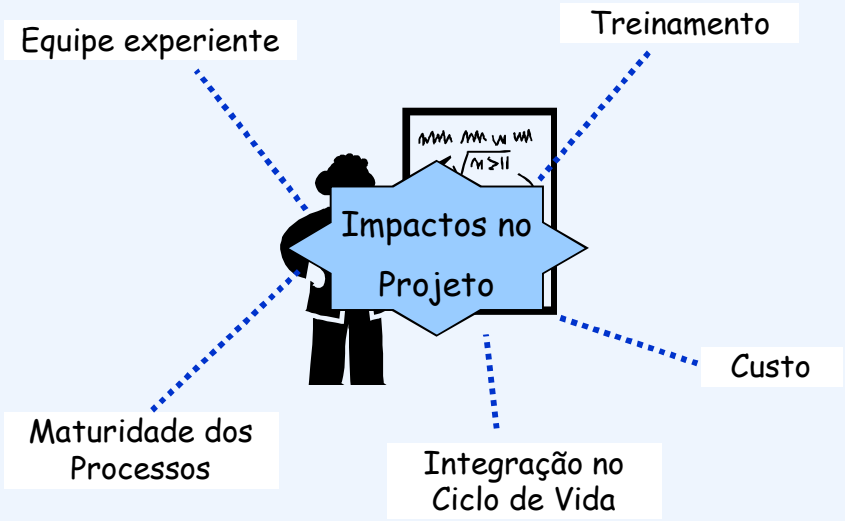
Nível 1
Especificação Formal
Lógica Matemática/Linguagem de Especificação

Nível 2
Prova de Correção Formal - Manual

Nível 3
Prova de Correção Formal - Ferramentas de SW



Métodos Formais



Métodos Formais x Testes

- Testes
  - Não são abandonados
  - Concentram-se em faixa mais restrita / problemática de casos de teste
  - Aplicáveis apenas após conclusão do programa
  - Verificar funcionamento p/ número finito valores
- Métodos Formais
  - Aplicável antes que o programa seja concebido
  - Verificar funcionamento p/ todos valores possíveis

Métodos Formais - Definições

Especificação Formal: descrição concisa do comportamento e das propriedades de um sistema, escrita em linguagem matemática, especificando o que o sistema deve realizar, da forma mais abstrata possível

Métodos Formais - Definições

Abstração: simplificação e descarte de informações não relevantes à especificação, permitindo que o foco esteja nos aspectos de maior importância

Métodos Formais - Definições

Prova Formal: argumentação completa e convincente sobre a validade de uma afirmação sobre a descrição de um sistema. A prova é composta por passos, cada um deles justificado

Classificação de Métodos Formais

- Métodos descritivos baseados em propriedades - definir o sistema por equações algébricas
  - VDM (Viena Development Method) – metodologia de desenvolvimento de software
  - Z – notação para especificação formal
  - B - método de especificação e implementação
  - LOTOS (Language of Temporal Ordering Specification)
  - CCS (Cauculus of Communicating System)
  - CSP (Communicating Sequential Process)
- E por equações lógicas
  - Lógicas Temporais
  - RTTL (Real Time Temporal Logic)

Até que ponto são necessários métodos Formais no projeto de um STR?

Lógicas de Tempo Real

Lógicas de Tempo Real

Linguagem de Especificação Formal: especificação dos requisitos

Sistema de Prova: verificar correção do sistema com relação aos requisitos

Conjunto de **heurísticas** e **linhas mestras** de uso

Operadores temporais

RTL – Real Time Logic

- @ - Função de Ocorrência

  @ (EVENTO, i) = tempo da i-ésima ocorrência de EVENTO
  (i: inteiros não negativos)

- Início e fim de um evento: ↑Evento, ↓Evento

RTL – Real Time Logic

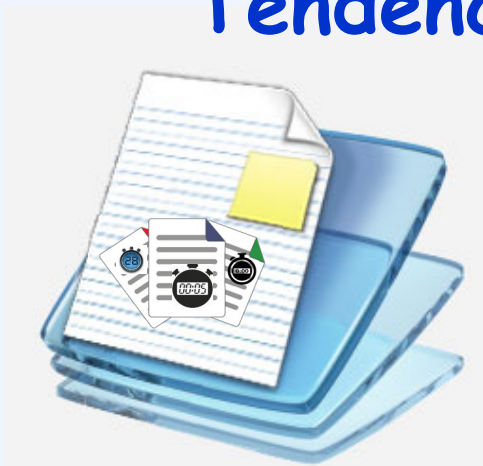
- Suponha que junto à pressão do botão A, uma computação Teste deve ser concluída em 30 seg.

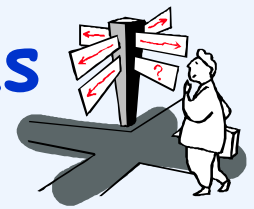
$$\forall x\ @(A, x) < @(\uparrow Teste, x)) \wedge\ (@(\downarrow Teste, x) \leq @(A,x) + 30)$$

- Duas ações A_1 e A_2 devem ser mutuamente exclusivas, isto é, durante a execução de A_1, A_2 não pode ser executada e vice-versa

$$\forall i, j\ ((@(\downarrow A_1, i) < @(\uparrow A_2, j) \vee (@(\downarrow A_2, j) < @(\uparrow A_1, i)))$$

Tendências



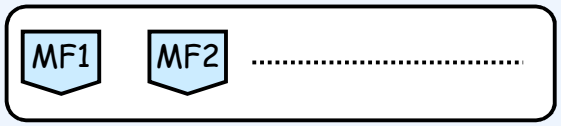


Tendências

- Grande campo de pesquisa
- Proliferação de formalismos
- Indústria
  - Métodos de projeto visuais / linguagens de TR
- Meio Acadêmico
  - Verificação / análise formal

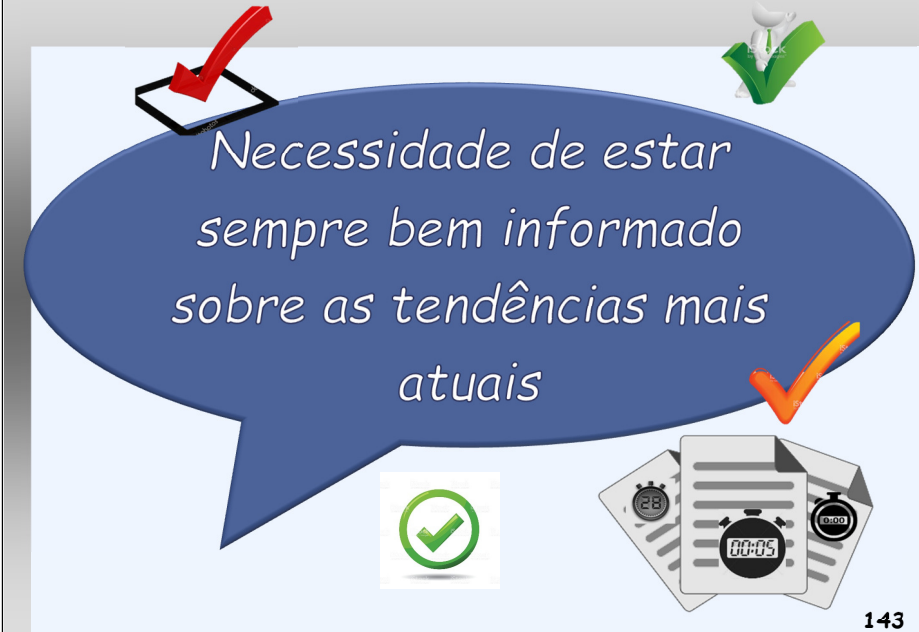
Indústria ⟺ Academia

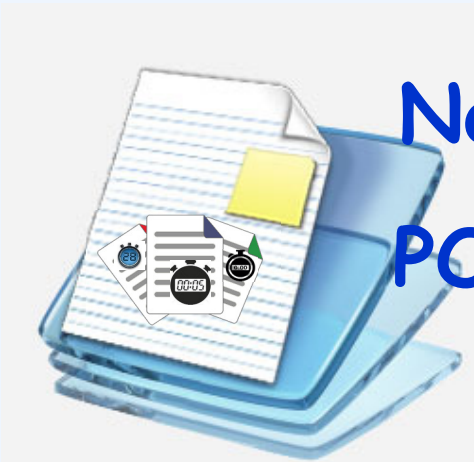
Tendências



Síntese / Análise formal e automática

Síntese / Análise semi-formal e semi-automática





Norma POSIX

POSIX – Portable Operating System Interface

- Padrão IEEEpara SO baseado no UNIX
- Define interfaces p/ SO – não a implementação
- Padroniza sintaxe de rotinas de bibliotecas
- 1003.1 – Funcionalidade básica, chamadas de sistema para gerência de processos, dispositivos, sistemas de arquivos e comunicação entre processos
- 1003.1b – Estende 1003.1 para TR, incluindo semáforos, escalonamento com prioridades, temporização

POSIX

IEEE Standard for Information Technology—Portable Operating System Interface (POSIX)—Part 1: System Application Program Interface (API)—Amendment 1: Realtime Extension [C Language]

IEEE Std 1003.1b-1993
(Formerly known as IEEE P1003.4)
(Includes IEEE Std 1003.1-1990)

Abstract: This amendment is part of the POSIX series of standards for applications and user interfaces to open systems. It defines the applications interface to basic system services for input/output, file system access, and process management. It also defines a format for data interchange. When options specified in the Realtime Extension are included, the standard also defines interfaces appropriate for realtime applications.

Keywords: API, application portability, C (programming language), data processing, information interchange, open systems, operating system, portable application, POSIX, programming language, realtime, system configuration computer interface

POSIX

This standard describes the external characteristics and facilities that are of importance to applications developers, rather than the internal construction techniques employed to achieve these capabilities. Special emphasis is placed on those functions and facilities that are needed in a wide variety of commercial applications and applications with realtime requirements.

The interfaces included were the set required to make this standard minimally usable to realtime applications on single processor systems. The scope is to take existing realtime operating system practice and add it to the standard. The definition of *realtime* used in defining the scope of this standard is:

> "Realtime in operating systems: the ability of the operating system to provide a required level of service in a bounded response time."

The key elements defining the scope are

(1) Defining a sufficient set of functionality to cover a significant part of the realtime application program domain, and

(2) Defining sufficient performance constraints and performance-related functions to allow a realtime application to achieve deterministic response from the system.

POSIX

As normas exercem um papel importante na modelagem e projeto de um STR

Técnicas de Verificação de Projeto de Sistemas de Tempo Real

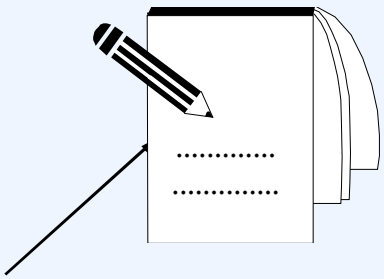
Checklists | Árv. Falhas | Árv. Eventos | HAZOP | FMEA | Anal. Erros Humanos

Técnicas de Verificação de Projeto de STR

- Checklists
- Árvore de Falhas (falha - eventos)
- Árvore de Eventos (eventos - falha)
- HAZOP - Hazard and Operability Analysis
- FMEA – Failure Mode and Effect Analysis
- Análise de Erros Humanos

Checklists

- Aproveitar experiência de projetos anteriores – não cometer mesmos erros
- Origem: normas/recomendações/experiência/ dados históricos



Lista com Principais aspetos ligados ao tempo

Árvores de Falhas

Análise de causas de **desvios temporais**

Obter combinações de falhas individuais que podem provocar um **desvio temporal**

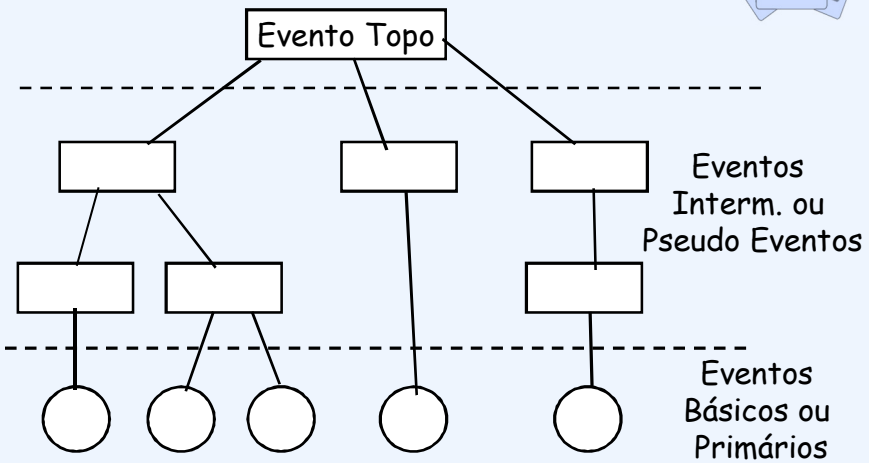
Evento resultante da combinação de outros eventos

Evento básico

AND gate

OR gate

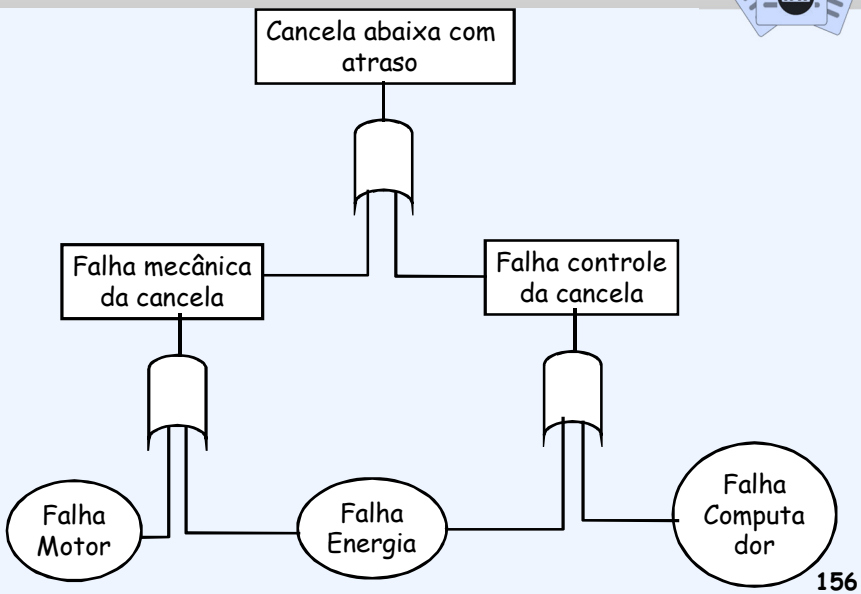
Árvores de Falhas



Árvores de Falhas

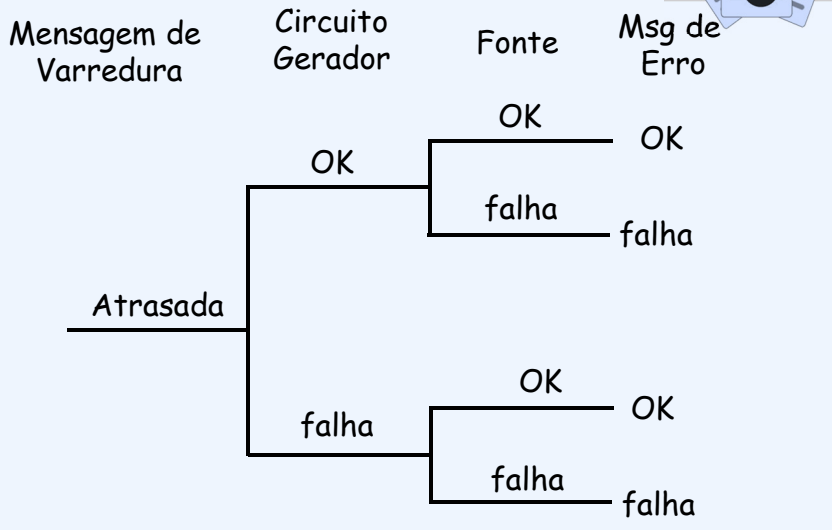
- Combinações de eventos para causar evento topo
  - Análise Qualitativa
  - Análise Quantitativa
    - Probabilidades de ocorrência nos eventos
    - Cálculo da probabilidade de ocorrência do evento topo
- Atenção a falhas de modo comum em eventos básicos

Árvore de Falhas



Árvore de Eventos

Árvore de Eventos



HAZOP

- **Hazard and Operability Analysis** ou **Análise de Operabilidade e Perigo**
- Análise qualitativa
- Origem: indústria química
- A cada desvio do processo: considerar causas potenciais e efeitos no sistema
- Palavras guia aplicadas a variáveis de interesse: fluxo, temperatura, tempo

HAZOP – Palavras Guia

- **NO, NOT, NONE**: resultado esperado não foi obtido
- **MORE/LESS**: mais (menos) do que alguma propriedade física relevante não ocorreu
- **EARLY/LATE**: sinal chega antecipadamente (atrasado) com relação a um clock
- **BEFORE/AFTER**: sinal chega antes/depois de uma sequência
- **PART OF**: apenas algumas das intenções de projeto foram obtidas
- **REVERSE**: ocorre o oposto ao esperado
- . . . . . .

HAZOP

Atributo	Palavra Guia	Causa	Conseq.	Recomendação
Sinal de Aplicação de Freio	NO	Código de velocidade incorreto	Descarrilamento	Rotina de Verificação
	LATE	Antena quebrada	Descarrilamento	
	EARLY	Código zero	Parada antecipada	

FMEA

- **Failure Modes and Effects Analysis**
  Análise de Modos de Falha e Efeitos
- Etapas
  - Identificar todos componentes e seus modos de falha
  - Para cada modo de falha: determinar seus efeitos em outros componentes e no sistema
  - Cálculo da probabilidade dos resultados de cada modo de falha

FMEA

- Taxas de falhas de componentes
  - Obtidas a partir de informações publicadas por centros de pesquisa / fabricantes
  - Base em valores médios coletados de um grande número de componentes

FMEA

CARTÃO FILTRO ANALÓGICO				
COMPONENTE	MODO DE FALHA	EFEITO IMEDIATO DA FALHA	EFEITO DA FALHA NA SAÍDA	OBSERVAÇÕES
R1	Aberto	CI1 tem entrada presa em 1	Nenhum	Seguro, detectável
	Curto	CI1 tem entrada presa em 0	Não é gerado sinal de sincronismo	Inseguro, detectável
CI1	Entradas presas em 1	.....	.....	.....
	Entradas presas em 0	.....	.....	.....
.....	.....	.....	.....	.....

Análise de Erros Humanos

Análise de Erros Humanos

OPERAÇÃO DO MOTOR				
TAREFA	**PERIGO**	**CAUSA**	**EFEITO**	**MEDIDAS CORRETIVAS**
Desligar motor	Deixar motor ligado por muito tempo	Falha de alarme	Aquecimento anormal do motor	Implementar alarme
Acionar bomba	Acúmulo de água	Falha da bomba	Possibilidade de inundação	Acrescentar bomba de reserva
.....	.....	.....	.....	.....