

# PSI 2591

PROJETO DE FORMATURA I

7ª Aula

Decomposição

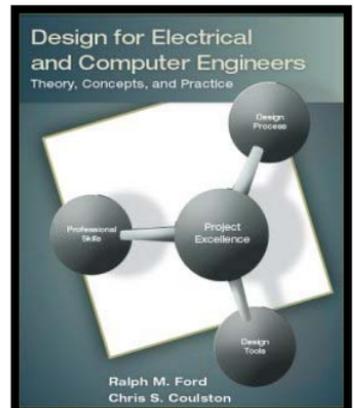
Funcional

2015



## Elaboração

- Prof. Sergio Takeo Kofuji
- Prof. Marcelo K. Zuffo
- Prof. Antonio C. Seabra
- Dra. Ramona M. Straube
  
- Livro Texto:





# MOTIVAÇÃO: DESIGN DE SISTEMAS

Times de engenheiros que constroem um Sistema necessitam de:

- Uma abstração do sistema
- Um meio de Comunicação consistente
- Uma maneira de descrever os subsistemas:
  - Entradas
  - Saídas
  - Comportamento
- Decomposição funcional
  - Função – transformação das entradas para as saídas
  - Decomposição – elaborar uma descrição através de módulos **tangíveis**

3



## Abordagens Bottom-Up e Top-Down

- Bottom-Up
  - Dados os elementos constituintes:
    - Desenvolver um Sistema que funcione
      - A partir de components, construir módulos para realizar tarefas específicas
      - Integrar módulos entre si formando um Sistema que funcione
    - Por exemplo
      - Dada uma oferta de portas E, OU e NÃO, construir um computador
    - Pros
      - Leva a um subsistema eficiente
      - É realista
      - Permite criatividade
    - Cons
      - A complexidade é difícil de gerenciar
      - Pouca preocupação em projetar módulos reutilizáveis
      - Ciclos de reprojeção difíceis

4

# Abordagens Bottom-Up e Top-Down

## • Top-Down

- Dada a especificação de um sistema

Desenvolver um Sistema que funcione

A partir dos requisitos, dividir o problema em módulos abstratos

Repetir o processo até obter partes tangíveis (“adquiríveis”)

Pros

Ciclo de projeto altamente previsível

Divisão eficiente de trabalho

Cons

Emprega mais tempo no planejamento

Pode barrar a criatividade (pensamento vertical, não lateral)

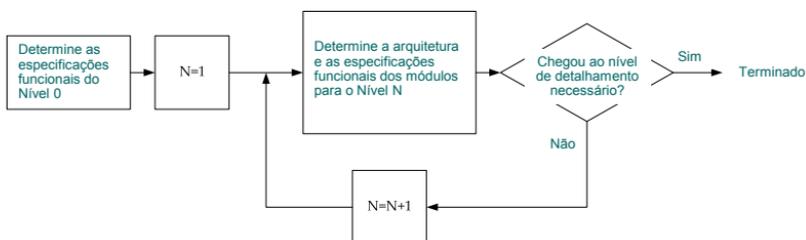
Nesta disciplina: Abordagem top-down com elementos bottom-up sempre que possível

5

# A Decomposição Funcional (Top-down) dos Requisitos de Engenharia

## • Dividir e conquistar de forma recursiva

- Divida um módulo em vários submódulos
- Defina a entrada, a saída e o comportamento  $\vec{y} = f(\vec{x})$
- Pare quando atingir components/blocos tangíveis



6

## Destaques

- O processo de projeto é iterativo
- Planejamento reduz o tempo de reprojeto posterior
- Submódulos devem possuir complexidade semelhante
- Especificações precisas de entradas, saídas, transformações (funcionalidades) e interconexões
- Busque inovação
- Não decomponha *ad infinitum*
- Utilize abstrações adequadas para descrever os módulos:
  - Não existe uma forma única de descrever os módulos. Quando necessário complemente a descrição funcional com fluxogramas, diagramas de estado, etc.
- Procure saber como costuma ser feito ou foi feito antes
- Utilize tecnologia existente/disponível
- Mantenha simples
- Comunique os resultados

7

## Aplicação em Projetos

- Projeto Nível 0
  - Apresente um módulo na forma de um único bloco com entradas e saídas identificadas (nomeadas) e com um título.
  - Apresente na forma de tabela os requisitos funcionais: entradas, saídas e funcionalidades.
- Projeto Nível 1
  - Apresente o diagrama do Nível 1 (arquitetura do sistema) com todos os módulos e interconexões bem visíveis.
  - Descreva a teoria de operação. Explique como os módulos trabalham juntos para alcançar os objetivos de funcionamento.
  - Apresente os requisitos funcionais na forma de tabela para cada modulo deste nível.
- Projeto Nível N (para  $N > 1$ )
  - Repita o processo empregado no Nível 1 tantas vezes quantas necessárias
- Alternativas de Projeto
  - Descreva as diferentes alternativas que foram consideradas, os compromissos (tradeoffs) e a justificativa para cada escolha. Baseie-se nos métodos de avaliação de conceitos (opções) apresentados na aula 5.

8

# A Decomposição Funcional dos Requisitos de Engenharia

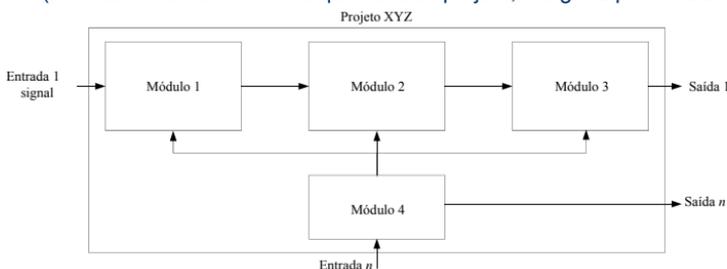
## Nível 0 (mais alto nível)



<b>Módulo</b>	Projeto
<b>Entradas</b>	Entrada 1, Entrada 2, Entrada $n$
<b>Saídas</b>	Saída 1, Saída 2, Saída $n$
<b>Funcionalidade</b>	Descrever em frases curtas ou figuras (diagramas de estado, de blocos, etc.) as ações ou transformações ou combinações que o módulo realiza com as informações vindas das entradas

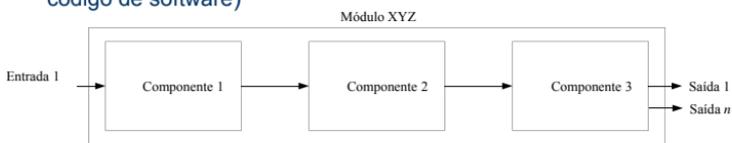
# A Decomposição Funcional dos Requisitos de Engenharia

## Nível 1 (normalmente descreve a arquitetura do projeto, em geral por módulos)



<b>Módulo n</b>	nome
<b>Entradas</b>	Entrada 1, Entrada 2, Entrada $n$
<b>Saídas</b>	Saída 1, Saída 2, Saída $n$
<b>Funcionalidade</b>	Descrever em frases curtas ou figuras (diagramas de estado, de blocos, etc.) as ações ou transformações ou combinações que o módulo realiza com as informações vindas das entradas
<b>Módulos Associados</b>	Indicar de quais módulos o Módulo n depende

**Nível 2** (em geral já descreve o detalhamento ao nível de componentes básicos ou código de software)



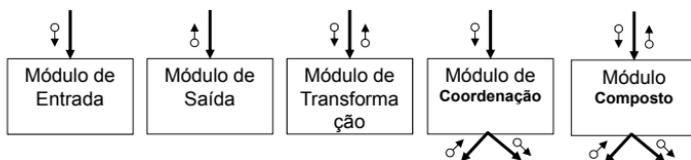
Submódulo ou componente n	nome
Entradas	Entrada 1, Entrada 2, Entrada n
Saídas	Saída 1, Saída 2, Saída n
Funcionalidade	Descrever em frases curtas ou figuras (diagramas de estado, de blocos, etc.) as ações ou transformações ou combinações que o módulo realiza com as informações vindas das entradas

## Nível n

- Chegue ao nível de detalhamento desejado e tangível (detailed design level)
- O número de níveis depende do detalhamento do projeto
- Não exagere no detalhamento, pare quando tiver algo tangível (software, bloco ou circuito)

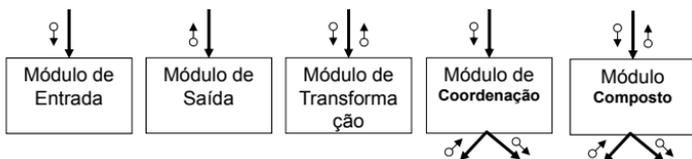
## Domínios de Aplicação

- Projetos Eletrônicos
- Projetos Digitais
- Projetos de Software (para linguagens funcionais ex. C)
  - Note que praticamente todas as linguagens de programação permitem a chamada de funções, subrotinas ou módulos
  - O projeto funcional simplifica o desenvolvimento de softwares, eliminando a necessidade de se criar códigos redundantes
  - Gráficos estruturados (structured charts) são diagramas de blocos específicos para visualizar Projetos de Software na forma funcional



# Domínios de Aplicação

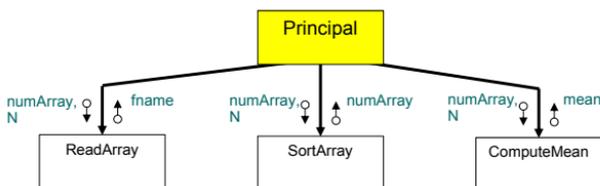
- Módulo de Entrada: Recebe informação
- Módulo de Saída: Retorna informação
- Módulo de Transformação: Recebe informação, a modifica e retorna a informação modificada
- Módulo de Coordenação: Coordena ou sincroniza as atividades entre módulos
- Módulos de Composição: Qualquer combinação dos quatro anteriores



13



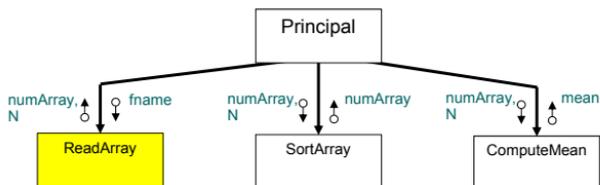
## Projeto de Software - Nível 1



<b>Módulo</b>	Principal
Tipo de Módulo	Coordenação
Argumentos de entrada	Nenhum
Argumentos de saída	Nenhum
Descrição	A função <b>principal</b> chama ReadArray() para ler o arquivo de entrada do disco, SortArray() para ler a matriz e ComputeMean() para determinar o valor médio dos elementos da matriz. É necessário a interação com o usuário para entrar o nome do arquivo. O valor da media é apresentado na tela.
Módulos invocados	ReadArray(), SortArray() e ComputeMean()

14

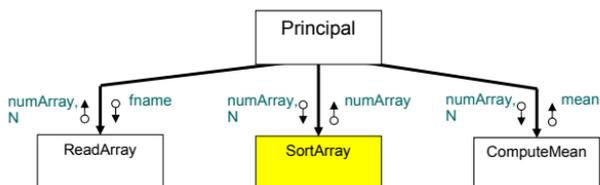
## Projeto de Software - Nível 1



<b>Módulo</b>	ReadArray()
Tipo de Módulo	Entrada e Saída
Argumentos de entrada	fname[]: matriz de caracteres com o nome do arquivo a ser lido
Argumentos de saída	numArray[]: matriz inteira com os elementos lidos do arquivo N: número de elementos de numArray[]
Descrição	Lê os dados do arquivo de dados de entrada e armazena os elementos na matriz numArray. O número de elementos lidos é colocado em N
Módulos invocados	Nenhum

15

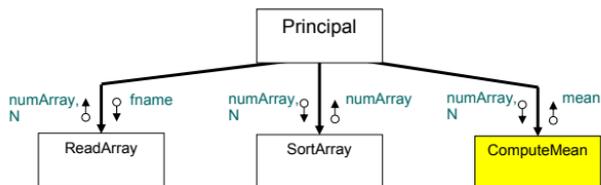
## Projeto de Software - Nível 1



<b>Módulo</b>	SortArray()
Tipo de Módulo	Transformation
Argumentos de entrada	numArray[]: matriz de números inteiros N: número de elementos de numArray[]
Argumentos de saída	numArray[]: matriz classificada de números inteiros
Descrição	Classifica os elementos da matriz usando um algoritmo de classificação. Armazena em disco a matriz classificada.
Módulos invocados	Nenhum

16

## Projeto de Software - Nível 1



<b>Módulo</b>	ComputeMean()
Tipo de Módulo	Entrada e Saída
Argumentos de entrada	numArray[]: matriz de números inteiros N: número de elementos de numArray[]
Argumentos de saída	mean: valor médio dos elementos na matriz
Descrição	Calcula o valor médio dos elementos inteiros da matriz
Módulos invocados	Nenhum

## Exemplo: O Termômetro Digital

# Requisitos de Engenharia e Conceito escolhido

O Sistema deve:

- Medir temperaturas entre 0 e 200°C.
- Possuir uma acurácia de 0.4% no fundo de escala
- Apresentar a temperatura digitalmente, com um dígito além do ponto decimal
- Ser alimentado por corrente alternada de 127V 60Hz
- Usar um RTD (dispositivo resistivo térmico) com acurácia de 0.55°C em toda a escala. A resistência do RTD varia linearmente com a temperature, de 100Ω a 0°C até 178Ω a 200°C

## Exemplo: 0 Termômetro Digital

### Nível 0

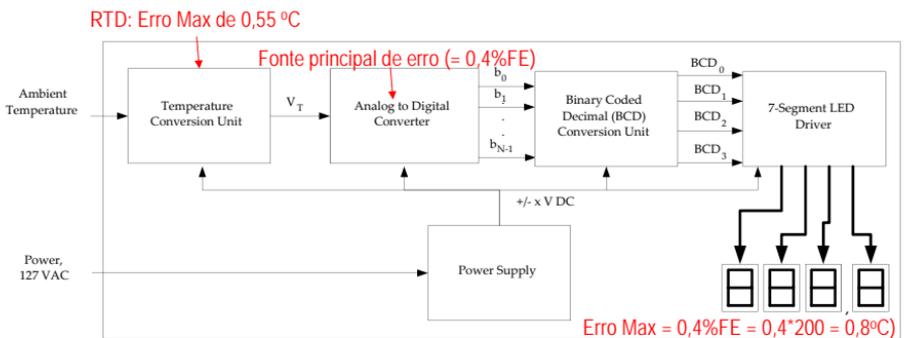
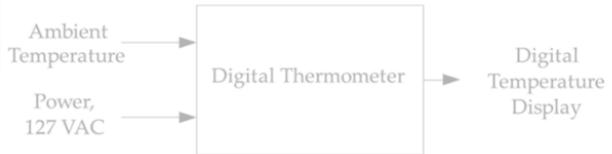


<i>Módulo</i>	Termômetro digital
<i>Entradas</i>	- Temperatura ambiente: 0-200°C. - Alimentação: 127V AC, 60Hz.
<i>Saídas</i>	- Mostrador digital de temperatura: Mostrador de 4 dígitos, com uma casa decimal.
<i>Funcionalidade</i>	Mostrar a temperatura para leitura digital com uma acurácia de 0.4% do fundo de escala.

19

## Exemplo: 0 Termômetro Digital

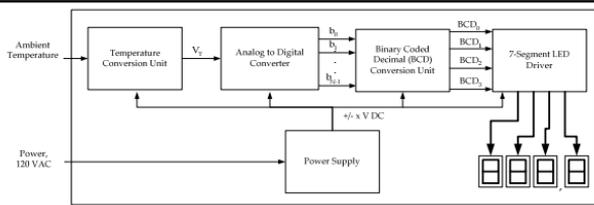
### Nível 1



20

## Exemplo: 0 Termômetro Digital

### Nível 1



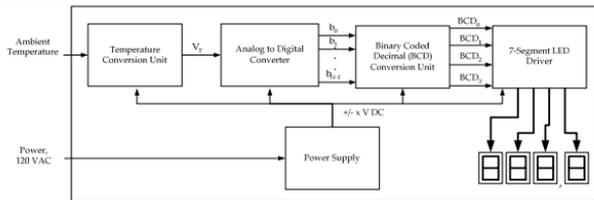
<i>Módulo</i>	Unidade de Conversão de Temperatura
<i>Entradas</i>	- Temperatura Ambiente: 0-200°C. - Alimentação: $\pm V$ DC (para alimentar a eletrônica).
<i>Saídas</i>	- $V_T$ : tensão proporcional à temperatura. $V_T = \alpha T$ , e vai de $\underline{V}$ à $\overline{V}$ .
<i>Funcionalidade</i>	Produz uma tensão de saída que é linearmente proporcional à temperatura. Precisa atingir uma acurácia de $\underline{\%}$ .

RTD: Erro Max de 0,55 °C  
 Erro estimado (razoável) para a UCT: 0,05 °C.

21

## Exemplo: 0 Termômetro Digital

### Nível 1



<i>Módulo</i>	Conversor A/D
<i>Entradas</i>	- $V_T$ : tensão proporcional à temperatura, variando de $\underline{V}$ à $\overline{V}$ . - Alimentação: $\pm V$ DC.
<i>Saídas</i>	- Representação binária de $\underline{\text{bits}}$ ( $b_{N-1} - b_0$ ) de $V_T$ .
<i>Funcionalidade</i>	Converte uma entrada analógica à uma saída digital binária.

RTD: Erro Max de 0,55 °C; UTC: Erro Max 0,05 °C  
 ADC Erro Max ? (Total = 0,4%FE)  
 Erro Max Total = 0,4%FE =  $0,4 \cdot 200 = 0,8^\circ\text{C}$

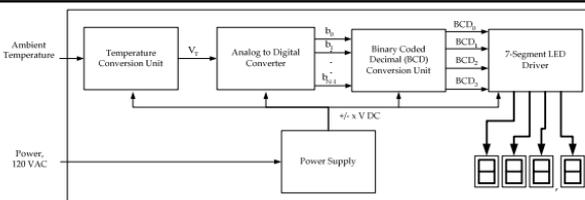
ADC Erro de 0,20°C

$$\text{ErroMax ADC} = \frac{\text{faixa}}{\text{número de intervalos}} = \frac{200^\circ\text{C}}{2^N} \geq 0,25^\circ\text{C} \Rightarrow N \geq 9,97 \text{ bits}$$

22

## Exemplo: O Termômetro Digital

### Nível 1



<i>Módulo</i>	BCD ... LED Driver ... Power Supply (IDEM)
<i>Entradas</i>	
<i>Saídas</i>	
<i>Funcionalidade</i>	

Com base na apreciação geral dos módulos, definir alimentação DC e seu ripple (5V?)

23

## Exemplo: O Termômetro Digital

### Detalhes de Projeto

- Como você determinaria os detalhes desconhecidos dos dois slides anteriores?

Exemplo: Com base na apreciação geral dos módulos, definir alimentação DC e seu ripple (5V?)

24

# Acoplamento e Coesão

## • O que é acoplamento?

- Considere o número de módulos no nível mais baixo e o número de conexões entre eles. P.ex, se dois módulos, no máximo 1 conexão (se 3, 3; se 4, 6; se 5, 10 – pense nisso)
- Portanto o número máximo de conexões cresce vertiginosamente com o número de módulos:

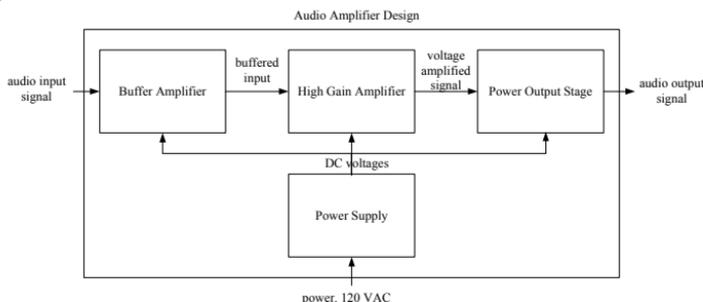
$$\text{Conexões}_{MAX} = \frac{n(n-1)}{2}$$

- Acoplamento indica até que ponto os módulos estão conectados entre si:
  - Sistemas altamente acoplados indicam que erro em um módulo impacta diretamente nos outros e torna a identificação do erro difícil

25

# Acoplamento

- O acoplamento dos módulos do Nível 1 do amplificador abaixo é relativamente baixo (excluindo-se a fonte)
- Note que os módulos não são desacoplados, uns dependem das impedâncias de entrada/saída dos outros



- Características de sistemas altamente acoplados
  - Falha em um módulo se propaga a outros
  - Difícil de reprojeter um módulo
- Características de sistemas pouco acoplados
  - Desencoraja a reutilização de módulos

26

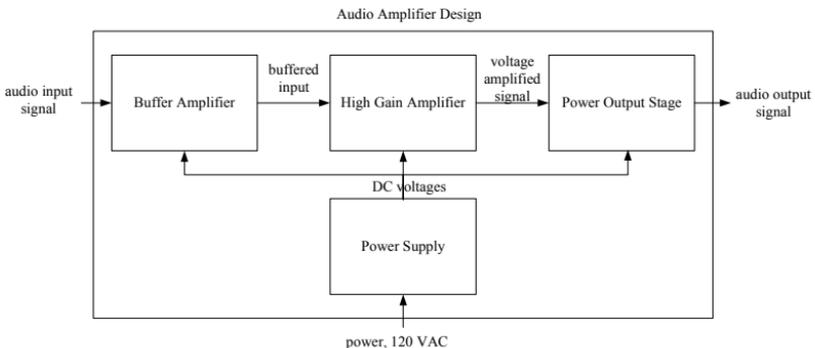
# Coesão

- O que é coesão?
  - Indica quão focado um módulo é. Em geral quanto mais coeso, menos acoplamento no Sistema
  - Existem tipos de coesão: lógica, temporal, funcional, etc.
- Características de um Sistema altamente coeso
  - Fácil de testar os módulos de maneira independente
  - Interface de controle simples (ou não existente)
- Características de um Sistema pouco coeso
  - Menos reuso dos módulos

27

# Acoplamento

- Quanta coesão existe nos módulos do Nível 1 do amplificador abaixo?



- O Sistema acima é altamente coeso, cada modulo realiza uma etapa específica da amplificação.
- Note que cada modulo poderia ser utilizado sozinho em outras aplicações

28



## Resumo

- Abordagens de Projeto: top-down e bottom-up
- Decomposição Funcional (mais top-down)
  - Decomposição iterativa
  - Entrada, saída e função
  - Aplicável a muitos domínios de problemas
- Acoplamento – interconectabilidade dos módulos
- Coesão – foco dos módulos