

# Arquitetura e Organização de Computadores (EIC0083)

Apresentação da unidade curricular

João Canas Ferreira

Setembro de 2019



# Tópicos

- 1 Computadores
- 2 Arquitetura e Organização de Computadores
- 3 Temas da unidade curricular
- 4 Funcionamento da unidade curricular

- 1 Computadores
- 2 Arquitetura e Organização de Computadores
- 3 Temas da unidade curricular
- 4 Funcionamento da unidade curricular

# Computadores não foram sempre automáticos



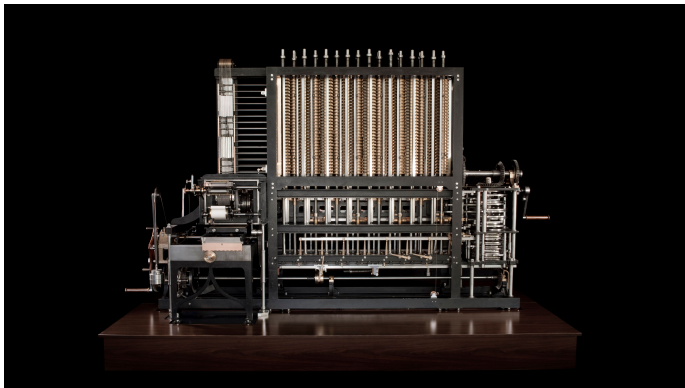
Dryden Flight Research Center E49-0053 Photographed 10/49  
Early "computers" at work. NASA photo



<https://www.dfrc.nasa.gov/Gallery/Photo/People/Medium/E49-0053.jpg>

# Computadores automáticos: o início

- ➡ *Difference engine* de Charles Babbage (1791–1871) terminado em 2002 (153 anos depois de projetado).



- ➡ 8000 peças; 5 toneladas; 3,4 m de largura.

# Computador automático

➡ O computador automático é uma máquina.

- Executa cálculos (processa informação)
- Para qualquer fim (*general purpose*)
- Segue uma receita (programa) que pode ser alterada.

## **A funcionalidade não é fixa**

- **“Programa” também é informação!**
- Pode ser realizado em várias tecnologias
  - mecânica (p. ex., mecanismos de rodas dentadas)
  - válvulas
  - transístores, etc.

O computador automático é uma máquina que *recebe, trata* (processa) e *produz* **informação**.

Para isso, também necessita de *preservar* informação (ter “memória”).

# A primeira programadora



Augusta Ada King-Noel,  
Countess of Lovelace  
(1815–1852)

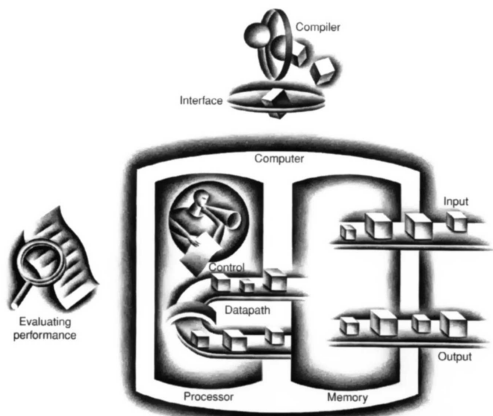
[https://en.wikipedia.org/  
wiki/Ada\\_Lovelace](https://en.wikipedia.org/wiki/Ada_Lovelace)

- 1 Computadores
- 2 **Arquitetura e Organização de Computadores**
- 3 Temas da unidade curricular
- 4 Funcionamento da unidade curricular



# Componentes clássicos de um computador

## The Five Classic Components of a Computer



1. Entradas
2. Saídas
3. Memórias
4. Percursos de Dados  
(barramentos)
5. Controlo  
(sequência de operações)

*Obs: Processador=4+5*

Fonte:  
Computer Organization  
and Design --  
-- Patterson & Hennessy

# Arquitetura / Organização

- **arquitetura:** descrição das capacidades e do modelo de programação de um computador (p. ex., a definição do seu conjunto de instruções), mas não de uma implementação em particular. É constituída apenas por atributos *visíveis* ao programador.

Descrição abstrata **do que é que o computador faz.**

- **organização:** descrição de uma implementação detalhada do computador (da sua estrutura funcional e respetivo comportamento durante execução do programa). Também designada por *microarquitetura*.

Descreve **como** o computador realiza a arquitetura por interligação de vários componentes.

- ➡ Exemplo: Nehalem e Haswell são nomes de código de duas microarquiteturas diferentes da arquitetura Intel 64.
- ➡ Em teoria, arquitetura e organização são independentes. Na prática, isso não acontece sempre, o que leva alguns autores a não dar grande ênfase à distinção.

- 1 Computadores
- 2 Arquitetura e Organização de Computadores
- 3 Temas da unidade curricular
- 4 Funcionamento da unidade curricular

# Resumo do programa (1/2)

▶▶▶ Como vamos abordar os assuntos?

# Resumo do programa (1/2)

▶▶▶ Como vamos abordar os assuntos?

▶ Dos componentes para os sistemas (complexidade crescente)

# Resumo do programa (1/2)

## ► Como vamos abordar os assuntos?

- ☞ Dos componentes para os sistemas (complexidade crescente)
- ☞ Usando modelos apropriados a cada nível de complexidade

# Resumo do programa (1/2)

## ▢▢▢▢ Como vamos abordar os assuntos?

- ▢▢ Dos componentes para os sistemas (complexidade crescente)
- ▢▢ Usando modelos apropriados a cada nível de complexidade

## 1 Representação de informação

- ▢▢▢▢ Como é representada a informação que os computadores recebem, processam e produzem?

# Resumo do programa (1/2)

## ▢▢▢▢ Como vamos abordar os assuntos?

- ▢▢ Dos componentes para os sistemas (complexidade crescente)
- ▢▢ Usando modelos apropriados a cada nível de complexidade

### 1 Representação de informação

- ▢▢▢▢ Como é representada a informação que os computadores recebem, processam e produzem?

### 2 Circuitos lógicos combinatórios

- ▢▢▢▢ Como é realizado o processamento da informação? (modelo lógico)



# Resumo do programa (1/2)

## ►►► Como vamos abordar os assuntos?

- ☞ Dos componentes para os sistemas (complexidade crescente)
- ☞ Usando modelos apropriados a cada nível de complexidade

### 1 Representação de informação

- Como é representada a informação que os computadores recebem, processam e produzem?

### 2 Circuitos lógicos combinatórios

- Como é realizado o processamento da informação? (modelo lógico)

### 3 Circuitos lógicos sequenciais

- O que é o tempo (para um sistema lógico)?
- Como é que os sistemas lógicos se “lembram” da informação?
- Como é que os resultados obtidos podem influenciar resultados posteriores?

## Resumo do programa (2/2)

### 4 Modelo básico conceptual de um computador

➡ O que é um programa? O que significa “executar um programa”?

## Resumo do programa (2/2)

- ④ Modelo básico conceptual de um computador
  - ▢➡ O que é um programa? O que significa “executar um programa”?
- ⑤ Desempenho
  - ▢➡ O que é o desempenho de um processador? Como avaliá-lo?

## Resumo do programa (2/2)

- 4 Modelo básico conceptual de um computador
  - ▢ O que é um programa? O que significa “executar um programa”?
- 5 Desempenho
  - ▢ O que é o desempenho de um processador? Como avaliá-lo?
- 6 Conjunto de instruções
  - ▢ O que são as instruções de um processador? Que informação é que contém?

## Resumo do programa (2/2)

- 4 Modelo básico conceptual de um computador
  - ➡ O que é um programa? O que significa “executar um programa”?
- 5 Desempenho
  - ➡ O que é o desempenho de um processador? Como avaliá-lo?
- 6 Conjunto de instruções
  - ➡ O que são as instruções de um processador? Que informação é que contém?
- 7 Programação nativa
  - ➡ Como especificar programas na “linguagem” do processador?

## Resumo do programa (2/2)

- 4 Modelo básico conceptual de um computador
  - ▢ O que é um programa? O que significa “executar um programa”?
- 5 Desempenho
  - ▢ O que é o desempenho de um processador? Como avaliá-lo?
- 6 Conjunto de instruções
  - ▢ O que são as instruções de um processador? Que informação é que contém?
- 7 Programação nativa
  - ▢ Como especificar programas na “linguagem” do processador?
- 8 Organização lógica de uma unidade de processamento
  - ▢ Como se pode realizar um processador com circuitos lógicos?

## Resumo do programa (2/2)

- 4 Modelo básico conceptual de um computador
  - ▢ O que é um programa? O que significa “executar um programa”?
- 5 Desempenho
  - ▢ O que é o desempenho de um processador? Como avaliá-lo?
- 6 Conjunto de instruções
  - ▢ O que são as instruções de um processador? Que informação é que contém?
- 7 Programação nativa
  - ▢ Como especificar programas na “linguagem” do processador?
- 8 Organização lógica de uma unidade de processamento
  - ▢ Como se pode realizar um processador com circuitos lógicos?
- 9 Memória de computadores
  - ▢ Que tipos de circuitos de memória existem?
  - ▢ Como é que os circuitos de memórias influenciam o desempenho?

- 1 Computadores
- 2 Arquitetura e Organização de Computadores
- 3 Temas da unidade curricular
- 4 Funcionamento da unidade curricular



# Métodos de ensino e atividades de aprendizagem

- ▀ A unidade curricular (UC) tem uma componente teórica baseada em aulas de exposição dos diversos temas que serão acompanhados da apresentação de exemplos e respetiva discussão.

# Métodos de ensino e atividades de aprendizagem

■▶ A unidade curricular (UC) tem uma componente teórica baseada em aulas de exposição dos diversos temas que serão acompanhados da apresentação de exemplos e respetiva discussão.

■▶ As aulas teórico-práticas (TP) incluem a apresentação, análise e resolução de um conjunto de questões e de casos de estudo.

■▶ Outras atividades de aprendizagem a realizar fora do período de aulas:  
Questionários de escolha múltipla sobre os diferentes assuntos.

Os questionários destinam-se **unicamente** à auto-avaliação e não contam para a avaliação final da UC.

# Avaliação

## ► Obtenção de frequência:

- Participação em, pelo menos, 75 % das aulas TP (número de faltas às aulas TP  $\leq 3$ )

► Só podem **obter aprovação** os estudantes que com frequência este ano ou no ano passado!

# Avaliação

## Obtenção de frequência:

- Participação em, pelo menos, 75 % das aulas TP (número de faltas às aulas TP  $\leq 3$ )

➡ Só podem **obter aprovação** os estudantes que com frequência este ano ou no ano passado!

## Avaliação escrita:

- Dois (2) testes (90 minutos cada)

# Avaliação

## ► Obtenção de frequência:

- Participação em, pelo menos, 75 % das aulas TP (número de faltas às aulas TP  $\leq 3$ )

## ► Só podem **obter aprovação** os estudantes que com frequência este ano ou no ano passado!

## ► Avaliação escrita:

- Dois (2) testes (90 minutos cada)

## ► Classificação final

- $N_{Final} = 0,5 \times T1 + 0,5 \times T2$

# Prova de repescagem

- ▶ Prova destinada exclusivamente a estudantes que obtiveram nota final  $< 10$  valores (após arredondamento).
- ▶ Podem ser repescadas **apenas** as componente T1 e/ou T2 com nota  $< 9,5$ .
- ▶ Em cada componente da repescagem (P1 e P2) a nota máxima será de 9,5 valores (em 20), substituindo a nota obtida anteriormente nessa componente.
- ▶ Esta prova não se destina a melhoria de classificação (estudantes que já obtiveram nota final  $\geq 10$  valores).

# Casos especiais de avaliação

## ■ **Faltas justificadas a componentes de avaliação**

Estudantes que tenham atempadamente apresentado justificação válida para eventuais faltas a alguma das componentes de avaliação, e desde que essas faltas sejam consideradas justificadas pela Direcção do Curso, poderão submeter-se a provas de avaliação de substituição em data a estabelecer pelos docentes.

## ■ **Formas de avaliação de casos especiais (TE, DA, ....)**

Estudantes dispensados da presença nas aulas teórico-práticas deverão obrigatoriamente realizar as provas de avaliação indicadas.

## ■ **Melhoria de classificação**

Como se trata de uma unidade curricular (UC) de avaliação distribuída sem exame final, a melhoria de classificação é possibilitada através da realização dos testes da UC no ano letivo seguinte.

Neste caso, a classificação final é calculada pela média aritmética das classificações dos testes.

## Equipa docente

► Os docentes estão disponíveis para esclarecer quaisquer dúvidas sobre a matéria e funcionamento da unidade curricular.

■ João Canas Ferreira	E-mail: <a href="mailto:jcf@fe.up.pt">jcf@fe.up.pt</a>	gabinete I 237
Horário de atendimento:		<b>quinta-feira, 14:30–18:00</b>
■ António Duarte Araújo	E-mail: <a href="mailto:aja@fe.up.pt">aja@fe.up.pt</a>	gabinete I 236
■ Bruno Miguel C. Lima	E-mail: <a href="mailto:bruno.lima@fe.up.pt">bruno.lima@fe.up.pt</a>	Lab I 122
■ Helder Avelar	E-mail: <a href="mailto:h2avelar@gmail.com">h2avelar@gmail.com</a>	Lab I 223
■ Daniel Granhão	E-mail: <a href="mailto:daniel.granhao@fe.up.pt">daniel.granhao@fe.up.pt</a>	Lab I 223

► Para marcar eventuais reuniões para esclarecimento fora dos horários de atendimento, os estudantes devem contactar diretamente o docente por e-mail.



# Elementos de apoio

## ➡ Tirar apontamentos!

➡ Livro: David A. Patterson, John L. Hennessy, Computer Organization and Design – ARM Edition, Elsevier, 2016, ISBN: 9780128017333

<http://booksite.elsevier.com/9780128017333/>

➡ Cópias das apresentações (pasta de conteúdos)

➡ Caderno de exercícios resolvidos e exercícios propostos (pasta de conteúdos)

➡ Simulador de circuitos digitais:

<https://github.com/hneemann/Digital>

➡ *Assembler*/emulador ARM: Development Studio 5 Community Edition

<http://bit.ly/2Dt2Ydw>