

Arquitetura de Von Neumann

Desenvolvimento de Sistemas Ciber Físicos

Felipe Krupa Sokulski

Henrique de Oliveira Zadereski

João Pedro Schmidt Dias

Nicolas Oswaldo Letti

Otavio Graczyki Belich

Theo Monteiro

Tiago Augusto Almeida Miglioli

Victor Henrico Schroeder



O Gênio que Reinventou a Computação: John von Neumann:

A Revolução Silenciosa:

Imagine uma calculadora que só faz multiplicação. Se você quiser somar, precisa construir outra máquina só para isso.

Agora você tem uma calculadora que pode fazer tudo – multiplicar, somar, subtrair – só trocando o "programa", sem mexer nas peças.

→ Base Científica Sólida:

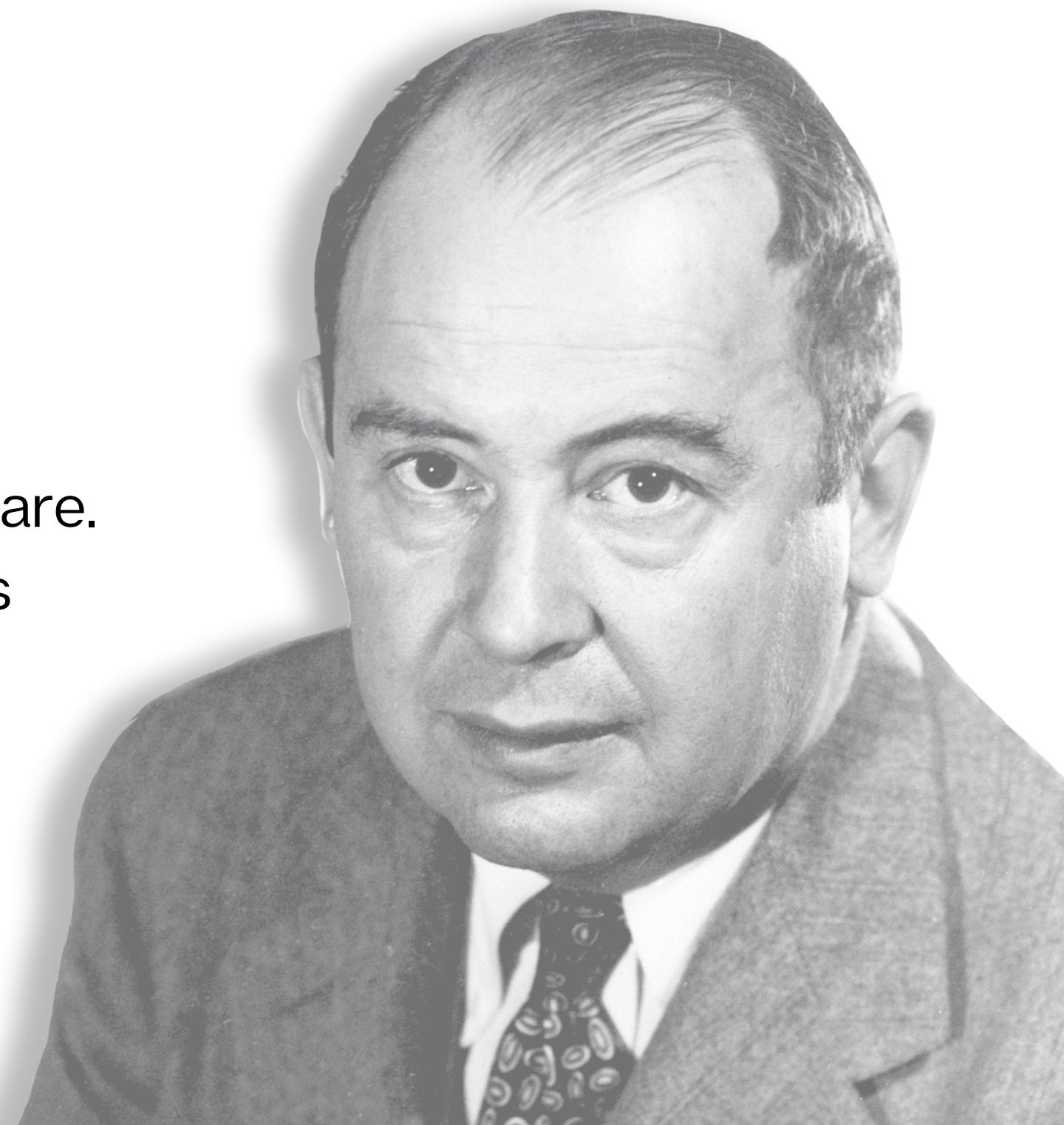
- 1945: "First Draft of a Report on the EDVAC" – Introduz a ideia de programa armazenado, permitindo reprogramação sem alterar o hardware.
- 1982 (Randell): Mostra como sua ideia sepultou máquinas como o ENIAC, que necessitava de recabeamento manual.

Analogias e Impacto:

- Steve Jobs da Computação: Não inventou o transistor, mas organizou componentes para criar algo revolucionário.
- Arquitetura Von Neumann = "iPhone" da computação moderna.

Por Que Isso Importa?

- **Flexibilidade radical:** Reprogramação sem alterar o hardware.
- **Implicações modernas:** Permitiu o surgimento de sistemas operacionais e IA generativa, como o ChatGPT.

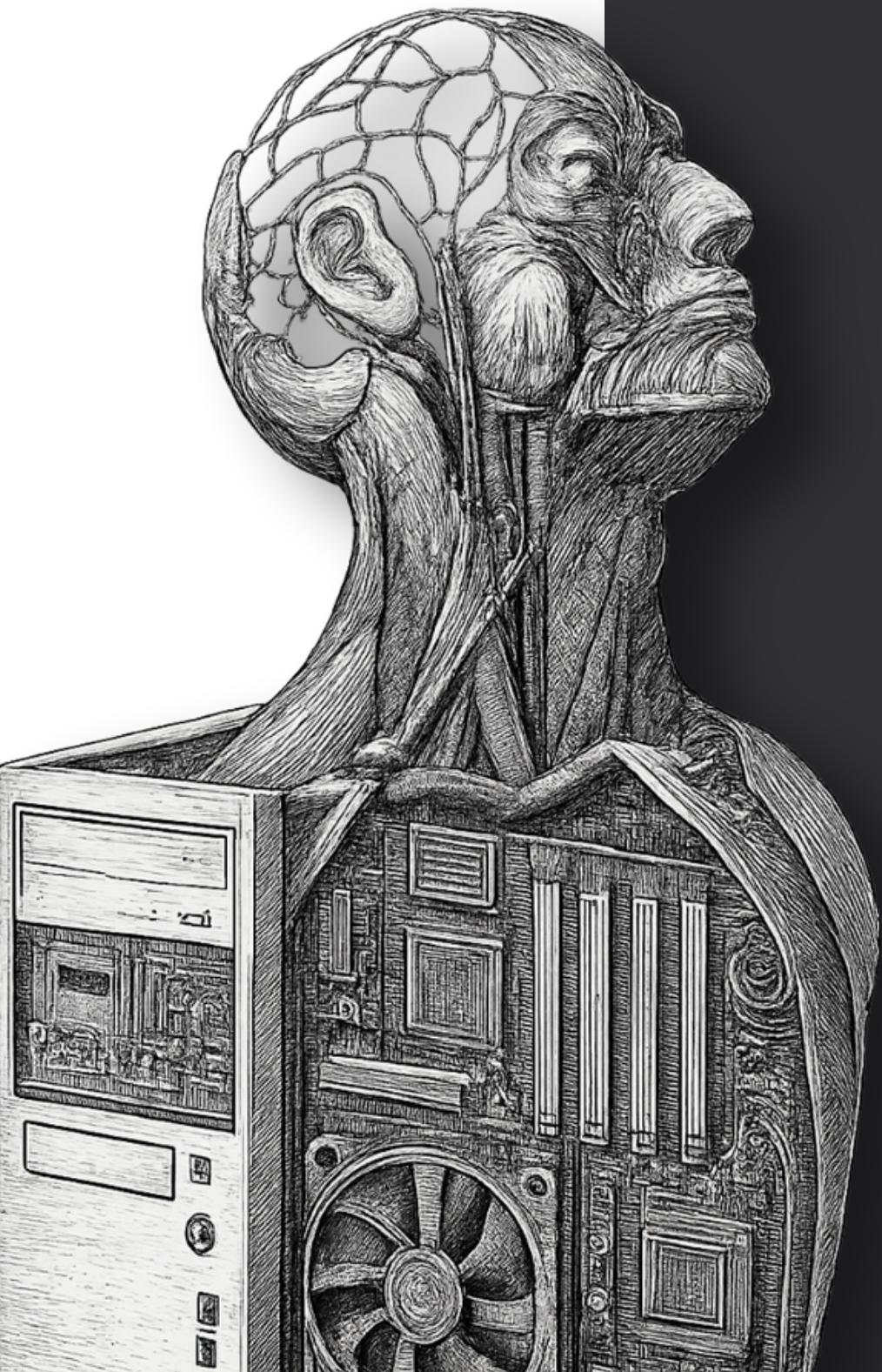


A Anatomia do Computador

Os 4 componentes:

Base Científica Avançada:

- * **Hennessy & Patterson (2017):**
Separação CPU/memória maximiza eficiência (prova matemática).
- * **Shiva (2008):**
Diferença prática: Arduino (Harvard) ≠ PC (Von Neumann).



Computador segundo Von Neumann

Um Computador Von Neumann é como um Restaurante 3 Estrelas Michelin:

CPU: Processador

- Chef executivo (decide, processa, cria)
- Executa decisões e cálculos

E/S: Comunicação externa

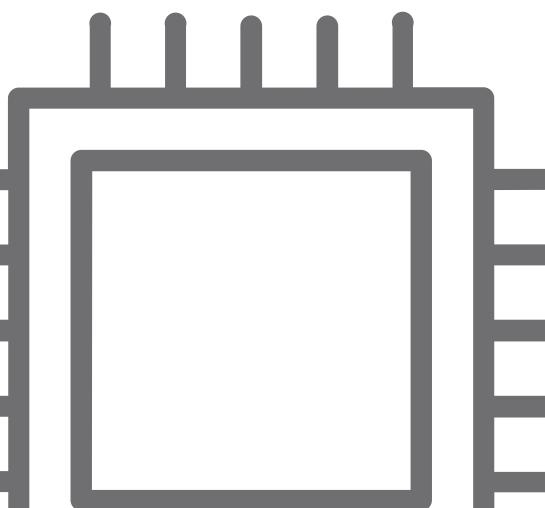
- Realiza a comunicação externa
- Garçons (entrada = pedidos, saída = pratos)

Memória: Armazena

- Dispensa (guarda ingredientes/dados)
- Armazena dados e instruções

Barramento: Comunicação interna

- Esteira que conecta tudo
- Esteira que conecta tudo



Impacto e Relevância Técnica

Porque isso é relevante?



Sistema Operacional: Gerência da Anatomia

- Administra múltiplos processos, memória e E/S
- Garante sincronização, escalonamento e isolamento entre programas
- Uma falha de gestão → deadlocks, perda de dados, corrupção

Segurança: Anatomia como superfície de ataque

- Vulnerabilidades como buffer overflow exploram a estrutura interna (pilha, heap, registradores)
- Má proteção anatômica → acesso indevido, execução arbitrária de código

Barramento:

→ O que é o barramento?

Linhas físicas que conectam CPU, RAM e periféricos;

Divisão em 3 partes:

- Barramento de Dados:
- Barramento de endereço:
- Barramento de controle:

Gargalo de Von Neumann: CPU espera dados da RAM via barramento compartilhado

→ CPU ociosa em aplicações intensivas da memória;

Soluções:

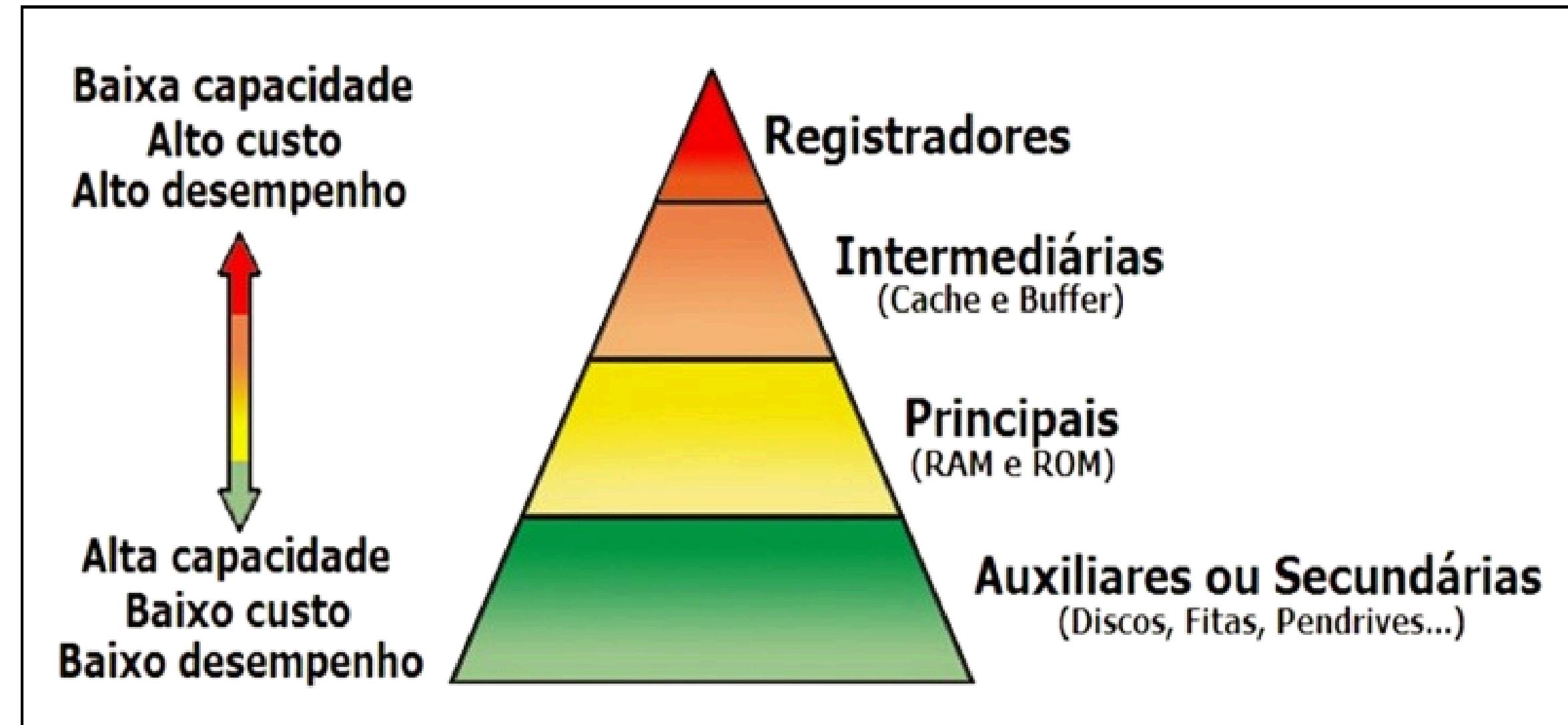
- Memórias cache (L1/L2/L3);
- Barramentos dedicados / multiplexação;
- Processamento na Memória (PIM);

Hierarquia de Memória:

- 1. Registradores
- 2. Cache
- 3. RAM
- 4. SSD/SD

→ Funções:

- 1. Reduzir o Gargalo de Von Neumann
- 2. Balancear Custo e Desempenho
- 3. Aproveitar a Localidade de Referência
- 4. Otimizar o Acesso à Memória

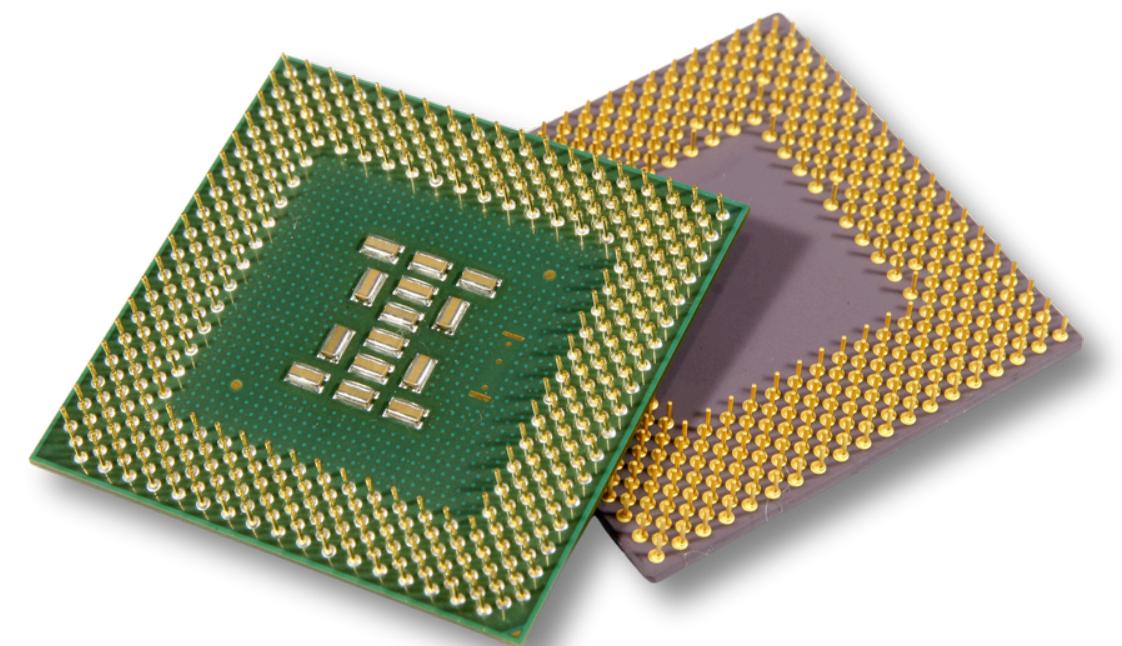


“Porque o computador se esquece?”

“Porque meu computador trava se acaba a RAM?”

CPU: Por que uma de 5GHz pode ser mais lenta que uma de 3GHz?

- Nem sempre uma CPU com frequência maior é mais rápida. O desempenho real depende da eficiência da arquitetura, e não só da velocidade do clock.
- Partes principais da CPU:
 - ULA (Unidade Lógica e Aritmética): executa operações como somas, comparações e cálculos.
 - Unidade de Controle: coordena o que será executado, em que ordem e por quais partes.
 - Pipeline: divide o processamento em etapas (buscar, decodificar, executar, etc.) e executa várias instruções em paralelo.



Problemas que afetam a performance:

- Hazards: quando uma instrução depende de outra ainda em execução.
- Branch misprediction: erro na previsão de um desvio (ex: if), que obriga a CPU a descartar instruções e reiniciar – perdendo tempo.
- Speculative execution: a CPU apostava em instruções futuras. Se errar, há desperdício de ciclos;

Programadores podem otimizar isso:

- Linguagens compiladas como C++ aproveitam melhor o pipeline.
- Técnicas como loop unrolling e branchless programming ajudam a reduzir desvios e manter a CPU eficiente.

Conclusão:

Mais GHz não significa mais desempenho. O que importa é o aproveitamento do pipeline e a previsibilidade do código – como um Neymar que pensa mais rápido do que corre.



Limitações: O Gargalo Que Ainda Nos Assombra

“De que adianta um supercérebro (CPU) se ele depende de um sistema nervoso lento (memória)?”

→ O Problema Matemático Por Trás do Gargalo

Lei de Amdahl: A melhoria de um sistema é limitada pela parte que não pode ser melhorada.

- P : proporção do tempo de execução.
- S : quanto essa parte foi acelerada

$$(1 - P) + \frac{P}{S}$$

O Que Isso Significa Para Nós?

- Carreira: Especialistas em otimização de memória.
- Pesquisa: Arquitetura de memória, chips híbridos, e novas tecnologias



Programa Armazenado:

→ O que é o programa armazenado?

- O programa é executado junto com os outros dados normais dentro da memória, ao invés de ter somente um uso

Exemplo:

Antes o computador tinha uma função específica como uma calculadora, após esse conceito aplicado se possibilitou a reprogramação e a execução de qualquer coisa

→ O que possibilitou/aplicações:

- Versatilidade
- O computador se tornou totalmente moldável
- Celulares, PCs, IAs, Apps e etc

A screenshot of the Microsoft Visual Studio Code (VS Code) interface. The main area shows a Python file named 'unit_test.py' with the following code:

```
class Test_TestIncrementDecrement(unittest.TestCase):
    def test_increment(self):
        self.assertEqual(test.increment(3), 4)
    # This test is designed to fail for demonstration purposes.
    def test_decrement(self):
        self.assertEqual(test.decrement(3), 4)
```

The status bar at the bottom indicates 'Python: Current File (VSCode-DataCamp)' and 'Jupyter'. On the right, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'JUPYTER'. The 'TERMINAL' tab shows the command 'Test run finished at 2/1/2023, 9:38:23 PM <'. The 'OUTPUT' tab shows 'Finished running tests!'. The 'DEBUG CONSOLE' tab shows 'Ln 11, Col 1 (31 selected) Spaces'.

Conclusão: Por Que Você Deveria Se Importar?

Todas as linguagens de programação seguem esse modelo (inclusive Python, que opera em uma máquina virtual).

Gargalos de desempenho em jogos, IA e bancos de dados originam-se aqui.

Segurança cibernética moderna (e seus ataques) é fruto dessa arquitetura.

"Von Neumann não é só um capítulo da história – é o DNA de toda computação . Entendê-lo não faz de você um espectador da revolução digital... faz de você um arquiteto do futuro ."

Perguntas:

1. Por que a ideia de programa armazenado foi uma revolução para a computação?

- a) Porque permitiu que os computadores ficassem menores
- b) Porque passou a usar energia elétrica
- c) Porque tornou possível mudar o que o computador faz sem trocar o hardware
- d) Porque acabou com o uso de fios nos circuitos

Perguntas:

1. Por que a ideia de programa armazenado foi uma revolução para a computação?

- a) Porque permitiu que os computadores ficassem menores
- b) Porque passou a usar energia elétrica
- c) Porque tornou possível mudar o que o computador faz sem trocar o hardware**
- d) Porque acabou com o uso de fios nos circuitos

Perguntas:

2. Como a memória influencia na velocidade com que um computador executa tarefas?

- a) Quanto menor a memória, mais rápido o computador
- b) Ela decide qual programa será aberto
- c) Quanto mais rápida a memória, mais rápido a CPU acessa os dados
- d) A memória serve apenas para armazenar fotos e vídeos

Perguntas:

2. Como a memória influencia na velocidade com que um computador executa tarefas?

- a) Quanto menor a memória, mais rápido o computador
- b) Ela decide qual programa será aberto
- c) Quanto mais rápida a memória, mais rápido a CPU acessa os dados**
- d) A memória serve apenas para armazenar fotos e vídeos

Perguntas:

3. O que é o “gargalo de Von Neumann”?

- a) Um problema causado pelo superaquecimento da CPU
- b) Um limite na comunicação entre a CPU e a memória
- c) Um erro no sistema operacional
- d) Um tipo de vírus que afeta computadores antigos

Perguntas:

3. O que é o “gargalo de Von Neumann”?

- a) Um problema causado pelo superaquecimento da CPU
- b) Um limite na comunicação entre a CPU e a memória**
- c) Um erro no sistema operacional
- d) Um tipo de vírus que afeta computadores antigos

Referências:

<https://olhardigital.com.br/2024/09/16/reviews/gargalo-de-von-neumann-o-que-e-e-por-que-ele-e-importante/#:~:text=Como%20resolver%20o%20Gargalo%20de,uma%20capacidade%20de%20armazenamento%20limitada.&text=Em%20muitos%20processadores%20comuns%2C%20encontramos,assim%20não%20superar%20o%20gargalo.>

<https://www.youtube.com/watch?v=tZ5W2LpdcEw>

<https://canaltech.com.br/hardware/o-que-e-arquitetura-de-von-neumann/>

[https://www.tme.com.br/pt/news/library-articles/page/56104/arquitetura-de-von-neumann-e-arquitetura-de-harvard/#:~:text=A%20arquitetura%20de%20von%20Neumann%20\(%20assim%20chamada%20em%20homenagem%20a,a%20velocidade%20de%20funcionamento%20limitada.](https://www.tme.com.br/pt/news/library-articles/page/56104/arquitetura-de-von-neumann-e-arquitetura-de-harvard/#:~:text=A%20arquitetura%20de%20von%20Neumann%20(%20assim%20chamada%20em%20homenagem%20a,a%20velocidade%20de%20funcionamento%20limitada.)

**Agradecemos a
atenção!**