ORGANIZAÇÃO E MANUTENÇÃO DE COMPUTADORES

Versão 0.1

JOÃO PAULO FERREIRA GUIMARÃES

Organização e Manutenção de Computadores — Versão 0.1
João Paulo Ferreira Guimarães
2026

Sumário

1 Fundamentos: o que é um computador

Neste capítulo você vai estudar a definição técnica de computador, sua evolução histórica desde os primeiros instrumentos de cálculo até o computador pessoal moderno, o conceito de modularidade que sustenta toda a disciplina de manutenção, e uma primeira introdução à hierarquia de memória.

1.1 Definição de computador

O termo *computador* deriva do verbo *computar*, originado do latim *calculus* (“pedrinha”), em referência ao uso histórico de pedrinhas para realizar contagens. Essa origem revela o sentido amplo do termo: em sua acepção mais larga, um computador é qualquer instrumento que auxilia a realizar cálculos — o que inclui dispositivos como o ábaco ou uma calculadora simples.

Essa definição ampla, no entanto, é insuficiente para descrever o computador digital moderno. Este livro adota a definição proposta pelo pesquisador Andrew Tanenbaum:

> “O computador digital é uma máquina que pode resolver problemas para as pessoas executando instruções que lhes são dadas.” [1]

Essa definição contém três elementos essenciais:

- **Máquina** — o computador é, antes de tudo, um dispositivo físico (hardware).
- **Resolve problemas de forma genérica** — diferentemente de uma calculadora, que executa um conjunto fixo de operações, o computador recebe um programa e pode, em princípio, resolver qualquer problema computável.
- **Executa instruções que lhe são dadas** — o software é ele próprio um dado de entrada, e não uma característica fixa do hardware.

Aplicação da definição. Uma calculadora de padaria realiza operações de soma, subtração, multiplicação e divisão, mas não pode receber um programa genérico — não é possível, por exemplo, instalar nela um aplicativo

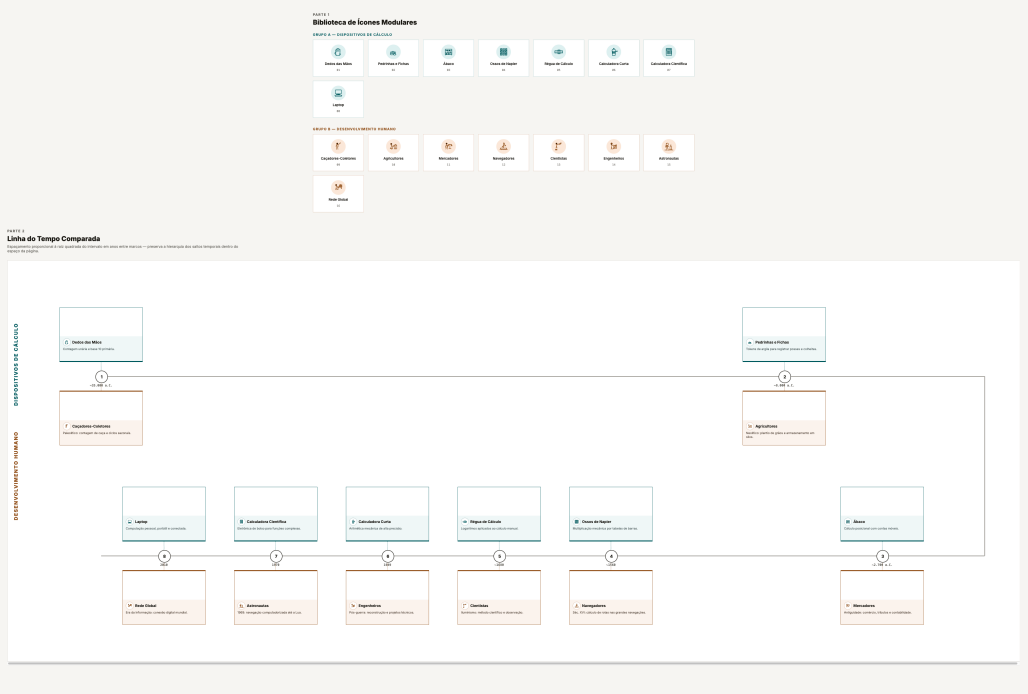

Figura 1.1: Linha do tempo comparada entre dispositivos de cálculo e estágios de desenvolvimento humano, em dois eixos paralelos. Eixo "dispositivos de cálculo": mãos (contagem unária) → pedrinhas e fichas → ábaco → ossos de Napier → régua de cálculo → calculadora Curta → calculadora científica → laptop. Eixo "desenvolvimento humano", pareado ponto a ponto com o eixo acima: caçador-coletor → agricultor → mercador → navegador → cientista → engenheiro → astronauta → era da rede global. As datas vão de aproximadamente 35.000 a.C. (mãos/caçador-coletor) a 2010 (laptop/rede global); o espaçamento entre marcos consecutivos encolhe de milênios, no início da linha, para décadas, no final — representando visualmente a aceleração do ritmo de inovação.

de mensagens. Por essa razão, ela se enquadra na definição ampla de computação, mas não na definição de computador digital de uso geral. Essa distinção entre "dispositivo que computa" e "computador de uso geral" é usada ao longo de todo o capítulo.

Repare como o espaçamento entre marcos consecutivos encolhe ao longo da linha do tempo acima: são muitos mil anos entre o uso das mãos para contar e as primeiras pedrinhas/fichas de contagem, mas apenas décadas entre a calculadora científica e o laptop conectado à rede global. Esse padrão de crescimento exponencial é o que ajuda a narrar como os grandes modelos de linguagem baseados em redes neurais estão a contribuir para a resolução de problemas matemáticos em aberto nos dias atuais. É também o pano de fundo de todo o restante deste livro: cada capítulo seguinte cobre uma fatia cada vez mais estreita — e cada vez mais recente — dessa mesma linha do tempo.

1.2 Evolução histórica: dos instrumentos mecânicos ao transistor

A tabela a seguir situa os principais marcos na evolução dos instrumentos de cálculo:

Período	Dispositivo	Característica
~2.500 a.C.	Ábaco	Contagem manual com contas móveis
1617	Ossos de Napier [2]	Auxílio a multiplicações
Até anos 1970	Régua de cálculo	Cálculo analógico por escalas logarítmicas
Anos 1940	Calculadora Curta	Mecanismo de engrenagens
—	Dispositivos eletromecânicos	Baseados em relés
—	Válvulas	Eletrônicos, porém volumosos e de alto consumo energético
—	Transistor	Chave eletrônica miniaturizada e de baixo custo

O **transistor** é o componente que viabilizou a computação moderna. Fisicamente, é constituído por um arranjo de material semicondutor dopado (uma junção N-P-N ou P-N-P), que funciona como uma chave eletrônica capaz de ligar e desligar em altíssima velocidade, ocupar espaço microscópico e ser fabricado em larga escala a baixo custo. A substituição progressiva de válvulas por transistores, ao longo da segunda metade do século XX, é o que permitiu a redução de tamanho e custo que tornou o computador pessoal viável.

Figura pendente

foto comparativa — ábaco / régua de cálculo / calculadora Curta / transistor em corte esquemático

1.3 Alan Turing e o conceito de máquina de propósito geral

Durante a Segunda Guerra Mundial, as forças alemãs utilizavam a máquina Enigma para criptografar comunicações militares. O matemático britânico Alan Turing foi recrutado para desenvolver métodos de decifração dessas mensagens [3].

A contribuição central de Turing para a computação não foi construir uma máquina especializada apenas em quebrar aquele código específico, mas conceber uma **máquina de propósito geral**: um dispositivo capaz de receber diferentes programas e, a partir deles, resolver diferentes classes de problemas. O primeiro problema resolvido por essa máquina foi, historicamente, a quebra do código Enigma.

Essa distinção — entre uma máquina que executa uma operação fixa e uma máquina que recebe o próprio algoritmo como entrada — é o critério que separa uma calculadora de um computador de uso geral, conforme apresentado na Seção 1.1.

Figura pendente

ilustração/foto da máquina Bombe de Turing

1.4 John von Neumann e o programa armazenado

De forma paralela e complementar ao trabalho de Turing, o conceito de **programa armazenado** foi formalizado por escrito no relatório *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument* (1946), assinado por Arthur Burks, Herman Goldstine e John von Neumann [4]: a ideia de que as instruções de um programa — e não apenas os dados que ele manipula — devem residir na mesma memória do computador. O nome consagrado do conceito, "arquitetura de von Neumann", é o que este livro adota, mas a formalização foi, de fato, um trabalho conjunto dos três autores.

Antes dessa formalização, o algoritmo existia apenas como um procedimento mental ou registrado em papel, executado passo a passo por um operador humano. Com o programa armazenado, o computador passa a executar a sequência completa de instruções de forma autônoma, sem intervenção humana a cada etapa.

O conceito de programa armazenado é a base da **arquitetura de von Neumann**, tratada em profundidade no Capítulo 3.

1.5 O modelo entrada–processamento–saída

Todo computador, para operar, requer três elementos: **entrada** de dados, **processamento** sobre esses dados e **saída** do resultado.

Exemplo. No cálculo da média entre duas notas (N1 e N2), a entrada consiste nos valores de N1 e N2; o processamento consiste na soma dos dois valores seguida da divisão por dois; a saída é o valor da média resultante.

Uma distinção relevante deve ser observada: ao realizar esse cálculo numa calculadora convencional, é o usuário humano quem executa o algoritmo, decidindo a sequência de operações. Num computador de uso geral,

o próprio algoritmo é tratado como um dado de entrada e é a máquina que o executa de forma autônoma — reforçando a definição apresentada na Seção 1.1.

Figura pendente

diagrama entrada → processamento → saída, com o exemplo da média

1.6 O sistema computacional: hardware, software e pessoas

Um sistema computacional não é composto apenas por hardware e software; inclui também as **pessoas** responsáveis por sua operação. Essa tríade está implícita na definição de Tanenbaum, que qualifica o computador como uma máquina que resolve problemas *para as pessoas*.

Em contextos de suporte técnico, é comum que um problema relatado como falha de hardware ou software seja, na verdade, decorrente de erro de operação — por exemplo, um cabo de alimentação desconectado. A identificação correta da origem do problema (hardware, software ou operação humana) é o primeiro passo de qualquer diagnóstico técnico — retomado como filtro central de toda manutenção no Capítulo 12 (§12.2).

1.6.1 Manutenção corretiva e preventiva

- **Manutenção corretiva**: intervenção realizada após a ocorrência de uma falha (por exemplo, substituir a pilha de um controle remoto somente depois que ele para de responder).
- **Manutenção preventiva**: intervenção realizada antes da ocorrência da falha, com base no ciclo de vida esperado do componente (por exemplo, substituir a pilha antes do fim de sua vida útil média).

Essa distinção é aprofundada, com condições reais de bancada e exemplos de periodicidade, no Capítulo 12 (§12.3).

1.6.2 Modularidade

O computador é um dispositivo **modular**, constituído por submódulos substituíveis (fonte, processador, memória, placa-mãe, entre outros). O procedimento padrão de diagnóstico consiste em:

1. Formular a hipótese de que um módulo específico está danificado.
2. Substituir esse módulo por um equivalente sabidamente funcional.
3. Observar se o sistema volta a operar corretamente.

Esse princípio de modularidade é o fundamento metodológico da disciplina de Manutenção de Computadores (2026.1).

Figura pendente

esquema “sistema computacional = hardware + software + pessoas”

1.7 Da computação corporativa ao computador pessoal

Até meados da década de 1970, a computação era predominantemente corporativa e institucional, realizada por grandes **mainframes** utilizados por bancos, governos, universidades e agências como a NASA.

A tabela a seguir reúne declarações históricas frequentemente citadas para ilustrar a dificuldade de prever a trajetória da computação pessoal:

Ano	Declaração	Autor/fonte
1949	“No futuro os computadores não pesarão mais do que uma tonelada e meia.” [6]	Revista Popular Mechanics
1977	“Não há razão para alguém querer um computador em casa.” [5]	Presidente da Digital Equipment Corporation

Nota sobre a citação de 1977. A declaração é real — foi feita por Ken Olsen, presidente e fundador da Digital Equipment Corporation, na World Future Society, em 1977 — mas costuma circular fora de contexto: Olsen se referia a um computador central controlando toda a casa (um único sistema automatizando iluminação, eletrodomésticos etc.), não ao PC pessoal como se tornaria comum. Ele próprio já possuía um computador doméstico na época [5].

1.7.1 Os primeiros computadores pessoais

- **Altair 8800** (1975): programado por meio de chaves binárias (posição alta = 1, posição baixa = 0), sem monitor, com saída representada por luzes indicadoras. Utilizava o processador **Intel 8080**, herdeiro da linhagem iniciada pelo **Intel 4004** (1971) — o primeiro microprocessador comercial, originalmente projetado para a calculadora Busicom — e continuada pelo **Intel 8008** (1972). Essa mesma linhagem leva, poucos anos depois, ao Intel 8086/8088 usado no IBM PC (Seção 1.8).
- **Apple I** (1976): placa de circuito artesanal, sem gabinete próprio, dependente de um televisor externo como monitor.

- **Apple II**, **Commodore** e **TRS-80** (1977): computadores já equipados com teclado integrado e gabinete, marcando o início da popularização comercial da computação pessoal.

Nota terminológica. O termo *bug*, usado para designar um defeito de software, tem origem num inseto encontrado literalmente causando um curto-circuito em um relé de um computador mainframe antigo.

Figura pendente

Altair 8800, Apple I e Apple II lado a lado, mesma escala

1.8 O IBM PC e a padronização da arquitetura pessoal

Em 1981, a IBM lançou o IBM PC, com o objetivo de atingir um preço de venda próximo a US$ 1.500. Para viabilizar esse custo, a equipe de desenvolvimento utilizou componentes já disponíveis no mercado, incluindo um processador (Intel 8088), uma variante de custo reduzido do Intel 8086 — com barramento de dados externo de 8 bits em vez de 16, o que barateava a memória (RAM/ROM) e a lógica de suporte necessárias na placa-mãe.

A IBM adotou um modelo de **hardware aberto**, publicando as especificações completas do computador e permitindo que terceiros fabricassem componentes e sistemas compatíveis. A exceção foi o chip de **BIOS** (*Basic Input/Output System*) — firmware responsável por inicializar o hardware e fornecer funções básicas de entrada e saída antes do carregamento de qualquer sistema operacional —, mantido como propriedade fechada da IBM.

Um terceiro realizou a engenharia reversa do BIOS da IBM e distribuiu uma versão funcionalmente equivalente e livre de restrições de licenciamento, o que reduziu significativamente o custo de produção de computadores compatíveis com o padrão IBM PC. A combinação entre hardware aberto e BIOS livre resultou na entrada de novos fabricantes no mercado — entre eles Compaq, Dell e HP —, consolidando o padrão IBM PC como referência da indústria.

Esse processo ilustra um efeito de mercado relevante para a área de tecnologia: plataformas com maior base de usuários tendem a atrair mais desenvolvimento de software, o que por sua vez amplia ainda mais sua base de usuários — um mecanismo análogo ao que hoje explica a predominância do desenvolvimento de aplicativos para a plataforma Android em relação a plataformas minoritárias.

Figura 1.2: Fotografia do IBM PC 5150 (1981), modelo original: gabinete horizontal bege com duas unidades de disquete 5,25”, teclado destacado na frente e monitor monocromático verde-sobre-preto apoiado sobre o gabinete, exibindo uma tela de texto de diagnóstico do sistema. Fonte: Wikimedia Commons [7]

1.9 Componentes mínimos de um computador desktop

São necessários quatro componentes para que um computador do tipo desktop seja capaz de ligar e operar minimamente:

1. **Fonte de alimentação** — fornece energia ao sistema (tratada em detalhe no Capítulo 8 e no Capítulo 11).
2. **Processador (CPU)** — unidade responsável pelo processamento.
3. **Memória principal (RAM)** — tratada na Seção 1.10.
4. **Placa-mãe** — promove a interconexão entre os demais componentes e a ligação com dispositivos de entrada e saída.

Componentes como placa de rede, impressora ou webcam não são necessários para o funcionamento mínimo do sistema, sendo classificados como módulos adicionais.

Procedimento de manuseio. Componentes eletrônicos devem ser manuseados pelas bordas, evitando o contato direto com os pontos de contato elétrico, e com atenção à descarga eletrostática acumulada pelo corpo humano.

Figura pendente

foto de placa-mãe com CPU, pente de RAM e conector de fonte identificados por setas

1.10 Introdução à hierarquia de memória

O computador utiliza dois tipos de memória de natureza distinta: **memória principal** (RAM) e **memória secundária** (armazenamento).

1.10.1 Memória RAM

RAM é a sigla para *Random Access Memory* (memória de acesso aleatório): o tempo necessário para ler ou escrever um dado independe da posição de memória acessada. Esse comportamento se opõe ao **acesso sequencial**, no qual o tempo de acesso depende da posição — como ocorre em dispositivos de armazenamento mecânico.

A célula de memória da RAM é construída com transistores e capacitores, otimizada para alta velocidade de acesso. Como consequência direta dessa construção, a RAM é uma memória **volátil**: seu conteúdo é perdido na ausência de energia elétrica.

1.10.2 Memória secundária

A memória secundária (armazenamento) é **não volátil** — mantém seus dados sem necessidade de energia contínua —, porém apresenta velocidade de acesso significativamente inferior à da memória RAM.

1.10.3 Pirâmide de hierarquia de memória

Da camada mais rápida (e mais cara) para a mais lenta (e mais barata):

1. **Registradores** — internos ao processador, capacidade da ordem de kilobytes.
2. **Memória cache (L1, L2, L3)** — associada ao processador, capacidade da ordem de megabytes; a cache L3 é tipicamente compartilhada entre múltiplos núcleos.
3. **Memória RAM** — capacidade da ordem de gigabytes.
4. **Armazenamento secundário** — capacidade da ordem de terabytes.

Essas unidades formam uma progressão: 8 bits formam um **byte**; 1024 bytes (2^{10}) formam um **kilobyte**; e cada unidade seguinte (megabyte, gigabyte, terabyte…) é 1024× a anterior — não 1000× como nas demais unidades do dia a dia (quilograma, quilômetro), porque a contagem nasce de potências de base 2, não de base 10. É essa progressão que explica por que a pirâmide se estreita: quanto mais alto o nível de memória, menor a capacidade típica disponível hoje — terabytes no armazenamento secundário, gigabytes na RAM, megabytes no cache, bytes no registrador.

1.10.4 Aplicações práticas

- O tempo de inicialização (*boot*) de um dispositivo corresponde à cópia de dados da memória secundária (lenta) para a memória RAM (rápida); por isso, destravar um dispositivo já ligado é mais rápido do que ligá-lo do zero.
- A substituição de um HD por um SSD reduz o tempo de inicialização por aumentar a velocidade da memória secundária.
- O conceito de *cache* é aplicado também fora do hardware local — por exemplo, em serviços de streaming de vídeo, que priorizam conteúdo popular em memória de acesso mais rápido.

1.10.5 Processadores especializados: CPU, GPU e NPU

Um computador moderno tipicamente integra mais de um tipo de processador — a Seção 1.11 aprofunda essa taxonomia e a estende a outros tipos de processador e a outros perfis de máquina além do desktop.

- **CPU** (*Central Processing Unit*) — processamento de propósito geral.
- **GPU** (*Graphical Processing Unit*) — processador dedicado a tarefas gráficas, com memória de alta velocidade própria (VRAM).

- **NPU** (*Neural Processing Unit*) — processador dedicado a cargas de trabalho de inteligência artificial, cada vez mais comum em dispositivos móveis e notebooks.

Figura pendente

pirâmide da hierarquia de memória com registradores, cache, RAM e armazenamento secundário] [IMAGEM: anúncio comentado de um processador e de uma placa de vídeo, com cache/núcleos/memória destacados

---

1.11 Processadores especializados e perfis de máquina

A Seção 1.9 apresentou os quatro componentes mínimos de um desktop, e a Seção 1.10 introduziu a hierarquia de memória e uma primeira taxonomia de processadores (CPU, GPU, NPU). Esta seção fecha o Capítulo 1 respondendo a uma pergunta que fica em aberto até aqui: um computador moderno raramente tem *um* processador — ele tem vários, cada um especializado numa tarefa —, e a combinação desses processadores muda radicalmente conforme o tipo de máquina considerado (um desktop, um notebook, um smartphone ou um servidor).

1.11.1 Taxonomia de processadores

- **CPU** (*Central Processing Unit*) — processamento de propósito geral; já apresentado na Seção 1.10.5.
- **GPU** (*Graphics Processing Unit*) — processador dedicado a tarefas gráficas, com memória de alta velocidade própria (VRAM); já apresentado na Seção 1.10.5.
- **NPU** (*Neural Processing Unit*) — processador dedicado a cargas de trabalho de inteligência artificial, cada vez mais comum em dispositivos móveis e notebooks; já apresentado na Seção 1.10.5.
- **APU** (*Accelerated Processing Unit*) — termo usado para designar um chip que combina, no mesmo encapsulamento, uma CPU e uma GPU integrada. Hoje praticamente todo processador de desktop e notebook tem vídeo integrado (Capítulo 8, §8.7.2), então "APU" deixou de ser uma categoria à parte e passou a descrever quase qualquer CPU moderna — o termo sobrevive principalmente como nome comercial de determinadas linhas de produto.
- **DSP** (*Digital Signal Processor*) — processador especializado em processar sinais contínuos (áudio, imagem, rádio) por meio de operações matemáticas repetitivas (somas e multiplicações em sequência) sobre grandes volumes de amostras. Diferente da CPU, que precisa lidar

com qualquer tipo de instrução, o DSP é otimizado apenas para esse tipo de cálculo — e por isso executa essas operações com muito mais eficiência energética. Está presente, por exemplo, no microfone e na câmera de um smartphone, processando o sinal bruto antes que ele chegue à CPU ou à NPU.

- **SoC** (*System on a Chip*, sistema em um único chip) — não é um tipo de processador, mas uma **estratégia de integração**: reunir, num único encapsulamento, CPU, GPU, controlador de memória, modem de rede e demais controladores que, num desktop, estariam espalhados entre processador, chipset e placa-mãe (Capítulo 10, §10.1.2, trata do chipset e da interconexão desses componentes). Praticamente todo smartphone, tablet e Raspberry Pi é organizado em torno de um SoC.

Figura pendente

diagrama de um SoC de smartphone mostrando CPU, GPU, NPU, DSP e modem integrados no mesmo chip, ao lado de um diagrama de desktop com CPU, GPU discreta e chipset como blocos separados

1.11.2 Hardware para diferentes perfis de uso

O mesmo conjunto de conceitos — CPU, memória, armazenamento, interconexão — se organiza de formas muito diferentes conforme o perfil de uso da máquina. Quatro perfis cobrem a maior parte do mercado: **desktop**, **notebook**, **dispositivo móvel** e **servidor**.

Característica	Desktop	Notebook	Dispositivo móvel	Servidor
Processador	CPU substituível, soquete próprio (Capítulo 8, §8.4)	CPU frequentemente soldada à placa	SoC (CPU+GPU+modem integrados)	Um ou mais soquetes de CPU na mesma placa-mãe
Memória RAM	Módulos substituíveis, geralmente Dual Channel (Capítulo 5, §5.7)	Módulos substituíveis ou soldados, conforme o modelo	Soldada/empacotada junto ao SoC	Módulos ECC (Capítulo 5, §5.15), muitos slots, capacidade na casa de centenas de GB a TB

Característica	Desktop	Notebook	Dispositivo móvel	Servidor
Alimentação	Fonte ATX interna (Capítulo 11, §11.2)	Bateria + fonte externa	Bateria interna	Uma ou mais fontes redundantes
Prioridade de projeto	Custo-benefício e desempenho bruto	Portabilidade e autonomia de bateria	Autonomia de bateria acima de tudo	Disponibilidade contínua (*uptime*) e capacidade de processar muitas requisições simultâneas
Forma física	Gabinete de mesa (torre, *desk*)	Chassi único e fino	Chassi único, sem abertura para manutenção	Chassi em formato *rack*, dimensionado em unidades **U** (1U, 2U, 4U) para empilhamento em um armário de rede

Servidores: redundância como princípio de projeto. Um servidor difere de um desktop não por processar "mais rápido" — muitas vezes o processador de um servidor tem clock por núcleo *menor* que o de um desktop, priorizando número de núcleos e eficiência energética sob operação contínua —, mas por ser projetado para **nunca parar**. Essa exigência se traduz em componentes redundantes: duas ou mais fontes de alimentação (se uma falha, a outra assume sem interrupção), discos organizados em arranjos com redundância de dados (de forma que a falha de um disco não cause perda de dados, tema aprofundado na disciplina de Manutenção de Computadores), placas-mãe capazes de hospedar mais de um processador físico simultaneamente, e memória **ECC** — tratada em detalhe, junto com o fenômeno físico que ela existe para corrigir (o *bit flip*), no Capítulo 5, §5.15.

Notebooks: compromisso entre modularidade e portabilidade. O princípio de modularidade apresentado na Seção 1.6.2 — trocar um módulo suspeito para diagnosticar uma falha — se aplica com menos liberdade a um notebook do que a um desktop. Para reduzir espessura, peso e consumo de energia, fabricantes de notebook soldam diretamente à placa componentes que, num desktop, seriam módulos substituíveis: a memória RAM e, cada vez mais, também o processador. Um técnico ainda consegue substituir bateria, SSD (quando não soldado) e tela na maioria dos modelos, mas a possibilidade de "testar trocando o módulo" (Seção 1.6.2) fica mais restrita

— e, quando o componente soldado falha, a reparação deixa de ser uma troca simples e passa a exigir retrabalho de solda em nível de placa, ou a substituição da placa-mãe inteira.

Dispositivos móveis: o extremo da integração. Um smartphone leva a lógica do notebook ao limite: por trás de um SoC único não há sequer um soquete de processador — CPU, GPU, NPU, modem e, com frequência, a própria memória, estão fisicamente empilhados no mesmo encapsulamento (uma técnica chamada *package-on-package*). Não existe, na prática, "abrir o aparelho e trocar a CPU" nesse perfil de máquina — a modularidade da Seção 1.6.2 se desloca inteiramente para fora do hardware, para o nível de aplicativos e serviços.

Figura pendente

fotos comparativas lado a lado — placa-mãe de desktop com CPU socketed e RAM em slots, placa de notebook com RAM soldada, e um SoC de smartphone isolado

---

1.12 Síntese do capítulo

Este capítulo apresentou a definição técnica de computador, sua origem histórica e evolução até o computador pessoal moderno, o princípio de modularidade que fundamenta a manutenção de computadores, uma primeira introdução à hierarquia de memória, e como esses mesmos componentes se recombinam em diferentes perfis de máquina — do desktop ao servidor. Esses conceitos serão retomados e aprofundados nos capítulos seguintes: memória (Capítulo 5), sistema operacional (Capítulo 6), instalação (Capítulo 7), hardware físico e montagem (Capítulo 8) e arquitetura de processadores (Capítulo 3).

1.13 Referências

1. TANENBAUM, Andrew S.; AUSTIN, Todd. *Organização Estruturada de Computadores*. 6. ed. São Paulo: Pearson Education do Brasil, 2013.
2. Data de 1617 para os "Ossos de Napier" (publicação de *Rabdologiae*, Edimburgo) confirmada em: MACTUTOR HISTORY OF MATHEMATICS ARCHIVE, University of St Andrews, verbete "John Napier"; SCIENCE MUSEUM GROUP, coleção de instrumentos de cálculo. Note-se que o livro-texto local de Tangon & Santos atribui erroneamente 1614 a este marco — essa é a data de outra obra de Napier (*Mirifici Logarithmorum Canonis Descriptio*, sobre logaritmos), não dos ossos.

3. HODGES, Andrew. *Alan Turing: The Enigma*. Londres: Burnett Books, 1983; Nova York: Simon & Schuster, 1983.
4. BURKS, A. W.; GOLDSTINE, H. H.; VON NEUMANN, J. *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*. Princeton: Institute for Advanced Study, 1946.
5. Sobre o contexto da declaração de Ken Olsen (1977, World Future Society): QUOTE INVESTIGATOR. “There Is No Reason for Any Individual to Have a Computer in Their Home”; TECHRADAR, reportagens sobre a origem e o contexto da citação.
6. POPULAR MECHANICS. “Brains That Click.” mar. 1949.
7. Fotografia do IBM PC 5150. Autor: Reseletti. Wikimedia Commons, 2022. Licença CC BY-SA 3.0. Disponível em: https://commons.wikimedia.org/wik

2 Do Transistor à Arquitetura de von Neumann

Neste capítulo você vai reconstruir, camada por camada, a ponte entre o software que se escreve e o hardware que efetivamente o executa: da plataforma de execução ao transistor, do transistor às portas lógicas, das portas lógicas aos circuitos aritméticos e de memória, e destes à hierarquia de linguagens que culmina na arquitetura de von Neumann e em sua limitação estrutural, o gargalo de von Neumann. Esses fundamentos sustentam os capítulos seguintes: a Arquitetura e Organização de Processadores (Capítulo 3) usa o vocabulário de arquitetura do conjunto de instruções construído aqui, e a Memória (Capítulo 5) usa diretamente o flip-flop apresentado na Seção 2.4.

2.1 Plataforma e abstração em camadas

Todo software é desenvolvido para ser executado em um determinado local — o que este livro chama de **plataforma**. A plataforma de um software pode ser um programa (por exemplo, um navegador específico), um sistema operacional, ou, em última instância, uma arquitetura de hardware.

Exemplo. Um bloqueador de anúncios desenvolvido para o navegador Firefox tem como plataforma o próprio Firefox: o software não funciona no Chrome nem no Safari. Por sua vez, o Firefox tem como plataforma diversos sistemas operacionais (Windows, macOS, Linux, Android, iOS). E cada um desses sistemas operacionais tem como plataforma hardwares fisicamente distintos — um processador Snapdragon num smartphone Android, um chip Apple M1 num MacBook, um processador Intel ou AMD num desktop Windows.

Essa organização em camadas — software sobre navegador, navegador sobre sistema operacional, sistema operacional sobre hardware — é chamada **abstração em camadas**. Ela é o motivo pelo qual um desenvolvedor de software para navegador não precisa se preocupar com qual câmera, qual placa de rede ou qual processador está instalado no dispositivo do usuário final: cada camada delega à camada abaixo dela a responsabilidade por lidar com a complexidade que está fora do seu escopo.

A estratégia de camadas não é exclusiva da relação hardware-software: a mesma lógica organiza, por exemplo, os protocolos de rede em camadas (física, enlace, transporte, aplicação). Este capítulo adota uma abordagem

top-down (do software visível ao usuário até o hardware que o executa) para, nas seções seguintes, reconstruir a mesma pilha de forma **bottom--up** (do transistor até a linguagem de programação).

Figura pendente

pilha de camadas — software aplicativo → navegador → sistema operacional → hardware, com setas de dependência apontando para baixo

2.2 Do transistor às portas lógicas

Para entender como um software é efetivamente executado por um hardware, é necessário responder a uma pergunta simples em aparência: como um circuito eletrônico realiza uma operação lógica ou aritmética?

2.2.1 O combinado entre bit e tensão

Um programa como `x = 2` instrui o computador a reservar um espaço de memória, nomeá-lo `x` e armazenar ali o valor 2. Esse número, em base decimal, é convertido para binário antes de ser armazenado, porque a eletrônica digital opera sobre um **combinado** (convenção) entre os símbolos binários (0 e 1) e grandezas elétricas mensuráveis. Na lógica TTL, por exemplo, o combinado usual é: 5 volts representa o bit 1, e 0 volts representa o bit 0 [1]. Esse combinado é arbitrário — poderia ser outro par de tensões — mas precisa ser fixado entre fabricante e desenvolvedor para que a informação seja interpretada de forma consistente. **Nota técnica:** na prática, os níveis TTL reais são faixas, não valores exatos — um nível alto válido é qualquer tensão a partir de aproximadamente 2,4 V, e um nível baixo válido vai até aproximadamente 0,4–0,5 V; “5 V e 0 V” é a simplificação didática usual para os valores nominais de saída.

2.2.2 O semicondutor e o transistor

Um **semicondutor** é um material que, dependendo das condições (como a temperatura), pode se comportar como isolante ou como condutor — o silício é o exemplo mais usado na indústria. Ao dopar o silício com impurezas específicas, obtêm-se dois tipos de material: o tipo **N** (com excesso de elétrons) e o tipo **P** (com déficit de elétrons, ou “buracos”). A junção entre um material tipo P e um tipo N forma um **diodo**, que conduz corrente em apenas um sentido quando polarizado corretamente.

Empilhando três camadas alternadas (P-N-P ou N-P-N), obtém-se o **transistor**: um dispositivo de três terminais — coletor, base e emissor — que se comporta como uma chave eletromecânica. Quando há nível lógico alto (1) na base, o transistor fecha o contato entre coletor e emissor, permitindo a passagem de corrente; quando há nível lógico baixo (0) na base, o contato permanece aberto.

2.2.3 Construindo portas lógicas com transistores

A partir dessa chave elementar, é possível construir as portas lógicas estudadas em eletrônica digital:

- **Porta AND**: dois transistores ligados em série. A saída só apresenta nível lógico 1 se ambas as entradas estiverem em 1 — cada transistor precisa estar "fechado" para que a tensão chegue até a saída.
- **Porta OR**: dois transistores ligados em paralelo. A saída apresenta nível lógico 1 se qualquer uma das entradas estiver em 1 — basta que um dos dois caminhos esteja fechado.
- **Porta NOT (inversora)**: um único transistor usado como chave que inverte o sinal de entrada — 1 na entrada produz 0 na saída, e vice-versa.

Nota técnica. Esse modelo (série = AND, paralelo = OR) é uma simplificação pedagógica — um "modelo de chave" coerente com a lógica CMOS moderna — e não corresponde exatamente à topologia de circuitos comerciais TTL/DTL, que tradicionalmente implementam AND/OR com lógica a diodo e usam um transistor inversor separado [2]. O modelo aqui é útil para a intuição, mas não deve ser confundido com a topologia exata de um circuito integrado comercial.

Figura pendente

três circuitos lado a lado — porta AND (transistores em série), porta OR (transistores em paralelo) e porta NOT (transistor inversor), com tabela-verdade de cada uma

2.3 Circuitos aritméticos: o meio-somador

Se é possível construir portas lógicas com transistores, também é possível construir um circuito capaz de somar dois números binários.

A adição em binário segue a mesma lógica da adição em decimal, mas com apenas dois símbolos disponíveis:

A	B	Soma	Vai-um (*carry*)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Observando essa tabela, nota-se que a coluna "Soma" corresponde exatamente à tabela-verdade da porta **OU exclusivo** (XOR), e a coluna "Vai-um" corresponde exatamente à tabela-verdade da porta **AND**.

O circuito que combina uma porta XOR (para o bit de soma) e uma porta AND (para o bit de vai-um), ambas recebendo as mesmas duas entradas A

e B, é chamado **meio-somador** (*half adder*). Ele soma dois bits e produz dois resultados: o bit de soma e o bit de transporte para a próxima posição. Encadeando vários meios-somadores (com o acréscimo da entrada de vai-um recebido da posição anterior, o que dá origem ao **somador completo**, ou *full adder*), constrói-se um circuito capaz de somar números de qualquer quantidade de bits — o tamanho da palavra binária que um processador consegue somar de uma vez (32 bits, 64 bits) é justamente definido por quantos desses circuitos estão encadeados no hardware.

Figura pendente

circuito do meio-somador — porta XOR e porta AND recebendo as entradas A e B, produzindo Soma e Carry

2.4 Memória: o flip-flop e os sinais de controle

Se as portas lógicas resolvem o problema das operações lógicas e aritméticas, resta um terceiro problema: como armazenar um valor de forma persistente, para além do instante em que ele é calculado.

A célula de memória mais básica é o **flip-flop**: um circuito construído a partir da combinação de portas lógicas (e, portanto, em última instância, de transistores) capaz de armazenar um bit e responder a duas operações — leitura ("qual valor você tem?") e escrita ("guarde este valor").

A tabela-verdade do flip-flop tipo JK, um dos modelos mais estudados, resume seu comportamento:

J	K	Comportamento
0	0	Mantém o valor atual (memória não é alterada)
0	1	Escreve 0
1	0	Escreve 1
1	1	Inverte o valor atual

O flip-flop só responde a essas entradas no instante em que recebe um sinal de **clock** — um pulso periódico que sincroniza a operação. É esse sinal de clock, gerado em alta frequência, que corresponde à frequência de operação anunciada para um processador (em megahertz ou gigahertz).

2.4.1 Sinais de controle

A operação de leitura e escrita não esgota o funcionamento de um chip de memória real. Um registrador construído com flip-flops — como o circuito integrado 74HC173, um registrador de 4 bits disponível comercialmente [3] — expõe também:

- **Reset**: zera o valor armazenado, independentemente do estado do clock.

- **Habilitação de saída** (*output enable*): liga ou desliga a conexão entre o valor armazenado e a saída do chip — relevante, por exemplo, quando a saída está ligada a um motor que precisa estar energizado antes de receber o sinal.
- **Habilitação de escrita** (*write enable*): determina se o chip está em modo de leitura ou de escrita naquele instante, mesmo que o sinal de clock continue chegando.

Exemplo. Um simulador de eletrônica digital utilizado em aula demonstra por que os sinais de controle são indispensáveis: ao conectar um circuito somador simples diretamente a um decodificador binário e um display de sete segmentos, o resultado exibido fica incorreto se o segundo operando for alterado antes de o primeiro ter sido efetivamente processado. Os sinais de controle são o que permite dizer ao circuito "carregue este valor agora" e "leia o resultado agora" em momentos distintos e bem definidos — sem eles, não há garantia de que a operação foi concluída antes da leitura.

Esses sinais — de ligar/desligar, de habilitar saída, de selecionar entre operações possíveis de um chip — reaparecem em praticamente todo circuito digital: um somador, um circuito de memória, um controlador de barramento. Entender que eles existem é o que permite compreender a camada seguinte da abstração, tratada na próxima seção.

Figura pendente

registrador de 4 bits com entradas de dados, clock, reset e habilitação de saída identificadas

2.5 A hierarquia de linguagens: de L0 a L5

A hierarquia de níveis apresentada nesta seção segue o modelo de máquinas multiníveis proposto por Tanenbaum [4]. A lógica digital descrita nas seções anteriores é poderosa, mas impraticável de programar diretamente: escrever um programa inteiro em sequências de 0 e 1 é uma tarefa inviável para qualquer problema não trivial. A solução histórica para esse problema foi a criação de sucessivas camadas de linguagem, cada uma mais próxima do raciocínio humano do que a anterior, e cada uma traduzida (por um interpretador, compilador ou circuito de controle) para a camada imediatamente abaixo.

Nível	Nome	Função
L0	Lógica digital	Portas lógicas, somadores, memórias (Seções 2.2–2.4)

Nível	Nome	Função
L1	Microarquitetura (microprograma)	Controla o fluxo de dados: o que é carregado, de onde, para onde, acionando a camada L0
L2	Arquitetura do conjunto de instruções (ISA)	Define se o processador é RISC ou CISC, e qual conjunto de instruções ele reconhece (x86, ARM, etc.)
L3	Sistema operacional	Permite a execução de múltiplos programas e o gerenciamento do hardware por eles
L4	Tradução (compiladores/montadores)	Converte código de alto nível em código de máquina executável pelo sistema operacional
L5	Linguagem orientada a problema	Python, C++, JavaScript — nível em que a maioria dos desenvolvedores trabalha

Nota terminológica. É comum que a imprensa especializada em hardware anuncie uma "nova arquitetura" ao descrever um novo processador quando, tecnicamente, o que mudou foi a **microarquitetura** (nível L1) — a forma como aquele conjunto de instruções é implementado internamente —, enquanto a arquitetura do conjunto de instruções (nível L2, por exemplo x86-64) permanece a mesma.

2.5.1 Exemplo: de Python ao circuito

Retomando o programa `x = 2; y = x + 1`: em L5 (Python), a atribuição e a soma são expressas em sintaxe legível por humanos. O interpretador ou compilador (L4) traduz essas instruções para código de máquina. O sistema operacional (L3) aloca memória e agenda a execução do processo. O processador consulta seu conjunto de instruções (L2) para decodificar cada instrução recebida. O microprograma (L1) aciona os circuitos específicos — leitura de memória, soma, escrita de memória — que fisicamente existem no nível de lógica digital (L0), executando a operação por meio de tensões elétricas.

Antes da existência de linguagens de alto nível, a única alternativa a programar diretamente em código de máquina (sequências binárias) era a **linguagem Assembly**: um conjunto de mnemônicos (como `ADD`, `MOV`) que correspondem quase diretamente às instruções do processador, mas ainda exigem gerenciamento manual explícito de registradores e endereços de memória — por isso, extremamente trabalhosa para programas complexos.

2.5.2 O nível além de L5

Um nível ainda não formalizado na literatura clássica de arquitetura de computadores, mas cada vez mais concreto na prática, é o de **linguagem natural**: comandos dados a um modelo de linguagem (LLM) que gera código executável em Python, C++ ou outra linguagem de nível L5 a partir de um pedido descrito em português ou inglês comum. Ferramentas como assistentes de voz (Alexa, Google Assistant) representaram uma primeira aproximação limitada desse nível; modelos de linguagem contemporâneos ampliam significativamente essa capacidade, embora ainda não exista uma camada intermediária formal, treinada especificamente para maximizar a taxa de acerto entre o pedido em linguagem natural e o código gerado.

Nota sobre o uso de ferramentas de IA. Um modelo de linguagem gera texto com alta probabilidade estatística de ser relevante — não é um interlocutor consciente, ainda que a fluência de suas respostas possa sugerir o contrário. Isso reforça um ponto prático para o técnico de informática: modelos de linguagem e agentes de IA são ferramentas, e a responsabilidade por uma tarefa delegada a eles continua sendo de quem concedeu essa liberdade — da mesma forma que dizer "o macaco trocou o pneu do carro" ignora que foi a pessoa quem usou o macaco.

2.6 A arquitetura de von Neumann

O conceito de **programa armazenado** — a ideia de que as instruções de um programa, e não apenas os dados que ele manipula, residem na mesma memória do computador — já apresentado como fundamento histórico da computação de uso geral, foi formalizado por escrito no relatório "First Draft of a Report on the EDVAC" (jun. 1945), de John von Neumann, e detalhado em 1946 em coautoria com Arthur Burks e Herman Goldstine [5]. A disputa de crédito mais discutida na literatura histórica não é com Alan Turing, mas com **Eckert e Mauchly**, engenheiros do grupo ENIAC/EDVAC que consideravam a ideia um resultado coletivo e ficaram descontentes por o relatório circular só com o nome de von Neumann. Turing, por sua vez, produziu um relatório equivalente sobre programa armazenado (o projeto ACE, apresentado ao National Physical Laboratory em fevereiro de 1946, já em tempos de paz) — o sigilo que de fato cercou o trabalho de Turing durante a guerra foi o da criptoanálise em Bletchley Park, não o do próprio projeto ACE [6]. Ainda assim, "arquitetura de von Neumann" é o nome consagrado que este livro adota para a formalização.

Essa arquitetura organiza o computador em quatro unidades funcionais:

- **Unidade de entrada** — capta dados do mundo externo (teclado, mouse, câmera).
- **Unidade de saída** — expõe dados processados ao mundo externo (monitor, alto-falante).
- **Unidade de processamento (CPU)** — subdividida em **unidade de controle** (coordena a sequência de operações) e **unidade lógica e aritmética — ULA** (executa as operações lógicas e aritméticas propriamente ditas).
- **Unidade de memória** — armazena tanto os dados quanto o próprio programa em execução.

No computador desktop moderno, a via de dados que interliga processador, memória e dispositivos de entrada e saída — o **barramento** — é fisicamente provida pela placa-mãe.

Figura pendente

diagrama da arquitetura de von Neumann — CPU (unidade de controle + ULA), memória, entrada e saída interligados por um barramento central

2.7 O gargalo de von Neumann

A separação física entre processador e memória — necessária, já que nenhum dispositivo conhecido realiza simultaneamente as funções de processamento e de armazenamento — traz uma limitação estrutural conhecida como **gargalo de von Neumann**.

Abordagens modernas de arquitetura, como o **System-on-Chip (SoC)**, aproximam fisicamente processador e memória dentro de um único encapsulamento, reduzindo a latência de comunicação entre eles. Essa aproximação mitiga o problema, mas não o elimina: processamento e memória continuam sendo entidades fisicamente distintas, e o gargalo de von Neumann permanece um problema em aberto na arquitetura de computadores.

2.8 Computação além do silício

A abstração em camadas construída ao longo deste capítulo — da tensão elétrica à porta lógica, da porta lógica ao circuito aritmético, do circuito à linguagem de programação — revela algo importante sobre a natureza da computação: o que importa é a estrutura lógica das operações, não o material físico que as implementa.

Exemplo. Utilizando o sistema de circuitos do jogo *Minecraft* (o mecanismo de *redstone*), é possível reproduzir portas lógicas, meios-somadores e células de memória, e a partir deles construir um computador funcional

dentro do próprio jogo — capaz, por exemplo, de executar um programa que calcula a sequência de Fibonacci. Um projeto ainda mais extremo levou essa ideia ao limite: como o computador construído em *redstone* é, em si, capaz de executar qualquer programa, um desenvolvedor conseguiu rodar uma cópia do próprio *Minecraft* dentro do computador que havia construído dentro do *Minecraft* — a um custo de desempenho da ordem de milhões de vezes mais lento, mas funcionalmente completo [7].

Outros projetos substituem o transistor por outros meios físicos para implementar as mesmas portas lógicas — por exemplo, circuitos hidráulicos que implementam portas AND, OR e NOT utilizando água e tubulações [8]. O romance de ficção científica *O Problema dos Três Corpos*, do autor chinês Liu Cixin, descreve um exército de seres humanos treinados para atuar, cada um, como uma porta lógica — formando coletivamente um computador funcional operado inteiramente por pessoas [9].

Esses exemplos, por mais lúdicos que pareçam, sustentam um ponto central deste capítulo: a computação é uma estrutura lógica de camadas de abstração, e o silício é apenas o meio físico mais eficiente conhecido atualmente para implementá-la — não o único possível.

2.9 Síntese do capítulo

Este capítulo reconstruiu a ponte entre software e hardware por meio da abstração em camadas: da plataforma de execução ao transistor, do transistor às portas lógicas, das portas lógicas aos circuitos aritméticos e de memória, e destes à hierarquia de linguagens que culmina na arquitetura de von Neumann e em sua limitação estrutural, o gargalo de von Neumann. Esses fundamentos — sobretudo a noção de que todo problema pode ser localizado numa camada específica dessa pilha, e de que o flip-flop e a porta lógica são os blocos de construção de qualquer circuito digital — sustentam os capítulos seguintes: a Arquitetura e Organização de Processadores (Capítulo 3) aprofunda o nível L2 (ISA); a Memória (Capítulo 5) usa diretamente o flip-flop aqui apresentado.

2.10 Referências

1. MALVINO, Albert Paul; BROWN, Jerald A. *Digital Computer Electronics*. 3. ed. Nova York: Glencoe/McGraw-Hill, 1993, Cap. 4 “TTL Circuits”.
2. MALVINO, Albert Paul; BROWN, Jerald A. *Digital Computer Electronics*. 3. ed. Nova York: Glencoe/McGraw-Hill, 1993, Cap. 2 (“2-1 Inverters”, “Diode OR Gate”, “2-3 AND Gates”, “Diode AND Gate”).

3. NEXPERIA. “74HC173; 74HCT173 — Quad D-type flip-flop; positive--edge trigger; 3-state.” Datasheet. Disponível em: https://assets.nexperia.com/documents/data-sheet/74HC_HCT173.pdf.
4. TANENBAUM, Andrew S.; AUSTIN, Todd. *Organização Estruturada de Computadores*. 6. ed. São Paulo: Pearson Education do Brasil, 2013, Seção 1.1.
5. BURKS, A. W.; GOLDSTINE, H. H.; VON NEUMANN, J. *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*. Princeton: Institute for Advanced Study, 1946.
6. COPELAND, B. Jack (org.). *Alan Turing’s Automatic Computing Engine*. Oxford: Oxford University Press, 2005.
7. MOJANG. “Minecraftception.” Minecraft.net. Disponível em: https://www.minecraft.net/en-us/article/-minecraftception.
8. “Hydraulic logic gates: building a digital water computer.” *European Journal of Physics*, v. 39, 2018. IOP Publishing. Disponível em: https://iopscience.iop.org/article/10.1088/1361-6404/aa97fc.
9. LIU, Cixin. *O Problema dos Três Corpos*. São Paulo: Aleph. (Tradutor e ano da edição a confirmar pelo autor.)

3 Arquitetura e Organização de Processadores

Neste capítulo você vai estudar a distinção entre arquitetura e organização de computadores, o aprofundamento da arquitetura de von Neumann introduzida no Capítulo 1, os dois grandes paradigmas de projeto de processadores — CISC e RISC —, a evolução da família x86/x64, as famílias ARM e AVR como exemplos de RISC, e uma revisão integrada de conceitos de firmware e diagnóstico que perpassam todo o livro.

3.1 Arquitetura versus organização de computadores

Os termos *arquitetura de computadores* e *organização de computadores* são, com frequência, usados como sinônimos — inclusive na literatura técnica e no vocabulário do mercado de tecnologia. Este livro, seguindo a distinção adotada por autores como Stallings [1], trata os dois termos como conceitos tecnicamente distintos.

Arquitetura é o conjunto de instruções que um processador disponibiliza para quem programa nesse processador. Esse conjunto de instruções é chamado, em inglês, de *Instruction Set Architecture* (ISA) — arquitetura do conjunto de instruções. A arquitetura define o que um processador é capaz de executar: quais operações existem, como os dados são endereçados, quais registradores estão disponíveis.

Organização é a implementação em hardware de uma dada arquitetura: as escolhas de projeto que determinam quantos núcleos um processador tem, qual a quantidade de memória cache, qual a frequência de operação, quanto de memória RAM ou de armazenamento um computador possui. Duas máquinas podem compartilhar a mesma arquitetura — o mesmo conjunto de instruções — e, ainda assim, ter organizações completamente diferentes.

Exemplo. Um processador Intel Core i3, um Core i5 e um Core i7 de uma mesma geração compartilham a mesma arquitetura x86/x64: executam exatamente o mesmo conjunto de instruções e, portanto, os mesmos programas. O que os diferencia — quantidade de núcleos, tamanho de cache, frequência de clock, consumo energético — são decisões de organização. O mesmo raciocínio se aplica a um smartphone vendido em versões 4G e 5G, ou a um mesmo modelo de notebook oferecido com capacidades

diferentes de SSD: a arquitetura do processador permanece idêntica; o que muda é a organização do hardware ao redor dele.

Essa distinção também explica por que o nome da disciplina que dá origem a este livro é *organização e montagem de computadores*: ao longo dos capítulos anteriores, o que foi estudado — processador, memória principal, memória secundária, placa-mãe e demais componentes — são, tecnicamente falando, decisões de organização, e não de arquitetura. Arquitetura é assunto do programador de baixo nível e do fabricante do processador; organização é assunto de quem monta, especifica e mantém computadores.

Nota de uso. No vocabulário de mercado — sites de notícias de tecnologia, materiais de fabricantes, embalagens de produto —, o termo "arquitetura" costuma ser aplicado de forma ampla, cobrindo também mudanças que, tecnicamente, são de organização (por exemplo, uma nova geração de processador com mais núcleos e cache é frequentemente anunciada como "nova arquitetura"). O leitor deve estar preparado para essa imprecisão terminológica ao consultar fontes não acadêmicas.

Figura pendente

comparativo de anúncios de processadores i3/i5/i7 de mesma geração, com núcleos e cache destacados

3.2 A arquitetura de von Neumann: instrução e dado são a mesma coisa

O Capítulo 2 (§2.6) formalizou a arquitetura de von Neumann e suas quatro unidades funcionais. Esta seção não repete essa formalização — explora, em vez disso, uma consequência dela que é central para entender a distinção entre arquitetura e organização apresentada na Seção 3.1.

Essa consequência raramente é percebida por quem está começando a estudar computação: o processador não distingue, por natureza, uma instrução de um dado, nem atribui significado ao que está processando. O processador é uma máquina que executa instruções mecanicamente, manipulando padrões binários e produzindo uma saída — sem qualquer noção do propósito daquela operação.

Exemplo. Do ponto de vista do processador, é absolutamente indiferente se a sequência de instruções que ele executa está calculando a nota final de um aluno ou controlando a fervura de um doce: ele apenas manipula dados e gera uma saída. Quem atribui sentido a essa saída — quem decide se o resultado representa uma média escolar ou uma receita culinária — é o software, não o hardware. O processador, isoladamente, "não sabe o que está fazendo": ele apenas executa.

Essa característica é o que torna a arquitetura de von Neumann ao mesmo tempo poderosa e genérica: como instruções são armazenadas e tratadas como qualquer outro dado na memória, o mesmo hardware pode executar qualquer programa que respeite o seu conjunto de instruções (a arquitetura,

na definição da Seção 3.1) — sem que o hardware precise ser fisicamente alterado a cada novo problema. Esse é o mesmo critério que, no Capítulo 1, distinguiu a máquina de Turing de propósito geral de uma calculadora de operação fixa.

Figura pendente

diagrama simplificado da arquitetura de von Neumann — processador, memória (instruções e dados), barramento, entrada/saída

3.3 CISC: retrocompatibilidade e a família x86/x64

Em 1981, a IBM lançou o IBM PC utilizando o processador Intel 8088, derivado do 8086 — um chip de 16 bits projetado como sucessor do 8080 para competir com os processadores de 16/32 bits da Zilog e da Motorola, escolhido pela IBM por seu baixo custo [2] (o Capítulo 1 apresenta o 8088 como variante de custo reduzido do 8086; a origem em calculadoras mencionada naquele capítulo se refere a chips anteriores da família Intel — o 4004 e o 8008 —, não ao 8086/8088). O sucesso comercial desse computador consolidou o conjunto de instruções do 8086 como padrão de mercado, e toda a evolução subsequente da família de processadores da Intel — e, mais tarde, da AMD — preservou esse conjunto de instruções original, apenas adicionando novas instruções a cada geração.

Esse comportamento decorre de uma exigência implícita do mercado de software: um fabricante que investiu no desenvolvimento de um programa para um determinado processador espera que esse programa continue funcionando — de preferência com desempenho melhor — nos processadores lançados posteriormente. A esse requisito dá-se o nome de **retrocompatibilidade**: a capacidade de um sistema mais novo executar, sem modificação, programas feitos para um sistema mais antigo da mesma família.

A tabela a seguir situa os principais marcos da evolução da família x86:

Processador	Observação
8086	16 bits; sucessor do 8080, projetado para competir com Zilog Z8000 e Motorola 68000; base do computador que a IBM adotou
8088	Versão de barramento externo simplificado do 8086, usada no IBM PC original (1981)
80286	Segunda geração da família; retrocompatível com o 8086

Processador	Observação
80386	Lançado em 1985 [3]; primeira geração de 32 bits da família, retrocompatível até o 8086
80486	Evolução do 386 com memória cache integrada ao processador
Pentium	Sucessor do 486; a Intel abandonou a numeração “586” e passou a usar um nome comercial [4]
Pentium II / III	Gerações seguintes da linha Pentium
Core 2 Duo	Geração seguinte, já multinúcleo
Core i3 / i5 / i7	Segmentação por faixa de desempenho, mantendo a mesma arquitetura x86/x64
Core Ultra / Core 5, 7, 9	Nomenclatura adotada a partir de 2023 (geração Meteor Lake), substituindo i3/i5/i7 após décadas de uso [5]

Cada geração dessa família mantém, como subconjunto, o conjunto de instruções de todas as gerações anteriores — o que pode ser representado como um diagrama de Venn de círculos concêntricos, em que as instruções do 8086 estão contidas nas do 286, que estão contidas nas do 386, e assim sucessivamente.

Ao conjunto de processadores que evoluem dessa forma — acumulando instruções ao longo do tempo, sem nunca descartar as anteriores — dá-se o nome de arquitetura **CISC** (*Complex Instruction Set Computer*, computador com conjunto de instruções complexo). O ponto forte dessa filosofia é justamente a retrocompatibilidade. Quanto ao consumo energético, um estudo de referência que mediu processadores ARM e x86 reais concluiu que as diferenças de consumo observadas na prática vêm principalmente de escolhas de microarquitetura e do ponto de projeto desempenho/eficiência (processadores ARM historicamente otimizados para baixo consumo; x86 para alto desempenho) — não do fato de o conjunto de instruções ser CISC ou RISC em si [6]. O que de fato é uma consequência direta do paradigma CISC é o acúmulo constante de complexidade no hardware ao longo de décadas de evolução — o que exige mais lógica de decodificação e microcódigo, ainda que o efeito disso sobre o consumo energético seja mais modesto do que costuma ser popularmente descrito.

Figura pendente

diagrama de Venn com círculos concêntricos representando 8086 ⊂ 286 ⊂ 386 ⊂ 486 ⊂ Pentium ⊂ Core

3.4 Endereçamento e a barreira dos 32 bits

A quantidade de bits que um processador utiliza para endereçar a memória determina diretamente a quantidade máxima de memória RAM que esse processador é capaz de acessar.

Um processador de arquitetura de 32 bits consegue endereçar, no máximo, cerca de 4 GB de memória RAM. Essa é uma limitação estrutural da arquitetura, não da quantidade de memória fisicamente instalada na placa-mãe.

Exemplo. É possível instalar 8 GB de memória RAM em um computador equipado com um sistema operacional de 32 bits; entretanto, esse sistema só conseguirá endereçar e utilizar até 4 GB — o restante permanece inacessível, por ausência de endereços suficientes para representá-lo. Esse foi historicamente um problema real de suporte técnico: computadores com memória fisicamente instalada, mas parcialmente inutilizável, por limitação da arquitetura do sistema operacional instalado, não do processador.

A superação dessa barreira motivou a extensão da arquitetura x86 para 64 bits. Em vez de propor uma arquitetura inteiramente nova, a AMD estendeu a arquitetura x86 de 32 bits existente, preservando toda a sua retrocompatibilidade e adicionando um modo de operação de 64 bits — resultando na arquitetura **AMD64**, posteriormente adotada por toda a indústria (incluindo a Intel, que tinha à época uma arquitetura de 64 bits concorrente e incompatível, a Itanium/IA-64, e só adotou a extensão da AMD em 2004, sob o nome “Intel 64”) sob a denominação genérica **x64** [7].

Uma consequência prática dessa relação de subconjunto (x86 $\subset$ x64) diz respeito à compatibilidade de software: um sistema operacional de 64 bits é capaz de executar tanto programas de 64 bits quanto programas de 32 bits, por manter a retrocompatibilidade; já um sistema operacional de 32 bits não é capaz de executar programas de 64 bits, por não reconhecer as instruções e os endereços estendidos dessa arquitetura mais recente.

Figura pendente

comparação de páginas de download de um mesmo programa, mostrando as versões disponíveis para x86, x64 e ARM

3.5 RISC: o paradigma da simplicidade

Em contraposição à filosofia CISC, existe uma segunda abordagem de projeto de processadores, baseada na premissa oposta: construir um processador cujo conjunto de instruções seja o mais simples e reduzido possível. A essa família dá-se o nome de **RISC** (*Reduced Instruction Set Computer*, computador com conjunto de instruções reduzido).

O ponto forte do RISC, em comparação ao CISC, é a simplicidade do hardware de decodificação. É comum associar essa simplicidade a um menor consumo energético — e, na prática, processadores RISC (como os da famí-

lia ARM) costumam ser mais eficientes energeticamente do que processadores CISC (x86/x64) — mas a pesquisa específica sobre o tema mostra que essa diferença vem majoritariamente de escolhas de microarquitetura e do mercado-alvo de cada família (ARM historicamente otimizada para dispositivos móveis; x86/x64 para desempenho bruto), não de uma propriedade intrínseca do conjunto de instruções em si [6]. A contrapartida real e direta da simplicidade do conjunto de instruções é que instruções complexas, que num processador CISC seriam executadas diretamente em hardware, precisam ser decompostas em uma sequência de instruções mais simples — transferindo parte da complexidade do hardware para o software (compiladores e demais camadas de programação).

Exemplo. A instrução "5 × 3" pode ser implementada de duas formas distintas. Em um processador com uma instrução de multiplicação dedicada em hardware, essa operação é resolvida diretamente. Em um processador cujo conjunto de instruções contém apenas soma, a mesma operação é obtida por meio de somas sucessivas (5 + 5 + 5), decompostas pelo compilador antes da execução. O resultado final é idêntico; o caminho até ele é que muda — e é essa diferença de caminho que distingue, na prática, um processador CISC de um processador RISC.

A principal família comercial de processadores RISC de propósito geral é a **ARM**, presente em smartphones, no Raspberry Pi e, desde 2020, também em notebooks (como os MacBooks equipados com chips da série M — M1, M2, M3, M4, M5, lançados a partir de novembro de 2020) e em processadores da linha Snapdragon [8].

Uma segunda família RISC, voltada não para computação de propósito geral, mas para **microcontroladores** embarcados, é a **AVR**, presente em chips como o ATmega328 (usado nas placas Arduino Uno) e nos microcontroladores PIC.

Diferentemente da relação entre x86 e x64 (em que uma arquitetura está contida na outra), ARM e AVR não compartilham conjuntos de instruções entre si: são duas famílias RISC distintas, sem relação de subconjunto uma com a outra, cada uma otimizada para um tipo de aplicação diferente.

Exemplo. Um semáforo de trânsito exige um hardware de controle que permaneça ligado ininterruptamente, com baixíssimo consumo energético e alta confiabilidade, mas sem necessidade de grande poder de processamento — um cenário típico de aplicação para microcontroladores AVR. Um notebook usado para tarefas de escritório e navegação em rede, por outro lado, demanda maior poder computacional e tipicamente adota arquiteturas x86/x64 ou ARM, conforme o caso de uso.

Figura pendente

fotos comparativas de um chip Cortex-A (Raspberry Pi), um Snapdragon (smartphone) e um ATmega328 (Arduino Uno)

3.6 Comparação entre paradigmas de arquitetura

A tabela a seguir resume as principais diferenças entre os paradigmas CISC e RISC:

Característica	CISC	RISC
Conjunto de instruções	Extenso e complexo, acumulado ao longo de décadas	Reduzido e simplificado, por projeto
Consumo energético	Historicamente mais alto (efeito majoritariamente de microarquitetura, não da ISA em si [6])	Historicamente mais baixo (idem)
Retrocompatibilidade	Ponto forte histórico (ex.: x86/x64)	Não é o foco de projeto
Onde reside a complexidade	No hardware	Transferida para o software/compilador
Exemplos de arquitetura	x86, x64	ARM, AVR
Aplicações típicas	Desktops, notebooks, servidores tradicionais	Smartphones, computação móvel, automação embarcada

Arquitetura (o paradigma de projeto) e implementação específica não devem ser confundidas: nem todo processador CISC pertence à família x86/x64, e nem todo processador RISC pertence à família ARM. A relação correta é de classe e subclasse.

CISC e RISC são, portanto, **paradigmas** — formas distintas de resolver o mesmo problema geral (processar instruções), da mesma maneira que bicicleta, carro e ônibus são meios de transporte que resolvem o mesmo problema (deslocamento) seguindo lógicas de projeto totalmente diferentes, cada uma adequada a um conjunto de circunstâncias.

3.7 Tendências de mercado: computação de borda e migração para ARM

A distinção entre arquitetura e organização (Seção 3.1) também explica decisões recentes de mercado. Um exemplo é a diferenciação entre versões "Pro" e "não Pro" de um mesmo smartphone: ambas podem compartilhar a mesma arquitetura de processador, mas a versão "Pro" pode incluir uma NPU (ver Capítulo 1, Seção 1.10.5) mais capaz, permitindo que tarefas de inteligência artificial — como transcrição e tradução simultânea de voz —

sejam processadas localmente no aparelho, sem depender de um servidor remoto. Essa diferenciação é uma decisão de **organização**, não de arquitetura: o processamento no dispositivo (computação de borda, ou *edge computing*) versus o processamento na nuvem é uma escolha de hardware e de projeto de produto, não uma mudança no conjunto de instruções do processador.

A adoção da arquitetura ARM tem se expandido para além de dispositivos móveis. O projeto **Asahi Linux**, em desenvolvimento desde o início da década de 2020, busca portar o sistema operacional Linux para funcionar de forma plena nos chips ARM da série M da Apple, originalmente destinados exclusivamente ao macOS [9].

Figura pendente

captura de tela de um site de hospedagem em nuvem mostrando opções de servidor com processador ARM ao lado de opções x86/x64

3.8 Síntese do capítulo

Este capítulo formalizou a distinção entre arquitetura (o conjunto de instruções de um processador) e organização (sua implementação em hardware), aprofundou a arquitetura de von Neumann introduzida no Capítulo 1, e apresentou os dois grandes paradigmas de projeto de processadores — CISC, representado pela família x86/x64, e RISC, representado pelas famílias ARM e AVR — situando-os no contexto de tendências atuais de mercado, como a computação de borda e a possível migração de servidores para arquitetura ARM.

O princípio de modularidade apresentado no Capítulo 1 — a ideia de que um computador é um conjunto de módulos substituíveis, e de que o diagnóstico técnico consiste em isolar qual módulo é responsável por uma falha — permanece como o fio condutor de toda a disciplina. Com a arquitetura do conjunto de instruções formalizada, o Capítulo 4 fecha o bloco “processador” (Capítulos 2–4) tratando da organização interna de um processador real e de sua evolução — Lei de Moore, calor, multicore — antes de o livro seguir para a memória, o sistema operacional, o hardware físico e a eletricidade que sustentam fisicamente tudo o que foi estudado até aqui. O Capítulo 13, mais adiante, retoma esses mesmos atributos sob a metodologia concreta de benchmark.

3.9 Referências

1. STALLINGS, William. *Arquitetura e Organização de Computadores*. 10. ed. São Paulo: Pearson Education do Brasil, 2018, seção 1.1 “Organização e Arquitetura”.

2. WIKIPEDIA. “Intel 8086.” Disponível em: https://en.wikipedia.org/wiki/Intel_8086; “Intel 4004.” Disponível em: https://en.wikipedia.org/wiki/Intel_4004; “Intel 8008.” Disponível em: https://en.wikipedia.org/wiki/Intel_8008.
3. STALLINGS, William. *Arquitetura e Organização de Computadores*. 10. ed. São Paulo: Pearson Education do Brasil, 2018.
4. TEDIUM. “Why Intel Couldn’t Trademark Numbers Anymore.” Disponível em: https://tedium.co/2017/05/18/intel-386-486-trademark-battle
5. TECHPOWERUP. “Intel Confirms ‘Core i-’ Getting Replaced by ‘Core Ultra’ For Upcoming Meteor Lake Processors.” Disponível em: https://www.techpowerup.com/308061/; ENGADGET. “Intel drops ‘i’ processor branding after 15 years, introduces ‘Ultra’ for higher-end chips.” Disponível em: https://www.engadget.com/intel-drops-i-processor-brand html.
6. BLEM, Emily; MENON, Jaikrishnan; SANKARALINGAM, Karthikeyan. “Power Struggles: Revisiting the RISC vs. CISC Debate on Contemporary ARM and x86 Architectures.” *HPCA 2013*. Disponível em: https://research.cs.wisc.edu/vertical/papers/2013/hpca13-isa-power-struggl pdf.
7. WIKIPEDIA. “x86-64.” Disponível em: https://en.wikipedia.org/wiki/X86-64.
8. Data de lançamento dos MacBooks com chip Apple M1 (novembro de 2020): Apple Newsroom (anúncio oficial); para o histórico mais amplo de tentativas anteriores de notebooks ARM, ver retrospectivas de imprensa especializada (Windows Central, How-To Geek) — link exato a confirmar pelo autor.
9. ASAHI LINUX. Página “About” do projeto. Disponível em: https://asahilinux.org/about/.

4 Processador: Organização Interna e Evolução

Neste capítulo você vai estudar a evolução histórica e os limites físicos do processador moderno — da lei de Moore ao problema do calor —, a distinção entre arquiteturas RISC e CISC, a organização interna de um processador multicore (núcleos, threads, cache e pipeline), e os critérios de soquete, geração e compatibilidade que determinam a longevidade de uma plataforma no mercado. O Capítulo 13 dá sequência a este, aplicando esses conceitos a ferramentas concretas de benchmark e dimensionamento.

4.1 O processador como peça central da especificação

Dentro da tarefa de especificar uma máquina para uma determinada demanda — um dos objetivos centrais desta disciplina —, o processador (CPU, *Central Processing Unit*) ocupa posição crítica. A escolha da CPU não é uma decisão isolada: ela impõe restrições sobre a placa-mãe (via soquete e chipset) e sobre a fonte de alimentação (via potência exigida, tema do Capítulo 11), e é frequentemente o fator que define o desempenho final da máquina — o chamado **gargalo** (*bottleneck*) da especificação.

Se o processador fosse tratado como uma classe de programação, seus atributos característicos seriam:

- **Arquitetura** (RISC ou CISC, tratada na Seção 4.2)
- **Litografia** (o processo de fabricação, tratado na Seção 4.3)
- **Fabricante**
- **Soquete** (o encaixe físico na placa-mãe, tratado na Seção 4.7)
- **Geração**
- **Número de núcleos** (*cores*)
- **Número de threads**
- **Potência** (em watts)
- **Clock** (frequência de trabalho)

Esses atributos formam a “ficha técnica” completa de um processador, tal como aparece nos anúncios comerciais. O restante deste capítulo desenvolve cada um desses atributos e apresenta a metodologia — baseada em *benchmark* — que permite comparar processadores diferentes de forma objetiva.

Figura pendente

anúncio real de processador com os atributos arquitetura, litografia, soquete, núcleos, threads, potência e clock destacados

4.2 RISC e CISC no mercado: da irrelevância à decisão de compra

O Capítulo 3 (§3.3–3.6) tratou em profundidade a distinção entre as arquiteturas CISC e RISC — retrocompatibilidade, famílias comerciais, eficiência energética. Esta seção não repete essa base: parte dela para mostrar como uma distinção que já foi irrelevante para o técnico de manutenção virou, nos últimos anos, uma decisão real de especificação.

Por muito tempo, essa escolha de arquitetura era irrelevante para o técnico de manutenção de desktops e notebooks: processadores RISC ficavam restritos a smartphones, onde a eficiência energética é decisiva para a autonomia de bateria. Esse cenário mudou. Hoje, processadores RISC (como os chips Apple Silicon e os processadores Snapdragon) também competem no mercado de desktops e laptops. Consequentemente, ao especificar computadores portáteis — por exemplo, para uma equipe de professores que leva o equipamento para dar aula fora do laboratório —, a arquitetura passa a ser uma escolha técnica relevante: um computador de arquitetura CISC de alto desempenho pode exigir tantos acessórios de suporte (fonte robusta, ventilação adicional) que se torna impraticável para uso móvel, enquanto um equivalente RISC entrega autonomia de bateria muito superior.

> **Atualidade de mercado — o chip RISC da NVIDIA para computação de IA local**
>
> Em 1º de junho de 2026, durante a feira Computex em Taipei, a NVIDIA — historicamente fabricante de placas de vídeo GeForce — anunciou o **RTX Spark**, seu primeiro chip de arquitetura RISC (especificamente Arm) voltado a computadores pessoais, com memória unificada (nos moldes do que a Apple já pratica em seus chips Apple Silicon) e capacidade de até 128 GB de memória compartilhada entre CPU, GPU e NPU [1]. A NVIDIA e a Microsoft, em conjunto, descreveram o lançamento como parte de uma “reinvenção do computador”.
>
> A novidade central não é apenas a arquitetura RISC em si, mas o fato de que o chip foi projetado com instruções voltadas ao chamado *loop agêntico*: da mesma forma que o ciclo clássico de busca–decodificação–execução (Seção 4.6) fundamenta o funcionamento de qualquer CPU, o RTX Spark inclui, em nível de processador, suporte a um ciclo de raciocínio–execução–validação voltado à execução local de agentes de inteligência artificial (assistentes de programação e automação que operam diretamente

na máquina do usuário, sem depender inteiramente de processamento em nuvem).

O chip já nasce com parcerias fechadas com fabricantes de notebooks como Lenovo, HP, Dell (Microsoft Surface), Asus e MSI, colocando pressão competitiva simultânea sobre Intel, AMD e Apple. Trata-se de uma notícia de mercado recente — o leitor deste livro deve tratá-la como indicativo de tendência, não como fato estático: a configuração exata de produtos comerciais baseados nesse chip deve ser verificada em fontes atualizadas no momento da leitura.

4.3 A Lei de Moore e os limites físicos da litografia

Historicamente, observou-se que os processadores lançados pela Intel dobravam a quantidade de transistores em relação ao modelo anterior a cada aproximadamente dois anos. Essa observação empírica foi formalizada como meta de desenvolvimento da indústria e ficou conhecida como **lei de Moore**.

Como o transistor é a unidade fundamental de construção de portas lógicas, circuitos aritméticos, circuitos de controle e circuitos de memória, um processador com o dobro de transistores tende a oferecer maior poder computacional. O gráfico histórico de contagem de transistores por chip, de 1970 a meados da década de 2010, mostra crescimento de milhares para dezenas de bilhões de transistores por chip.

Fabricante/arquitetura	Litografia aproximada
Intel, 12ª geração	10 nm
AMD Zen 3 (Ryzen série 5000)	7 nm
AMD Zen 4 (Ryzen série 7000)	5 nm
Apple Silicon (M1)	5 nm
Qualcomm Snapdragon (gerações recentes)	~3 nm

Dados de litografia levantados em 2026; a litografia real muda a cada nova geração de produto [3].

Reduzir a litografia traz dois benefícios simultâneos e vinculados por física básica: como a velocidade de deslocamento do elétron é aproximadamente constante, diminuir a distância física que o elétron percorre entre terminais do transistor reduz o tempo de processamento (maior velocidade) e reduz a perda de energia por efeito Joule (maior eficiência energética). Um processador fabricado numa litografia menor é, por concepção de projeto, mais rápido e mais eficiente do que um equivalente fabricado numa litografia maior.

O processo de fabricação. O silício, elemento com quatro elétrons na camada de valência, é abundante — extraído da areia, não de jazidas ra-

ras — e forma cristais estáveis por ligação covalente entre átomos vizinhos. Após purificado e cristalizado, o silício é fatiado em discos finos chamados *wafers*. Sobre o *wafer*, um processo óptico de precisão chamado **litografia** projeta, com o auxílio de lasers e lentes especiais, a imagem do circuito do processador, "queimando" no silício o desenho dos transistores já interligados em portas lógicas. O processo de **dopagem** (introdução controlada de impurezas, como boro, na estrutura cristalina) cria as regiões de tipo N e tipo P que formam diodos e transistores. Concluída a gravação, o *wafer* é fatiado em unidades individuais (os *dies*), que recebem encapsulamento protetor e se tornam o chip vendido no mercado, no formato que se conecta ao soquete da placa-mãe.

Historicamente, a Intel controlava tanto o projeto quanto a fabricação (fundição) de seus próprios chips. Outros fabricantes — AMD, Apple, Qualcomm — projetam seus chips mas terceirizam a fabricação para fundições especializadas, principalmente a TSMC (*Taiwan Semiconductor Manufacturing Company*), sediada em Taiwan. A dificuldade da Intel em reduzir sua própria litografia abaixo de 10 nm é um fator relevante da perda de competitividade da empresa nos últimos anos, a ponto de ela também ter passado a recorrer a fundições terceirizadas para parte de sua produção. A concentração geográfica dessa capacidade de fabricação de semicondutores de ponta em Taiwan é, adicionalmente, um fator de tensão geopolítica relevante, dado o contexto de disputa territorial entre Taiwan e a China — um ponto que tem desdobramentos diretos sobre preço e disponibilidade de chips no mercado global.

Figura pendente

fotografia de um wafer de silício antes e depois do processo de litografia] [IMAGEM: gráfico da contagem de transistores por chip ao longo do tempo, 1970–2016 — se adaptado de fonte de terceiros (ex.: Our World in Data), creditar a fonte e confirmar a licença antes de publicar; alternativa: construir com dados brutos próprios/públicos para evitar qualquer dúvida de direito autoral

4.4 Calor, thermal throttling e o fim do núcleo único

Alocar mais transistores no mesmo espaço físico tem uma consequência inevitável: mais calor gerado na mesma área. Processadores das gerações mais antigas (até o início dos anos 2000) nem sempre possuíam proteção térmica — e a diferença entre ter ou não essa proteção podia ser dramática: um vídeo de demonstração histórico e amplamente citado (Tom's Hardware, ~2000–2001) comparou um Intel Pentium III e um AMD Athlon "Thunderbird", ambos sem *cooler*, sob carga. O Pentium III, equipado com um monitor térmico que desligava o processador ao atingir a temperatura crítica, travou o sistema mas sobreviveu; o Athlon, sem qualquer proteção térmica,

literalmente queimou e soltou fumaça em menos de um segundo, chegando a 370 °C no núcleo [4].

Os processadores modernos resolvem esse risco com o mecanismo de **thermal throttling** (acelerador térmico): ao detectar que a temperatura se aproxima de um limite pré-definido, o processador reduz automaticamente sua frequência de processamento; ao esfriar, ele volta a acelerar. O resultado é um ciclo contínuo de aceleração e desaceleração, governado pela temperatura.

O *thermal throttling* deixou de ser apenas uma medida de emergência e passa a ser usado como característica de projeto: um MacBook sem ventoinha reduz o clock progressivamente conforme aquece, enquanto um MacBook Pro com ventoinha ativa consegue sustentar clocks mais altos por mais tempo, adiando o momento do *throttling*.

O limite que originou o multicore. No início dos anos 2000, a Intel chegou a planejar um Pentium 4 de 4 GHz, mas cancelou o lançamento em outubro de 2004: a dissipação de calor necessária tornava o produto inviável para os computadores da época [5]. A solução encontrada pela indústria não foi continuar aumentando a densidade de transistores num único núcleo, mas **duplicar o número de núcleos de processamento** dentro do mesmo encapsulamento — cada um mais simples e mais frio do que seria um único núcleo hipertrofiado. Nasceu assim a era **multicore** no mercado de desktop, por volta de 2005, com o lançamento do Pentium D pela Intel e do Athlon X2 pela AMD [6]. Essa mudança de direção arquitetural se mantém até hoje: não houve retorno ao paradigma de núcleo único, apenas a adição de novas unidades de processamento especializadas (GPU, NPU), tratadas ao final deste capítulo.

Figura pendente

gráfico comparando temperatura e FPS de um processador com e sem cooler durante um jogo

4.5 A era multicore: núcleos, cache e hyper-threading

Um **core** (núcleo) é uma unidade de processamento física — hardware. Um processador multicore contém, dentro de um único encapsulamento, múltiplas cópias completas da estrutura de processamento (unidade lógico-aritmética, registradores, unidades de controle de fluxo), cada uma com sua própria memória cache de nível 1 (L1) e nível 2 (L2). Externamente aos núcleos, mas ainda dentro do chip, fica a cache de nível 3 (L3), compartilhada entre todos os núcleos — funcionando como uma mesa de trabalho comum para a troca de dados entre eles quando uma tarefa é dividida entre múltiplos núcleos.

Nota sobre controle de qualidade. Nem todo processador de uma mesma linha de produção nasce igual: após a fabricação, cada chip passa

por uma bancada de testes. Esse processo, chamado *binning*, é real e bem documentado na indústria [7]: às vezes, um chip com todos os núcleos aprovados no controle de qualidade é vendido como a linha superior (por exemplo, um "i7"), enquanto um chip do mesmo *die*, com núcleos defeituosos desabilitados, é vendido como um produto de linha inferior ("i5" ou "i3"). Essa não é, porém, a explicação universal da diferenciação entre linhas: em boa parte dos casos — especialmente em gerações mais recentes — i5 e i3 são fabricados a partir de *dies* fisicamente menores e diferentes do i7, não do mesmo chip com núcleos desabilitados.

Uma **thread**, por sua vez, é uma unidade lógica — uma simulação em software da existência de mais processadores do que os fisicamente presentes. A tecnologia responsável por essa simulação é chamada de **hyper-threading** (nomenclatura da Intel) ou **SMT**, *simultaneous multi-threading* (nomenclatura da AMD). Um núcleo físico com hyper-threading de grau 2 é enxergado pelo sistema operacional como dois processadores lógicos.

Exemplo. Um anúncio real de processador Intel Core i5-12400F (12ª geração) especifica 6 núcleos físicos (*cores*) com hyper-threading, totalizando 12 threads [8] — ou seja, o sistema operacional identifica 12 processadores lógicos disponíveis, embora fisicamente existam apenas 6.

Essa complexidade, no entanto, tem um custo que sobe para as camadas de software: programação paralela (a técnica de dividir um algoritmo para ser executado simultaneamente em múltiplos núcleos) é uma disciplina distinta e mais difícil do que a programação sequencial convencional, e a maioria dos programas aplicativos comuns não é otimizada para tirar proveito de mais do que quatro a seis núcleos simultâneos. A consequência prática desse descompasso — poder computacional disponível, mas ocioso — é discutida no Capítulo 13.

Figura pendente

fotografia (die shot) de um processador com quatro núcleos, cache L1/L2 por núcleo e cache L3 compartilhada identificados

4.6 Pipeline: sobreposição de estágios de execução

O ciclo básico de funcionamento de uma CPU consiste em três etapas repetidas continuamente: **busca** da instrução na memória, **decodificação** dessa instrução e **execução** da instrução decodificada. A técnica de **pipeline** consiste em sobrepor essas etapas para diferentes instruções simultaneamente, em vez de executar o ciclo completo de uma instrução antes de iniciar a próxima.

Analogia (adaptada da clássica analogia da lavanderia de Patterson & Hennessy [9]). Considere uma lavanderia com quatro estações — lavar, secar, dobrar e guardar —, em que cada etapa leva uma hora. Numa execução sem pipeline, o ciclo completo (lavar → secar → dobrar → guardar)

leva 4 horas, e o próximo ciclo só começa depois que o anterior termina inteiramente: quatro ciclos completos consumiriam 16 horas (4 etapas × 1 hora × 4 ciclos), com cada máquina ficando ociosa na maior parte do tempo. Com pipeline, assim que a máquina de lavar termina uma carga e a transfere para a secadora, ela já recebe uma nova carga de roupa suja — nenhum equipamento fica parado. Nesse esquema, em sete horas obtêm-se quatro ciclos completos de saída, contra as dezesseis horas necessárias sem sobreposição — um ganho de mais do que o dobro de produtividade utilizando exatamente o mesmo hardware, apenas evitando ociosidade.

É essa mesma lógica de aproveitamento de estágios ociosos do circuito de busca–decodificação–execução que permite ao hyper-threading (Seção 4.5) fazer um único núcleo físico atender a duas instruções em estágios diferentes do pipeline ao mesmo tempo, simulando dois processadores lógicos. O termo *pipeline* não é exclusivo da arquitetura de computadores — é usado de forma equivalente em engenharia de produção industrial para descrever qualquer processo organizado em estágios reaproveitáveis e encadeados.

Figura pendente

diagrama de linha do tempo mostrando quatro estágios de pipeline sobrepostos ao longo de sete unidades de tempo

4.7 Soquete, geração e compatibilidade de mercado

O Capítulo 8 (§8.4) tratou o soquete do ponto de vista físico — o conector, a compatibilidade entre fabricante e geração, as diferenças entre os soquetes Intel e AMD. Esta seção parte dessa base para tratar a mesma compatibilidade sob a ótica de mercado: o impacto que a estratégia de soquete de cada fabricante tem sobre a longevidade de uma plataforma e sobre a recomendação de compra ao cliente.

Historicamente, a AMD tem mantido o mesmo soquete por múltiplas gerações de processador — o soquete AM4 atendeu várias gerações consecutivas de Ryzen, e o soquete AM5, lançado mais recentemente, teve seu suporte estendido pela AMD até o ano de 2029 [10]. Isso significa que uma placa-mãe AM5 fabricada hoje continuará aceitando processadores lançados anos depois. A Intel, por sua vez, tende a alterar o soquete a cada uma ou duas gerações.

Consequência prática para o técnico. Diante de uma placa-mãe com defeito comprovado, cujo processador continua funcional, a estratégia de diagnóstico modular (transferir o processador para outra placa-mãe compatível) é significativamente mais viável em um ecossistema AMD — em que placas-mãe compatíveis com processadores mais antigos continuam disponíveis no mercado — do que em um ecossistema Intel, no qual encontrar uma placa-mãe compatível com um processador de poucos anos atrás pode

ser difícil e caro o suficiente para tornar mais econômico descartar o processador funcional e comprar um conjunto novo (processador e placa-mãe).

Esse tipo de escolha de projeto — que reduz a vida útil economicamente viável de um componente ainda funcional — se aproxima do fenômeno mais amplo de **obsolescência programada**, observável em outras indústrias. Do ponto de vista puramente técnico, isso reforça por que a análise de soquete e longevidade de plataforma deve fazer parte da recomendação de compra ao cliente, e não apenas a comparação de desempenho bruto — que é tratada, com apoio de benchmark, no Capítulo 13. Cabe notar, entretanto, que a Intel frequentemente compensa essa desvantagem de longevidade praticando preços mais agressivos, de modo que a escolha entre os dois fabricantes é sempre uma decisão de custo-benefício, não uma resposta única.

Figura pendente

foto comparativa de soquete AMD (AM4/AM5) e soquete Intel (LGA) com os pinos/contatos visíveis

---

4.8 Síntese do capítulo

Este capítulo apresentou o processador como peça central da tarefa de especificação de um computador: a distinção entre arquiteturas RISC e CISC, os limites físicos impostos pela litografia e pelo calor — e a mudança de rumo que esse limite provocou, do núcleo único ao multicore —, a organização interna de um processador multicore (núcleos, threads, cache e pipeline) e os critérios de soquete, geração e compatibilidade que determinam a longevidade de uma plataforma no mercado. O Capítulo 13 dá sequência direta a este, aplicando esses mesmos atributos — arquitetura, litografia, núcleos, soquete — à metodologia concreta de benchmark que permite compará-los de forma objetiva e dimensionar um computador dentro de um orçamento.

4.9 Referências

1. NVIDIA NEWSROOM/GEFORCE NEWS. “NVIDIA at COMPUTEX 2026: NVIDIA RTX Spark…” Disponível em: `https://www.nvidia.com/en-us/geforce/news/computex-2026-nvidia-geforce-rtx-announcements/`; TOM’S HARDWARE. “Nvidia unveils RTX Spark Superchip for laptops and desktop PCs at Computex 2026.” Disponível em: `https://www.tomshardware.com/laptops/nvidia-unveils-rtx-spark-superchip-at-comp`

2. Dados de litografia por fabricante/arquitetura conferidos via TechInsights/TrendForce/roteiro de nós de processo TSMC; cobertura em notebookcheck.net e techpowerup.com.
3. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). "A sheet of paper is about 100,000 nanometers thick." Disponível em: https://x.com/NIST/status/1792590000179064919; NATIONAL NANOTECHNOLOGY INITIATIVE. "Just How Small Is Nano?" Disponível em: https://www.nano.gov/about-nanotechnology/just-how-small-is-nano
4. TOM'S HARDWARE. "Hot Spot: How Modern Processors Cope With Heat Emergencies." Disponível em: https://www.tomshardware.com/reviews/hot-spot,365-4.html.
5. CHANNEL INSIDER. "Intel Cancels 4-GHz Pentium 4." Disponível em: https://www.channelinsider.com/news-and-trends/intel-cancels-4-ghz- PC PERSPECTIVE. "Intel Cancels the 4GHz Prescott." Disponível em: https://pcper.com/2004/10/intel-cancels-the-4ghz-prescott/.
6. INTEL NEWSROOM. "Dual Core Era Begins, PC Makers Start Selling Intel-Based PCs." 18 abr. 2005. Disponível em: https://www.intel.com/pressroom/archive/releases/2005/20050418comp.htm.
7. Documentação técnica geral sobre segmentação de produto e *binning* de CPUs (Intel/AMD); artigos técnicos sobre *yield* de semicondutores — fonte específica a confirmar pelo autor.
8. INTEL. Especificações oficiais, "Intel® Core™ i5-12400F Processor (18M Cache, up to 4.40 GHz)." Disponível em: https://www.intel.com/content/www/us/en/products/sku/134587/intel-core-i512400f-proces specifications.html.
9. PATTERSON, David A.; HENNESSY, John L. *Computer Organization and Design: The Hardware/Software Interface* — RISC-V Edition. Cambridge, MA: Morgan Kaufmann, 2017. ISBN 978-0-12-812275-4.
10. TECHPOWERUP. "AMD Announces Socket AM5 Longevity till 2029." Disponível em: https://www.techpowerup.com/349541/amd-announces-socket VIDEOCARDZ. Disponível em: https://videocardz.com/newz/amd-extends-am

5 Memória

Neste capítulo você vai aprofundar os diversos tipos de memória de um computador moderno — da memória RAM volátil, passando pela hierarquia de cache dentro do processador, até a memória secundária persistente em suas variantes magnética, de estado sólido (flash) e ótica — e a relação direta entre a natureza física da célula de memória e o desempenho do sistema. O Capítulo 1 já introduziu a distinção entre memória RAM e armazenamento secundário e apresentou, na Seção 1.10, a pirâmide de hierarquia de memória; este capítulo não repete essa introdução, mas aprofunda cada uma de suas camadas.

5.1 Da célula de memória às características da memória

Um princípio orienta toda a discussão deste capítulo: **as características de uma memória — velocidade, custo, volatilidade, durabilidade — não são arbitrárias; elas decorrem diretamente da natureza física da sua célula de memória**, isto é, do mecanismo usado para armazenar um único bit.

Uma célula de memória é, em essência, um combinado: um dispositivo físico ao qual se atribui o significado “0” ou “1”. Esse combinado pode ser implementado de formas radicalmente diferentes — um circuito com transistores, um capacitor carregado, uma região magnetizada, um ponto que reflete ou não reflete luz —, e é exatamente essa diferença de implementação que separa memória primária de memória secundária, e que separa, dentro da própria memória secundária, um HD de um SSD ou de uma mídia ótica.

Para transformar uma célula isolada num chip de memória, basta replicá-la em grade — cada célula recebe um endereço próprio (uma organização linha/coluna, como uma planilha invertida: primeiro a linha, depois a coluna) — e conectar todas as células aos mesmos barramentos de endereço e de dado. É esse compartilhamento físico de barramento que explica por que a RAM tem acesso *aleatório*: o endereço e o dado chegam, por tensão elétrica, simultaneamente a todas as células ao mesmo tempo; apenas a célula cujo endereço foi ativado responde, gravando ou lendo o dado — não há deslocamento físico até a posição, ao contrário do que ocorre no HD (Seção 5.10). Por isso o tempo de acesso é o mesmo, não importa qual endereço seja escolhido.

Nota prática. Sempre que um chip de hardware exibir uma estrutura visivelmente repetitiva de blocos idênticos — como os retângulos pretos de um pente de RAM —, é sinal de que aquele componente é memória: cada bloco é a réplica de uma mesma célula básica, repetida até atingir a capacidade total do chip.

Figura pendente

esquema comparativo das quatro células de memória estudadas no capítulo — flip-flop, capacitor, domínio magnético, floating gate

5.2 Memória estática (SRAM)

Da década de 1950 até o início dos anos 1970, a tecnologia dominante de memória primária foi a **memória de núcleo magnético**; a partir de então, esse papel passou para a memória dinâmica (DRAM, Seção 5.3), situação que permanece até hoje [1]. A memória **estática**, apresentada nesta seção, nunca foi, portanto, a tecnologia dominante de memória primária de propósito geral — seu uso sempre se concentrou nas memórias cache (Seção 5.8), justamente pelo motivo de custo/densidade explicado adiante nesta seção. Numa célula estática, o bit é armazenado em um circuito chamado **flip-flop**, construído pela combinação de portas lógicas (AND, OR, NAND, NOT etc.), que por sua vez são construídas com transistores — chaves eletrônicas feitas de semicondutores dopados (tipo N e tipo P), cuja matéria-prima básica é o silício, um dos elementos mais abundantes do planeta.

Uma vez que um valor é escrito num flip-flop, ele permanece estável indefinidamente enquanto houver energia — daí o nome “estática”. Essa estabilidade é o que torna a **SRAM** (*Static RAM*) extremamente rápida.

A contrapartida da SRAM é o custo: por unidade de armazenamento, uma célula estática é significativamente mais cara e menos densa do que a alternativa dinâmica apresentada a seguir. Por isso, hoje a SRAM não é mais usada como memória primária de um computador — sua aplicação atual se restringe às memórias **cache** dentro do processador (Seção 5.8).

5.3 Memória dinâmica (DRAM) e a operação de refresh

A memória **dinâmica** (**DRAM**, *Dynamic RAM*) armazena cada bit em um capacitor — um componente que guarda carga elétrica, e não um valor lógico estável por natureza. Um capacitor perde carga ao longo do tempo, o que significa que a informação armazenada se degrada progressivamente.

Esse “completar periodicamente” é a operação de ***refresh***: em intervalos regulares, o controlador de memória relê e regrava cada célula da DRAM para restaurar a carga elétrica antes que ela caia a ponto de tornar ambíguo se o bit é 0 ou 1. O termo é o mesmo usado para a atualização de uma página

web (F5): uma releitura periódica que traz o conteúdo de volta ao estado esperado.

A necessidade de *refresh* torna a DRAM mais lenta do que a SRAM. Em compensação, sua célula é mais simples (um único capacitor, em vez de um circuito completo de portas lógicas), o que permite maior **densidade de memória** — mais bits armazenados no mesmo espaço físico — e, consequentemente, um custo por bit muito menor.

É essa relação custo–densidade–velocidade que explica por que a DRAM, apesar de mais lenta que a SRAM, tornou-se a tecnologia usada como memória RAM principal de praticamente todos os computadores modernos.

5.4 Sincronismo e a família SDRAM/DDR

A geração seguinte de memória dinâmica introduziu o **sincronismo**: em vez de operar de forma assíncrona, a memória passou a coordenar suas operações com um sinal de **clock** — um sinal digital que alterna regularmente entre nível alto e nível baixo. As operações de leitura e escrita só são processadas quando o clock está em um determinado estado, o que evita instabilidades elétricas durante a transição de valores no circuito. Essa é a **SDRAM** (*Synchronous Dynamic RAM*): dinâmica (usa capacitores e precisa de *refresh*) e síncrona (usa clock).

A frequência desse clock, medida em megahertz (MHz) ou gigahertz (GHz), determina quantas operações de leitura/escrita a memória realiza por segundo — é o número de frequência de trabalho estampado na embalagem de qualquer memória RAM vendida no mercado.

A partir da SDRAM, o mercado consolidou a tecnologia **DDR** (*Double Data Rate*), cujas gerações sucessivas — DDR, DDR2, DDR3, DDR4, DDR5 — dobram, a cada geração, o número de operações realizadas por ciclo de clock, além de aumentar a densidade da célula e reduzir o consumo de energia. A tabela a seguir resume as principais características discutidas em aula:

Geração	Taxa de transferência (aprox.)	Tensão de operação	Observação
DDR3	~1,6 Gb/s	1,5 V	Tecnologia madura, hoje mais cara por menor oferta
DDR4	~3,2 Gb/s	1,2 V	Tecnologia dominante no mercado atual

Geração	Taxa de transferência (aprox.)	Tensão de operação	Observação
DDR5	~4,8–6,4 Gb/s	1,1 V	Maior capacidade por módulo; ainda mais cara por menor adoção

Valores de taxa de transferência e tensão conforme especificação JEDEC [2].

CAS Latency (CL). Além da frequência de trabalho, memórias RAM anunciam também um conjunto de números chamado *timings*, o mais citado sendo o **CL** (*CAS Latency*, de *Column Address Strobe Latency*): o número de ciclos de clock que a memória leva entre receber o endereço de uma coluna de dados e efetivamente disponibilizar esse dado. Como o CL é medido em ciclos — não em tempo absoluto —, comparar o CL de memórias com frequências de trabalho diferentes não é direto: o mesmo CL pode representar tempos reais bem distintos dependendo da frequência. Na prática, diferenças de CL costumam ser imperceptíveis em uso cotidiano, ficam um pouco mais visíveis em jogos, e só se tornam significativas (até ~15% de ganho) em aplicações profissionais que manipulam grandes volumes de dados [12].

A tensão de operação cai a cada geração porque, segundo a relação P = U·I, uma tensão menor implica uma corrente menor para a mesma potência — e uma corrente menor produz menos perda de energia por efeito Joule. Como as trilhas de um circuito de memória percorrem distâncias muito curtas (ao contrário de uma linha de transmissão de energia de longa distância, onde se eleva a tensão para reduzir a corrente), reduzir a tensão é a estratégia mais eficiente para economizar energia num computador.

Exemplo. Em pesquisa de preços real feita em aula num site de varejo de hardware, uma memória DDR3 de 4 GB a 1600 MHz custava R$ 169,99, enquanto uma memória DDR4 de 8 GB — o dobro da capacidade e de uma tecnologia mais recente — custava R$ 129,49, um valor menor. A explicação não é técnica, mas de mercado: pela lei da oferta e da procura, uma tecnologia em fim de vida (DDR3) fica mais cara à medida que sua oferta diminui, mesmo sendo tecnologicamente inferior a uma tecnologia mais nova e mais barata (DDR4) que está no auge de sua adoção. O mesmo padrão se repetia com a DDR5: um kit de 2×16 GB custava R$ 1.139 — cerca do dobro do preço por gigabyte de um kit DDR4 equivalente —, mostrando que montar um computador de última geração tem um custo-benefício pior no curto prazo, embora garanta maior disponibilidade de peças de reposição no médio prazo.

Nota prática. Um computador cliente de servidor que roda sobre memória DDR3 é, em geral, mais barato de manter trocando apenas o módulo de memória (da ordem de R$ 100) do que substituindo toda a máquina por uma

DDR4 (da ordem de R$ 10.000 a R$ 20.000) — uma decisão de manutenção baseada em custo-benefício, e não apenas em desempenho bruto.

Atualização de mercado (2026). A partir da expansão de cargas de trabalho de IA e da demanda de servidores por memória em larga escala, o preço de módulos DDR5 subiu a ponto de o ganho de desempenho sobre DDR4 deixar de compensar a diferença de custo — por isso, mesmo em computadores novos, DDR4 seguiu sendo, no mercado brasileiro de 2026, o "sweet spot" de custo-benefício, e o ciclo de vida comercial do DDR4 se estendeu mais do que o normalmente esperado para uma geração de memória.

Cada geração tecnológica também define um **limite de capacidade por módulo**: no segmento de desktop/consumidor, um slot de memória DDR4 comporta tipicamente um módulo de até 32 GB (módulos de capacidade maior existem, mas apenas na forma de módulos registrados — RDIMM/LRDIMM — voltados a servidores, com engenharia e custo diferentes) [3]. Numa placa-mãe de desktop com quatro slots, a capacidade total possível de memória primária é, portanto, de até 128 GB — mas apenas dentro da mesma geração tecnológica: não é possível combinar um módulo DDR4 com um DDR5 na mesma máquina, tanto por incompatibilidade elétrica quanto física (Seção 5.6).

Figura pendente

linha do tempo DDR, DDR2, DDR3, DDR4, DDR5 com taxa de transferência e tensão

5.5 VRAM: a mesma tecnologia aplicada à GPU

A GPU, como qualquer processador, não realiza computação sem memória associada — e a indústria resolveu essa necessidade com uma memória RAM dedicada exclusivamente ao processamento gráfico, chamada **VRAM** (*Video RAM*). Em essência, a VRAM é a mesma coisa que a RAM principal — uma memória de acesso aleatório, construída com a mesma família de tecnologias — só que otimizada especificamente para o padrão de acesso do hardware gráfico.

Por serem submódulos de hardware independentes — CPU e RAM principal de um lado, GPU e VRAM de outro —, cada par evolui no seu próprio ritmo de mercado: é perfeitamente possível uma máquina ter RAM DDR3 convivendo com uma VRAM de geração bem mais recente, porque a GPU segue sua própria linha de evolução, batizada de família **GDDR** (*Graphics DDR*) em vez de DDR.

Nota prática (overhead de comunicação). Sempre que um dado precisa ser processado pela GPU, ele tem que ser copiado da RAM principal para a VRAM, processado ali, e o resultado trazido de volta — esse ir e vir consome tempo, chamado de ***overhead*** de comunicação. Em aplicações que programam a GPU diretamente (como CUDA, da NVIDIA — usado, por

exemplo, em processamento de imagens de tomografia computadorizada), esse gerenciamento de memória é feito manualmente pelo desenvolvedor, ao contrário de um programa comum que roda inteiramente na CPU. É por isso que uma VRAM maior importa na prática: quanto mais dados cabem de uma vez na VRAM, menos vezes esse overhead de ida e volta precisa acontecer (ver também Seção 5.7 sobre memória unificada, uma estratégia alternativa que elimina esse overhead aproximando fisicamente RAM e VRAM).

5.6 Overclock, underclock e a tríade memória–placa-mãe–processador

Sobrecarregar a frequência de trabalho de um componente acima de sua especificação nominal é chamado de ***overclock***; o inverso — reduzir a frequência de trabalho abaixo do especificado — é o ***underclock***.

A frequência de trabalho de uma memória RAM não é uma propriedade isolada: ela precisa estar compatibilizada com a placa-mãe e com o processador. Se a memória suporta uma frequência mais alta do que a placa-mãe ou o processador conseguem sustentar, o sistema realiza um underclock automático de todo o conjunto para garantir estabilidade — em vez de operar na velocidade máxima anunciada da memória, o conjunto passa a operar na velocidade máxima que o elo mais fraco da cadeia suporta.

Nota prática. Um computador instável, travando com frequência, pode ter como causa uma memória configurada para trabalhar acima da frequência que a combinação placa-mãe/processador suporta de forma confiável. Reduzir manualmente essa frequência (um underclock deliberado) costuma resolver o travamento ao custo de uma perda de desempenho geralmente irrelevante na prática.

Nota prática (diagnóstico de módulo defeituoso). Quando um computador não passa no POST (Capítulo 9, §9.1.1) e a memória RAM é uma das hipóteses em aberto, o diagnóstico começa pelo mais físico e barato antes de qualquer ferramenta de software — exatamente o princípio de priorizar a hipótese mais barata de verificar, discutido no Capítulo 12 (§12.4):

1. **Reencaixar o módulo.** Um encaixe parcial, sem os contatos dourados totalmente assentados no slot, já é suficiente para impedir o POST — e é a causa mais comum e mais barata de verificar.
2. **Limpar os contatos dourados** com uma borracha branca comum, num único sentido, removendo oxidação ou poeira que prejudique o contato elétrico entre o módulo e o slot.
3. **Testar módulo a módulo**, numa placa-mãe com mais de um módulo instalado: isolar cada módulo em slots diferentes para descobrir se o defeito acompanha o módulo ou fica associado a um slot específico da placa-mãe.
4. **Testar em outra placa-mãe compatível** (mesma geração DDR), quando disponível: se o POST volta a funcionar, confirma-se que o defeito é do

módulo, não da placa-mãe original.

Cada passo elimina uma hipótese antes de avançar para a próxima, do mais simples (reencaixar) para o mais trabalhoso (testar em outra máquina) — a mesma lógica de isolamento por hipóteses do Capítulo 12 aplicada especificamente à memória RAM.

5.7 Compatibilidade física entre gerações: o chanfro

Módulos de memória de gerações DDR diferentes não são apenas eletricamente incompatíveis (cada geração opera numa tensão diferente); eles são também **fisicamente incompatíveis por projeto**. Cada módulo de memória possui um entalhe — o **chanfro** — posicionado em um ponto específico ao longo do conector de pinos; o slot correspondente na placa-mãe possui uma saliência física alinhada apenas com o chanfro daquela geração específica de memória.

O objetivo é puramente preventivo: como módulos de gerações diferentes têm a mesma pinagem física mas tensões de operação distintas, conectar um módulo incompatível poderia danificar tanto a memória quanto a placa-mãe. O chanfro torna essa conexão errada fisicamente impossível — não por os módulos terem tamanhos diferentes (todos os módulos DIMM de desktop têm dimensões físicas praticamente idênticas, para simplificar o projeto das placas-mãe), mas exclusivamente pela posição do chanfro, que muda a cada geração (DDR, DDR2, DDR3, DDR4, DDR5).

Nota prática. Se um técnico sentir que está fazendo força para encaixar um componente de hardware, isso é sinal de que algo está errado — nenhum procedimento de manuseio de memória, processador ou demais componentes eletrônicos deve exigir força. É necessário parar e reavaliar o encaixe.

Existe ainda uma diferença de tamanho físico entre memórias de notebook e de desktop — as memórias de notebook são fisicamente menores, para otimizar espaço —, o que significa que um desktop pode, em certos casos, usar memória de notebook, mas o inverso nunca é possível.

Figura pendente

fotografia de dois módulos DDR de gerações diferentes lado a lado, com o chanfro em posições distintas destacado

5.8 Dual Channel e Flex Mode

Uma mesma geração de memória tem um limite de velocidade determinado pela tecnologia da sua célula. O **Dual Channel** é uma estratégia para superar esse limite sem depender de uma nova geração tecnológica: em vez de

escrever um dado inteiro sequencialmente em um único chip de memória, o sistema divide o dado (técnica chamada em inglês de *striping*) e o escreve simultaneamente em dois ou mais módulos de memória idênticos, através de canais distintos.

Exemplo. Considere uma memória capaz de escrever a 10 MB por segundo e um arquivo de 100 MB a ser carregado da memória secundária para a memória primária. Usando um único módulo, a escrita levaria 10 segundos. Dividindo o arquivo ao meio e escrevendo 50 MB em cada um de dois módulos idênticos simultaneamente, cada metade leva 5 segundos — e, como as duas escritas ocorrem em paralelo, o tempo total cai de 10 para 5 segundos. O sistema operacional arca com um pequeno custo adicional de software para fragmentar e depois recompor o dado, mas o ganho de desempenho compensa amplamente esse custo.

Para o Dual Channel funcionar de forma ideal, o cenário recomendado é usar módulos **idênticos** — mesmo fabricante, mesma frequência de trabalho, mesma tensão de operação — e instalá-los nos slots corretos indicados no manual da placa-mãe. Quando não há dois módulos idênticos disponíveis, é possível usar o **Flex Mode**: módulos de capacidades ou frequências diferentes trabalham parcialmente em Dual Channel, mas a frequência de trabalho do conjunto fica limitada pelo módulo mais lento e a capacidade em Dual Channel, pelo módulo menor.

Exemplo (Flex Mode). Considere um módulo de 10 GB e outro de 5 GB, ambos com a mesma taxa de transferência (10 MB/s). Ao escrever um arquivo de 100 MB, o sistema consegue fazer o split de 50 MB para cada módulo — e os dois terminam em 5 segundos, desde que o módulo menor ainda tenha espaço correspondente disponível. Na prática, isso significa que apenas 10 GB dessa configuração de 15 GB totais (o dobro da capacidade do módulo menor) se beneficiam da velocidade dobrada do Dual Channel; os 5 GB excedentes do módulo maior funcionam como espaço comum, sem ganho de velocidade — o sistema nivela por baixo, limitado pelo módulo menor tanto em capacidade combinada quanto em frequência de trabalho.

Nota prática (quiz de mercado). Existe diferença entre usar um único módulo de 8 GB ou dois módulos de 4 GB somando os mesmos 8 GB? Sim: se a placa-mãe suportar Dual Channel, dois módulos entregam desempenho sensivelmente melhor do que um único módulo da mesma capacidade total. Curiosamente, o mercado às vezes cobra mais caro por um kit de dois módulos idênticos (por exemplo, 2×16 GB) do que por um único módulo de capacidade equivalente (1×32 GB) — o que não invalida a vantagem técnica do Dual Channel, apenas reflete dinâmica de preços e demanda.

Atualização de mercado (setembro de 2026). Pesquisa de preços real: um módulo único de 32 GB custava R$ 3.800, um kit de 2×16 GB custava R$ 4.300, e um kit de 4×8 GB custava R$ 2.890 — confirmando que a relação de preço entre configurações não segue uma lógica simples de “quanto mais peças, mais caro”: nesse caso, o kit mais fragmentado saiu mais barato que as outras duas opções.

Nota real (arquiteturas modernas). Dual Channel não é a única estratégia da indústria para superar limites de desempenho de memória —

apenas a mais comum em placas-mãe convencionais. Um exemplo recente e notável é a **memória unificada** (*Unified Memory*), adotada pela Apple a partir da linha de processadores M (M1, M2, M3…): em vez de módulos de RAM soldados numa posição distante do processador e conectados por um barramento tradicional, os chips de memória são posicionados fisicamente ao lado do próprio processador dentro do mesmo encapsulamento (*package*) e compartilhados entre CPU, GPU e demais núcleos especializados do chip — em vez de a GPU ter sua própria memória dedicada e separada da RAM do sistema, como é comum em arquiteturas com placa de vídeo discreta. Essa proximidade física reduz a latência de acesso e elimina a cópia de dados entre memórias separadas de CPU e GPU, à custa de menor flexibilidade: a memória deixa de ser um módulo substituível pelo usuário, ficando soldada e com capacidade definida já na compra do equipamento [11].

Em números concretos, a evolução de largura de banda dessa arquitetura ilustra o ritmo da corrida: o Apple M1 original (2020), pioneiro do conceito, oferecia até 16 GB de memória unificada a ~68 GB/s; a variante "Ultra" da mesma família chegou a 128 GB a ~800 GB/s. A AMD seguiu a mesma estratégia com sua linha Ryzen AI Max, oferecendo até 128 GB a ~256 GB/s — para efeito de comparação, uma configuração convencional de RAM DDR5 em *dual channel* opera na faixa de ~100 GB/s.

Por que não escala indefinidamente. Colocar quatro módulos numa placa com controlador **Quad Channel** não multiplica a velocidade por quatro — o ganho é mais próximo de outro fator de 2 sobre o Dual Channel, limitado pela capacidade do próprio controlador de memória de coordenar os canais simultaneamente. É um exemplo do princípio mais geral de que, em engenharia, ganho de desempenho não aparece do nada: a complexidade de coordenar múltiplos canais é deslocada para um controlador mais sofisticado — por isso placas-mãe com Quad Channel são bem mais caras e, em geral, voltadas a servidores.

Nota prática (armadilha comum). Conectar módulos de memória em slots que não formam um par de Dual Channel válido (fora da combinação indicada no manual da placa-mãe) é um erro silencioso: o sistema operacional continua enxergando a capacidade total corretamente, mas sem o ganho de desempenho do canal duplo — um técnico que não verificar essa configuração pode entregar um computador funcionando "normal", porém performando abaixo do potencial pago pelo cliente.

5.9 Cache e o princípio da localidade

O termo *cache* designa, ao mesmo tempo, um **conceito** e um **componente de hardware** específico — e é importante não confundir os dois.

Como conceito, *cache* é a estratégia de trazer, antecipadamente, dados de uma memória mais lenta para uma memória mais rápida, com base na expectativa de que esses dados serão necessários em breve. Essa estratégia se apoia no **princípio da localidade**: quando um programa acessa uma determinada posição de memória, há alta probabilidade de que ele também

precise, em seguida, de posições adjacentes a ela.

Exemplo. Ao abrir um arquivo PDF, o sistema não lê apenas o byte solicitado no exato instante da requisição: como o disco tem acesso sequencial e o arquivo está fisicamente concentrado numa mesma região do disco (justamente por causa do princípio da localidade), o sistema aproveita que a cabeça de leitura já está posicionada ali e lê um bloco inteiro ao redor da posição pedida, guardando-o numa memória rápida — antecipando pedidos futuros em vez de repetir o processo de acesso sequencial, que é caro em tempo.

Esse mesmo princípio é usado fora do hardware de um computador local: um serviço de streaming de vídeo, por exemplo, antecipa a demanda por um lançamento popular trazendo seus dados de um armazenamento mais lento para um armazenamento mais rápido antes mesmo de a maior parte dos usuários assistir.

Quando o dado antecipado é efetivamente solicitado em seguida, ocorre um ***cache hit***; quando o dado solicitado não está na memória cache e precisa ser buscado na memória mais lenta, ocorre um ***cache miss***.

Como **hardware**, a *cache da CPU* é a memória física, construída em tecnologia SRAM (Seção 5.2), localizada dentro do próprio processador. Ela é organizada em níveis:

- **Cache L1** — a menor e mais rápida, exclusiva de cada núcleo, geralmente subdividida em cache de instrução (o código do programa) e cache de dado.
- **Cache L2** — maior e um pouco mais lenta que a L1, também exclusiva de cada núcleo.
- **Cache L3** — a maior das três, compartilhada entre todos os núcleos do processador.

Exemplo. Ao solicitar um dado, o processador primeiro verifica a cache L1; se houver *cache miss*, verifica a L2; se houver novo *miss*, verifica a L3; e apenas se também não encontrar ali, vai buscar o bloco de dados na memória RAM (usando Dual Channel, se disponível), trazendo-o de volta para as camadas de cache. O AMD Ryzen 9 5900HX, por exemplo, especifica 16 MB de cache L3 compartilhada entre seus 8 núcleos e, para cada núcleo, 512 KB de cache L2 e 64 KB de cache L1 (32 KB de dados + 32 KB de instrução) — números que aparecem diretamente na página de especificações técnicas do fabricante [4].

Cache versus Dual Channel. Embora ambos aumentem o desempenho da memória, operam em níveis e por mecanismos diferentes: o Dual Channel atua ao nível da memória RAM, dividindo (*striping*) e recompondo um mesmo dado entre módulos distintos para ganhar velocidade de escrita e leitura simultânea. O cache atua ao nível da CPU, antecipando e mantendo próximos dados que provavelmente serão requisitados, com base no princípio da localidade. Um não substitui o outro — ambos coexistem no mesmo sistema, otimizando pontos diferentes do caminho entre o processador e o armazenamento.

Recurso	Nível de atuação	Mecanismo	Objetivo
Cache (L1/L2/L3)	CPU	Antecipa dados prováveis com base no princípio da localidade	Reduzir tempo de espera por dados já previstos
Dual Channel	Memória RAM	Divide e recompõe (*striping*) um dado entre módulos simultâneos	Aumentar a taxa de leitura/escrita da RAM

Figura pendente

diagrama da hierarquia L1/L2/L3 dentro de um processador multi-núcleo, com fluxo de cache miss subindo até a RAM

5.10 Memória virtual (swap)

Todo processo em execução precisa estar carregado, ao menos parcialmente, em memória RAM. Quando a soma da memória exigida pelos programas em execução excede a capacidade física de RAM instalada, o sistema operacional recorre a um recurso chamado **memória virtual**: ele reserva um espaço de armazenamento secundário e o trata como se fosse uma extensão da memória primária, "mentindo" para o sistema sobre a quantidade de RAM realmente disponível. No Linux, essa área reservada é chamada de **swap**.

O grande custo desse recurso é o desempenho: a memória secundária usada como swap é ordens de grandeza mais lenta do que a RAM, então o sistema como um todo fica visivelmente mais lento sempre que depende de memória virtual.

Exemplo. Ao processar um programa de linguagem natural que precisava tokenizar grandes volumes de texto, o professor precisou expandir sucessivamente a memória RAM disponível: começando com 16 GB, depois testando com 32 GB e 64 GB emprestados, até finalmente montar um computador dedicado com 128 GB de RAM. Mesmo assim, o sistema operacional precisou recorrer à memória virtual, chegando a usar cerca de 210 GB no pico da execução — 128 GB de RAM física somados a aproximadamente 82 GB de espaço em SSD alocados como swap — antes de estabilizar em torno de 30 GB de uso real após a conclusão da etapa mais pesada do processamento. Uma alternativa mais barata, sugerida posteriormente por um aluno em sala, seria configurar um SSD disponível inteiramente como área de swap: o programa rodaria de forma extremamente lenta, mas sem exigir a compra de um novo computador.

Nota prática. Computadores mais antigos, com pouca memória RAM, dependiam fortemente de memória virtual mesmo em tarefas cotidianas (como alternar entre poucos aplicativos abertos), o que explica boa parte da lentidão percebida nesses equipamentos. Um caso particularmente relevante é o de notebooks modernos com memória RAM reduzida e soldada à placa (não expansível): como recorrem a memória virtual com maior frequência, o uso constante de swap desgasta progressivamente as células da memória flash usada como área de troca — um problema discutido em detalhe na Seção 5.11.

Diferenças entre sistemas operacionais. A política de quando e quanto usar memória virtual é uma característica de software que varia entre sistemas operacionais e entre versões — não há uma regra fixa documentada de forma definitiva. Em linhas gerais: o macOS tende a ser mais agressivo, antecipando o uso de swap antes de a RAM se esgotar por completo, de forma que o usuário dificilmente perceba; o Windows usa o espaço livre disponível sem reservar uma área fixa com antecedência; e o Linux, no processo de instalação, tradicionalmente reserva uma partição de swap dedicada — embora distribuições mais recentes cada vez mais ofereçam, como padrão, um arquivo de swap ou o recurso *zram* em vez de uma partição fixa.

Nota histórica. O Windows já ofereceu, através do recurso **ReadyBoost** (introduzido no Windows Vista, em 2006, e aprimorado no Windows 7, em 2009), a possibilidade de usar um pendrive como área de swap — a memória flash do pendrive, mais rápida que um HD, acelerava computadores com pouca RAM [13]. O recurso perdeu relevância depois que SSDs (também baseados em memória flash) se tornaram o padrão de armazenamento secundário.

5.11 Armazenamento magnético: o disco rígido (HD)

O **disco rígido** (HD, *hard disk*) é a memória secundária mais tradicional, e sua célula de memória é de natureza **magnética**. Fisicamente, um HD é composto por:

- Um ou mais **pratos** (*platters*) — discos rígidos girando em alta rotação, onde os dados são efetivamente gravados.
- Uma **cabeça de leitura e escrita** eletromagnética, posicionada na ponta de um **braço atuador**, que se movimenta sobre a superfície do prato.

A escrita ocorre porque uma corrente elétrica variável na cabeça de leitura/escrita gera um campo eletromagnético, que magnetiza uma região microscópica do prato numa polaridade norte-sul ou sul-norte — cada orientação corresponde a um bit 0 ou 1.

Uma vez magnetizada, a região do disco mantém sua orientação **sem necessidade contínua de energia** — daí a não volatilidade do HD. O dado

só é perdido se um novo campo magnético externo suficientemente forte for aplicado sobre a região gravada, o que reescreve (ou corrompe) a informação ali contida. Essa é, inclusive, uma técnica legítima de destruição segura de dados.

Como a cabeça de leitura/escrita precisa se posicionar fisicamente sobre a trilha correta e aguardar a rotação do prato até o setor desejado passar por baixo dela, o HD tem **acesso sequencial**: o tempo de acesso depende da posição física do dado no disco, ao contrário do acesso aleatório da memória RAM. Cada face de cada prato é organizada em **trilhas** (círculos concêntricos) subdivididas em **setores**; o endereço de um dado no disco é dado pela combinação face/trilha/setor.

Um HD com múltiplos pratos aumenta a complexidade do processo de busca: a cabeça de leitura/escrita precisa identificar não só a trilha e o setor, mas também qual prato e qual face daquele prato contém o dado — mais uma camada de endereçamento sobre a já discutida combinação face/trilha/setor.

A parte interna de um HD tem propositalmente muito pouca eletrônica visível: a maior parte dos componentes eletrônicos fica concentrada numa placa controladora externa, isolada da região magnética, para evitar contaminação eletromagnética dos dados gravados.

Por depender de peças móveis — motor, prato girando, braço atuador —, o HD tem vulnerabilidades de nascença: desgaste mecânico ao longo do tempo (fadiga de peças em movimento), risco de dano por impacto físico enquanto o disco está girando, e um tempo de vida útil finito determinado pela durabilidade dos componentes mecânicos. Em compensação, sua carcaça é lacrada hermeticamente — sem entrada de poeira ou partículas —, e, ao contrário das mídias óticas (Seção 5.12), o prato interno não é exposto a arranhões no uso normal.

Nota prática. Reparar internamente um HD com segurança normalmente exige um ambiente com nível de partículas por metro cúbico comparável ao de uma sala de cirurgia — profissionais de recuperação de dados costumam operar em salas limpas certificadas (ISO Classe 5) justamente por isso [5]. Abrir a carcaça fora desse tipo de ambiente controlado é arriscado e pode inutilizar o disco, já que poeira que entre compromete a leitura — mas parte do próprio setor de recuperação de dados argumenta que uma bancada com fluxo de ar filtrado localizado pode reduzir esse risco sem exigir uma sala limpa completa [5]. Por isso, HDs trazem parafusos e adesivos de segurança que evidenciam violação.

Nota prática (avançada). Entre dois HDs idênticos, é possível, em alguns casos, recuperar um HD com a placa controladora queimada substituindo-a pela placa de um HD gêmeo saudável — desde que a mecânica e a gravação magnética original estejam intactas. Em alguns modelos, é necessário também transferir um chip de calibração específico entre as placas. É uma técnica de manutenção avançada, pouco praticada fora de laboratórios especializados de recuperação de dados.

Figura pendente

HD aberto com prato, braço atuador e cabeça de leitura/escrita identificados] [IMAGEM: esquema de endereçamento face/trilha/setor de um disco magnético

5.12 Unidade de alocação: clusters e metadados

Todo sistema de armazenamento secundário organiza o espaço em disco numa unidade mínima de alocação chamada **cluster**. Um cluster funciona como uma gaveta de tamanho fixo: mesmo que um arquivo ocupe uma fração mínima dessa gaveta, o espaço inteiro do cluster fica reservado para aquele arquivo.

Exemplo. Um arquivo de texto simples (`.txt`) contendo apenas a letra "J" — o equivalente, em ASCII, a 8 bits — ocupa, ainda assim, um cluster inteiro no disco: com o tamanho de cluster padrão do NTFS (4 KB, para a maioria dos tamanhos de partição), esse arquivo de 1 byte ocupa 4 KB [6]. Adicionar mais conteúdo ao mesmo arquivo (por exemplo, mais uma letra) continua ocupando o mesmo cluster, até que o conteúdo exceda sua capacidade — só então um segundo cluster é alocado. **Atenção:** essa relação vale para um arquivo de texto simples; um documento `.docx` equivalente ocupa vários KB a mais, mas essa diferença extra vem do formato do arquivo em si (o `.docx` é um contêiner ZIP/XML com metadados e estrutura de pacote própria), não do tamanho do cluster — são dois fenômenos distintos que não devem ser confundidos.

O tamanho do cluster é uma escolha de engenharia com consequências em ambas as direções: clusters maiores reduzem o esforço de endereçamento (menos gavetas para gerenciar), mas desperdiçam mais espaço em arquivos pequenos; clusters menores reduzem o desperdício, mas aumentam o número de endereços que o sistema precisa gerenciar. Esse tema será retomado, junto com o processo de formatação, no Capítulo 6.

Associado a cada cluster alocado existe um conjunto de **metadados** — dados sobre o próprio dado: nome do arquivo, extensão, programa associado, e se aquele espaço está ou não disponível para uso. Os metadados ficam registrados no disco separadamente do conteúdo do arquivo propriamente dito, numa estrutura chamada **tabela de partição** (também aprofundada no Capítulo 6).

Nota técnica. É comum haver confusão entre a capacidade anunciada de um dispositivo de armazenamento e a capacidade reconhecida pelo sistema operacional. Isso ocorre porque fabricantes costumam anunciar capacidade em múltiplos de 1.000 (o padrão do Sistema Internacional de Unidades — kilo, mega, giga como potências de dez), enquanto sistemas operacionais tradicionalmente calculam capacidade em múltiplos de 1.024 (potências de dois, já que o computador trabalha em base binária). Um dispositivo anun-

ciado com “16 GB” aparece, portanto, com uma capacidade ligeiramente menor quando visualizado pelo sistema operacional.

5.13 Memória flash

A **memória flash** é a tecnologia por trás do SSD (*Solid State Disk* — disco de estado sólido), de pendrives e da memória de armazenamento da grande maioria dos smartphones atuais. Sua célula de memória é construída com um tipo específico de transistor de efeito de campo chamado **MOSFET**, dotado de uma estrutura adicional chamada **porta flutuante** (*floating gate*).

A presença ou ausência de elétrons retidos na porta flutuante altera a tensão necessária para que corrente passe entre os dois terminais do transistor (chamados de dreno e fonte) — esse comportamento é o que permite implementar, em hardware, o equivalente a um teste “se-então” (*if*) que determina se a célula representa um bit 0 ou 1. Escrever um bit consiste em aplicar uma tensão que empurra elétrons para dentro ou para fora da porta flutuante, atravessando a fina camada de óxido isolante.

É exatamente essa travessia repetida de elétrons que explica as duas principais fraquezas da memória flash:

1. **Número limitado de operações de leitura e escrita.** Cada vez que elétrons atravessam a camada de óxido, ela se desgasta um pouco — de forma análoga a uma borracha que, ao apagar repetidamente, vai afinando o papel até rasgá-lo. Depois de um número finito de ciclos de escrita (de cerca de mil ciclos nas tecnologias mais densas e baratas até cerca de 100 mil ciclos nas tecnologias mais duráveis, dependendo de como a célula é fabricada), a célula perde a capacidade de reter dados de forma confiável.
2. **Necessidade de reenergização periódica.** Mesmo sem novas escritas, a carga de elétrons retida na porta flutuante vaza lentamente ao longo do tempo, de forma análoga ao esvaziamento gradual de um capacitor. Um dado gravado numa memória flash e deixado sem uso por um período muito longo pode se tornar ilegível, porque a diferença de carga que originalmente distinguia um bit 0 de um bit 1 se dissipa.

Nota prática. Um pendrive guardado por décadas sem ser conectado a um computador não terá seus dados automaticamente preservados — a memória flash não é permanente da mesma forma que a magnetização de um HD; ela precisa ser periodicamente reenergizada para reter seu conteúdo.

Apesar dessas fraquezas, a memória flash apresenta vantagens decisivas para determinadas aplicações: ausência de partes móveis (logo, resistência a impactos e vibração), baixo consumo energético e volume físico reduzido — características que “casam” perfeitamente com dispositivos móveis. É por isso que praticamente 100% dos smartphones usam memória flash como armazenamento primário de dados: por não ter partes móveis,

o iPhone — lançado já com memória flash — é significativamente mais robusto à vibração e a impactos do que seria um dispositivo móvel equivalente baseado em HD.

Nota prática. Computadores modernos com memória flash soldada à placa e pouca RAM disponível (como certos notebooks lançados a partir de 2020) dependem fortemente de memória virtual (Seção 5.9), usando a própria memória flash soldada como área de swap. Como a memória flash tem um número finito de ciclos de escrita, esse uso intensivo acelera o desgaste da célula ao longo dos anos — e, como a memória é soldada (não modular), não há como substituí-la isoladamente quando ela falha. Esse é um dos fatores que explica a desvalorização acentuada desses equipamentos à medida que envelhecem. Um exemplo real e bastante discutido é o MacBook Air M1 (2020), vendido majoritariamente com apenas 8 GB de RAM soldada: o uso constante de memória virtual sobre o próprio SSD soldado acelera o desgaste da célula flash ao longo dos anos, num aparelho sem qualquer possibilidade de upgrade de RAM ou substituição isolada do armazenamento.

Do ponto de vista de organização interna, múltiplas células de memória flash formam **páginas**, e múltiplas páginas formam **blocos** — a unidade típica de apagamento na memória flash. Um SSD contém, além das próprias células flash, um chip controlador e, tipicamente, uma pequena quantidade de memória DRAM interna usada como cache: blocos de dados recentemente acessados são mantidos nessa DRAM interna (mais rápida que a própria flash) seguindo o mesmo princípio da localidade discutido na Seção 5.8.

A memória flash é historicamente descendente da família de memórias **ROM** (*Read Only Memory*): a ROM original era gravada uma única vez na fábrica; a **PROM** (*Programmable ROM*) podia ser gravada uma vez fora da fábrica; a **EPROM** (*Erasable PROM*) podia ser apagada por exposição à luz ultravioleta e regravada; e a **EEPROM** (*Electrically Erasable PROM*) podia ser apagada eletricamente, sem necessidade de luz ultravioleta — sendo essa a ancestral direta da memória flash moderna [7]. É também por descender dessa linhagem que a memória flash substituiu a ROM em aplicações como o firmware da placa-mãe (BIOS/UEFI, estudado no Capítulo 6), que hoje pode ser atualizado justamente porque está gravado numa memória flash regravável.

Figura pendente

corte esquemático de um MOSFET de porta flutuante, com as camadas de óxido isolante destacadas] [IMAGEM: HD aberto ao lado de um SSD aberto, evidenciando ausência de partes móveis no SSD

5.14 Mídias óticas

As mídias óticas — CD, DVD e Blu-ray — armazenam dados através de uma célula de memória completamente distinta das anteriores: um ponto da su-

perfície do disco que **reflete ou não reflete** um feixe de laser. Um traço longo ou curto gravado na superfície corresponde a um bit 1 ou 0; um laser lê essa superfície e um fototransistor detecta se houve ou não reflexão, convertendo isso de volta em dados binários.

Nota conceitual. Um CD não deve ser confundido com um disco de vinil, apesar da semelhança física: o disco de vinil grava a informação de forma **analógica** — o sulco físico reproduz diretamente a forma de onda sonora, amplificada mecanicamente pela agulha. O CD grava informação de forma **digital**: cada ponto da superfície representa exclusivamente um 0 ou um 1.

Assim como o HD, a mídia ótica tem **acesso sequencial**: a leitura começa por uma posição combinada entre fabricante e leitor (próxima ao centro do disco) e prossegue radialmente para fora. Isso explica, por exemplo, por que gravar apenas metade da capacidade de um CD reduz visivelmente a área gravada (mais clara) em relação à área não utilizada.

A evolução de CD para DVD e para Blu-ray consistiu em reduzir o tamanho físico de cada célula de memória, permitindo mais dados no mesmo espaço — o que exige um laser de comprimento de onda menor (medido em nanômetros) e mais preciso tanto para gravar quanto para ler. O nome *Blu-ray* vem justamente do fato de seu laser operar no espectro azul, de comprimento de onda mais curto que o laser do DVD (vermelho, ~650 nm) e do CD (infravermelho próximo, ~780 nm — tecnicamente fora da faixa visível, ainda que próximo do vermelho) [8].

Em termos de capacidade, um CD armazena algo em torno de 700–800 MB; um DVD de camada única, cerca de 4,7 GB; e um Blu-ray, 25 GB numa única camada — chegando a 50 GB com dupla camada [13].

Por não estar protegida por uma carcaça lacrada como o HD, a superfície de uma mídia ótica é vulnerável a arranhões, que interferem diretamente na reflexão do laser e corrompem a leitura — o “CD riscado” clássico. Técnicas informais de reparo (como aplicar verniz ou pasta de polimento para remover uma fina camada superficial e expor uma superfície de reflexão menos danificada) funcionam apenas de forma limitada, já que removem também parte da camada onde o próprio dado está gravado.

Por fim, distingue-se o **CD-ROM** (gravado uma única vez na fábrica, sem possibilidade de regravação), o **CD-R** (gravável uma única vez pelo usuário) e o **CD-RW** (regravável), com a mesma lógica de gravação valendo para DVD e Blu-ray.

5.15 Nuvem, backup e estratégias de segurança de dados

O armazenamento **em nuvem** não constitui um novo tipo de célula de memória: é, na prática, um computador remoto (ou um conjunto de servidores) conectado à internet, armazenando dados nas mesmas tecnologias já estudadas neste capítulo — predominantemente HD e SSD, organizados dentro de sua própria pirâmide de hierarquia de memória, como qualquer outro

computador. Provedores como Google Drive, OneDrive e Dropbox mantêm réplicas dos dados em servidores fisicamente distribuídos em diferentes localizações, o que aumenta a confiabilidade ao custo, por vezes, de menor velocidade de acesso.

Uma pergunta recorrente em sala foi: qual mídia de armazenamento secundário é a mais segura? A resposta é uma provocação: **não existe uma mídia mais segura em termos absolutos**. Cada tecnologia estudada neste capítulo tem um ponto fraco de natureza distinta:

Mídia	Ponto fraco característico
HD (magnético)	Vulnerável a campos magnéticos fortes próximos; peças móveis se desgastam
SSD / pendrive (flash)	Número limitado de operações de escrita; perda de carga ao longo do tempo
CD/DVD/Blu-ray (ótica)	Vulnerável a arranhões na superfície de leitura
Nuvem	Depende de senha/acesso; exposta a questões de privacidade e disponibilidade de rede

A segurança real de um dado não vem de escolher a "melhor" mídia, mas de criar **redundância**: manter cópias em mídias de natureza física diferente e em **localizações físicas diferentes**, já que cada tecnologia falha por um motivo diferente e um único desastre local não deve comprometer todas as cópias simultaneamente.

Exemplo. Um estudante de pós-graduação mantinha backups de sua dissertação de mestrado no laptop, num HD externo e num pendrive — mas todos os três dispositivos estavam guardados dentro da mesma mochila, que foi roubada. Apesar de ter três cópias redundantes em três mídias diferentes, a ausência de separação geográfica entre elas fez com que todas fossem perdidas simultaneamente, obrigando-o a refazer o trabalho do zero. O princípio de backup exige não apenas diversidade de mídia, mas também diversidade de localização.

Figura pendente

esquema de backup redundante em três mídias e duas localizações físicas diferentes

5.16 Bit flip e memória ECC

Bit flip: causa física. Um ***bit flip*** (também chamado ***soft error*** ou *single-event upset*, SEU) é a inversão espontânea do valor armazenado numa célula de memória — um 0 que passa a 1, ou vice-versa —, sem que a célula tenha sofrido qualquer dano físico permanente: se regravada com o valor correto, ela volta a funcionar normalmente. A literatura técnica — a partir de um estudo seminal da Intel, em 1978, que primeiro identificou o fenômeno em DRAM [9] — aponta **duas** causas físicas bem estabelecidas, ambas formas de radiação ionizante, hoje normatizadas por um padrão da indústria de testes de memória [10]:

1. **Raios cósmicos secundários.** Partículas de altíssima energia vindas do espaço colidem com a atmosfera terrestre e produzem um chuveiro de partículas secundárias — na superfície da Terra, majoritariamente **nêutrons**. Um nêutron não tem carga elétrica e não perturba um circuito diretamente, mas, ao colidir com o núcleo de um átomo de silício do chip, pode gerar partículas carregadas secundárias, que então depositam carga suficiente para inverter o estado de um capacitor de DRAM (Seção 5.3) ou de um flip-flop de SRAM (Seção 5.2).
2. **Partículas alfa de contaminantes radioativos no encapsulamento.** Os próprios materiais usados para embalar e soldar o chip (a "casca" plástica ou cerâmica do processador ou do módulo de memória) contêm, em quantidade mínima, traços de elementos radioativos residuais dos processos de mineração e refino usados para produzi-los. Esses traços emitem, de forma constante e previsível, partículas alfa — e, por estarem fisicamente muito próximos da própria célula de memória, essas partículas depositam carga da mesma forma que um nêutron secundário.

Bit flip não é o mesmo fenômeno que a vulnerabilidade do HD a campos magnéticos (Seção 5.10) nem que a perda de dados discutida na Seção 5.14. Um HD armazena dados por meio da orientação magnética de uma região do prato — por isso um ímã suficientemente forte de fato ameaça um HD, ao reescrever essa orientação. Uma célula de RAM, em contraste, armazena dado como **carga elétrica** (DRAM) ou como **estado lógico de um circuito** (SRAM) — não existe, na literatura sobre soft errors, um mecanismo estabelecido de bit flip por campo magnético externo; um ímã comum, mesmo forte, não tem como induzir diretamente a carga necessária para inverter um bit de RAM da forma como raios cósmicos e partículas alfa fazem. As duas ameaças — magnética para o HD, radiação ionizante para a RAM — têm mecanismos físicos distintos e não devem ser confundidas.

Memória ECC. Servidores costumam usar módulos de memória **ECC** (*Error-Correcting Code*), um tipo de RAM capaz de detectar e corrigir automaticamente um erro de bit flip de um único bit por palavra de memória, usando bits extras de paridade/verificação gravados junto com o dado. Quanto menor a litografia do chip de memória (o Capítulo 4 trata da litografia em pro-

fundidade) e quanto maior a quantidade total de memória instalada, maior a probabilidade estatística de que um bit flip ocorra em algum lugar do sistema. Num desktop doméstico, um bit flip ocasional é, na pior hipótese, uma tela azul isolada; num servidor operando 24 horas por dia com centenas de gigabytes de RAM, o mesmo tipo de erro, acumulado ao longo de meses, pode corromper silenciosamente um banco de dados inteiro — daí o uso de ECC ser padrão nesse perfil de máquina, e praticamente inexistente em desktops e notebooks comuns (Capítulo 1, §1.11.2).

A probabilidade de um bit flip também aumenta com a altitude, pela menor espessura de atmosfera disponível para blindar a radiação cósmica — data centers instalados em altitudes mais elevadas registram, mensuravelmente, taxas de falha um pouco maiores por esse motivo.

5.17 Síntese do capítulo

Este capítulo aprofundou a hierarquia de memória introduzida no Capítulo 1, mostrando que cada nível dessa pirâmide — registradores e cache, memória RAM, e as diferentes formas de armazenamento secundário — deriva suas características de desempenho, custo e volatilidade diretamente da natureza física de sua célula de memória. Foram estudadas a evolução da memória primária (da SRAM estática à família DDR síncrona e dinâmica), os recursos que ampliam seu desempenho (cache, Dual Channel, memória virtual), as três grandes famílias de armazenamento secundário — magnética, flash e ótica —, cada uma com vantagens e fragilidades próprias, e as estratégias de backup e redundância que protegem esses dados. O capítulo fechou com o *bit flip* — a inversão espontânea de um bit por radiação ionizante — e a memória ECC que o corrige, aplicando à própria memória primária o mesmo princípio de redundância contra falha discutido a propósito do backup. Esses conceitos formam a base necessária para o Capítulo 6, no qual o sistema operacional passa a ser estudado em detalhe: a formatação de um disco, a criação de sistemas de arquivos e o gerenciamento de clusters e metadados — apenas introduzidos aqui na Seção 5.11 — são, na prática, a camada de software que organiza e dá sentido à memória secundária estudada neste capítulo.

5.18 Referências

1. COMPUTER HISTORY MUSEUM. “1970: MOS Dynamic RAM Competes with Magnetic Core Memory on Price.” *Memory & Storage — Timeline of Computer History*. Disponível em: `https://www.computerhistory.`

org/storageengine/; HENNESSY, John L.; PATTERSON, David A. *Computer Architecture: A Quantitative Approach*. 6. ed. Morgan Kaufmann, 2017, seção “SRAM Technology”/“DRAM Technology”.
2. JEDEC SOLID STATE TECHNOLOGY ASSOCIATION. *JESD79-4: DDR4 SDRAM Standard*; *JESD79-5: DDR5 SDRAM Standard*. Arlington, VA: JEDEC. Disponível em: https://www.jedec.org/standards-documents.
3. HP. “DDR4 RAM: A Comprehensive Guide.” Disponível em: https://www.hp.com/us-en/shop/tech-takes/what-is-ddr4-ram-and-how-to-instal KINGSTON TECHNOLOGY. “What is DDR4 Memory?” Disponível em: https://www.kingston.com/en/memory/ddr4-overview.
4. AMD. Página de especificações técnicas do processador AMD Ryzen 9 5900HX.
5. DRIVESAVERS. “Certified ISO Class 5 Cleanroom.” Disponível em: https://drivesaversdatarecovery.com/why-us/certified-iso-class-5-cl ROSSMANN GROUP. “You Do Not Need a Cleanroom for Data Recovery.” Disponível em: https://rossmanngroup.com/data-recovery-myths/cleanroom-not-required-for-data-recovery.
6. MICROSOFT. “Default cluster size for NTFS, FAT, and exFAT.” Disponível em: https://mskb.pkisolutions.com/kb/140365.
7. MALVINO, Albert Paul; BROWN, Jerald A. *Digital Computer Electronics*. 3. ed. Glencoe/McGraw-Hill, 1993, seções “9-2 PROMS AND EPROMS” e “EEPROM”; MONTEIRO, Mario A. *Introdução à Organização de Computadores*. 5. ed. Rio de Janeiro: LTC, seções “PROM”/“EPROM e EEPROM”.
8. TANENBAUM, Andrew S.; AUSTIN, Todd. *Organização Estruturada de Computadores*. 6. ed. São Paulo: Pearson Education do Brasil, 2013, seção 2.3.11 “Blu-ray”; COMPUTER HISTORY MUSEUM. “2000: Prototype blue laser disc stores HD video.” Disponível em: https://www.computerhistory.org/storageengine/prototype-blue-laser-disc-stores-
9. MAY, T. C.; WOODS, M. H. A new physical mechanism for soft errors in dynamic memories. In: *Proceedings of the 16th Annual Reliability Physics Symposium*. IEEE, 1978. p. 33–40. Publicado também como: MAY, T. C.; WOODS, M. H. Alpha-particle-induced soft errors in dynamic memories. *IEEE Transactions on Electron Devices*, v. 26, n. 1, p. 2–9, jan. 1979.
10. JEDEC SOLID STATE TECHNOLOGY ASSOCIATION. *JESD89B: Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Errors in Semiconductor Devices*. Arlington, VA: JEDEC, 2021.
11. APPLE. “Apple unveils M1, the first in a groundbreaking family of chips for Mac.” *Apple Newsroom*, 2020. Disponível em: https://www.apple.com/newsroom/2020/11/apple-unveils-m1-the-first-in-a-groundbr
12. TOM’S HARDWARE. “What Is CAS Latency in RAM? CL Timings Explained.” Disponível em: https://www.tomshardware.com/reviews/cas-latency-ram-cl-timings-glossary-definition,6011.html.
13. Blu-ray Disc Association (BDA). Especificação padrão de capacidade: 25 GB (camada única) e 50 GB (dupla camada); CD-DA (700–800 MB)

e DVD-5 (~4,7 GB) conforme especificações originais dos formatos.
14. SOURCEDADDY. “Windows ReadyBoost.” *Windows 7 Tutorial*. Disponível em: https://sourcedaddy.com/windows-7/windows-readyboost.html; MICROSOFT. “ReadyBoost changes in Windows 7.” *Microsoft Learn (archive)*. Disponível em: https://learn.microsoft.com/en-us/archive/blogs/7/readyboost-changes-in-windows-7.

6 Sistema Operacional e Sistemas de Arquivo

Neste capítulo você vai estudar o papel do sistema operacional como camada de interface entre hardware, software e usuário; a forma como os sistemas de arquivo organizam o armazenamento secundário em unidades chamadas clusters; a operação de formatação e suas implicações sobre a permanência real dos dados; o conceito de partição e as duas soluções de tabela de partição em uso — MBR e GPT; e o procedimento de inicialização do computador, do POST ao carregamento do sistema operacional. O Capítulo 7 dá sequência direta a este, tratando da instalação propriamente dita.

6.1 O sistema operacional como interface

Um sistema operacional (SO, do inglês *Operating System*, OS) é um software cuja função central é atuar como **interface** entre o hardware do computador e os demais softwares e usuários. O termo *interface* designa aquilo que se coloca entre duas faces, mediando a relação entre elas sem se confundir com nenhuma delas.

Em termos de camadas, o hardware ocupa a base do sistema; sobre ele executa o sistema operacional, cujo núcleo é chamado de **kernel**; e acima do sistema operacional executam os demais programas — desde o próprio ambiente gráfico (o menu iniciar, os ícones, as janelas) até os aplicativos que o usuário abre, como um navegador ou um editor de texto.

Figura pendente

diagrama em camadas — hardware na base, kernel do sistema operacional no meio, aplicativos e usuário no topo

6.1.1 Abstração e plataforma

O Capítulo 2 (§2.1) apresentou os conceitos de **plataforma** e **abstração em camadas** a partir do exemplo do navegador. Aplicados ao sistema operacional: um desenvolvedor de software não precisa saber qual é o modelo da memória RAM instalada, a marca do disco ou a velocidade da placa de vídeo do computador em que seu programa vai rodar — ele apenas chama

funções do sistema operacional, e é o SO quem trata da comunicação efetiva com aquele hardware específico. Um programa feito para a plataforma Windows não roda nativamente em outra plataforma, a menos que exista uma versão também disponível para ela — o que caracteriza um software **multiplataforma**; no caso de uma aplicação web, a própria plataforma é o navegador, e não o sistema operacional subjacente.

Antes da existência do sistema operacional, um programa era escrito diretamente para o hardware que o executaria — de forma semelhante ao que ocorre hoje com uma placa Arduino, cujo programa precisa ser reescrito se a placa mudar. O sistema operacional resolveu esse problema criando uma camada intermediária estável, da qual decorre uma segunda propriedade fundamental — e essa sim exclusiva do sistema operacional, sem paralelo na discussão do Capítulo 2: a possibilidade de executar **múltiplos programas ao mesmo tempo**, alternando (fazendo o chaveamento) entre eles os recursos de hardware disponíveis.

6.1.2 Da era monofunção à multitarefa

Exemplo. Antes dos smartphones modernos, cada função dependia de um dispositivo dedicado: um iPod para ouvir música, uma câmera para fotografar, um telefone para ligar, um bloco de papel para anotar. Esses aparelhos eram **monofunção**. O smartphone moderno reúne um hardware capaz de realizar todas essas tarefas e, sobre ele, instala um sistema operacional capaz de gerenciar múltiplos programas simultaneamente — permitindo, por exemplo, ouvir música no Spotify enquanto se recebe uma mensagem e um e-mail ao mesmo tempo. É o sistema operacional que faz a interface entre esses vários programas aplicativos, o usuário e o hardware do aparelho.

6.2 Sistemas de arquivo e a unidade mínima de alocação: o cluster

O **sistema de arquivo** é o protocolo — o conjunto de combinados — pelo qual o sistema operacional organiza e localiza dados dentro de uma unidade de armazenamento secundário. Sistemas operacionais diferentes utilizam, em geral, sistemas de arquivo diferentes: o Windows usa hoje o **NTFS** (e usou historicamente o FAT); distribuições Linux usam tipicamente **EXT4**; Android, iOS e macOS têm, cada um, seus próprios sistemas de arquivo.

A unidade mínima de alocação de espaço em disco para arquivos e diretórios é o **cluster**.

Ao formatar uma unidade de armazenamento, o sistema operacional solicita, entre outras informações, o **tamanho da unidade de alocação** (o tamanho do cluster). Valores típicos oferecidos pelo Windows incluem 8.192 bytes, 16 KB, 32 KB e 64 KB, além de um tamanho padrão sugerido para o dispositivo [1].

Figura pendente

janela de formatação do Windows mostrando a escolha do sistema de arquivo (FAT32, NTFS, exFAT) e do tamanho da unidade de alocação

6.2.1 File slack

Como cada arquivo ocupa um número inteiro de clusters, raramente o tamanho lógico de um arquivo coincide exatamente com o espaço físico que ele ocupa em disco. Essa diferença é chamada de **file slack**.

Exemplo. Nas propriedades de um arquivo de áudio real usado em demonstração, o tamanho do arquivo (tamanho lógico) era de 27.550.336 bytes (26,2 MB), enquanto o tamanho em disco (espaço físico ocupado) era de 27.553.792 bytes — uma diferença decorrente de o arquivo não preencher por completo o último cluster que lhe foi atribuído.

6.2.2 Fragmentação

A escolha do tamanho do cluster no momento da formatação gera compromissos entre quatro cenários possíveis, resumidos na tabela a seguir.

Cenário	Cluster	Arquivo	Efeito
1	Pequeno	Pequeno	Cenário ideal, mas irreal em um computador de uso geral
2	Pequeno	Grande	Fragmentação: o arquivo é quebrado em muitos pedaços
3	Grande	Pequeno	File slack elevado: grande parte do disco é desperdiçada
4	Grande	Grande	Cenário aceitável, mas também irreal isoladamente

Como um computador moderno lida simultaneamente com arquivos pequenos e grandes, a situação real está sempre mais próxima dos cenários 2 e 3, exigindo um meio-termo na escolha do tamanho do cluster. O cenário 3 — cluster grande com arquivo pequeno — é considerado o mais prejudicial, por gerar o maior desperdício proporcional de espaço em disco.

A **fragmentação** ocorre porque, à medida que arquivos são apagados e recriados de tamanhos distintos, os espaços livres deixados por exclusões

(buracos) nem sempre comportam o próximo arquivo a ser gravado por inteiro, obrigando o sistema operacional a dividir um mesmo arquivo em blocos não contíguos no disco. A ferramenta de **desfragmentação** existe justamente para reorganizar esses blocos e reduzir esse efeito.

6.3 A operação de formatação

Formatar uma unidade de armazenamento significa atribuir a ela um sistema de arquivo — ou seja, "colocá-la em um formato", um combinado que o sistema operacional passa a reconhecer e utilizar. Antes da formatação, uma partição sem sistema de arquivo atribuído não pode ser usada pelo sistema operacional: nenhum programa consegue gravar ou ler dados nela.

Um efeito colateral direto da formatação é a perda de todos os arquivos e diretórios anteriormente existentes naquela unidade.

6.3.1 O que a formatação realmente faz: metadados, exclusão e recuperação de dados

Cada dado gravado em disco é acompanhado de **metadados**: informações sobre o próprio dado, como o nome do arquivo, o momento de criação, de modificação e de último acesso, atributos de somente leitura ou oculto, e assinatura digital, entre outros. Um mecanismo semelhante a uma tabela de referências indica onde cada arquivo começa e onde termina dentro do disco.

Exemplo. Suponha uma sequência de células de memória em que o valor 1001 foi gravado, seguido do valor 101. Sem uma marcação adicional, não é possível saber onde termina um número e começa o outro. A solução é registrar, para cada dado, uma referência com a posição inicial e o comprimento (por exemplo: "o dado A começa aqui e tem comprimento 4"). Apagar um arquivo consiste, nesse esquema, simplesmente em remover essa referência — não em reescrever os bits do dado propriamente dito.

Essa é a razão pela qual formatar ou apagar um arquivo não desgasta uma memória flash (como um pendrive ou SSD) na mesma proporção que reescrever cada bit: a operação normalmente descarta apenas a tabela de referências, preservando o conteúdo bruto até que aquele espaço seja reutilizado.

A **lixeira** do sistema operacional é uma lista de arquivos cuja referência está marcada como "pode ser removida no futuro", mas ainda não foi de fato eliminada — uma camada extra de segurança contra exclusões acidentais. Enquanto o dado permanecer fisicamente gravado, softwares de recuperação de dados podem restaurá-lo, mesmo após a formatação: eles percorrem o disco bit a bit em busca de cabeçalhos característicos de cada tipo de arquivo (por exemplo, os bytes iniciais que identificam um `.docx`) e,

pelo princípio da localidade, reconstroem o conteúdo entre o início e o fim identificados.

Aplicação prática. Se um computador estiver infectado por um malware capturando dados do usuário, formatar o disco elimina o malware — mas também elimina, junto com ele, todos os demais dados do usuário, incluindo aqueles que se desejaria preservar. É, na expressão usada em aula, "matar uma mosca com uma bazuca": resolve o problema, mas com um custo desproporcional se não houver backup prévio (Capítulo 7, §7.2.1).

Figura pendente

esquema comparando a tabela de referências antes e depois da exclusão de um arquivo

---

6.4 Partições: divisão lógica do disco

Uma **partição** é uma divisão lógica de um disco físico — não uma divisão física real.

Todo disco precisa ter **ao menos uma partição** para que o sistema operacional possa atribuir a ele um sistema de arquivo e utilizá-lo — mesmo que essa única partição ocupe a totalidade do espaço físico disponível.

6.4.1 Sistemas de arquivo por partição e o conceito de dual boot

Cada partição pode ter atrelado a si um sistema de arquivo próprio, independente das demais partições do mesmo disco. Essa propriedade tem duas consequências práticas relevantes:

- **Modularização de uso.** É possível reservar cotas de espaço distintas para finalidades diferentes — por exemplo, dividir um mesmo disco entre dois usuários de uma mesma máquina.
- **Coexistência de sistemas operacionais.** Como sistemas operacionais diferentes exigem sistemas de arquivo diferentes (o Windows opera sobre NTFS; uma distribuição Linux como o Ubuntu opera sobre EXT4), um único disco físico pode conter partições distintas, cada uma com seu próprio sistema operacional instalado.

Quando um computador com múltiplos sistemas operacionais instalados é ligado e apresenta uma tela de escolha entre eles, esse mecanismo é chamado de **dual boot** (ou *multi boot*, se houver mais de dois sistemas). *Boot* é o termo em inglês para o procedimento de inicialização; dual boot significa, portanto, que há mais de uma forma possível de inicializar aquele computador, cada uma delas carregando um sistema operacional diferente, tratado em detalhe no Capítulo 7, §7.5.

Sistema operacional	Sistema de arquivo típico
Windows	NTFS (historicamente, FAT)
Linux (ex.: Ubuntu)	EXT4 (historicamente, EXT3)
Pendrives / mídias removíveis	FAT32 ou exFAT

Figura pendente

captura do gerenciador de disco do Windows mostrando um disco físico dividido em múltiplas partições

6.5 Tabelas de partição: MBR e GPT

As informações sobre quantas partições um disco possui, onde cada uma começa e termina, e qual sistema de arquivo está atrelado a cada uma constituem, elas próprias, um conjunto de metadados que precisa ser gravado em algum lugar do disco. A estrutura responsável por essa organização é chamada de **tabela de partições**.

Existem duas soluções de tabela de partição amplamente utilizadas: **MBR** (mais antiga) e **GPT** (mais recente).

6.5.1 Master Boot Record (MBR)

O **MBR** (*Master Boot Record*) grava a tabela de partições em um setor específico no início do disco, usando endereçamento de **32 bits**. Dessa limitação de endereçamento decorrem duas restrições centrais:

- O tamanho máximo de uma partição é de **2 TB**.
- É possível criar no máximo **quatro partições primárias** [2].

Para superar o limite de quatro partições, uma das partições primárias pode ser convertida em **partição estendida**, dentro da qual é possível criar até **128 partições lógicas**. Uma consequência prática dessa regra é que múltiplas partições lógicas devem estar todas contidas dentro de uma única partição estendida — não é possível, por exemplo, dividir 64 partições lógicas entre duas partições primárias diferentes.

Exemplo. Um disco já dividido em quatro partições primárias atingiu o limite da tabela MBR. Para criar uma quinta divisão, uma das quatro partições primárias precisa ser apagada e recriada como partição estendida; somente dentro dela é possível abrir novas partições lógicas adicionais.

6.5.2 GUID Partition Table (GPT)

O **GPT** (*GUID Partition Table*) foi desenvolvido para superar as limitações do MBR, mantendo **retrocompatibilidade** com ele — isto é, softwares e firmwares mais antigos, mesmo sem reconhecer o GPT, ainda conseguem ler as informações essenciais gravadas no mesmo local histórico do disco.

Cada disco identificado em GPT recebe um **GUID** (*Globally Unique Identifier*), um identificador único análogo a um endereço IP em uma rede. Usando endereçamento de **64 bits**, o GPT permite partições na ordem de zettabytes [3], até **128 partições** — o padrão adotado pelo Windows; a especificação GPT em si permite um número de partições configurável, tipicamente maior [4] — sem a necessidade do artifício de partições estendidas, e inclui um mecanismo de **redundância**: como historicamente ataques que reescreviam apenas o setor da tabela de partições eram suficientes para inutilizar o acesso a um disco inteiro (sem apagar os dados propriamente ditos, mas destruindo a referência para encontrá-los), o GPT mantém cópias redundantes dessa informação.

Característica	MBR	GPT
Época de criação	Mais antiga	Mais recente
Endereçamento	32 bits	64 bits
Tamanho máximo de partição	2 TB	Na casa de zettabytes
Partições primárias	Até 4	Até 128 (sem partição estendida)
Partições lógicas	Até 128, dentro de uma partição estendida	Não se aplica
Redundância da tabela	Não	Sim
Firmware associado historicamente	BIOS	UEFI

Figura pendente

diagrama do layout de um disco em MBR — código de inicialização, tabela de partições e partições de dados

6.6 BIOS, UEFI e o procedimento de inicialização

O **BIOS** (*Basic Input/Output System*), introduzido no Capítulo 1 a propósito do IBM PC, é o conjunto de softwares gravado na placa-mãe responsável por inicializar o hardware e oferecer uma interface básica de entrada e saída antes de qualquer sistema operacional ser carregado. Ele é composto,

entre outros elementos, por dois programas centrais: o **POST** e o **Setup**. O **UEFI** (*Unified Extensible Firmware Interface*) é a evolução moderna desse firmware, com interface gráfica navegável por mouse — em contraste com as telas de texto do BIOS tradicional.

6.6.1 POST

O **POST** (*Power On Self Test*, autoteste de inicialização) é o primeiro programa executado quando o computador é ligado. Sua função é varrer os componentes de hardware — processador, memória, teclado, entre outros — em busca de falhas, antes de liberar o controle da CPU para qualquer outro software.

Se algum componente essencial falhar no teste, o POST comunica o erro por meio de sinais sonoros (bips), já que, sem memória funcional, ele não tem como exibir uma mensagem na tela — mostrar algo na tela já é, em si, uma operação de software que depende de memória disponível. A quantidade e o padrão de bips indicam, conforme o manual da placa-mãe, qual componente falhou (por exemplo, ausência ou defeito de memória RAM); o Capítulo 9 aprofunda o POST sob a ótica dos componentes de hardware que ele avalia.

Do ponto de vista do diagnóstico técnico, um POST bem-sucedido indica que processador, memória e placa-mãe estão minimamente funcionais — o que não exclui problemas de hardware que só se manifestem sob carga (como superaquecimento durante o uso), nem problemas de software, que só podem ocorrer depois que o POST é concluído com sucesso.

6.6.2 Setup

O **Setup** é o programa que permite alterar as configurações de hardware do computador — frequência do processador e da memória (overclock), habilitação ou desabilitação de funcionalidades, data e hora do sistema, e a **ordem de inicialização** (*boot order*), entre outras.

Essas configurações — incluindo a informação de qual dispositivo de armazenamento contém o sistema operacional a ser carregado — precisam ser preservadas mesmo com o computador desligado. Por isso, ficam gravadas em uma pequena memória flash não volátil na própria placa-mãe, dedicada a esse fim; o Capítulo 9 trata da bateria que mantém essa memória energizada com o computador desligado da tomada.

6.6.3 Boot: carregamento do sistema operacional

Concluído o POST com sucesso, o próximo passo padrão é o **boot** (inicialização) do sistema operacional: a cópia do sistema operacional da memória secundária, onde está instalado, para a memória primária (RAM), de onde ele passa a ser executado — retomando o conceito de hierarquia de me-

mória apresentado no Capítulo 1 (Seção 1.10) e aprofundado no Capítulo 5.

Para saber onde procurar o sistema operacional entre as possivelmente várias partições e discos existentes, o computador consulta a variável de ordem de inicialização gravada na memória da placa-mãe (Seção 6.6.2). É possível interromper esse fluxo padrão e forçar a inicialização a partir de outro dispositivo — como um pendrive — de duas formas: alterando permanentemente a ordem de boot dentro do Setup, ou acionando, na tela do POST, um atalho de teclado que abre o chamado **boot menu**, uma lista de dispositivos disponíveis para inicialização imediata (nas máquinas descritas em aula, a tecla de atalho variava entre **F9**, **F11**/**F12** ou a sequência **10 → F12**, dependendo do fabricante).

Figura pendente

tela de POST/BIOS de um computador real, mostrando o logotipo do fabricante e a instrução para acessar o boot menu

Figura pendente

tela de Setup/BIOS com a configuração de ordem de inicialização (boot order) destacada

6.6.4 Segurança e ética do acesso físico

Um ponto central para a formação de um técnico de informática é a compreensão de que **o acesso físico a uma máquina muda todos os paradigmas de segurança** — formulação que corresponde à "Lei nº 3" das clássicas "10 Immutable Laws of Security" da Microsoft [5]. Se é possível interromper o boot padrão e carregar, em vez do sistema operacional instalado, um programa alternativo a partir de um pendrive — por exemplo, um Live CD/USB (Capítulo 7, §7.3) —, é possível se tornar administrador daquela máquina sem conhecer nenhuma senha, e a partir daí acessar, copiar ou apagar qualquer dado nela contido, independentemente das proteções de software configuradas pelo usuário original.

Essa mesma técnica que permite, de forma legítima, recuperar o acesso a uma máquina cujo usuário esqueceu a senha, pode ser usada de forma ilegítima para violar dados de terceiros sem autorização. Por essa razão, deixar um pendrive ou dispositivo USB como primeira opção na ordem de inicialização é considerado uma **falha de segurança grave**: qualquer pessoa com acesso físico breve à máquina pode assumir controle administrativo total sobre ela. O uso ético dessas técnicas — e a orientação de nunca acessar dados sensíveis de um cliente sem que ele esteja presente e ciente do procedimento — é parte inseparável da formação técnica apresentada neste capítulo.

6.7 Síntese do capítulo

Este capítulo apresentou o sistema operacional como a camada de interface entre hardware, software e usuário, detalhando como essa camada organiza o armazenamento secundário — introduzido no Capítulo 5 em termos de blocos, páginas e setores — por meio de clusters, sistemas de arquivo e partições. Foram estudadas as duas soluções de tabela de partição em uso, MBR e GPT, e o procedimento completo de inicialização do computador: POST, Setup/BIOS e boot. Esses conceitos formam a base necessária para o Capítulo 7, no qual o processo completo de instalação de um sistema operacional é tratado em detalhe — da preparação da mídia à instalação de drivers, passando por backup, dual boot, diagnóstico via Live CD/USB e a manutenção contínua dessa camada de software.

6.8 Referências

1. MICROSOFT. “NTFS overview.” Disponível em: https://learn.microsoft.com/en-us/windows-server/storage/file-server/ntfs-overview.
2. MICROSOFT. “Windows support for hard disks exceeding 2 TB.” Disponível em: https://learn.microsoft.com/en-us/troubleshoot/windows-server/backup-and-storage/support-for-hard-disks-exceeding-2-tb; UEFI FORUM. “FAQ: Drive Partition Limits.” Disponível em: https://uefi.org/sites/default/files/resources/UEFI_Drive_Partition_Limits_Fact_Sheet.pdf.
3. UEFI FORUM. “FAQ: Drive Partition Limits.” Disponível em: https://uefi.org/sites/default/files/resources/UEFI_Drive_Partition_Limits_Fact_Sheet.pdf.
4. MICROSOFT. “Windows and GPT FAQ.” Disponível em: https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-and
5. MICROSOFT. “10 Immutable Laws of Security (Version 2.0).” Disponível em: https://learn.microsoft.com/en-us/archive/blogs/rhalbheer/ten-immutable-laws-of-security-version-2-0.

7 Instalação e Manutenção de Sistemas Operacionais

Neste capítulo você vai estudar o processo completo de instalação de um sistema operacional — da preparação da mídia ao particionamento, ao backup e à instalação em dual boot —, o uso de um Live CD/USB como ferramenta de diagnóstico, a instalação de drivers, a atualização contínua do sistema operacional e do firmware da placa-mãe, e dois problemas recorrentes de manutenção corretiva de software: malware e a decisão de reinstalar o sistema operacional.

7.1 Preparação da mídia de instalação

Para instalar um sistema operacional, é necessário preparar previamente uma mídia de instalação — tipicamente um pendrive — contendo o instalador em um formato reconhecível pelo firmware do computador de destino. O procedimento envolve duas etapas conceitualmente distintas: baixar a imagem do sistema operacional (arquivo `.iso`) e, em seguida, gravar essa imagem no pendrive de forma que ele se torne inicializável — o que exige, ele próprio, criar uma tabela de partições e um sistema de arquivo válidos na mídia.

Procedimento de segurança. A imagem de instalação (ISO) deve ser obtida diretamente do fabricante — no caso do Windows, do site oficial da Microsoft. Imagens obtidas de fontes intermediárias não confiáveis podem vir acompanhadas de software malicioso embutido no próprio instalador.

7.1.1 BIOS/MBR e UEFI/GPT

Como o UEFI é retrocompatível com o BIOS (Capítulo 6, §6.5.2), um pendrive gravado em **MBR** funciona tanto em computadores com firmware BIOS quanto UEFI, enquanto um pendrive gravado em **GPT** funciona apenas em computadores com UEFI. Por essa razão, ao preparar uma única mídia de instalação para um parque de máquinas heterogêneo — com computadores antigos e recentes —, a opção mais segura é gravar em MBR, garantindo compatibilidade universal.

Esquema gravado	Funciona em BIOS	Funciona em UEFI
MBR	Sim	Sim
GPT	Não	Sim

Procedimento de identificação. Ao inicializar o computador e entrar no *setup* (menu de configuração do firmware, acessado durante o POST): se o menu responde ao mouse, o firmware é UEFI/GPT; se o menu só é navegável pelo teclado, o firmware é BIOS/MBR legado.

Aplicação prática. Se um dado computador não reconhece o pendrive de instalação, ou apresenta uma mensagem de erro mencionando GPT e EFI (ou MBR), a causa mais provável é uma incompatibilidade entre o esquema de partição gravado na mídia e o tipo de firmware daquele computador.

7.1.2 Ferramentas de criação de mídia

Ferramenta	Uso recomendado
Assistente de instalação da Microsoft (*Media Creation Tool*)	Gera diretamente um pendrive UEFI/GPT (acerta para a maioria dos computadores novos) ou permite apenas baixar o arquivo ISO para uso posterior com outra ferramenta
Rufus	Ferramenta leve (poucos megabytes), gratuita e de código aberto [1], que permite escolher explicitamente o tipo de sistema-alvo (BIOS/MBR ou UEFI/GPT); recomendada para gravar uma única imagem por vez com controle preciso do padrão de destino
YUMI	Permite reunir múltiplos instaladores (por exemplo, várias distribuições Linux e versões do Windows) em um único pendrive, funcionando como um “GRUB de instaladores”; menos direta de configurar que o Rufus
Ventoy	Alternativa multi-imagem semelhante ao YUMI

Especificações e comportamento de cada ferramenta conforme suas páginas oficiais [5].

O Assistente de instalação da Microsoft, quando usado no modo “unidade flash USB”, automaticamente grava a mídia como UEFI/GPT — o que funciona para a grande maioria dos computadores novos, mas falha em máquinas antigas com BIOS/MBR. Para instalar em uma máquina legada, é necessário baixar apenas o arquivo ISO e utilizar o Rufus, selecionando manualmente o esquema de partição BIOS/MBR compatível com aquele hardware.

Figura pendente

captura de tela do Rufus com os campos de esquema de partição (MBR/GPT) e sistema de destino (BIOS/UEFI) destacados

---

7.2 Instalação do sistema operacional

O processo de instalação de um sistema operacional pode ser descrito, de forma resumida, em três etapas:

1. Interromper o fluxo normal de boot (Capítulo 6, §6.6.3) e direcioná-lo para a mídia de instalação (Seção 7.1), por meio do boot menu ou de uma alteração no Setup.
2. Executar o instalador: aceitar os termos de uso, escolher a partição de destino e, se necessário, formatá-la (Capítulo 6, §6.3) para realizar uma **instalação limpa** — isto é, uma instalação que parte do zero, sem preservar dados anteriores daquela partição.
3. Aguardar a conclusão da cópia dos arquivos do sistema operacional para a partição de destino e a reinicialização automática da máquina, que volta a carregar normalmente a partir do disco — sem necessidade de nova intervenção no teclado.

Um ponto de atenção recorrente é que, ao chegar pela primeira vez à tela do instalador, costuma ser necessário pressionar qualquer tecla para confirmar a inicialização a partir da mídia externa; esse gesto não deve ser repetido após a primeira reinicialização automática do processo de instalação, sob pena de reiniciar a instalação do zero (Seção 7.2.2).

Também é preciso observar que, uma vez que o sistema operacional em execução está instalado em determinada partição, essa mesma partição não pode ser formatada enquanto está em uso — de forma análoga a tentar remover o alicerce sobre o qual se está de pé. Formatar a partição ativa do sistema em execução interrompe o próprio funcionamento do computador.

7.2.1 Backup antes de reinstalar

Como a formatação é uma etapa típica da instalação limpa, qualquer dado do usuário presente na partição de destino é perdido no processo, salvo se houver uma cópia prévia. Essa cópia de segurança é chamada de **backup**.

Antes de reinstalar um sistema operacional em uma máquina com dados relevantes do usuário, o procedimento recomendado é:

1. Criar (ou reaproveitar) uma partição separada, que não será formatada durante a instalação.
2. Copiar para essa partição os dados que precisam ser preservados.
3. Realizar a instalação limpa apenas na partição de destino do sistema operacional, deixando intacta a partição de backup.

Aplicação prática — reparo de software. Quando o hardware de uma máquina está íntegro e o problema diagnosticado está no sistema operacional (arquivos corrompidos, desempenho degradado, incompatibilidades), a instalação limpa — precedida de backup dos dados necessários — é um dos procedimentos de reparo mais comuns em manutenção de computadores, por "reiniciar o ciclo de vida" do software.

Um cuidado adicional é necessário quando a causa do problema é um software malicioso (**malware**): se o backup incluir arquivos infectados, o malware é copiado junto com os dados legítimos e pode se propagar para o próximo computador em que esse backup for restaurado. Por isso, um procedimento de backup responsável inclui a verificação e remoção de arquivos contaminados antes da restauração.

7.2.2 O loop de instalação infinito

Um problema comum na primeira experiência com instalação de sistemas operacionais ocorre quando o dispositivo USB de instalação permanece como primeira opção na ordem de boot (Capítulo 6, §6.6.3). Nesse cenário, ao final da instalação, o computador reinicia, identifica novamente o pendrive como primeira opção de boot, e reinicia o processo de instalação do zero — repetindo esse ciclo indefinidamente, sem nunca chegar a carregar o sistema recém-instalado a partir do disco interno.

A correção consiste em, após concluída a instalação, remover a mídia USB ou reconfigurar a ordem de boot no Setup para priorizar o disco interno (HD ou SSD) sobre a mídia externa.

Figura pendente

tela típica de instalador solicitando "pressione qualquer tecla para iniciar a partir do CD ou DVD"

7.3 Live CD/USB como ferramenta de diagnóstico

Uma distribuição Linux como o Ubuntu, ao ser inicializada a partir de um pendrive, oferece tipicamente duas opções: **instalar** o sistema no disco,

ou **experimentar/testar** (*Live CD* ou *Live USB*) o sistema operacional sem instalá-lo.

No modo Live CD/USB, o sistema operacional completo é executado diretamente a partir da memória RAM, carregado da mídia externa, sem necessidade de gravar nada no disco interno da máquina — nem sequer é necessário que a máquina possua um disco de armazenamento funcional. Nesse modo, o usuário tem privilégios de administrador daquela sessão, o que abre uma série de possibilidades diagnósticas.

Aplicação prática — diagnóstico hardware versus software. Se o som não funciona no Windows instalado em determinada máquina, inicializar um Live USB e testar o som nele permite isolar a causa: se o som funcionar no Live USB, o problema está no software (driver ou configuração do Windows); se não funcionar em nenhum dos dois ambientes, a causa mais provável é hardware. O mesmo raciocínio se aplica a problemas de conectividade de rede ou de qualquer outro subsistema.

Por oferecer privilégios administrativos completos sobre uma máquina sem exigir senha, o Live CD/USB é também a ferramenta central por trás do risco de segurança descrito no Capítulo 6, §6.6.4: qualquer pessoa com acesso físico e um pendrive de instalação preparado pode acessar dados de um disco sem autenticação.

Figura pendente

tela de boas-vindas de um instalador Linux com as opções “Experimentar” (Try) e “Instalar” (Install)

7.4 Instalação do Linux: ponto de montagem raiz e partição swap

A instalação de uma distribuição Linux introduz um nível adicional de complexidade em relação à instalação do Windows, decorrente de sua organização de diretórios e do uso de uma partição de memória virtual.

No Linux, ao contrário do Windows — que atribui letras (C:, D:…) a cada unidade —, toda a estrutura de arquivos parte de uma única **raiz**, representada pela barra (`/`). Diretórios de sistema relevantes incluem `/dev` (arquivos que representam dispositivos conectados, como discos e pendrives), `/home` (pastas pessoais dos usuários) e `/tmp` e `/var` (arquivos temporários e variáveis do sistema), entre outros.

Discos conectados via SATA são identificados com o prefixo `sd` seguido de uma letra por disco (`sda` para o primeiro disco, `sdb` para o segundo, e assim por diante) e um número por partição dentro desse disco (`sda1`, `sda2`…).

Durante a instalação, é necessário indicar em qual partição o sistema deseja **montar** (*mount*) a raiz `/` — isto é, associar aquela partição ao ponto de entrada de toda a estrutura de diretórios do sistema. É essa exigência —

decidir “onde vai a barra” — que costuma ser o ponto de maior dificuldade para quem instala Linux pela primeira vez.

Além da partição raiz, a instalação típica de uma distribuição Linux requer uma segunda partição, chamada **swap** (área de troca): um espaço reservado em disco para funcionar como extensão da memória RAM, retomando o conceito de memória virtual e a hierarquia de memória apresentados no Capítulo 5. O dimensionamento da partição swap é função da quantidade de RAM instalada na máquina — não um valor fixo independente dela [2]; para uma máquina com 8–16 GB de RAM, uma faixa comum de referência é reservar entre 8 GB e 10 GB para a partição swap, destinando o restante do espaço disponível à partição raiz.

Exemplo de particionamento típico para instalação do Ubuntu: partição de swap de 8–10 GB e partição raiz (/) ocupando o espaço restante — por exemplo, em um espaço livre de 100 GB, aproximadamente 90 GB para / e 10 GB para swap.

Figura pendente

tela do particionador avançado de um instalador Linux, mostrando a criação da partição swap e a seleção do ponto de montagem /

7.5 Dual boot e o gerenciador de inicialização GRUB

Ao instalar uma distribuição Linux como o Ubuntu, é instalado também, por padrão, um **gerenciador de inicialização** chamado **GRUB**. O GRUB é o software responsável por, após o POST, apresentar ao usuário uma lista dos sistemas operacionais (ou das versões de kernel) disponíveis para carregamento, permitindo a escolha entre eles — o mecanismo de dual boot descrito no Capítulo 6, §6.4.1.

Essa lista de opções inclui não apenas diferentes sistemas operacionais, mas também diferentes versões do **kernel** do Linux, o núcleo do sistema, atualizado periodicamente. Como programas podem depender de uma versão específica do kernel para funcionar corretamente, a possibilidade de escolher, no momento do boot, qual versão carregar é uma funcionalidade prática relevante do GRUB.

Para instalar Windows e Linux em dual boot no mesmo disco, a **ordem de instalação é determinante**: o Windows deve ser instalado **antes** do Linux. Isso ocorre porque o instalador do Windows sobrescreve o setor de inicialização do disco (a região do MBR descrita no Capítulo 6, §6.5.1) sem preservar um gerenciador de boot alternativo ali presente [3]. Se o GRUB for instalado primeiro (ao instalar o Linux) e o Windows for instalado depois, a instalação do Windows sobrescreve o GRUB, tornando o Linux inacessível

na inicialização — embora os arquivos do sistema Linux continuem fisicamente intactos no disco, apenas inacessíveis por falta do menu de entrada. Recuperar o GRUB nessa situação é um procedimento de manutenção mais avançado.

Sequência recomendada para dual boot:

1. Instalar o Windows normalmente.
2. Reduzir (diminuir) a partição do Windows para liberar espaço não alocado.
3. Inicializar a mídia do Linux nesse espaço livre, criando a partição raiz (/) e a partição swap.
4. Ao final, o GRUB é instalado automaticamente e passa a apresentar, a cada boot, a opção de carregar Windows ou Linux.

Figura pendente

tela do menu GRUB listando as opções de inicialização Windows e Ubuntu

---

7.6 Drivers: a interface entre hardware e sistema operacional

Concluída a instalação do sistema operacional, um computador plenamente funcional ainda depende da instalação dos **drivers** apropriados. Um driver é um software, desenvolvido pelo fabricante de determinado componente de hardware, responsável por fazer a interface específica entre aquele hardware e o sistema operacional.

Sistemas operacionais diferentes exigem versões diferentes do mesmo driver — por exemplo, uma placa de vídeo tem uma versão de driver para Windows 10 e outra para Windows 11, desenvolvidas e distribuídas separadamente pelo fabricante da placa. Isso ocorre porque, embora o hardware seja o mesmo, o software que faz a comunicação com ele precisa ser compatível com a arquitetura interna de cada sistema operacional.

Exemplo. O procedimento padrão de instalação do Windows instala automaticamente **drivers genéricos** — versões simplificadas, capazes de operar minimamente qualquer hardware compatível, mas sem extrair seu desempenho pleno. Uma placa de vídeo de alto desempenho, adquirida separadamente do fabricante (por exemplo, uma GPU de gama alta), só entrega sua capacidade real de processamento gráfico depois que o driver **específico** fornecido pelo fabricante é instalado.

Um efeito visível e didático da instalação de um driver de vídeo específico é o aumento na **resolução** disponível para a tela — a contagem de pixels em largura e altura que a placa de vídeo consegue endereçar corretamente (por

exemplo, o padrão Full HD, de 1920×1080 pixels, seguido pelos padrões 2K e 4K, múltiplos dessa resolução).

Historicamente, drivers eram distribuídos em mídias físicas (CD, disquete) que acompanhavam o hardware adquirido; com a popularização da internet, passaram a ser baixados diretamente do site do fabricante e, mais recentemente, os próprios sistemas operacionais passaram a identificar e instalar automaticamente o driver recomendado (por exemplo, via Windows Update), desde que haja conexão à internet disponível no momento.

Aplicação prática. Um computador sem driver de placa de rede Wi-Fi instalado enfrenta um problema de dependência circular: não é possível baixar o driver da internet, porque o driver necessário para acessar a internet é justamente o que está faltando. Nesse cenário, o procedimento é obter o driver em outro computador com acesso à internet, transferi-lo por mídia física (pendrive) e instalá-lo localmente.

Do ponto de vista de diagnóstico técnico, muitos sintomas atribuídos erroneamente a defeito de hardware — ausência de som, Wi-Fi ou Bluetooth não funcionando, touchpad sem responder a gestos — são, na realidade, causados pela ausência ou desatualização do driver correspondente, sendo corrigidos pela instalação ou reinstalação do software apropriado, sem qualquer intervenção física no componente.

Figura pendente

gerenciador de dispositivos do Windows, mostrando a lista de hardware detectado e o status dos drivers instalados

---

7.7 Atualização do sistema operacional e do firmware

7.7.1 Atualização do sistema operacional

Um sistema operacional instalado (Seção 7.2) não é um produto estático: o fabricante publica, periodicamente, **atualizações** que corrigem falhas de segurança recém-descobertas, corrigem defeitos de funcionamento (*bugs*) e, com menor frequência, adicionam funcionalidades novas ou trocam a versão do kernel disponível (Seção 7.5, a propósito do GRUB). Do ponto de vista do usuário, a rotina de manutenção preventiva de software (aprofundada no Capítulo 12) inclui manter essas atualizações em dia — a maior parte das infecções por malware (Seção 7.2.1) explora falhas de segurança já corrigidas pelo fabricante, mas não aplicadas pelo usuário.

Nota prática. Uma atualização de sistema operacional, sobretudo uma troca de versão maior (por exemplo, de uma versão do Windows para a seguinte), pode alterar a compatibilidade de drivers já instalados (Seção 7.6) — um hardware antigo, sem driver atualizado disponível pelo fabricante,

pode deixar de funcionar corretamente após a atualização. Por essa razão, uma atualização importante deve ser precedida de backup (Seção 7.2.1), da mesma forma que uma reinstalação completa.

7.7.2 Atualização de firmware: por que o BIOS passou a ser atualizável

O Capítulo 6, §6.6, apresentou o BIOS/UEFI como o firmware responsável por inicializar o hardware antes do sistema operacional. Historicamente, esse firmware era gravado numa memória do tipo **ROM** — na acepção mais restrita do termo (Capítulo 5, §5.12): gravada uma única vez na fábrica, sem possibilidade de alteração posterior. Um defeito ou limitação identificado depois da fabricação da placa-mãe (por exemplo, não reconhecer um processador lançado meses depois) não tinha correção possível por software — a única solução era substituir fisicamente o chip de BIOS, quando o fabricante disponibilizava essa opção.

A mesma evolução de memória descrita no Capítulo 5 (§5.12) — de ROM para PROM, EPROM, EEPROM e, por fim, memória **flash** — chegou também ao chip de BIOS/UEFI das placas-mãe modernas: hoje ele é gravado em memória flash, **eletricamente regravável**, sem qualquer equipamento especial. Esse é o motivo técnico exato pelo qual o procedimento popularmente chamado de "**flashar a BIOS**" existe: atualizar o firmware da placa-mãe passou a ser tão simples quanto rodar um utilitário fornecido pelo fabricante (às vezes acessível diretamente de dentro do próprio Setup, Capítulo 6, §6.6.2) que regrava o conteúdo daquela memória flash com uma versão mais nova.

Por que atualizar o firmware. As razões mais comuns para atualizar a BIOS/UEFI de uma placa-mãe são: oferecer suporte a um processador lançado depois da placa (Capítulo 8, §8.4, sobre compatibilidade de soquete e geração), corrigir uma vulnerabilidade de segurança do próprio firmware, ou resolver um problema de estabilidade documentado pelo fabricante.

Atenção — risco de "brickar" a placa-mãe. Diferente de uma atualização de sistema operacional, que ocorre com o sistema já carregado e com redundância de software, a atualização de firmware sobrescreve a única cópia do programa que a placa-mãe usa para sequer iniciar o POST (Capítulo 9, §9.1). Uma interrupção no meio desse processo — uma queda de energia, por exemplo — pode corromper essa memória de forma irrecuperável pelos meios usuais, inutilizando a placa-mãe (um problema popularmente chamado de "*brickar*" o equipamento, numa referência a transformar o dispositivo num "tijolo" sem função). Por essa razão, atualizar o firmware exige alimentação estável (idealmente com um nobreak, tema tratado a propósito da fonte de alimentação no Capítulo 11) e nunca deve ser interrompida manualmente.

7.7.3 Recuperação de firmware corrompido: regravador externo e troca de chip

Quando uma atualização de firmware falha e a placa-mãe deixa de dar POST (Capítulo 9, §9.1.4), existem, em ordem crescente de intervenção física, três caminhos de recuperação:

1. **Recuperação assistida pela própria placa** — muitas placas-mãe modernas guardam, além da memória flash principal, uma pequena rotina de recuperação (frequentemente chamada de *BIOS Flashback* ou nome equivalente do fabricante) capaz de regravar o firmware a partir de um pendrive, mesmo sem processador, memória ou vídeo instalados — o próprio chipset (Capítulo 10, §10.1.2) executa essa rotina de resgate de forma independente do restante do sistema.
2. **Regravador externo (*programmer*).** Quando esse recurso não existe ou também falha, o chip de memória flash da BIOS — fisicamente soquetado na maioria das placas-mãe, tipicamente num encapsulamento SOIC-8 — pode ser removido com cuidado e regravado fora da placa-mãe, usando um dispositivo dedicado chamado **regravador** (ou *programmer*, o mesmo tipo de equipamento historicamente usado para gravar uma EPROM antes da existência da memória flash — Capítulo 5, §5.12). O regravador se conecta a um computador auxiliar, recebe o arquivo de firmware correto e o grava diretamente nos terminais do chip, sem depender de o sistema já defeituoso conseguir executar qualquer software. Depois de regravado, o chip é reencaixado no soquete da placa-mãe.
3. **Troca do chip.** Quando o chip de BIOS está fisicamente danificado (e não apenas com o conteúdo corrompido), ou quando não há regravador disponível, a alternativa é substituí-lo por um chip novo, da mesma referência, já gravado de fábrica com o firmware correto — um serviço oferecido por fornecedores especializados em peças de reposição de placa-mãe. Esse procedimento só é possível em placas cujo chip de BIOS é soquetado (destacável); em placas nas quais o chip é soldado diretamente à placa-mãe, a troca exige retrabalho de solda em nível de componente, um serviço mais especializado e caro.

7.7.4 BIOS duplo (Dual BIOS)

Algumas placas-mãe de gama mais alta mitigam o risco descrito acima já em nível de projeto, incorporando um **BIOS duplo**: dois chips de memória flash fisicamente independentes na mesma placa, um designado **principal** e outro **de backup**, ambos capazes de armazenar o firmware completo. Durante o boot, um pequeno circuito de controle valida a integridade do chip principal (por meio de uma soma de verificação, ou *checksum*); se essa validação falhar — por exemplo, após uma atualização interrompida por queda de energia —, a placa alterna automaticamente para o chip de backup, que assume a inicialização e, em muitos modelos, oferece a opção de regravar

o chip principal corrompido a partir do próprio backup, sem exigir regravador externo (Seção 7.7.3) nem substituição física de peça. Algumas placas expõem ainda um interruptor físico dedicado, permitindo alternar manualmente entre os dois chips — por exemplo, para testar deliberadamente uma versão de firmware mais nova no chip secundário, mantendo o chip principal como versão estável conhecida.

Esse mecanismo é uma aplicação, em hardware e em pequena escala, do mesmo princípio de redundância apresentado a propósito de servidores no Capítulo 1 (§1.11.2): duplicar um componente crítico para que a falha de uma cópia não interrompa o funcionamento do sistema como um todo.

Figura pendente

foto de uma placa-mãe de gama alta evidenciando os dois chips de BIOS lado a lado, com o interruptor de seleção entre eles

---

7.8 Malware, vírus e cavalos de troia

Um problema de software identificado durante o diagnóstico pode ter origem em **software malicioso** (*malware*) instalado no computador sem o conhecimento ou consentimento do usuário.

- **Vírus**: software malicioso cuja característica definidora é a **replicação** — a capacidade de se propagar de um computador para outro, contaminando novas máquinas.
- **Trojan** (cavalo de troia): software que se apresenta como um programa legítimo, mas que traz embutido um código malicioso oculto, executado em segundo plano após a instalação.

Exemplo. Em 2023, o maior canal de tecnologia do YouTube, o Linus Tech Tips, teve seus canais sequestrados após um funcionário da área de publicidade abrir um PDF contaminado recebido como proposta comercial. O software malicioso obteve acesso ao navegador da máquina infectada e, através das permissões já concedidas àquele computador, passou a publicar em todos os subcanais do grupo vídeos promovendo um golpe de criptomoeda associado à imagem de uma figura pública. Levou cerca de 24 horas para a equipe recuperar o controle dos canais [4] — todo o alcance da rede havia sido redirecionado para o conteúdo malicioso antes que o acesso fosse restabelecido.

Esse caso ilustra por que a origem de um software instalado importa: um malware pode chegar embutido em qualquer instalador, mesmo em arquivos aparentemente inofensivos como um documento PDF.

7.9 Reinstalação do sistema operacional como manutenção corretiva

Diante de um problema de software cuja causa exata não foi identificada, uma prática comum — ainda que nem sempre a mais eficiente — é reinstalar o sistema operacional por completo, apagando o disco e recomeçando do zero.

Ainda assim, a reinstalação continua sendo uma ferramenta legítima de manutenção corretiva, sobretudo quando o tempo de diagnóstico pontual excederia o tempo do próprio procedimento de reinstalação. Duas implicações são obrigatórias sempre que esse caminho é adotado:

1. **Perda de dados**: a reinstalação apaga o disco. É dever do técnico alertar o usuário e obter confirmação de que uma cópia de segurança (backup) dos dados relevantes foi realizada antes de iniciar o procedimento.
2. **A reinstalação só resolve problemas de software**: se o sintoma reaparecer após uma reinstalação bem-sucedida, a hipótese de defeito de hardware sobe de prioridade.

Figura pendente

fluxograma de decisão — sintoma relatado → hipótese hardware/software/usuário → teste da hipótese mais barata → correção pontual ou reinstalação

---

7.10 Síntese do capítulo

Este capítulo apresentou o processo integral de instalação de um sistema operacional, da preparação da mídia à instalação de drivers, passando por backup, dual boot e diagnóstico via Live CD/USB. O capítulo fechou com a manutenção contínua dessa camada de software e firmware — atualização do sistema operacional, atualização e recuperação do firmware da placa-mãe, a redundância de hardware que o BIOS duplo oferece contra esse risco — e com dois problemas recorrentes de manutenção corretiva de software, malware e a decisão de reinstalar o sistema operacional. Esses procedimentos dependem diretamente dos componentes físicos tratados nos Capítulos 8 a 10 — a placa-mãe, seus barramentos e os próprios dispositivos de armazenamento — e do firmware gravado nesses componentes, cuja relação com a arquitetura de processadores foi aprofundada no Capítulo 3.

7.11 Referências

1. RUFUS. Página oficial. Disponível em: https://rufus.ie/; repositório de código-fonte: https://github.com/pbatard/rufus.
2. RED HAT. “Recommended system swap space.” *Red Hat Enterprise Linux 8 Documentation*. Disponível em: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/managing_storage_devices/getting-started-with-swap_managing-storage-devices.
3. UBUNTU COMMUNITY HELP WIKI. “WindowsDualBoot.” Disponível em: https://help.ubuntu.com/community/WindowsDualBoot.
4. TECHSPOT. “YouTube channel Linus Tech Tips terminated after it was hacked to show crypto-scam videos.” 2023. Disponível em: https://www.techspot.com/news/98047-; DIGITAL TRENDS. “Linus Tech Tips restored after crypto scam hack.” Disponível em: https://www.digitaltrends.com/computing/linus-tech-tips-offline-after-cryptosca
5. Páginas oficiais: RUFUS, https://rufus.ie; VENTOY, https://www.ventoy.net; YUMI, https://www.pendrivelinux.com.

8 Montagem Física: Componentes, CPU e Refrigeração

Neste capítulo você vai estudar os componentes físicos de um computador desktop e o procedimento de desmontagem e remontagem do gabinete, o painel frontal e o botão liga/desliga, os critérios de compatibilidade entre processador e placa-mãe (fabricante, soquete e geração), os sistemas de refrigeração a ar e a líquido — incluindo a função física da pasta térmica —, e a placa-mãe como objeto físico: seu fator de forma e seus jumpers.

8.1 Componentes de um computador desktop e modularidade aplicada

Conforme apresentado no Capítulo 1, o computador é um dispositivo **modular**: constituído por submódulos substituíveis que trabalham em conjunto. Um computador do tipo **desktop** — assim chamado por ter sido projetado para operar sobre uma mesa (*desk*) — reúne esses submódulos dentro de um gabinete.

A tabela a seguir relaciona os principais submódulos de um desktop e sua essencialidade para o funcionamento mínimo do sistema:

Submódulo	Função	Essencial para ligar?
Gabinete	Estrutura metálica que abriga e organiza os demais componentes	Não
Fonte de alimentação	Converte corrente alternada (rede elétrica) em corrente contínua nas tensões exigidas pelos componentes internos	Sim
Placa-mãe	Interconecta todos os demais submódulos e os dispositivos de entrada e saída	Sim

Submódulo	Função	Essencial para ligar?
Processador (CPU)	Executa as instruções dos programas	Sim
Memória primária (RAM)	Armazena, durante a execução, os dados e instruções em uso	Sim
Memória secundária (HD, SSD, unidade óptica)	Armazenamento não volátil de longo prazo	Não
Sistema de refrigeração (cooler)	Dissipa o calor gerado pelos componentes, em especial a CPU	Não (mas crítico à operação contínua)
GPU	Processamento gráfico — pode estar integrada ao próprio chip da CPU ou ser um componente externo	Depende (não essencial se houver GPU integrada)

Exemplo — a fonte de alimentação. A rede elétrica residencial fornece corrente alternada (no Brasil, tipicamente 127 V ou 220 V). Todos os componentes internos do computador, no entanto, operam em corrente contínua, em diversas tensões específicas. A fonte de alimentação é o submódulo responsável por essa conversão. Sem ela, nenhum outro componente recebe energia, e o computador simplesmente não liga.

Figura pendente

gabinete desktop aberto com fonte, placa-mãe, CPU, RAM e armazenamento secundário identificados por setas

8.1.1 Interconexões internas

A fonte de alimentação entrega energia diretamente à placa-mãe por meio de conectores dedicados (um conector principal mais robusto e um conector adicional para a CPU). A própria placa-mãe, por sua vez, distribui energia à CPU e à memória RAM — ou seja, esses dois submódulos não recebem alimentação diretamente da fonte, mas sim através da placa-mãe.

A memória secundária (HD, SSD ou unidade óptica) recebe dois cabos distintos quando conectada via interface **SATA** (*Serial ATA*): um cabo de dados, que liga o dispositivo à placa-mãe, e um cabo de energia, que vem diretamente da fonte. Já os dispositivos de armazenamento no padrão **NVMe** (conectados em um slot M.2) dispensam o cabo de energia externo, pois recebem alimentação diretamente da própria placa-mãe.

8.2 Desmontagem e remontagem física de um desktop

O procedimento de desmontagem de um desktop segue uma ordem lógica que reflete as dependências elétricas descritas na Seção 8.1.1.

Procedimento de segurança. Antes de qualquer manuseio interno, o computador deve estar desligado e desconectado da tomada. Componentes eletrônicos devem ser manuseados pelas bordas, evitando o contato direto com pontos de contato elétrico, com atenção à descarga eletrostática acumulada pelo corpo humano.

Ferramenta necessária. A abertura do gabinete é feita com uma chave de fenda do tipo **Philips** — popularmente chamada de chave “estrela”.

Sequência de desmontagem:

1. Desconectar a fonte de alimentação de qualquer unidade de armazenamento externo (leitor de CD/DVD, por exemplo) e o cabo de energia da fonte.
2. Desconectar os cabos de dados (SATA) e de energia da memória secundária, e removê-la.
3. Remover os módulos de memória RAM dos seus encaixes na placa-mãe.
4. Retirar os parafusos que fixam a placa-mãe ao gabinete e removê-la, mantendo o cooler acoplado à CPU (a remoção da CPU e do cooler é tratada separadamente, nas Seções 8.4 a 8.6).

A remontagem segue a **ordem inversa**: fixar a placa-mãe no gabinete, encaixar a memória RAM, conectar a fonte à placa-mãe, reconectar a memória secundária e, por fim, fechar o gabinete.

Procedimento de manuseio. Componentes como módulos de memória RAM possuem uma chave física (um entalhe no conector) que permite o encaixe em uma única orientação. Se for necessário aplicar força excessiva para encaixar um componente, isso indica, na maioria dos casos, que ele está sendo inserido de forma invertida — o procedimento correto é interromper, reavaliar a orientação da peça e não insistir com força.

Figura pendente

sequência fotográfica da desmontagem — fonte, armazenamento secundário, RAM, placa-mãe] [IMAGEM: detalhe do encaixe chaveado (entalhe) de um módulo de memória RAM no slot da placa-mãe

8.2.1 Instalação de placas de expansão

Diferente da memória RAM e da CPU — que se conectam a um encaixe específico e único na placa-mãe (Seções 8.1 e 8.4) —, uma placa de expansão (GPU, placa de captura, placa de rede adicional) se conecta a um dos slots PCIe descritos no Capítulo 10, §10.1.4, e o procedimento de instalação é o mesmo independentemente da função da placa:

1. Com o computador desligado e desconectado da tomada, remover a tampa metálica correspondente na parte traseira do gabinete (o local por onde os conectores da placa ficarão expostos).
2. Liberar a presilha de retenção do slot PCIe escolhido — normalmente uma pequena trava plástica na extremidade do slot, que impede a placa de se soltar por vibração.
3. Alinhar o conector dourado da placa ao slot e pressionar verticalmente e de forma uniforme, dos dois lados da placa, até sentir e (quando o slot tiver esse recurso) ouvir o encaixe da presilha.
4. Fixar a placa ao gabinete com o parafuso correspondente à tampa removida no passo 1, garantindo que ela não se desloque quando cabos forem conectados a ela.
5. Quando aplicável — como é o caso de GPUs de médio e alto desempenho —, conectar o(s) conector(es) de alimentação adicional vindos diretamente da fonte, que suprem uma placa cujo consumo excede o que o próprio slot PCIe é capaz de fornecer (o dimensionamento de fonte para acomodar esse consumo extra é tratado em profundidade no Capítulo 11, §11.4).

Nota prática. Os mesmos cuidados de manuseio já apresentados neste capítulo se aplicam aqui: nunca forçar o encaixe, manusear a placa pelas bordas, e observar a orientação correta indicada pelo próprio formato do conector — uma placa PCIe, assim como um módulo de memória (Seção 8.2), só se encaixa fisicamente numa única orientação.

8.3 O painel frontal e o botão liga/desliga

Como decorrência do modelo de hardware aberto adotado desde o IBM PC (Capítulo 1), cada componente de um desktop é tipicamente fabricado por uma empresa diferente: o gabinete — e, portanto, o painel frontal nele embutido — não é necessariamente do mesmo fabricante da placa-mãe. Por essa razão, qualquer funcionalidade presente no painel frontal (botão de energia, botão de reset, indicadores luminosos, portas USB) só opera se estiver **fisicamente conectada** à placa-mãe por um cabo: não existe comunicação sem fio entre painel frontal e placa-mãe.

8.3.1 Funcionamento elétrico do botão

O botão de liga/desliga é, mecanicamente, um interruptor momentâneo com apenas dois terminais. Em repouso, uma mola mantém os dois terminais desconectados. Ao ser pressionado, o botão conecta momentaneamente os dois terminais, permitindo a passagem de corrente entre eles; ao ser solto, a mola desfaz essa conexão.

O sinal relevante para a placa-mãe é a **transição momentânea** de "desconectado" para "conectado" — não é necessário manter o botão pressionado. Uma pressão breve é interpretada como o comando padrão de ligar (ou desligar de forma controlada, se o sistema já estiver ligado); manter o botão pressionado por um período prolongado é interpretado como um comando distinto, tipicamente o desligamento forçado.

8.3.2 Diagnóstico por curto controlado

Como o botão de energia nada mais é do que um par de terminais que se conectam momentaneamente, é possível ligar o computador **sem o painel frontal**, encostando um objeto condutor (por exemplo, a ponta metálica de uma chave de fenda) diretamente nos dois pinos correspondentes ao botão de energia na placa-mãe — reproduzindo o mesmo efeito elétrico do acionamento do botão.

Esse procedimento é usado como técnica de diagnóstico para isolar o painel frontal como possível causa de falha ao ligar. Por envolver contato direto com a placa-mãe energizada, deve ser realizado com os seguintes cuidados:

- Uso de calçado fechado e de uma ferramenta com cabo isolado.
- Contato restrito exclusivamente aos dois pinos corretos do conector de energia (identificados no manual da placa-mãe) — o toque em pontos incorretos pode danificar o equipamento.

O painel frontal também disponibiliza, tipicamente, um segundo botão equivalente: o botão de **reset**.

Figura pendente

detalhe do conector de pinos do painel frontal na placa-mãe, com o par correspondente ao botão de power identificado

8.4 CPU: fabricantes, soquetes e gerações

O processador é, em geral, um dos componentes mais caros de um computador desktop. O mercado de CPUs para desktop é dominado por dois fabricantes: **Intel** e **AMD** — ao contrário do mercado de notebooks, no qual, cada vez mais, a CPU vem soldada diretamente à placa-mãe, eliminando a possibilidade de substituição. A imensa maioria dos computadores desktop disponíveis no mercado utiliza, em 2026, processadores da arquitetura **x64**, cujos detalhes de conjunto de instruções são tratados no Capítulo 3.

8.4.1 Compatibilidade entre CPU e placa-mãe: o soquete

O **soquete** é o conector físico da placa-mãe no qual o processador é encaixado. Diferentemente de um módulo de memória RAM — cujo soquete é

padronizado e aceita módulos de fabricantes distintos (por exemplo, Kingston ou Samsung) —, o soquete de CPU é **específico do fabricante do processador**: uma placa-mãe projetada para processadores Intel não aceita processadores AMD, e vice-versa.

Além da compatibilidade entre fabricantes, é preciso observar a compatibilidade entre **gerações** de processador dentro de um mesmo fabricante:

- A **Intel**, historicamente, altera o soquete a cada uma ou duas gerações de processador. Em alguns casos, duas gerações consecutivas compartilham o mesmo soquete físico, mas ainda assim são incompatíveis entre si — a placa-mãe precisa ter sido projetada especificamente para aquela geração.
- A **AMD** tende a manter um mesmo soquete (por exemplo, o soquete AM4, mantido de 2016 a 2022) por várias gerações consecutivas de processadores, por vezes exigindo apenas uma atualização de BIOS para suportar uma geração mais recente [1].

Exemplo real. O soquete LGA1155, usado em algumas placas-mãe Intel, suporta processadores de 2ª geração (nome de código Sandy Bridge) e de 3ª geração (Ivy Bridge) [2]. Identificar qual geração uma determinada placa-mãe suporta é informação disponível no manual do fabricante — não é conhecimento a ser memorizado, mas sim consultado no momento da especificação ou do reparo. **Atenção:** o soquete compartilhado não garante, por si só, compatibilidade total — o chipset **H61**, por exemplo, é LGA1155 mas não suporta Ivy Bridge mesmo com atualização de BIOS.

8.4.2 Diferenças físicas entre os soquetes Intel e AMD

Um processador é conectado à placa-mãe por meio de um conjunto de contatos elétricos. Historicamente, os dois fabricantes adotam estratégias opostas quanto à localização física desses contatos:

- **Intel**: os pinos ficam alojados no **soquete da placa-mãe**; o processador possui apenas contatos planos (sem pinos salientes).
- **AMD** (na abordagem tradicional): os **pinos ficam no próprio processador**, e o soquete da placa-mãe possui os furos correspondentes.

Atenção: a AMD abandonou essa abordagem tradicional a partir do soquete **AM5** (lançado em 2022, usado pelos Ryzen 7000 em diante), que passou a ser LGA — os pinos ficaram na placa-mãe, como na Intel [3]. A descrição acima vale para os soquetes AMD anteriores ao AM5 (como o AM4).

Em ambos os casos, um pino entortado é um dano frequentemente irreparável: quando localizado na placa-mãe, compromete a placa inteira; quando localizado no processador, compromete o processador. Por isso, o manuseio de processadores e soquetes exige cuidado e nunca deve envolver força excessiva.

Todo processador possui uma marca de alinhamento (tipicamente um pequeno triângulo em um dos cantos) que corresponde a uma marca equivalente no soquete da placa-mãe. Esse par de marcas garante que o processador só possa ser encaixado em uma única orientação correta.

Figura pendente

fotos comparativas de um soquete Intel (LGA, com pinos na placa) e um soquete AMD tradicional (PGA, com pinos no processador)] [IMAGEM: detalhe da marca de alinhamento (triângulo) no canto do processador, alinhada à marca correspondente no soquete

8.4.3 Critérios de especificação: não existe “o melhor”

Além do fabricante, geração e soquete, os processadores variam amplamente em preço e capacidade de processamento — de modelos de entrada a modelos de milhares de reais. Uma capacidade de processamento maior não se traduz automaticamente em melhor desempenho para toda e qualquer aplicação: o desempenho real depende também de como o software em uso foi projetado para tirar proveito do hardware disponível.

Exemplo hipotético. Imagine um usuário que investisse um valor elevado em um processador de alto número de núcleos, esperando desempenho superior em um jogo popular. O resultado, nesse cenário, ficaria aquém do esperado: se o jogo em questão foi desenvolvido para utilizar poucos núcleos simultaneamente, o excesso de núcleos disponíveis não poderia ser aproveitado — o gargalo estaria na interdependência entre hardware e software, não na capacidade bruta do processador.

Esse caso ilustra um princípio central da disciplina: **não existe “o melhor” componente em termos absolutos — existe o componente mais adequado a um orçamento e a uma finalidade específicos.** O mesmo raciocínio se aplica a outras decisões de hardware e rede: uma rede Wi-Fi em 2,4 GHz oferece maior alcance com menor velocidade, enquanto uma rede em 5 GHz oferece maior velocidade com menor alcance — trata-se de uma escolha de compromisso, não da existência de uma opção objetivamente superior. Da mesma forma, um processador de maior capacidade de processamento tende a consumir mais energia, o que é indesejável em um dispositivo móvel dependente de bateria. Essas relações de compromisso estão associadas à **arquitetura** do processador, tema aprofundado no Capítulo 3.

8.5 Sistemas de refrigeração: ar e líquido

Todo processador em operação gera calor, o que exige um sistema de refrigeração para manter sua temperatura dentro de limites seguros. Um sistema de refrigeração não **controla** a temperatura para um valor espe-

cífico — ele apenas **remove** calor continuamente, independentemente da temperatura ambiente.

8.5.1 Componentes ativo e passivo do cooler a ar

O sistema de refrigeração a ar (*air cooler*) mais comum é composto por dois elementos:

- **Componente passivo** — o **dissipador**: uma peça metálica de alta condutividade térmica, fixada em contato direto com o processador, que aumenta a área de contato com o ar por meio de aletas.
- **Componente ativo** — a **ventoinha**: um ventilador alimentado eletricamente (conectado à placa-mãe ou à fonte por um conector de três pinos) que força a circulação de ar através das aletas do dissipador.

8.5.2 Condutividade térmica dos materiais

A eficiência de um sistema de refrigeração depende diretamente da condutividade térmica do meio utilizado para transferir o calor. A tabela a seguir apresenta valores relativos de condutividade térmica para os materiais discutidos nesta seção [4]:

Material	Condutividade térmica (ordem de grandeza relativa)
Prata	428
Cobre	398
Alumínio	247
Tungstênio	178
Ferro	80
Vidro	0,8
Água	0,57
Tijolo	0,66
Madeira	0,11
Fibra de vidro	0,04
Ar	0,026

O ar é um condutor térmico deficiente — a água é aproximadamente **22 vezes mais condutiva termicamente do que o ar** [4]. É por essa razão que sistemas de refrigeração líquida conseguem remover calor de forma mais eficiente do que sistemas a ar, especialmente em processadores de alto consumo energético.

8.5.3 Refrigeração líquida (water cooling)

Em um sistema de refrigeração líquida, um fluido (água destilada ou um líquido refrigerante específico) circula em um circuito fechado: passa por uma base em contato com o processador, absorve calor, segue por um tubo

até um **radiador** — onde uma área de contato maior com o ar permite a dissipação do calor absorvido — e retorna, já resfriado, ao ponto de partida. Uma bomba e uma ou mais ventoinhas, ambas exigindo alimentação elétrica, mantêm esse ciclo em funcionamento contínuo.

Observação de segurança. O líquido refrigerante não entra em contato direto com o processador: ele circula por uma base metálica que, por sua vez, está em contato com a CPU. Essa separação é intencional, já que muitos líquidos conduzem eletricidade, o que tornaria o contato direto com os circuitos do processador um risco de curto-circuito.

Figura pendente

esquema de um sistema de refrigeração a ar (dissipador + ventoinha) lado a lado com um sistema de water cooling (bloco, tubos, radiador, bomba)

8.6 Pasta térmica: função física e procedimento de troca

Mesmo com um dissipador (a ar ou líquido) corretamente instalado, a superfície de contato entre o processador e a base do dissipador não é perfeitamente lisa em nível microscópico: pequenas imperfeições geram **microlacunas preenchidas por ar**. Como visto na Seção 8.5.2, o ar é um péssimo condutor térmico — nessas microlacunas, ele atua como um **isolante**, prejudicando a transferência de calor entre o processador e o dissipador.

A **pasta térmica** é um composto de alta condutividade térmica aplicado sobre a superfície do processador antes da instalação do dissipador, com a função específica de preencher essas microlacunas, substituindo o ar (isolante) por um material condutor.

8.6.1 Procedimento de troca de pasta térmica

1. **Remoção do dissipador.** Desconectar o cooler da placa-mãe (conector de alimentação da ventoinha) e liberar as presilhas ou parafusos mecânicos que o fixam.
2. **Remoção da pasta térmica antiga.** Limpar o excesso com papel; em caso de resíduo endurecido, utilizar **álcool isopropílico**, substância recomendada para limpeza de componentes internos por ser altamente volátil (evapora rapidamente, sem deixar resíduo). O uso de produtos como “limpa-contato” deve ser evitado nessa etapa: por serem mais abrasivos — formulados para remover oxidação —, não são a opção adequada para essa limpeza específica.
3. **Remoção e reinstalação da CPU** (quando aplicável). Identificar a marca de alinhamento do processador (Seção 8.4.2) e alinhá-la à marca correspondente do soquete antes de reencaixá-lo, sem aplicar força.

4. **Aplicação da nova pasta térmica.** A quantidade necessária é pequena — comparável à quantidade de pasta de dente usada em uma escovação (bem menos do que costuma ser mostrado em comerciais). Para pastas em bisnaga, o padrão usual é aplicar em forma de "X" ou cruz no centro do processador; a própria pressão do dissipador, ao ser instalado, espalha a pasta uniformemente pela superfície, preenchendo as microlacunas sem necessidade de espalhamento manual prévio.
5. **Reinstalação do dissipador**, com fixação firme e uniforme das presilhas ou parafusos, garantindo contato mecânico rígido entre processador e dissipador — uma fixação frouxa reintroduz os espaços de ar que a pasta térmica deveria eliminar.

Nota sobre qualidade e custo. Pastas térmicas variam amplamente em preço (por exemplo, entre R$ 19,99 e R$ 59,90, a depender da marca e da formulação), refletindo diferenças no grau de condutividade térmica. Como regra prática, qualquer pasta térmica aplicada corretamente supera, em desempenho, a ausência completa de pasta.

Figura pendente

sequência fotográfica — remoção do dissipador, limpeza da pasta antiga, aplicação em "X" da nova pasta térmica, reinstalação do dissipador] [IMAGEM: foto aproximada da marca de alinhamento (chanfro/triângulo) da CPU sendo posicionada no soquete

---

8.7 Placa-mãe: fator de forma e organização interna

As seções anteriores deste capítulo trataram a placa-mãe do ponto de vista funcional — o que ela conecta (Seção 8.1), como ela se relaciona com o soquete do processador (Seção 8.4). Esta seção trata da placa-mãe como objeto físico: seu tamanho padronizado e os pequenos componentes de configuração manual que ela expõe.

8.7.1 Fator de forma (form factor)

O **fator de forma** de uma placa-mãe é o seu padrão de dimensões físicas e posicionamento de furos de fixação — um combinado (no mesmo sentido de "combinado" usado no Capítulo 5 a propósito da célula de memória) entre fabricantes de placa-mãe e fabricantes de gabinete, que garante que qualquer placa de um determinado fator de forma se encaixe em qualquer gabinete compatível com esse mesmo padrão.

Fator de forma	Dimensões aproximadas [5]	Slots de expansão (observação de mercado, não parte da especificação)	Uso típico
ATX	305 × 244 mm	Mais slots (tipicamente 4 a 7)	Desktop de uso geral, estações de trabalho
microATX (mATX)	244 × 244 mm	Menos slots (tipicamente 2 a 4)	Desktop compacto, custo reduzido
Mini-ITX	170 × 170 mm	Geralmente 1 slot	Computadores muito compactos, *home theater PC*, projetos de nicho

O número de slots de expansão não é fixado pela especificação do fator de forma em si — depende do projeto de cada fabricante de placa-mãe; as faixas acima são uma observação de mercado, não uma regra normativa. Quanto menor o fator de forma, menor o gabinete que ele permite montar — mas menos espaço físico sobra para slots de expansão (Capítulo 10, §10.1.4), conectores de alimentação e, com frequência, para soquetes adicionais de memória. A escolha do fator de forma é, portanto, um compromisso entre compacidade e capacidade de expansão futura, e deve ser decidida antes da compra do gabinete: um gabinete ATX aceita placas ATX, microATX e Mini-ITX (por retrocompatibilidade de posicionamento de furos), mas um gabinete Mini-ITX aceita **somente** placas Mini-ITX.

Figura pendente

três placas-mãe (ATX, microATX, Mini-ITX) fotografadas lado a lado na mesma escala, evidenciando a diferença de tamanho

8.7.2 Onboard versus offboard

Um recurso é dito **onboard** quando sua funcionalidade está integrada diretamente ao chipset ou à própria placa-mãe, sem exigir uma placa de expansão dedicada; é dito **offboard** quando depende de uma placa de expansão separada, conectada a um slot (Capítulo 10, §10.1.4).

Exemplo. Toda placa-mãe moderna inclui vídeo onboard (herdado do processador, quando este tem GPU integrada — Seção 1.10.5), áudio onboard (um chip dedicado de áudio, presente na quase totalidade das placas atuais) e rede onboard (controlador Ethernet e, em muitos modelos, Wi-Fi).

Um usuário com necessidades gráficas mais intensas (jogos, edição de vídeo) costuma instalar uma GPU offboard, conectada a um slot de expansão (Capítulo 10, §10.1.4), mesmo já possuindo vídeo onboard — nesse caso, o sistema geralmente desabilita automaticamente a saída de vídeo onboard em favor da placa offboard, embora ambas continuem fisicamente presentes.

Historicamente, antes da integração em larga escala promovida pelo chipset moderno (Capítulo 10, §10.1.2), até mesmo o controlador de rede e o de som eram tipicamente offboard — daí o nome "placa de som" e "placa de rede" ainda usados no vocabulário popular, mesmo quando o recurso em questão hoje é onboard na maioria dos computadores vendidos.

8.7.3 Jumpers

Um **jumper** é um pequeno conector plástico, revestido internamente por metal condutor, que une fisicamente dois pinos adjacentes de um conjunto de pinos expostos na placa-mãe — fechando um circuito simples e sinalizando, em hardware, uma escolha binária de configuração (ligado/desligado, modo A/modo B).

Exemplo mais comum atualmente: o jumper Clear CMOS. O Capítulo 9 apresenta a memória CMOS, que retém as configurações do *setup* graças à bateria da placa-mãe. Quando essas configurações ficam corrompidas — por exemplo, após uma tentativa de overclock malsucedida que impede o computador de sequer completar o POST — a correção mais confiável não é remover a bateria (o que, em muitas placas modernas, não é suficiente para descarregar completamente o capacitor de retenção), mas usar o jumper "Clear CMOS" (às vezes rotulado CLRTC ou similar): com o computador desligado e desconectado da tomada, move-se o conector plástico da posição padrão para a posição adjacente por alguns segundos, forçando a descarga da memória CMOS, e depois o devolve à posição original. O resultado é equivalente ao de uma placa nova de fábrica: todas as configurações de *setup* voltam ao padrão do fabricante.

Nota prática. A posição exata do jumper Clear CMOS — e de qualquer outro jumper presente numa placa específica — varia por fabricante e modelo, e deve ser sempre conferida no manual da placa-mãe antes de qualquer manuseio, seguindo a mesma recomendação já feita a propósito da pinagem do painel frontal (Seção 8.3) e dos conectores de alimentação do processador (Capítulo 11, §11.5).

Jumpers já foram mais comuns no passado — por exemplo, para selecionar manualmente a posição *master/slave* de um HD conectado por interface IDE, um padrão anterior ao SATA (Capítulo 10, §10.1.5) — e hoje sobrevivem principalmente no Clear CMOS e em funções avançadas de diagnóstico em placas de servidor e de entusiasta.

Figura pendente

foto aproximada de um bloco de pinos Clear CMOS na placa-mãe, com o jumper na posição padrão e, ao lado, na posição de reset

---

8.8 Síntese do capítulo

Este capítulo tratou dos componentes físicos de um computador desktop, do procedimento de desmontagem e remontagem do gabinete, do funcionamento elétrico do botão liga/desliga e do painel frontal, dos critérios de compatibilidade entre processador e placa-mãe (fabricante, soquete e geração), e dos sistemas de refrigeração a ar e a líquido, incluindo a função física da pasta térmica. O capítulo também situou a placa-mãe como objeto físico — seu fator de forma e seus jumpers. Os critérios de especificação de CPU introduzidos aqui — número de núcleos, desempenho por núcleo, consumo energético — dependem diretamente do conceito de **arquitetura de processadores**, aprofundado no Capítulo 3. O Capítulo 9 retoma o POST — já mencionado aqui como pano de fundo do jumper Clear CMOS — sob a ótica dos componentes de hardware que ele avalia, e o Capítulo 10 trata a placa-mãe como via de comunicação: a hierarquia de barramentos e os protocolos que ligam processador, memória, armazenamento e periféricos.

8.9 Referências

1. TECHPOWERUP. “AMD Socket AM5 an LGA of 1,718 Pins with DDR5 and PCIe Gen 4.” Disponível em: https://www.techpowerup.com/282532/amd-socket-am5-an-lga-of-1-718-pins-with-ddr5-and-pcie-gen-4
2. WIKIPEDIA. “LGA 1155.” Disponível em: https://en.wikipedia.org/wiki/LGA_1155; documentação Intel ARK para os chipsets específicos.
3. TECHPOWERUP. “AMD Socket AM5 an LGA of 1,718 Pins with DDR5 and PCIe Gen 4.” Disponível em: https://www.techpowerup.com/282532/amd-socket-am5-an-lga-of-1-718-pins-with-ddr5-and-pcie-gen-4
4. PRÄSS, Alberto Ricardo. “Condutividade Térmica — Constantes Físicas.” fisica.net. Disponível em: https://www.fisica.net/constantes/condutividade-termica-(k).php.
5. WIKIPEDIA. “ATX.” Disponível em: https://en.wikipedia.org/wiki/ATX; “MicroATX.” Disponível em: https://en.wikipedia.org/wiki/MicroATX; “Mini-ITX.” Disponível em: https://en.wikipedia.org/wiki/Mini-ITX.

9 POST, CMOS e Setup

Neste capítulo você vai revisitar, a partir dos componentes de hardware envolvidos, os três elementos do firmware que preparam o computador para o carregamento do sistema operacional: o POST (o autoteste que verifica se o hardware vital está funcional), a bateria CMOS (que retém as configurações desse firmware entre desligamentos) e o Setup (o programa que permite alterá-las) — já apresentados do ponto de vista do software no Capítulo 6.

9.1 O POST sob a ótica dos componentes de hardware

O Capítulo 6 apresentou o POST (*Power-On Self-Test*, autoteste de inicialização) do ponto de vista do software: o primeiro programa executado após a energização do sistema, responsável por verificar o hardware antes de transferir o controle do processador para o sistema operacional (ou para um instalador, no caso de uma instalação de sistema operacional). Este capítulo revisita o POST a partir dos componentes de hardware que ele avalia.

O POST é parte de um firmware maior, o **BIOS** (*Basic Input/Output System*), também referido pela designação mais atual **UEFI** (*Unified Extensible Firmware Interface*). Esse firmware reside fisicamente na placa-mãe, junto com o programa de *setup* — a interface usada para configurar parâmetros como a ordem de inicialização (*boot*) e a data e hora do sistema, tratada em detalhe na Seção 9.3.

9.1.1 Componentes vitais ao POST

Para que o POST seja executado com sucesso, quatro submódulos precisam estar operacionais:

Componente	Motivo
Fonte de alimentação	Sem energia, nenhum programa pode ser executado.
CPU	As instruções do POST precisam ser processadas por um processador funcional.

Componente	Motivo
Memória RAM	Executar um programa exige carregar suas instruções na memória primária.
Placa-mãe (incluindo o BIOS)	Promove a interconexão entre fonte, CPU e RAM, e é onde o próprio programa POST reside.

9.1.2 Componentes que não afetam o POST

Diversos componentes, apesar de relevantes ao uso pleno do computador, **não** interferem no resultado do POST:

- **Memória secundária** (HD, SSD, unidades ópticas): sua ausência não impede o POST, pois ela não é necessária à execução do próprio programa de autoteste.
- **Periféricos** (teclado, mouse, caixa de som, conexão de rede): são dispositivos de entrada e saída (E/S) auxiliares, sem relação causal com o POST — da mesma forma que um smartphone liga mesmo sem sinal de rede.
- **Painel frontal** (tratado em detalhe no Capítulo 8, §8.3): dispensável ao POST, embora seja o meio usual de acionar a energização do sistema.
- **Bateria CMOS** (tratada na Seção 9.2): não impede o POST, mas afeta a retenção de configurações.

9.1.3 Sinalização sonora (beep codes)

Quando a falha ocorre antes que qualquer saída de vídeo seja possível — por exemplo, na ausência de memória RAM —, a placa-mãe não tem como exibir uma mensagem de erro na tela. Nesses casos, um **alto-falante** (*speaker*) interno à placa-mãe emite sinais sonoros (bipes) para comunicar o resultado do autoteste ao técnico: um padrão de bipes indica sucesso, e outros padrões indicam falhas específicas. Esse mecanismo permite diagnosticar um problema mesmo quando o monitor ainda não foi inicializado ou está ausente.

9.1.4 Metodologia de diagnóstico

A ausência de POST indica, necessariamente, uma falha em um dos quatro componentes vitais listados na Seção 9.1.1 — nunca em memória secundária, periféricos ou rede. O procedimento de diagnóstico sistemático (aprofundado no Capítulo 12) parte sempre do ponto mais externo do sistema — a tomada elétrica — e avança progressivamente em direção aos componentes internos.

Figura pendente

fluxo energização → POST → carregamento do sistema operacional (ou instalador), com a fronteira hardware/software destacada

9.2 A bateria CMOS e a retenção de configurações

A placa-mãe mantém, em uma pequena memória volátil (a memória **CMOS**), as configurações do *setup*: data e hora do sistema, ordem de inicialização (*boot*) e demais parâmetros de firmware. Essa memória é alimentada por uma **bateria** dedicada, independente da fonte de alimentação principal, o que permite que as configurações persistam mesmo com o computador desligado da tomada.

A ausência ou descarga dessa bateria não impede o POST, mas faz com que as configurações do setup sejam perdidas a cada desligamento — por exemplo, a data e a hora voltam a um valor padrão, exigindo reconfiguração a cada inicialização.

Exemplo. Do ponto de vista de programação, o comportamento pode ser descrito por uma variável de controle (uma *flag*) que indica se a data e a hora já foram configuradas. Sem a bateria, essa flag não é preservada entre desligamentos: a cada nova inicialização, o sistema encontra a flag "não configurada" e sinaliza a ausência de data e hora válidas.

Figura pendente

foto da bateria CMOS (formato moeda) instalada na placa-mãe, com o soquete de encaixe visível

9.3 Setup: configuração de firmware

O **Setup** é o programa, parte do mesmo firmware BIOS/UEFI do POST, que permite alterar as configurações de hardware do computador — frequência do processador e da memória (overclock), habilitação ou desabilitação de funcionalidades, data e hora do sistema, e a **ordem de inicialização** (*boot order*), entre outras. É justamente essa memória de configurações que a bateria CMOS (Seção 9.2) mantém energizada entre desligamentos.

O Capítulo 6, §6.6.2, trata o Setup em detalhe do ponto de vista do procedimento de inicialização — como acessá-lo, como ele se relaciona com o boot menu, e sua relação com a segurança do acesso físico a uma máquina.

9.4 Síntese do capítulo

Este capítulo revisitou, sob a ótica dos componentes de hardware envolvidos, os três elementos do firmware que antecedem o carregamento do sistema operacional: o POST — os quatro componentes vitais à sua execução, os beep codes que sinalizam falha antes que haja vídeo disponível, e a metodologia de diagnóstico sistemático que sua ausência aciona —, a bateria CMOS que retém as configurações de firmware entre desligamentos, e o Setup que permite alterá-las. Esses fundamentos sustentam tanto o diagnóstico de hardware tratado no Capítulo 8 (o jumper Clear CMOS, por exemplo, existe justamente para contornar a bateria descrita aqui) quanto o procedimento de boot detalhado no Capítulo 6.

10 Barramentos, Entrada e Saída e Periféricos

Neste capítulo você vai estudar como a placa-mãe organiza a comunicação entre processador, memória, armazenamento e periféricos: a hierarquia de barramentos, da arquitetura clássica de ponte norte/sul à integração progressiva dessas funções dentro do próprio processador, os protocolos SATA, PCIe e NVMe que materializam essa comunicação hoje, e os princípios de entrada e saída que regem a comunicação com teclado, mouse e demais periféricos.

10.1 Barramentos: do PCI ao PCIe

10.1.1 O que é um barramento

Um **barramento** (*bus*) é um conjunto de linhas condutoras compartilhadas por meio das quais múltiplos componentes de um computador trocam dados. Um barramento é fisicamente organizado em três grupos de linhas:

- **Linhas de dados** — transportam a informação propriamente dita.
- **Linhas de endereço** — indicam a origem ou o destino daquela informação (qual posição de memória, qual dispositivo).
- **Linhas de controle** — sincronizam a operação, indicando, por exemplo, se a operação em curso é de leitura ou de escrita.

Como diversos dispositivos compartilham o mesmo conjunto de linhas, é necessário um **controlador de barramento**, responsável por arbitrar o acesso: decidir, a cada instante, qual dispositivo tem permissão para transmitir. Esse compartilhamento tem um custo estrutural: quanto maior o número de dispositivos conectados a um mesmo barramento, maior tende a ser seu comprimento físico, maior o atraso de propagação do sinal ao longo dele, e maior o tempo médio que cada dispositivo espera até obter o controle do barramento. A solução histórica para esse gargalo foi organizar o computador numa **hierarquia de barramentos** — vários barramentos menores e especializados, em vez de um único barramento universal —, o assunto do restante desta seção.

Nota conceitual. Essa forma de comunicação, na qual várias linhas transportam bits simultaneamente em paralelo, é chamada de **comunicação paralela** e se opõe à **comunicação serial**, na qual os bits trafegam

um após o outro por uma única linha (ou par de linhas), a uma frequência muito mais alta. Contra a intuição inicial, um link serial moderno normalmente transporta mais dados por segundo do que um barramento paralelo equivalente — a alta frequência de operação de uma única linha bem projetada compensa, e supera, a vantagem teórica de transmitir vários bits "ao mesmo tempo" em paralelo, que sofre mais com interferência entre linhas adjacentes (*crosstalk*) à medida que a frequência sobe. É por essa razão que a evolução dos barramentos de expansão, tratada na Seção 10.1.4, caminhou do paralelo (ISA, PCI) para o serial (PCI Express) — e a mesma lógica explica a evolução do armazenamento de IDE (paralelo) para SATA (serial), na Seção 10.1.5.

Figura pendente

comparação esquemática entre um barramento paralelo (várias linhas lado a lado) e uma conexão serial (uma linha, alta frequência)

10.1.2 Ponte norte e ponte sul: a arquitetura clássica do chipset

O **chipset** — já mencionado no Capítulo 11 (§11.5) como o "conjunto de chips" que interliga os controladores da placa-mãe — foi, por muitos anos, fisicamente dividido em dois chips distintos, cada um responsável por uma metade da hierarquia de barramentos do computador.

- **Ponte norte** (*northbridge*, também chamada *IO hub*): conectada diretamente ao processador por um barramento de altíssima velocidade (chamado **QPI** — *QuickPath Interconnect* — pela Intel, e **HyperTransport** pela AMD, historicamente) [1], a ponte norte intermediava o acesso à memória RAM e à placa de vídeo — os dois componentes que mais exigem largura de banda e menor latência possível em relação ao processador.
- **Ponte sul** (*southbridge*), hoje frequentemente chamada **PCH** (*Platform Controller Hub*, nomenclatura Intel) ou **FCH** (*Fusion Controller Hub*, nomenclatura AMD): conectada à ponte norte (nunca diretamente ao processador), reunia os controladores de dispositivos que toleram maior latência — portas USB, SATA (Seção 10.1.5), áudio, rede, o chip de BIOS/UEFI (Capítulo 6, §6.6) e os demais slots PCI/PCIe de expansão.

10.1.3 A migração para dentro do processador

A arquitetura de duas pontes começou a ser desmontada à medida que os processadores modernos passaram a incorporar, dentro do próprio encapsulamento da CPU, funções que antes pertenciam à ponte norte: o **controlador de memória** (Capítulo 5 trata a RAM em profundidade) e, em processadores mais recentes, um conjunto próprio de **pistas PCIe** (Seção

10.1.4, adiante) dedicadas à placa de vídeo e ao armazenamento NVMe (Seção 10.1.5).

O resultado é que a "ponte norte" propriamente dita desapareceu como chip separado na maioria dos desktops modernos: o processador se conecta diretamente à RAM e à GPU, e o que resta da comunicação com o restante da placa-mãe passa por um único link de alta velocidade até a ponte sul — chamado **DMI** (*Direct Media Interface*) pela Intel e **UMI** (*Unified Media Interface*) pela AMD [2]. Esse link concentra hoje o tráfego de tudo que ainda depende da ponte sul: USB, SATA, PCIe de menor prioridade, áudio, rede — e é, ele mesmo, um ponto de atenção em especificação de hardware, porque todo esse tráfego compartilha a largura de banda de um único link, ainda que cada dispositivo individual pareça ter sua própria conexão dedicada.

Nota prática. Essa reorganização explica por que a especificação de um processador (Capítulo 4) hoje frequentemente informa "quantas pistas PCIe" ele oferece diretamente — um dado que, antes da migração do controlador para dentro da CPU, seria uma característica do chipset, não do processador.

Figura pendente

dois diagramas lado a lado — arquitetura clássica (CPU → ponte norte → ponte sul) e arquitetura atual (CPU com controlador de memória e PCIe integrados, ligada à ponte sul por DMI/UMI)

10.1.4 Slots de expansão: de ISA a PCI Express

A tabela a seguir situa a evolução dos barramentos de expansão — as conexões da placa-mãe às quais placas adicionais (de vídeo, de som, de rede, entre outras) são fisicamente conectadas [3]:

Padrão	Tipo de comunicação	Situação atual
ISA (*Industry Standard Architecture*)	Paralela	Obsoleto, presente apenas em computadores muito antigos
AGP (*Accelerated Graphics Port*)	Paralela, dedicada a vídeo	Obsoleto, substituído pelo PCIe
PCI (*Peripheral Component Interconnect*)	Paralela, barramento compartilhado	Praticamente obsoleto em placas novas
PCI Express (PCIe)	Serial, ponto a ponto	Padrão atual

O **PCI Express** rompe com a lógica de barramento compartilhado das gerações anteriores: em vez de vários dispositivos disputando o mesmo conjunto de linhas (como discutido na Seção 10.1.1), cada slot PCIe estabelece uma conexão **ponto a ponto** exclusiva com o controlador — não é,

tecnicamente, um “barramento” no sentido estrito da Seção 10.1.1, embora o uso corrente do mercado continue chamando-o assim.

Essa conexão é organizada em **pistas** (*lanes*), cada uma constituída por um par de linhas seriais full-duplex (transmitindo e recebendo simultaneamente). Um slot PCIe pode agrupar diferentes quantidades de pistas — identificadas como **x1**, **x4**, **x8** e **x16** —, e quanto mais pistas agrupadas, maior a largura de banda disponível para o dispositivo conectado. Uma placa de vídeo moderna, por exigir grande volume de dados, normalmente ocupa um slot x16; um SSD NVMe (Seção 10.1.5) costuma usar o equivalente a quatro pistas (x4).

Compatibilidade física. Um slot PCIe de maior tamanho físico (por exemplo, x16) aceita, por projeto, uma placa de tamanho físico menor (x1, x4, x8) encaixada nele — a placa menor simplesmente não utiliza todas as pistas disponíveis no slot. O inverso não é possível sem um slot aberto na extremidade (um recurso presente em algumas placas-mãe): uma placa fisicamente x16 não encaixa, por padrão, num slot x1.

Cada geração do padrão PCIe (identificada por um número de versão — PCIe 3.0, 4.0, 5.0 e assim sucessivamente) dobra, em relação à geração anterior, a largura de banda disponível por pista [4], mantendo o mesmo princípio físico de conexão — um padrão de evolução comparável ao da família DDR de memória (Capítulo 5, §5.4).

Figura pendente

foto de uma placa-mãe evidenciando slots PCIe de tamanhos diferentes (x16, x4, x1) lado a lado

10.1.5 SATA e NVMe: interfaces de armazenamento

O Capítulo 8, §8.1.1, já introduziu, brevemente, as interfaces **SATA** e **NVMe** ao tratar da conexão física da memória secundária. Esta seção aprofunda essa distinção agora que o conceito de PCIe foi apresentado.

SATA (*Serial ATA*) é o sucessor serial do antigo padrão **IDE** (*Integrated Drive Electronics*, também chamado **PATA**, *Parallel ATA*) — mais um caso, como discutido na Seção 10.1.1, da migração geral de interfaces paralelas para seriais. O SATA usa seu próprio controlador (parte da ponte sul, Seção 10.1.2) e seu próprio protocolo de comunicação, projetado nos anos 2000 [5] tendo o HD mecânico (Capítulo 5, §5.10) como dispositivo de referência — um dispositivo cujo gargalo real de desempenho está na mecânica do prato girante e do braço atuador, não na interface elétrica em si.

NVMe (*Non-Volatile Memory Express*) é um protocolo de comunicação desenvolvido especificamente para memória flash (Capítulo 5, §5.12), projetado para eliminar exatamente essa limitação histórica do SATA. Em vez de usar o controlador da ponte sul e um protocolo pensado para discos mecânicos, um dispositivo NVMe se conecta **diretamente às pistas PCIe** (Seção 10.1.3) — com frequência às pistas oferecidas diretamente pelo processador, e não pela ponte sul —, dispensando a camada de tradução SATA/AHCI

e aproveitando a largura de banda muito maior do PCIe. Fisicamente, um dispositivo NVMe de consumo típico se conecta a um slot **M.2** na placa-mãe — um conector compacto que dispensa os cabos de dados e energia exigidos por um dispositivo SATA (Capítulo 8, §8.1.1) — embora nem todo slot M.2 seja necessariamente NVMe: alguns slots M.2 mais antigos transportam o protocolo SATA sobre o mesmo conector físico, uma fonte comum de confusão na hora de especificar um SSD compatível.

Nota de desempenho. A diferença de velocidade entre SATA e NVMe não é sutil: um SSD SATA está limitado à taxa de transferência máxima da interface SATA (da ordem de 600 MB/s) [6], enquanto um SSD NVMe, por usar múltiplas pistas PCIe diretamente, alcança taxas de vários gigabytes por segundo — uma ordem de grandeza acima. Essa diferença só se torna relevante, na prática, para cargas de trabalho que de fato saturam a interface (cópia de grandes volumes de arquivo, carregamento de jogos com texturas pesadas); para o uso cotidiano de um computador, a diferença perceptível entre os dois costuma ser pequena.

10.2 Entrada, saída e periféricos

10.2.1 Classificação e o problema da heterogeneidade

Um dispositivo de entrada e saída (E/S, ou *I/O*) é classificado, do ponto de vista do sistema, pelo sentido do fluxo de dados em relação ao computador:

- **Dispositivos de entrada** — enviam dados para o computador (teclado, mouse, *touchpad*, microfone, câmera, *scanner*).
- **Dispositivos de saída** — recebem dados do computador (monitor, caixas de som, impressora).
- **Dispositivos híbridos** — operam nos dois sentidos (tela sensível ao toque, impressora multifuncional com *scanner*, headset com microfone).

O desafio central de projetar a comunicação entre a CPU e esse universo de periféricos é a **heterogeneidade**: cada dispositivo tem conexão física própria, sentido de conexão próprio, velocidade própria, requisitos próprios (alguns toleram atraso, outros não) e é produzido por um fabricante diferente, sem garantia alguma de padronização espontânea entre eles. A solução estrutural para esse problema é a mesma adotada em outras camadas já estudadas neste livro: um **hardware controlador dedicado** para cada família de periférico, expondo à CPU uma **interface** simples e uniforme — poupando o processador de conhecer os detalhes elétricos específicos de cada fabricante, no mesmo espírito da abstração que o sistema operacional oferece ao software (Capítulo 6, §6.1.1).

10.2.2 Estudo de caso: CPU e memória secundária

A comunicação entre a CPU e um HD ou SSD (Capítulo 5) passa por uma controladora que desempenha quatro funções:

1. **Conexão física** — o conector elétrico propriamente dito (SATA ou NVMe, Seção 10.1.5).
2. **Conversão de protocolo de comunicação** — traduz entre o protocolo interno do computador e o protocolo específico daquele padrão de armazenamento.
3. **Conversão de tipos de dado** — organiza os dados na granularidade que o barramento espera (blocos, setores — Capítulo 5, §5.10).
4. **Buffer** — uma pequena área de armazenamento temporário que absorve diferenças momentâneas de velocidade entre a CPU (rápida) e o dispositivo de armazenamento (mais lento), evitando que a CPU precise esperar ociosa por cada byte individual.

10.2.3 Estudo de caso: CPU, teclado e mouse

A comunicação entre a CPU e dispositivos de entrada simples como teclado e mouse é resolvida majoritariamente em **software**, não em hardware dedicado de alta complexidade. Antigamente, essa comunicação básica de baixo nível era resolvida diretamente por rotinas do **BIOS** (Capítulo 6, §6.6); atualmente, a pilha é mais sofisticada e passa por três camadas, de baixo para cima: o **chipset** (Seção 10.1.2), que recebe o sinal elétrico bruto do dispositivo; o **driver** (Capítulo 7, §7.6), fornecido pelo sistema operacional ou pelo fabricante, que traduz esse sinal para um formato que o sistema entende; e o **gerenciador de dispositivos** do sistema operacional, que expõe esse dado já tratado aos programas em execução.

10.2.4 Modos de transferência de dados

Um requisito comum a qualquer transferência de E/S é comunicar três elementos: um **endereço** (para onde vai, ou de onde vem, o dado), **comandos** (o que fazer com ele) e os **dados** propriamente ditos. Existem três estratégias para a CPU realizar essa comunicação [7]:

- **Comunicação programada** — a própria CPU, executando uma rotina de software, é responsável por mover cada dado entre o periférico e a memória. Duas variantes existem: **espera ocupada** (a CPU fica presa num laço, checando repetidamente se o dispositivo está pronto, sem fazer mais nada nesse intervalo) e ***polling*** (a CPU verifica periodicamente, mas intercalando outras tarefas entre uma verificação e outra). Em ambas, a CPU desperdiça parte de sua capacidade de processamento apenas esperando ou verificando.
- **Comunicação por interrupção** — em vez de a CPU perguntar repetidamente "já terminou?", o próprio hardware do dispositivo **avisa** a

CPU (por meio de um sinal elétrico chamado interrupção) no exato instante em que há um dado pronto para transferência. A CPU fica livre para executar outras tarefas entre um aviso e outro, interrompendo o que está fazendo apenas quando o periférico efetivamente precisa de atenção.

- **DMA** (*Direct Memory Access*, acesso direto à memória) — para transferências de grande volume (como copiar um arquivo inteiro do SSD para a RAM), mesmo a comunicação por interrupção geraria overhead excessivo se a CPU precisasse mediar byte a byte. O DMA delega essa cópia a um controlador dedicado, que move os dados diretamente entre o dispositivo e a memória RAM sem ocupar o processador durante a transferência — a CPU apenas inicia a operação e é avisada, por uma única interrupção, quando ela termina por completo.

10.2.5 Estudo de caso: USB

O **USB** (*Universal Serial Bus*, barramento serial universal) ilustra, num único padrão amplamente conhecido, como diferentes tipos de periférico exigem diferentes garantias de entrega de dados. A especificação USB define quatro tipos de transferência [8]:

Tipo de transferência	Uso típico	Garantia
Controle	Inicialização do dispositivo ao ser conectado; comandos administrativos	Entrega garantida
Em massa (*bulk*)	Grandes volumes de dados sem urgência de tempo (impressoras, *scanners*, pendrives)	Entrega garantida, tempo não garantido
Interrupção	Pequenos volumes de dados sensíveis a tempo (teclado, mouse)	Entrega garantida, pequenos atrasos tolerados
Isócrona	Transmissão em tempo real (áudio, vídeo)	Ritmo de entrega garantido, mas dados individuais podem ser perdidos

Note que essa tabela reflete diretamente o princípio de heterogeneidade apresentado na Seção 10.2.1: um teclado não pode tolerar que uma tecla pressionada demore segundos para ser reconhecida (por isso usa transferência de interrupção, com prioridade sobre atraso), enquanto um fluxo de áudio tolera perder uma amostra ocasional, mas não tolera que o ritmo de entrega varie (por isso usa transferência isócrona) — e uma impressora tolera esperar, desde que nenhum byte do documento se perca (por isso usa transferência em massa). O mesmo protocolo físico (USB) acomoda,

portanto, contratos de entrega completamente diferentes, escolhidos pelo fabricante do periférico conforme a natureza do dado transmitido.

Figura pendente

diagrama de um cabo USB com quatro balões apontando para exemplos de periférico — teclado (interrupção), pendrive (massa), headset (isócrona), dispositivo genérico sendo conectado (controle)

10.2.6 Especificação de periféricos

Assim como processador, memória e armazenamento têm critérios objetivos de especificação (Capítulo 5 e Capítulo 8, §8.4.3), cada família de periférico tem seu próprio conjunto de critérios relevantes — geralmente ligados à natureza do dado que aquele dispositivo transmite (Seção 10.2.1):

Periférico	Critério de especificação mais relevante
Monitor	Resolução e taxa de atualização (Hz) — quantos quadros por segundo o painel exibe
Teclado	Tipo de acionamento (membrana ou mecânico) e disposição de teclas
Mouse	Resolução do sensor (DPI) e taxa de resposta
Impressora	Tecnologia de impressão (jato de tinta ou laser) e custo por página impressa
Caixas de som / headset	Resposta de frequência e potência (o Apêndice A trata potência elétrica em profundidade)

Nota prática. Como qualquer especificação de hardware (Capítulo 8, §8.4.3), não existe “o melhor” periférico em termos absolutos — um teclado mecânico de alto custo é desperdício de orçamento para um usuário de escritório, da mesma forma que uma impressora a laser monocromática não atende quem precisa imprimir fotos coloridas. O critério de especificação de periféricos segue o mesmo princípio de adequação à finalidade já estabelecido para os demais componentes.

10.3 Síntese do capítulo

Este capítulo situou a placa-mãe como via de comunicação: a hierarquia de barramentos que liga processador, memória, armazenamento e perifé-

ricos, da arquitetura clássica de ponte norte/sul à integração progressiva dessas funções dentro do próprio processador, e os protocolos SATA, PCIe e NVMe que materializam essa comunicação hoje. A segunda parte do capítulo tratou dos princípios de entrada e saída que regem a comunicação com teclado, mouse e demais periféricos — a heterogeneidade que todo controlador de E/S precisa resolver, os três modos de transferência de dados (programada, por interrupção, DMA) e o estudo de caso do USB, que ilustra como um único protocolo físico acomoda contratos de entrega completamente diferentes.

10.4 Referências

1. STALLINGS, William. *Arquitetura e Organização de Computadores*. 10. ed. São Paulo: Pearson Education do Brasil, 2018 (seção sobre QPI); MONTEIRO, Mario A. *Introdução à Organização de Computadores*. 5. ed. Rio de Janeiro: LTC (seção D.3.4.2, Tecnologia HyperTransport).
2. WIKIPEDIA. “Direct Media Interface.” Disponível em: `https://en.wikipedia.org/wiki/Direct_Media_Interface`; INTEL. “What Is the Direct Media Interface (DMI) of Intel® Processors?” Disponível em: `https://www.intel.com/content/www/us/en/support/articles/000094185/processors.html`; WIKIPEDIA. “Unified Media Interface.” Disponível em: `https://en.wikipedia.org/wiki/Unified_Media_Interface`.
3. PCI-SIG. Especificações oficiais PCI/PCI Express. Disponível em: `https://pcisig.com`.
4. PCI-SIG. Especificações oficiais PCI Express (PCIe 3.0/4.0/5.0). Disponível em: `https://pcisig.com`.
5. SEAGATE. “Serial ATA: High Speed Serialized AT Attachment, Revision 1.0, 29-August-2001.” Disponível em: `https://www.seagate.com/support/disc/manuals/sata/sata_im.pdf`.
6. SATA-IO. Especificação SATA. Disponível em: `https://www.sata-io.org`.
7. STALLINGS, William. *Arquitetura e Organização de Computadores*. 10. ed. São Paulo: Pearson Education do Brasil, 2018 (Capítulo 7, E/S programada, por interrupção e DMA).
8. USB IMPLEMENTERS FORUM. Especificação USB. Disponível em: `https://www.usb.org`; MONTEIRO, Mario A. *Introdução à Organização de Computadores*. 5. ed. Rio de Janeiro: LTC (seção D.3.4.1).

11 Fonte de Alimentação: ATX e Dimensionamento

Neste capítulo você vai estudar a diferença entre fontes lineares e fontes chaveadas, a arquitetura da fonte ATX e seus sinais de controle, os critérios de eficiência energética e fator de potência usados para classificar fontes comercialmente, e a metodologia de dimensionamento e diagnóstico de uma fonte de computador. As grandezas elétricas fundamentais e os princípios de aterramento sobre os quais este capítulo se apoia estão reunidos no Apêndice A.

11.1 Fontes chaveadas e modulação por largura de pulso (PWM)

A imensa maioria das fontes de alimentação usadas em computadores atuais — de smartphones a servidores — são **fontes chaveadas**. Em vez de reduzir a tensão de forma linear e contínua, esse tipo de fonte liga e desliga um circuito em altíssima frequência, controlando a fração do tempo em que o circuito permanece ligado. Essa técnica é chamada de **PWM** (*Pulse Width Modulation*, ou modulação por largura de pulso).

O princípio do PWM é simples: uma onda de tensão fixa (por exemplo, 10 V de amplitude) é ligada e desligada periodicamente. Se essa onda permanece "ligada" durante 50% de cada período, uma carga conectada a esse circuito "enxerga" uma tensão efetiva de 5 V — a metade da amplitude máxima. Se a onda permanece ligada apenas 23% do tempo, a carga enxerga aproximadamente 2,3 V.

Exemplo. Para gerar uma saída de 2,3 V a partir de uma onda de amplitude máxima de 10 V, basta manter o sinal em nível alto durante 23% de cada período (*duty cycle* de 23%) e em nível baixo nos 77% restantes. Circuitos com comportamento predominantemente passivo — como motores e LEDs — respondem a essa alternância rápida como se a tensão fosse, de fato, constante e igual à fração correspondente da amplitude máxima.

A vantagem central do chaveamento sobre outras formas de redução de tensão (como um simples divisor resistivo) é a **eficiência**: um divisor de tensão dissipa parte da energia em forma de calor através dos resistores, enquanto o chaveamento — ligando e desligando o circuito, em vez de dissipar energia continuamente — perde muito menos energia nessa conversão. É por isso que as fontes de computador atuais são chamadas de **fontes**

chaveadas: internamente, elas empregam circuitos de chaveamento em alta frequência (controlados por PWM) para produzir as diversas tensões contínuas exigidas pelos componentes, de forma muito mais compacta e eficiente do que uma fonte linear equivalente (Apêndice A, §A.6).

Figura pendente

forma de onda PWM com diferentes duty cycles (10%, 50%, 90%) e a tensão média efetiva resultante em cada caso

---

11.2 A fonte ATX

11.2.1 Padrão ATX e o conector principal

A **fonte ATX** é o padrão universal de fonte de alimentação para computadores desktop. Diferentemente de fontes com uma chave seletora de tensão (110 V/220 V) — cuja posição incorreta pode queimar o equipamento se ligado na tensão errada —, boa parte das fontes modernas de melhor qualidade é do tipo **full range** (ou *auto switch*): elas aceitam qualquer tensão alternada de entrada dentro de uma faixa ampla, tipicamente de 100 V a 240 V, ajustando-se automaticamente sem necessidade de seleção manual.

Atenção. Em fontes com chave seletora de tensão, a posição da chave deve sempre ser conferida antes de conectar o equipamento à tomada. Ligar uma fonte selecionada para 115 V em uma tomada de 220 V resulta, tipicamente, em dano imediato e irreversível ao equipamento — o mesmo princípio descrito no Apêndice A, §A.1.1.

A fonte ATX se conecta à placa-mãe por meio de um conector principal de **20 ou 24 pinos** [1] (dependendo do modelo da placa-mãe), fornecendo simultaneamente diversas tensões contínuas distintas, identificadas por um padrão de cores nos fios [2]:

Cor do fio	Tensão / sinal	Função
Laranja	+3,3 V	Alimentação de baixa tensão (memórias, lógica moderna)
Vermelho	+5 V	Alimentação de componentes diversos
Amarelo	+12 V	Alimentação de motores (ventoinhas, discos) e do processador

Cor do fio	Tensão / sinal	Função
Roxo	+5 V (auxiliar/*standby*)	Alimentação permanente, mesmo com o computador desligado
Preto	COM (referência/terra)	Referência de potencial (0 V) para todas as demais tensões
Verde	PS_ON (*Power Supply On*)	Sinal de comando da placa-mãe para a fonte
Cinza	Power OK	Sinal de confirmação da fonte para a placa-mãe

Além dos fios listados, a fonte também disponibiliza tensões negativas (-12 V e -5 V) para funções específicas de compatibilidade histórica. Um segundo conector, tipicamente de 4 ou 8 pinos, fornece alimentação dedicada ao processador — algumas placas-mãe mais recentes exigem inclusive **dois** conectores de 8 pinos para essa finalidade, tornando necessário verificar a compatibilidade entre placa-mãe e fonte antes da montagem.

O padrão ATX substituiu o padrão anterior, chamado **AT**, cuja principal limitação era a ausência de comunicação eletrônica entre a placa-mãe e a fonte para o desligamento: em computadores com fonte AT, o comando de desligar (pelo sistema operacional) apenas exibia uma mensagem informando que o computador já podia ser desligado com segurança, mas o desligamento físico ainda dependia de o usuário acionar uma chave mecânica. O padrão ATX introduziu o desligamento controlado por software, que se tornou padrão em todos os computadores modernos [3].

Figura pendente

conector ATX de 24 pinos com o código de cores dos fios sobreposto

11.2.2 Sinais de controle: PS_ON e Power OK

O procedimento de inicialização de um desktop ilustra como a fonte, o botão de energia e a placa-mãe se comunicam eletricamente.

O botão de *power* do painel frontal do gabinete é, fisicamente, apenas um par de fios que, quando o botão é pressionado, se conectam momentaneamente — fechando um pequeno curto-circuito entre dois pinos específicos no conector do painel frontal da placa-mãe. Esse contato momentâneo é interpretado por um circuito da placa-mãe como um comando de ligar.

Ao reconhecer esse comando, a placa-mãe sinaliza à fonte que deve energizar seus circuitos de saída, unindo eletricamente o pino verde (**PS_ON**) ao pino preto (**COM**) — ou seja, colocando o pino de comando no mesmo nível de potencial que a referência de terra. Quando a fonte detecta essa

condição, ela liga seus circuitos internos e passa a fornecer as tensões de saída (3,3 V, 5 V, 12 V etc.).

Esse acionamento não é instantâneo: existe um pequeno intervalo de tempo entre o comando de ligar e o momento em que todas as tensões de saída atingem seus valores nominais e estáveis. Ao final desse intervalo, a fonte sinaliza à placa-mãe, através do pino **Power OK**, que as tensões estão estabilizadas e prontas para uso. Somente após receber esse sinal a placa-mãe libera a energização dos demais componentes e inicia o procedimento de *boot*, chamando o primeiro programa executado pela placa-mãe — o POST.

Exemplo (metodologia de diagnóstico). Esse mecanismo fornece um procedimento sistemático para isolar problemas de energização em um computador que não liga:

1. Testar o botão físico do painel frontal, substituindo-o por um curto manual (por exemplo, com uma chave de fenda) diretamente nos pinos correspondentes da placa-mãe. Se o computador ligar dessa forma, o problema estava no botão ou em sua fiação — não na fonte nem na placa-mãe.
2. Se o curto manual no botão não resolver, desconectar a fonte da placa-mãe e testar a fonte isoladamente, unindo o pino verde (PS_ON) a qualquer pino preto (COM) no próprio conector da fonte. Se a fonte ligar (ventoinha gira, LEDs acendem), o problema está na placa-mãe. Se a fonte não ligar, o problema está na fonte.

Esse procedimento exemplifica, no contexto elétrico, a metodologia geral de diagnóstico por isolamento de módulos já apresentada no Capítulo 12 (§12.4): identificar o submódulo defeituoso testando cada elo da cadeia (botão → placa-mãe → fonte) separadamente.

Atenção. A fonte ligar durante esse teste — girando a ventoinha e produzindo as tensões nominais — indica apenas que ela consegue entregar tensão. Não garante que ela consegue entregar a **potência** total sob carga real: uma fonte deteriorada pode fornecer tensões corretas em vazio e ainda assim falhar ao alimentar componentes de alto consumo, como uma placa de vídeo. O teste do jumper é uma condição necessária, mas não suficiente, para validar uma fonte.

Figura pendente

fluxo botão → pinos do painel frontal → sinal PS_ON → fonte → sinal Power OK → placa-mãe, com indicação dos pontos de teste para diagnóstico

11.2.3 VRM: uma segunda fonte na placa-mãe

As tensões fornecidas pela fonte ATX (3,3 V, 5 V, 12 V) não correspondem, na maioria dos casos, às tensões efetivamente exigidas pelo processador

e por outros componentes modernos, que frequentemente operam em tensões muito mais baixas e específicas. Por esse motivo, a própria placa-mãe contém uma "segunda fonte" interna: o **VRM** (*Voltage Regulator Module*), responsável por converter as tensões recebidas da fonte ATX nas tensões finais exigidas por cada componente. De forma equivalente, alguns processadores contam com um módulo de conversão próprio, às vezes chamado de PPM (*Processor Power Module*) [4].

A razão histórica para manter essa conversão adicional na placa-mãe — e não na fonte — é a compatibilidade: o padrão de saída da fonte ATX permaneceu estável ao longo de gerações de processadores e memórias, permitindo que um usuário reaproveite a mesma fonte em upgrades de placa-mãe, processador ou memória, desde que as novas exigências específicas de tensão sejam resolvidas pela conversão adicional na própria placa-mãe.

11.3 Eficiência energética e fator de potência

11.3.1 Selos de eficiência (80 Plus e ETA-Lambda)

Toda conversão de energia envolve perdas: parte da energia retirada da tomada é dissipada em forma de calor durante as sucessivas transformações (AC–AC, AC–DC, DC–DC) realizadas dentro da fonte. A **eficiência** de uma fonte é definida como a razão entre a potência efetivamente entregue aos componentes do computador e a potência total consumida na tomada.

Órgãos de certificação independentes, como o **80 Plus**, realizam ensaios de bancada e atribuem selos de eficiência às fontes comerciais. Mais recentemente, a Cybenetics Labs passou a manter dois programas de certificação distintos: **ETA** (eficiência energética) e **LAMBDA** (ruído acústico produzido pela fonte sob diferentes cargas) — são dois selos separados, avaliados independentemente; uma fonte pode ter um sem o outro [5]. Esses ensaios avaliam a eficiência em três níveis de carga, expressos como percentual da potência nominal da fonte:

- **Carga leve** — cerca de 20% da potência nominal.
- **Carga típica** — cerca de 50% da potência nominal.
- **Carga máxima** — 100% da potência nominal.

O selo 80 Plus é concedido em diferentes categorias (Standard, Bronze, Silver, Gold, Platinum, Titanium, em ordem crescente de exigência), cada uma exigindo um patamar mínimo de eficiência nas três cargas. Por exemplo, o selo **Standard** exige 80% de eficiência nas três cargas (80/80/80); o selo **Platinum** exige patamares mais altos, como 90% em carga leve, 92% em carga típica e 89% em carga máxima [6].

Exemplo. Uma fonte é tipicamente projetada para atingir sua eficiência máxima justamente na região de carga típica (em torno de 50% da potência nominal) — motivo pelo qual o dimensionamento recomendado de uma fonte

para um computador (Seção 11.4) busca deixar o consumo real do sistema próximo dessa faixa.

Um fator adicional observado experimentalmente é que uma fonte tende a ser mais eficiente quando alimentada em 220 V do que em 110 V, para a mesma carga [7] — consequência direta do mesmo princípio de efeito Joule discutido no Apêndice A, §A.2: em uma tensão de entrada mais alta, a corrente de entrada correspondente é menor, reduzindo as perdas internas.

Selos de eficiência não devem ser confundidos com selos de potência: uma fonte de "650 W" descreve a potência que ela é capaz de fornecer, e sua eficiência (Standard, Bronze, Gold etc.) descreve a proporção dessa energia que é efetivamente aproveitada, e não perdida em calor, na conversão a partir da tomada.

11.3.2 Fator de potência e PFC

O **fator de potência** é uma segunda métrica de qualidade de uma fonte, distinta da eficiência. É definido como a razão entre a **potência ativa** (a potência que efetivamente realiza trabalho) e a **potência aparente** (a potência total que o circuito precisa "reservar" para operar, incluindo a energia armazenada temporariamente em componentes reativos, como capacitores e indutores, sem realizar trabalho útil):

$$\text{Fator de potência} = \frac{P_{\text{ativa}}}{S_{\text{aparente}}}$$

Fontes com correção de fator de potência (**PFC**, *Power Factor Correction*) empregam circuitos adicionais — passivos (bancos de capacitores e indutores) ou ativos (circuitos eletrônicos dedicados) — para reduzir essa parcela reativa e aproximar a potência aparente da potência ativa. No Brasil, consumidores residenciais não são cobrados por energia reativa (essa cobrança se aplica a consumidores do Grupo A — média/alta tensão, o que inclui grandes consumidores industriais e comerciais, não apenas industriais) [8], de forma que a presença de PFC em uma fonte doméstica não reduz a conta de energia do usuário — mas é, ainda assim, um indicador de qualidade construtiva do circuito.

Atenção. O fator de potência e a eficiência de uma fonte são métricas independentes: uma fonte com PFC ativo não é, por esse motivo isolado, necessariamente mais eficiente — a eficiência é determinada pelo ensaio de conversão de energia (Seção 11.3.1), e não pelo fator de potência.

11.4 Dimensionamento e diagnóstico de fontes

11.4.1 Trilhos de potência (rails)

Ao especificar uma fonte, o fabricante não apenas informa a potência total nominal (por exemplo, 650 W), mas também detalha, em uma tabela de especificações, a corrente máxima que cada **trilho de tensão** (*rail*) — 3,3 V, 5 V, 12 V, além dos trilhos negativos e auxiliares — é capaz de fornecer individualmente.

Exemplo. Considere uma fonte nominal de 650 W com a seguinte especificação de trilhos:

Trilho	Tensão	Corrente máxima	Potência do trilho
A	3,3 V	25 A	82,5 W
B	5 V	25 A	125 W
A + B (combinados)	—	—	150 W (máximo compartilha
C	12 V	52 A	624 W
D	12 V (auxiliar)	0,5 A	6 W
E	5 V (auxiliar)	2,5 A	12,5 W

Somando a potência máxima teórica de todos os trilhos individualmente ($150 + 624 + 6 + 12{,}5 = 792{,}5$ W), o resultado ultrapassa os 650 W nominais da fonte. Isso não significa que o fabricante esteja anunciando uma especificação incorreta: significa que os 650 W nominais são **compartilhados** entre os trilhos, e que não é possível extrair a corrente máxima de todos os trilhos simultaneamente. Na prática, o trilho de 12 V (tipicamente o que alimenta processador e placa de vídeo, os dois maiores consumidores de um desktop) concentra a maior reserva de potência da fonte, e os demais trilhos (3,3 V e 5 V, tipicamente usados por memória, armazenamento e portas USB) compartilham uma parcela menor do total.

Figura pendente

tabela de trilhos de uma fonte real com tensões e correntes máximas por rail

11.4.2 Metodologia de dimensionamento

Ao dimensionar uma fonte para um computador desktop, os dois maiores consumidores de energia — e, portanto, os fatores decisivos no dimensionamento — são o **processador** e a **placa de vídeo**: uma GPU de alto desempenho pode consumir, sozinha, uma potência muito superior à soma de todos os demais componentes do sistema. Componentes adicionais (memórias, unidades de armazenamento, ventoinhas) contribuem com potências

individuais menores, mas devem ser somados quando existem em grande quantidade — como em servidores com múltiplos discos.

Fabricantes de fontes disponibilizam calculadoras de dimensionamento *on-line* que, a partir da lista de componentes do sistema (processador, GPU, memória, armazenamento, ventoinhas), estimam o consumo total e recomendam uma potência de fonte adequada. Essas calculadoras tipicamente recomendam uma margem confortável acima do consumo estimado do sistema — não porque o sistema vá efetivamente consumir aquele valor, mas para que o consumo real do sistema recaia na faixa de carga típica (em torno de 50% da potência nominal), onde a fonte opera com eficiência máxima (Seção 11.3.1).

Atenção. Capacitores de grande capacitância no interior de uma fonte podem reter carga elétrica significativa por dias mesmo após o equipamento ser desconectado da tomada. Uma fonte nunca deve ser aberta sem os devidos cuidados de segurança, e o escopo deste curso não inclui a manutenção interna do circuito da fonte (troca de capacitores, indutores ou outros componentes) — apenas o diagnóstico que permite decidir pela substituição do módulo completo.

11.4.3 Roteiro prático de diagnóstico

Somando os conceitos apresentados neste capítulo, o diagnóstico de um computador que apresenta suspeita de falha na fonte segue, tipicamente, esta sequência:

1. **Inspeção visual** — verificar conexões soltas, a posição correta da chave seletora de tensão (em fontes não *full range*) e sinais evidentes de dano (capacitores estufados, vazamentos, queimaduras).
2. **Isolamento do ambiente de teste** — testar o computador com um monitor e uma tomada sabidamente funcionais, para que qualquer falha observada seja atribuída ao computador, e não ao ambiente de teste.
3. **Observação dos indicadores de energização** — ao ligar o computador, observar se a ventoinha da fonte gira, se a ventoinha do processador gira e se LEDs de atividade acendem.
4. **Verificação do POST** — a ausência do bipe característico do POST pode indicar um problema anterior ao teste do BIOS (fonte, botão, conexão) ou, simplesmente, um *speaker* (alto-falante interno de aviso) desconectado ou invertido em polaridade — o *speaker* é um componente polarizado (positivo e negativo) e deve ser conectado na orientação correta indicada na placa-mãe.
5. **Teste isolado da fonte**, conforme descrito na Seção 11.2.2, unindo o pino PS_ON (verde) a um pino COM (preto) diretamente no conector da fonte.
6. **Manutenção preventiva de contatos** — mau contato nos módulos de memória RAM é uma causa frequente de falha intermitente no POST (por exemplo, o computador não bipar em uma tentativa e bipar nor-

malmente na tentativa seguinte). O procedimento de limpeza consiste em remover os módulos de memória, aplicar um produto de limpeza de contatos eletrônicos (álcool isopropílico ou produto similar, altamente volátil) sobre os pontos de contato dourados, aguardar a evaporação completa e reinserir o módulo, certificando-se de que o encaixe nas travas do slot esteja completo (indicado por um clique audível em ambos os lados do slot).

Esse roteiro exemplifica, de forma concreta, a metodologia de manutenção corretiva por formulação e teste de hipóteses apresentada no Capítulo 12 (§12.4): cada etapa isola uma parte do sistema, permitindo concluir, por eliminação, em qual submódulo está o problema.

11.5 Integração da placa-mãe: chipset, painel frontal e aterramento do gabinete

Como visto nas seções anteriores, a fonte se conecta à placa-mãe em dois pontos (o conector principal e o conector de alimentação do processador). A placa-mãe, por sua vez, é responsável por distribuir essa energia — e por integrar todos os demais componentes do computador — através de um conjunto de circuitos controladores conhecido como **chipset**.

O chipset é, como o nome sugere, um *conjunto de chips* que provê as funcionalidades de interconexão da placa-mãe: controladores de portas USB, de rede Ethernet, de Wi-Fi (quando presente), de linhas de memória, entre outros. Um mesmo chipset (identificado por um código de modelo, como "B860") pode ser implementado por diferentes fabricantes de placa-mãe, cada um oferecendo uma organização física e um conjunto de recursos adicionais diferentes — explicando por que placas-mãe de fabricantes distintos, baseadas no mesmo chipset, podem ter preços significativamente diferentes: a diferença normalmente está em funcionalidades extras (como Wi-Fi integrado) e não necessariamente em qualidade superior de um fabricante sobre o outro.

Manual da placa-mãe. O manual técnico fornecido pelo fabricante de cada placa-mãe é a fonte de referência definitiva para identificar a pinagem de conectores, os requisitos de alimentação do processador (por exemplo, se são necessários um ou dois conectores de 8 pinos) e os procedimentos específicos daquele modelo, como o Clear CMOS (Capítulo 8, §8.7.3). Um técnico deve sempre consultar o manual de cada placa-mãe individualmente, em vez de presumir que todos os modelos seguem exatamente o mesmo layout de pinos.

11.5.1 Painel frontal

O conector de **painel frontal** (*front panel* ou *system panel header*) da placa-mãe recebe os fios provenientes do gabinete: o botão de energia, o botão

de reinicialização (*reset*), o LED de energia e o LED de atividade do disco. Como o gabinete e a placa-mãe são fabricados por empresas diferentes, a correspondência exata de pinos varia de modelo para modelo e deve ser conferida no manual da placa-mãe.

Uma distinção importante deve ser observada nesses conectores: **botões** (power e reset) não possuem polaridade — o fio pode ser conectado em qualquer uma das duas orientações possíveis, sem alterar o funcionamento. **LEDs**, por serem diodos emissores de luz, possuem polaridade definida (um terminal positivo e um negativo): conectados na orientação invertida, simplesmente não acendem, sem dano ao componente.

11.5.2 Aterramento do gabinete

O chassi metálico do gabinete deve manter continuidade elétrica com o condutor de aterramento da instalação — tanto através do cabo de alimentação da fonte quanto da fixação mecânica da própria fonte e da placa-mãe ao gabinete. Essa continuidade permite que eventuais excessos de carga acumulados no chassi (Apêndice A, §A.4.1) sejam escoados de forma segura para a terra, em vez de se acumularem e serem sentidos pelo usuário ao tocar o gabinete.

Atenção. A verificação de continuidade entre o terra da tomada e o chassi do gabinete deve sempre ser feita com um multímetro, nunca por contato direto. Uma instalação elétrica com fase e terra invertidos por erro de instalação pode energizar todo o chassi de um computador aterrado incorretamente — situação já registrada em ambientes reais, inclusive institucionais, e potencialmente fatal.

Pulseiras antiestáticas de aterramento, usadas para escoar cargas eletrostáticas do próprio corpo do técnico durante o manuseio de componentes sensíveis, dependem da confiabilidade da instalação elétrica ao qual são conectadas: um técnico deve avaliar, caso a caso, se confia o suficiente na instalação disponível antes de se conectar diretamente a ela por meio de uma pulseira — preferindo, em caso de dúvida, outras formas de controle eletrostático que não impliquem conexão direta a uma instalação de aterramento não verificada.

Figura pendente

diagrama do painel frontal de uma placa-mãe com os pinos de power switch, reset, HDD LED e power LED identificados, indicando polaridade dos LEDs

---

11.6 Síntese do capítulo

Este capítulo apresentou a distinção entre fontes lineares e chaveadas, a arquitetura e os sinais de controle da fonte ATX, e os critérios de eficiência,

fator de potência e dimensionamento que orientam a escolha e o diagnóstico de uma fonte real. Do ponto de vista metodológico, o capítulo reforça e aplica em contexto elétrico o princípio de diagnóstico por isolamento de módulos apresentado no Capítulo 12 (§12.4) — testar cada elo da cadeia tomada–fonte–placa-mãe isoladamente para localizar a origem de uma falha.

A energia entregue e regulada pela fonte, tratada aqui, é justamente o que torna possível o funcionamento do componente estudado no Capítulo 4: o processador. Os conceitos de tensão, corrente e dissipação de potência por efeito Joule (Apêndice A) retornarão, em outra escala, na discussão sobre consumo energético e dissipação térmica de processadores (Capítulo 4) — onde o mesmo compromisso entre potência entregue e eficiência de conversão, já estabelecido neste capítulo para a fonte, reaparece na análise de benchmark e desempenho da CPU (Capítulo 13).

11.7 Referências

1. WIKIPEDIA. “ATX.” Disponível em: `https://en.wikipedia.org/wiki/ATX`.
2. INTEL CORPORATION. *ATX12V Power Supply Design Guide*. (Versão a confirmar pelo autor.) Como fonte secundária: eTechnophiles. “ATX Power Supply Connector Pinout (20 & 24 pins).” Disponível em: `https://www.etechnophiles.com/atx-power-supply-connector-pinout/`.
3. WIKIPEDIA. “ATX.” Disponível em: `https://en.wikipedia.org/wiki/ATX`; COMPUTER HOPE. “What Is ATX?” Disponível em: `https://www.computerhope.com/jargon/a/atx.htm`.
4. WIKIPEDIA. “Voltage regulator module.” Disponível em: `https://en.wikipedia.org/wiki/Voltage_regulator_module`.
5. CYBENETICS LABS. “ETA — PSU Efficiency Certification.” Disponível em: `https://www.cybenetics.com/index.php?option=eta`; “LAMBDA — PSU Noise Level Certification.” Disponível em: `https://www.cybenetics.com/index.php?option=lambda-%28power-supplies%29`.
6. WIKIPEDIA. “80 Plus.” Disponível em: `https://en.wikipedia.org/wiki/80_Plus`.
7. ANANDTECH. “The Seasonic Focus Plus Gold 750FX 750W PSU Review.” Disponível em: `https://www.anandtech.com/show/14338/the-seasonic` 3.
8. AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA (ANEEL). Resolução Normativa nº 414, de 9 de setembro de 2010. (Edição/atualizações a confirmar pelo autor.)

12 Manutenção de Computadores: Definição e Método

Neste capítulo você vai estudar a manutenção de computadores como disciplina técnica — sua definição, sua divisão em manutenção corretiva e preventiva, as condições físicas de trabalho que afetam a qualidade de um reparo, e o raciocínio de diagnóstico que sustenta o trabalho do técnico de informática. Os capítulos anteriores construíram o conhecimento técnico necessário — do transistor à arquitetura de von Neumann, passando por processador, memória, sistema operacional, hardware físico e eletricidade; este capítulo dá nome e método ao ofício que aplica esse conhecimento na prática, servindo de ponte para o Capítulo 14, que trata da gestão desse trabalho em escala.

12.1 Manutenção de computadores: definição e escopo

Este livro adota a seguinte definição, comum na literatura técnica de manutenção industrial e plenamente aplicável à informática:

> “Manutenção é a combinação de todas as ações técnicas e administrativas, incluindo supervisão, destinadas a manter ou recolocar um item em estado no qual possa desempenhar uma função requerida.” [1]

Essa definição contém uma implicação frequentemente ignorada por quem está começando na área: manutenção não é sinônimo de conserto. Todo computador precisa de manutenção — não apenas os que já apresentam defeito. Essa distinção é o que separa a **manutenção corretiva** (agir depois que a falha ocorreu) da **manutenção preventiva** (agir antes que ela ocorra), tratadas em detalhe na Seção 12.3.

12.2 O sistema computacional como filtro de diagnóstico

A tríade hardware, software e pessoas — apresentada no Capítulo 1 (§1.6) — é, para a manutenção, o primeiro filtro de qualquer diagnóstico: diante de um chamado técnico, a primeira pergunta não é "qual é o defeito?", mas "em qual dessas três frentes está o problema?".

Exemplo. Um chamado técnico relata que "o mouse não está funcionando". Existem, ao menos, três hipóteses plausíveis, uma em cada frente do sistema computacional:

Frente	Hipótese	Diagnóstico
Software	O driver do *trackpad* do notebook não está instalado	O sistema operacional não reconhece o dispositivo apontador
Hardware	O mouse físico está com defeito ou sem pilha	O componente em si não opera
Pessoa (usuário)	O mouse é sem fio e não está pareado com o computador	O hardware e o software estão corretos; falta uma ação do usuário

Repare que, no terceiro caso, não há defeito de hardware nem de software: o problema está inteiramente na operação. É comum que problemas relatados como falha técnica sejam, na prática, erro de uso — por isso o técnico precisa investigar as três frentes antes de concluir qualquer diagnóstico.

Figura pendente

diagrama "sistema computacional = hardware + software + pessoas" com o exemplo do mouse ramificando nas três hipóteses

12.3 Manutenção corretiva e manutenção preventiva

- **Manutenção corretiva**: conjunto de ações tomadas para sanar um problema já identificado. O computador chega com sintomas de um comportamento indevido, e a tarefa do técnico é, a partir desses sintomas, identificar o defeito e realizar a substituição ou correção — seja em hardware (troca de um módulo), seja em software (reinstalação ou reconfiguração de um programa).

- **Manutenção preventiva**: conjunto de ações tomadas para reduzir a probabilidade de que um problema venha a ocorrer, executadas antes de qualquer falha se manifestar.

Não existe uma periodicidade única para a manutenção preventiva — ela depende do tipo de ação e das condições de uso do equipamento.

Exemplo. No ambiente de laboratório de um campus, os computadores costumam ser reinstalados uma vez por ano letivo, prática que reduz o índice de problemas ao longo do semestre; sob uso muito mais intenso, esse intervalo poderia cair para seis meses. Já a cópia de segurança (*backup*) de dados de uso profissional é uma ação de manutenção preventiva que deve ser executada diariamente: um computador sem rotina de backup que sofre uma falha grave perde integralmente os dados que não foram copiados.

Manutenção corretiva e preventiva se aplicam às três frentes do sistema computacional (Seção 12.2): existem ações preventivas relacionadas a hardware (limpeza, inspeção), a software (backup, atualização) e também a pessoas — treinamento e orientação do usuário para reduzir erros de operação recorrentes.

12.3.1 Condições reais e ideais de trabalho

A qualidade de uma manutenção não depende só do diagnóstico correto (Seção 12.4): depende também das condições físicas em que o reparo é executado. Três fatores concentram a maior parte do risco evitável numa bancada de manutenção.

Controle eletrostático (ESD). O corpo humano acumula, por atrito simples (caminhar sobre um carpete, por exemplo), uma carga eletrostática que pode chegar a milhares de volts — imperceptível ao toque, mas suficiente para danificar permanentemente um circuito integrado sensível, cuja tolerância a descargas é medida em dezenas ou centenas de volts [2]. A condição ideal de bancada inclui uma superfície de trabalho **antiestática** (um tapete condutor aterrado) e, quando disponível uma instalação de aterramento confiável (Apêndice A, §A.4), uma pulseira antiestática conectando o técnico a essa mesma referência de terra — a mesma ressalva já feita no Capítulo 11 (§11.5.2) se aplica aqui: só vale a pena se conectar a um aterramento em que se confia.

Umidade e risco de condensação. Um componente eletrônico transportado de um ambiente frio (por exemplo, um veículo com ar-condicionado) para um ambiente quente e úmido pode sofrer condensação de água em sua superfície — o mesmo fenômeno que embaça um copo gelado num dia quente. Água condensada sobre um circuito energizado é um risco direto de curto-circuito. A prática recomendada é deixar o equipamento estabilizar à temperatura ambiente, ainda desligado, antes de energizá-lo.

Organização como parte do método, não só do ambiente. Ao desmontar um equipamento com múltiplos parafusos e cabos semelhantes entre si (o Capítulo 8, §8.2, trata o procedimento de desmontagem em detalhe), separar e identificar cada peça na ordem de remoção evita o erro mais

comum de reparo malsucedido: a remontagem incorreta. Essa organização não é uma questão de estética de bancada — é uma extensão direta do método de diagnóstico por hipóteses (Seção 12.4): um técnico que não sabe de onde veio cada parafuso não consegue isolar, com confiança, se um problema após a remontagem foi causado pelo reparo em si ou por uma montagem incorreta.

12.4 O raciocínio diagnóstico: hipóteses e método científico

A habilidade central desenvolvida ao longo desta disciplina é a de **isolar um problema** dentro do sistema computacional. Essa habilidade, como programar ou nadar, não se aprende apenas lendo — desenvolve-se pela prática repetida.

O procedimento subjacente ao diagnóstico técnico reproduz o método científico: a partir de uma observação (o sintoma relatado), formula-se uma **hipótese** que explique aquele comportamento, testa-se essa hipótese e verifica-se se ela é válida ou deve ser descartada em favor de outra.

Exemplo. Um computador entra em ciclo de reinicialização (POST bem-sucedido, tela de instalação do sistema operacional trava e reinicia). As hipóteses possíveis incluem: defeito na memória secundária, imagem de instalação corrompida, problema de configuração da máquina, ou defeito de hardware específico daquele computador. Um teste direcionado — por exemplo, conectar o disco em outra máquina e rodar um software de diagnóstico de saúde do disco (percentual de blocos defeituosos) — permite confirmar ou descartar a hipótese do disco sem precisar investigar as demais.

Um princípio prático orienta a ordem em que as hipóteses devem ser testadas: **teste primeiro a hipótese mais barata de verificar**, não necessariamente a mais provável. Tempo é um recurso escasso no trabalho técnico; investigar uma hipótese complexa (por exemplo, desmontar o computador para testar um componente em outra máquina) antes de descartar uma hipótese simples (por exemplo, trocar a imagem de instalação e tentar de novo) desperdiça tempo caso a causa fosse, de fato, a mais simples.

Esse tipo de raciocínio — geração de hipóteses, priorização por custo de verificação, teste e confirmação ou descarte — não é adquirido de uma vez; desenvolve-se ao longo da prática cotidiana como técnico, à medida que se acumula repertório de problemas já vistos e resolvidos. É esse mesmo raciocínio, aplicado em escala organizacional em vez de a um único chamado, que estrutura o Capítulo 14.

12.5 Síntese do capítulo

Este capítulo apresentou a manutenção de computadores como disciplina fundamentada na distinção entre manutenção corretiva e preventiva, nas condições físicas ideais de bancada (controle eletrostático, umidade, organização), no raciocínio de diagnóstico por hipóteses e no reconhecimento do sistema computacional como uma tríade de hardware, software e pessoas. Esses fundamentos — sobretudo a noção de que todo diagnóstico começa por isolar em qual das três frentes está o problema, e que hipóteses devem ser testadas da mais barata para a mais cara — sustentam o capítulo seguinte, que trata da gestão desse trabalho em escala organizacional.

12.6 Referências

1. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 5462: Confiabilidade e mantenabilidade — Terminologia*. Rio de Janeiro: ABNT, 1994.
2. ESD ASSOCIATION. *Fundamentals of Electrostatic Discharge*. Disponível em: `https://www.esda.org`; ou ABNT NBR IEC 61340-5-1 (controle eletrostático em ambientes que manuseiam dispositivos sensíveis).

13 Ferramentas de Benchmark e Dimensionamento de Computadores

Neste capítulo você vai estudar a relação entre resolução de vídeo e demanda de processamento, a metodologia de benchmark que permite comparar hardware de fabricantes e gerações diferentes, as ferramentas de diagnóstico em campo (CPU-Z e HWMonitor), o conceito de gargalo e o método completo de dimensionamento de um computador dentro de um orçamento, e — aplicando na prática o raciocínio diagnóstico do Capítulo 12 — a manutenção preventiva e corretiva específica de notebooks.

13.1 Resolução, taxa de quadros e demanda gráfica

Um monitor é, fisicamente, uma matriz de milhões de **pixels**, cada um capaz de assumir uma cor por meio da combinação aditiva das cores primárias de luz — vermelho, verde e azul (RGB, *Red-Green-Blue*). A síntese aditiva parte do preto (monitor desligado) e soma luz até formar as demais cores; o processo é diferente da impressão em papel, que parte do branco e usa síntese subtrativa de tinta (ciano, magenta, amarelo).

A **resolução** de um monitor descreve a quantidade estática de pixels — largura por altura:

Nome comercial	Resolução (pixels)
HD	1280 × 720 (também chamado 720p)
Full HD	1920 × 1080 (também chamado 1080p)
2K / QHD	2560 × 1440
4K	3840 × 2160

Quanto maior a resolução, maior o número de pixels que o hardware precisa calcular, transportar e copiar continuamente — trabalho que recai, em última instância, sobre o processador (e sobre a GPU, tratada na Seção 13.2).

A segunda variável relevante é a **taxa de quadros por segundo** (FPS, *frames per second*): quantas vezes por segundo a imagem inteira é atualizada.

O olho humano percebe movimento contínuo a partir de aproximadamente 24 atualizações por segundo — o mesmo princípio de um desenho animado feito quadro a quadro num caderno.

Exemplo. Para estimar o volume de dados de vídeo que o hardware precisa processar por segundo, multiplica-se a quantidade de pixels de um quadro pela taxa de quadros por segundo:

- 720p a 30 FPS: $1.280 \times 720 \times 30 \approx 27{,}6$ milhões de pixels processados por segundo.
- Full HD (1080p) a 60 FPS: $1.920 \times 1.080 \times 60 \approx 124{,}4$ milhões de pixels processados por segundo.

A diferença entre os dois cenários é de aproximadamente 4,5 vezes — um salto de resolução de 720p para Full HD, combinado com o dobro da taxa de quadros, quase quintuplica a demanda computacional. Esse cálculo explica por que jogos e aplicações gráficas listam requisitos mínimos e recomendados atrelados tanto a uma resolução quanto a uma taxa de FPS específicas.

Vale registrar que nem toda queda de desempenho percebida em jogos multijogador tem origem no processamento local: quando o computador atua como cliente de um servidor remoto (típico de jogos *online*), atrasos de rede (Wi-Fi, latência do provedor, disponibilidade do servidor) também produzem sensação de travamento, independentemente da capacidade da CPU e da GPU locais.

Figura pendente

comparação lado a lado da mesma imagem renderizada em diferentes resoluções, evidenciando o tamanho dos pixels

13.2 Benchmark: metodologia de comparação e dimensionamento

Ao especificar um computador para uma finalidade concreta — por exemplo, atender aos requisitos mínimos de um jogo —, o técnico se depara com um problema: os requisitos publicados pelo fabricante do software costumam ser descritivos (sistema operacional, um modelo específico de CPU, quantidade de memória RAM, um modelo específico de GPU, espaço de armazenamento), e não numéricos. Memória RAM e armazenamento são diretamente comparáveis em gigabytes — qualquer módulo de 8 GB atende a um requisito de 8 GB. CPU e GPU não: não existe uma unidade simples que permita comparar diretamente um processador de um fabricante, arquitetura e geração com outro processador de fabricante, arquitetura e geração diferentes.

A solução adotada pela indústria é o **benchmark** — literalmente, "bancada de testes". Todo processador ou placa de vídeo submetido ao mesmo teste padronizado recebe uma nota (*score*) comparável.

As duas notas de CPU. Ferramentas de benchmark para processador (a mais usada em sala é o PassMark, cujo software CPU-Z é abordado na Seção 13.3) produzem duas notas distintas:

- **Single-thread rating** — desempenho de um único núcleo trabalhando sozinho.
- **Multi-thread rating** — desempenho de todos os núcleos e threads trabalhando simultaneamente.

Essa distinção é decisiva na prática: a maioria dos softwares de uso comum (planilhas, navegadores) e a maioria dos jogos não são otimizados para dezenas de núcleos simultâneos — eles dependem, sobretudo, do desempenho *single-thread*. Cargas de trabalho de servidor, renderização e simulação, por outro lado, se beneficiam diretamente do desempenho *multi-thread*.

Exemplo. Uma comparação real conduzida em sala, entre um processador Intel (lançado em 2015) e um processador AMD Ryzen (lançado em 2017), ilustra o ponto: o processador Intel obteve nota *single-thread* de aproximadamente 2.315 pontos contra aproximadamente 2.000 pontos do AMD (uma diferença de cerca de 10% a favor da Intel); já na nota *multi-thread*, o AMD obteve cerca de 12.000 pontos contra 6.305 pontos da Intel — quase o dobro. *(Nota: dado autoral de demonstração em sala; os modelos exatos de CPU não são nomeados aqui — vale documentá-los, ou anexar a captura de tela do PassMark usada na aula, para que o exemplo seja reproduzível.)* A conclusão prática: para um computador cuja finalidade é jogar (dependente de desempenho *single-thread*), o processador Intel seria a escolha mais adequada, apesar de mais antigo; para um servidor cuja carga se distribui por múltiplas threads simultâneas, o AMD seria a escolha correta. O benchmark permite essa comparação mesmo entre processadores de fabricantes, arquiteturas, gerações, caches e potências diferentes, porque ambos foram submetidos exatamente à mesma prova.

Ganho geracional. Comparações de longo prazo dentro de uma mesma linha de produto evidenciam o efeito acumulado da miniaturização (Capítulo 4, §4.3) e de melhorias de arquitetura. Um exemplo apresentado em aula, de uma mesma família de processador ao longo de gerações sucessivas: em 2017, um modelo de 65 W entregava cerca de 2.000 pontos *single-thread*; em 2018, a geração seguinte, operando a 105 W, chegou a cerca de 2.400 pontos; já em 2024, uma geração mais recente voltou a operar a 65 W (a mesma potência de 2017) e alcançou aproximadamente 4.500 pontos *single-thread* e cerca de 30.000 pontos *multi-thread* — mais do que o dobro do desempenho *single-thread* de sete anos antes, com potência igual à do ponto de partida. *(Nota: família de processador específica a documentar pelo autor, para que a série seja rastreável.)* Esse é o retrato numérico do que o Capítulo 4, §4.3, descreve de forma qualitativa: litografias menores entregam, ao mesmo tempo, mais desempenho e mais eficiência energética.

Benchmark de GPU e custo-benefício. O mesmo tipo de ferramenta existe para placas de vídeo. Ao comparar duas GPUs reais de gerações

próximas, por exemplo, uma custando cerca de R$ 2.890 com nota de aproximadamente 22.000 pontos, contra outra custando cerca de R$ 4.600 com nota de aproximadamente 28.000 pontos, *(GPUs específicas e data de consulta de preço a documentar pelo autor, já que preços em reais mudam rapidamente)* o técnico consegue avaliar objetivamente se o ganho de desempenho (cerca de 27%) justifica o acréscimo de custo (cerca de 60%) para aquele cliente específico — e ainda precisa verificar se a fonte de alimentação do computador suporta a potência adicional exigida pela placa mais cara, sob risco de o upgrade da GPU obrigar também um upgrade da fonte.

Uma forma particularmente útil de visualizar essas comparações é o gráfico de dispersão (*scatter plot*), com a nota de benchmark no eixo horizontal e o preço no eixo vertical: quanto mais à direita e mais abaixo estiver um produto nesse gráfico, melhor o seu custo-benefício. O cálculo de custo-benefício em si é simples — nota de benchmark dividida pelo preço do componente.

Exemplo — RISC contra CISC, lado a lado. A comparação mais didática de eficiência energética entre arquiteturas usa um processador RISC de origem móvel (o Apple A18 Pro, originalmente projetado para iPhone e reaproveitado num MacBook) contra um processador CISC de laptop equivalente: o A18 Pro obteve cerca de 4.000 pontos *single-thread* e quase 12.000 pontos *multi-thread*, consumindo entre 4 W e 10 W [1]; o processador CISC equivalente consumiu cerca de dez vezes mais potência para entregar uma nota inferior *(modelo CISC específico a documentar pelo autor)*. É esse tipo de comparação, quantificada por benchmark, que explica por que a autonomia de bateria de dispositivos com chips RISC é tão superior — e por que a indústria de desktops e laptops caminha na direção descrita no quadro do Capítulo 4, §4.2.

Figura pendente

captura de tela do PassMark comparando dois processadores lado a lado, com notas single-thread e multi-thread destacadas] [IMAGEM: gráfico de dispersão (scatter plot) de nota de benchmark por preço, com pontos coloridos por fabricante

13.3 Diagnóstico em campo: CPU-Z e HWMonitor

Duas ferramentas de software, de uso corrente entre técnicos, permitem levantar as características de um processador e monitorar seu comportamento sob carga sem a necessidade de desmontar o computador.

CPU-Z lê e apresenta, a partir do sistema operacional em execução, informações detalhadas de hardware: nome comercial e codinome do processador, litografia, potência máxima, família de instruções, faixa de clock em operação, tamanho das caches L1/L2/L3, contagem de núcleos e threads,

fabricante e versão de BIOS da placa-mãe, configuração de canais de memória (single ou dual channel) e as GPUs disponíveis no sistema. Uma aba específica, chamada *bench*, aplica o teste de benchmark descrito na Seção 13.2 diretamente no computador em uso.

HWMonitor é um painel de sensores em tempo real — o equivalente a um monitor cardíaco preso a um atleta correndo em uma esteira. Ele reporta, para cada componente monitorado, os valores atual, mínimo e máximo registrados de temperatura, potência (watts) e outras grandezas.

Exemplo (demonstração em sala). Em um notebook equipado com processador Intel i7 de 13ª geração e 32 GB de RAM, em repouso a temperatura do pacote de processamento ficava próxima de 50–59 °C. Ao aplicar um teste de estresse (o mesmo tipo de ferramenta de benchmark da Seção 13.2, usada aqui para forçar a carga máxima), a potência consumida saltou para a faixa de 45–51 W e a temperatura chegou a 96–97 °C — no limite de segurança do fabricante. Nesse ponto, o *thermal throttling* (Capítulo 4, §4.4) reduziu a potência entregue para cerca de 19 W, e o ciclo de aceleração e desaceleração térmica ficou visível em tempo real no HWMonitor, com a potência oscilando repetidamente entre esses dois extremos. Ao conectar o notebook a um carregador de 100 W, o sistema operacional liberou automaticamente o "modo de alto desempenho" — deixando de gerenciar a bateria — e o processador passou a sustentar potências mais altas por mais tempo antes de sofrer *throttling* novamente. Esse comportamento — desempenho diferente conforme o notebook está ou não conectado à tomada — é característico de notebooks Windows; processadores Apple Silicon, por comparação, mantêm o mesmo desempenho ligados ou desligados da tomada, justamente por serem RISC e demandarem uma fração da potência.

Esse teste de estresse cumpre dupla função de diagnóstico: verifica, simultaneamente, se a **fonte de alimentação** (ou a bateria, no caso de notebooks) é capaz de entregar a potência de pico exigida pelo hardware sob carga máxima — tema do Capítulo 11 —, e se a **refrigeração** é adequada para sustentar essa carga sem superaquecimento. É o procedimento indicado, por exemplo, para validar se a troca de pasta térmica resolveu um quadro de reinicializações intermitentes atribuídas a superaquecimento.

Figura pendente

captura de tela do CPU-Z mostrando núcleos, threads, caches e clock de um processador real] [IMAGEM: captura de tela do HWMonitor durante um teste de estresse, evidenciando o ciclo de thermal throttling

13.4 Gargalo e dimensionamento orientado à aplicação

O **gargalo** de um sistema computacional é o componente que, em um dado momento, limita o desempenho do conjunto — e ele não é fixo: ao melhorar

um componente, o gargalo simplesmente se desloca para outro ponto do sistema (processador, GPU, armazenamento, memória RAM ou rede).

Exemplo. Um usuário que investiu uma quantia elevada em um processador de altíssimo desempenho, mas obteve resultado pior do que o esperado em um jogo específico, ilustra bem o problema: o jogo em questão não estava otimizado para tirar proveito de dezenas de núcleos de processamento, de modo que apenas uma pequena fração dos núcleos disponíveis era efetivamente utilizada, e o restante do investimento em poder computacional ficou ocioso.

Exemplo. Em um cenário de download de arquivos em uma rede universitária extremamente rápida, o gargalo real não estava na velocidade da internet, nem na memória RAM, nem no processador, mas na velocidade de escrita do disco rígido (HD) mecânico, incapaz de gravar os dados recebidos na mesma velocidade em que chegavam pela rede. Substituir esse HD por um SSD, de escrita muito mais rápida, não elimina o conceito de gargalo — apenas desloca o ponto de estrangulamento para outro componente do sistema, tipicamente a rede ou o servidor remoto.

Método de dimensionamento. A metodologia apresentada ao longo deste capítulo pode ser resumida em cinco passos:

1. Definir a **aplicação** do computador (jogo, estação de trabalho, servidor, uso de escritório etc.), pois é ela que determina quais notas de benchmark (single-thread, multi-thread, GPU) são relevantes.
2. Levantar os requisitos mínimos e recomendados publicados pelo fabricante do software-alvo.
3. Converter esses requisitos descritivos em notas de benchmark comparáveis, usando ferramentas como o PassMark.
4. Pesquisar, dentro do orçamento disponível, componentes cuja nota de benchmark atenda ou supere a meta, comparando custo-benefício entre alternativas de fabricantes e gerações diferentes.
5. Verificar se a escolha de um componente mais potente não desloca o gargalo, ou a exigência de potência, para outro ponto do sistema — tipicamente a fonte de alimentação.

Exemplo — orçamento de potência de um servidor. Considere um servidor local com processador de 150 W de consumo máximo, duas GPUs idênticas de 250 W cada, oito discos de armazenamento de 15 W cada, e um consumo combinado de placa-mãe, memória RAM e ventoinhas de 80 W:

Componente	Consumo máximo
Processador	150 W
GPUs (2 × 250 W)	500 W
Discos (8 × 15 W)	120 W
Placa-mãe, RAM e ventoinhas	80 W
Total de pico	**850 W**

Esse valor de 850 W representa o **pico teórico**, isto é, a soma dos consumos máximos individuais — um cenário raro na prática, já que dificilmente todos os componentes operam simultaneamente no limite. Em uso típico, um servidor como esse opera perto de 40% da carga de pico (cerca de 340 W neste exemplo). Retomando o Capítulo 11: a eficiência elétrica de uma fonte ATX é máxima justamente perto de 50% de sua carga nominal. Escolher uma fonte de exatos 850 W deixaria a operação típica numa faixa de baixa eficiência (cerca de 40% de 850 W) e sem margem alguma para upgrades futuros. A escolha tecnicamente correta é uma fonte de capacidade nominal maior — por exemplo, entre 1.000 W e 1.500 W —, de modo que a operação típica do servidor caia próxima da faixa de melhor eficiência da fonte, com folga de segurança para os momentos de pico.

Esse raciocínio fecha o ciclo entre o dimensionamento do processador (e dos demais componentes de processamento) e o dimensionamento da fonte de alimentação: nenhuma escolha de hardware de processamento pode ser tomada de forma isolada da capacidade de entrega de energia do sistema.

Figura pendente

tabela de orçamento de potência de um servidor, com a soma dos componentes e a faixa de operação típica marcada sobre a curva de eficiência de uma fonte ATX

13.5 Manutenção preventiva e corretiva em notebooks

As seções anteriores deste capítulo trataram o processador de forma geral, aplicável tanto a desktops quanto a notebooks. Esta seção fecha o capítulo tratando especificamente do que muda quando o computador em questão é portátil — um recorte que soma, mas não substitui, tudo o que já foi visto sobre refrigeração (Capítulo 8), fonte de alimentação (Capítulo 11) e thermal throttling (Capítulo 4, §4.4), aplicando na prática o raciocínio de manutenção corretiva e preventiva formalizado no Capítulo 12.

13.5.1 Bateria: o componente que só o notebook tem

A bateria de um notebook é uma célula de íon de lítio (ou polímero de lítio), sujeita a um processo de degradação química irreversível — diferente de qualquer componente puramente eletrônico já estudado neste livro, que falha por defeito, não por desgaste químico natural. Duas práticas de manutenção preventiva reduzem essa degradação:

- **Evitar ciclos completos de descarga.** Descarregar a bateria até 0% com frequência acelera sua degradação química, de forma análoga

(guardadas as proporções) ao desgaste por ciclo de escrita da memória flash (Capítulo 5, §5.12): a bateria de íon de lítio se degrada menos quando mantida, na maior parte do tempo, numa faixa intermediária de carga (por exemplo, entre 20% e 80%) do que quando ciclada repetidamente entre 0% e 100%.

- **Evitar calor prolongado.** A degradação química da bateria acelera com a temperatura — manter o notebook conectado à energia e sob carga de processamento pesada por longos períodos, numa superfície que restrinja a ventilação (uma cama, um sofá), combina justamente os dois fatores que mais aceleram esse desgaste: calor e carga elétrica sustentada.

Diagnóstico corretivo. Uma bateria degradada não perde a capacidade de operação instantaneamente — ela perde **capacidade total** (a autonomia cai progressivamente) e pode passar a apresentar inchaço físico, visível como uma leve deformação do chassi ou do touchpad. Um notebook com a bateria fisicamente inchada não deve continuar em uso: o inchaço indica acúmulo de gases internos por decomposição química da célula, um risco real de incêndio caso a célula seja perfurada ou continue sendo carregada.

13.5.2 Refrigeração: o desafio do espaço reduzido

A Seção 7.7 apresentou dissipador e ventoinha como os dois componentes do resfriamento a ar. Num notebook, o mesmo princípio se aplica, mas dentro de um volume drasticamente menor — o que tem duas consequências práticas diretas:

- **Acúmulo de poeira em menos espaço, com efeito proporcionalmente maior.** As aletas do dissipador de um notebook são mais finas e mais próximas entre si do que as de um cooler de desktop, para caber no chassi fino. Uma mesma quantidade de poeira acumulada obstrui, proporcionalmente, uma fração muito maior da área de passagem de ar — o que explica por que notebooks tendem a apresentar thermal throttling (Capítulo 4, §4.4) por acúmulo de poeira num intervalo de tempo menor do que um desktop equivalente. A limpeza preventiva do dissipador e da ventoinha (normalmente com ar comprimido, sem desmontar o notebook por completo) é, por essa razão, mais frequente em notebooks do que em desktops.
- **Repasse de pasta térmica mais delicado.** O procedimento de troca de pasta térmica (Capítulo 8, §8.6) segue o mesmo princípio físico num notebook, mas a desmontagem para acessar o processador é, em geral, mais invasiva: exige remover a placa inteira do chassi em muitos modelos, em vez de apenas liberar um dissipador preso por presilhas externas como num desktop. Por isso, esse é um procedimento tipicamente reservado a um técnico com experiência prévia em desmontagem daquele modelo específico — um manual de serviço (*service manual*) do fabricante, quando disponível, é a referência mais confiável para essa etapa.

13.5.3 Componentes soldados: o que muda no diagnóstico por substituição

O Capítulo 1 (§1.11.2) já apresentou o compromisso entre modularidade e portabilidade: memória RAM e, cada vez mais, o próprio processador vêm soldados à placa num notebook. Isso tem uma consequência direta sobre o método de diagnóstico por substituição de módulo (Capítulo 1, §1.6.2): quando o componente suspeito está soldado, não é possível simplesmente trocá-lo por um equivalente para testar a hipótese. As alternativas de diagnóstico corretivo, nesse cenário, são:

- Testar o mesmo sintoma num live USB (Capítulo 7, §7.3) para isolar hardware de software, já que essa técnica não depende de trocar peça alguma.
- Testar o componente suspeito em outro notebook do mesmo modelo, quando disponível (por exemplo, numa bancada com máquinas de reposição), como forma indireta de aplicar o mesmo princípio de substituição sem precisar desoldar nada.
- Quando a suspeita se confirma e o componente é de fato o defeituoso, a correção deixa de ser uma troca de módulo e passa a ser retrabalho de solda em nível de placa — fora do escopo de manutenção de bancada convencional na maioria dos casos, e que costuma justificar, financeiramente, a comparação entre o custo do reparo e o custo de um equipamento novo (a mesma lógica de obsolescência econômica já discutida a propósito de soquetes de CPU, Capítulo 4, §4.7).

13.5.4 Reparos mais comuns: tela e teclado

Diferente do processador e da memória, tela e teclado são, na prática de bancada, os componentes de notebook mais frequentemente substituídos — não por serem tecnologicamente mais frágeis, mas por estarem fisicamente mais expostos a dano por impacto, derramamento de líquido e desgaste de uso repetitivo. Em praticamente todo modelo de notebook, ambos são módulos destacáveis (presos por parafusos e conectores do tipo *ribbon cable*, uma fita plana de contatos), o que os torna reparáveis por substituição direta mesmo em modelos que soldam processador e RAM — uma peça de reposição compatível, comprada do fabricante ou de um fornecedor terceirizado, restaura o equipamento sem exigir retrabalho de solda.

Figura pendente

notebook com a tampa inferior removida, evidenciando bateria, dissipador miniaturizado e conectores tipo ribbon cable da tela e do teclado

13.6 Síntese do capítulo

Este capítulo apresentou a relação entre resolução de vídeo e demanda computacional, e a metodologia de benchmark que transforma requisitos descritivos de software em notas numéricas comparáveis entre fabricantes, arquiteturas e gerações diferentes — aplicada tanto a CPU quanto a GPU, e sistematizada nas ferramentas de diagnóstico em campo CPU-Z e HWMonitor. O exemplo de orçamento de potência da Seção 13.4 retoma diretamente o dimensionamento de fonte de alimentação estudado no Capítulo 11, evidenciando que a escolha de processador, GPU e armazenamento nunca é independente da capacidade de entrega de energia do sistema. O capítulo fechou aplicando esses mesmos princípios — refrigeração, energia, diagnóstico por substituição, manutenção corretiva e preventiva — ao caso específico do notebook, cujo espaço reduzido e componentes soldados exigem adaptações do método geral. Os conceitos de diagnóstico por eliminação de módulos, teste de estresse e identificação de gargalo, exercitados aqui, serão retomados e sistematizados no Capítulo 14, dedicado ao atendimento e suporte técnico.

13.7 Referências

1. PASSMARK/CPU BENCHMARK. “Apple A18 Pro (MacBook Neo) Benchmark.” Disponível em: `https://www.cpubenchmark.net/cpu.php?cpu=Apple+A18+Pro+%28MacBook+Neo%29&id=7232`.

14 Suporte e Gestão de Serviços

Neste capítulo você vai estudar a organização do atendimento técnico ao usuário: a central de serviços (*service desk*) como ponto único de entrada das demandas, a categorização de chamados em evento, incidente e solicitação, os critérios de priorização por impacto e urgência, os níveis de suporte e o custo crescente de cada escalonamento, e as ferramentas e princípios éticos que regem o acesso remoto a máquinas de terceiros.

14.1 Central de serviços (service desk)

Quando um problema técnico surge — por exemplo, a internet de um laboratório para de funcionar —, o primeiro passo do diagnóstico é distinguir se a causa é **local** (a máquina específica) ou de **infraestrutura** (algo fora do alcance daquele posto de trabalho, que depende do setor de Tecnologia da Informação). Uma vez identificado que o problema é de infraestrutura, a demanda deve ser encaminhada a um ponto único de contato.

Esse ponto único é a **central de serviços**, também chamada de *service desk*. Trata-se de um hub para onde convergem todas as demandas de um setor ou organização, e a partir do qual elas recebem tratamento e encaminhamento adequados. Em vez de o usuário procurar diretamente um técnico específico, toda solicitação passa por esse canal centralizado, que a registra, categoriza e distribui.

Exemplo. Um caso amplamente divulgado na imprensa e em redes sociais em Natal/RN ilustra o que ocorre quando uma central de serviços é sobrecarregada ou mal gerida: em 2026, clientes da operadora **Alares** relataram instabilidade prolongada de internet, TV e telefone em bairros como Ponta Negra, Pitimbu e Candelária, sem conseguir contato pelos canais oficiais nem para suporte técnico nem para cancelamento de serviço — a ponto de precisarem procurar a sede da empresa pessoalmente —, o que motivou fiscalização do Procon-RN na sede da empresa [2]. O episódio mostra que a central de serviços não é um detalhe administrativo: é a peça que determina se a organização consegue, de fato, responder às demandas de seus usuários. *(Nota: caso identificado por correspondência forte com a descrição do capítulo — confirmar com o autor se é este o episódio pretendido.)*

No âmbito do IFRN, a central de serviços é operada pelo sistema **SUAP** (Sistema Unificado de Administração Pública) [1], tratado em detalhe na Seção 14.6.

Figura pendente

fluxograma — usuário identifica problema → local ou infraestrutura? → abertura de chamado na central de serviços → triagem

14.2 Categorização das demandas: evento, incidente e solicitação

Nem toda demanda que chega a uma central de serviços tem a mesma natureza. A prática de gestão de serviços de TI — estudada em profundidade em cursos de Administração e Engenharia de Produção, sob frameworks como o ITIL (*Information Technology Infrastructure Library*, atualmente em transição da versão 4 para a versão 5, lançada em fevereiro de 2026) [3] — classifica as demandas em três categorias.

14.2.1 Evento

Seguindo o vocabulário de gerenciamento de eventos do ITIL [4], um **evento** é uma ocorrência detectável que **não interrompe** o trabalho do usuário, mas gera um sinal de atenção. Está fortemente associado à manutenção preventiva: o evento é o alerta que permite agir antes que um problema real se manifeste.

Exemplo. O uso de CPU de um servidor atinge 80% de ocupação; o nível de tinta ou toner de uma impressora cai abaixo de determinado percentual e o equipamento emite uma notificação; uma rotina de backup semanal projeta que o disco de armazenamento ficará sem espaço disponível dali a três semanas. Em nenhum desses casos o usuário deixou de trabalhar — mas em todos eles existe um sinal que, se ignorado, tende a se converter em um problema real.

14.2.2 Incidente

De acordo com o glossário oficial do ITIL [5], um **incidente** é uma interrupção não planejada ou uma redução da qualidade de um serviço de TI. Ao contrário do evento, o incidente **interrompe** o fluxo de trabalho do usuário e está associado à manutenção corretiva: exige uma ação de resposta, geralmente com maior urgência.

Exemplo. O computador não liga; a impressora trava; o usuário perde a conexão com a internet. Em todos esses casos, alguém que precisa de um recurso computacional está impedido de usá-lo.

14.2.3 Solicitação

No mesmo vocabulário do ITIL [6], uma **solicitação de serviço** é um pedido previsível, que não decorre de uma falha. Pode ser periódica ou não,

mas segue o fluxo padrão de atendimento, sem a urgência associada a um incidente.

Exemplo. Recuperar uma senha esquecida; solicitar a instalação de uma impressora nova; pedir a expansão do sinal de Wi-Fi em um setor; solicitar a troca da conexão de um computador de rede sem fio para rede cabeada; os procedimentos de matrícula e criação de e-mail institucional de um aluno recém-ingresso.

14.2.4 Quando uma solicitação vira incidente

A fronteira entre as três categorias depende do critério de **interrupção do fluxo do usuário**, e não apenas da natureza superficial do pedido. Um exemplo recorrente: solicitar a substituição de um mouse com defeito é, em princípio, uma solicitação de serviço. Mas se aquele mouse é a ferramenta de trabalho de um funcionário que, sem ele, não consegue produzir, a demanda deixa de ser uma solicitação comum e passa a ser tratada como **incidente** — porque há um custo real (o salário daquela pessoa, parada) sendo desperdiçado a cada minuto sem solução.

A tabela a seguir resume os três critérios de categorização:

Categoria	Interrompe o trabalho do usuário?	Tipo de manutenção associada	Exemplos
Evento	Não	Preventiva	CPU do servidor em 80%; toner baixo; projeção de disco cheio
Incidente	Sim	Corretiva	Computador não liga; impressora travada; sem internet
Solicitação	Não (em geral)	Preventiva ou corretiva, conforme o caso	Redefinição de senha; instalação de software; expansão de Wi-Fi

14.3 Priorização: impacto, urgência e prioridade

Como as demandas de uma central de serviços não chegam de forma organizada nem podem ser todas atendidas simultaneamente, é preciso um critério para decidir a ordem de atendimento. Esse critério não é a ordem

de chegada — nem todo incidente é atendido na sequência em que foi registrado. O que determina a ordem de atendimento é a **prioridade**, calculada a partir de dois fatores:

- **Impacto**: quantas pessoas são afetadas pelo problema e quão crítico é o serviço afetado.
- **Urgência**: quão rapidamente a situação precisa ser resolvida antes de gerar prejuízo maior.

A combinação desses dois fatores define a prioridade de atendimento, seguindo o modelo da matriz de prioridade do ITIL (impacto × urgência) [7]:

Impacto \ Urgência	Urgência baixa	Urgência alta
Impacto baixo	Prioridade baixa — ex.: substituir teclados antigos, sujos, mas ainda funcionais	Prioridade moderada
Impacto alto	Prioridade alta — ex.: trocar cadeiras que causam problema de postura para toda uma equipe	Prioridade crítica — ex.: internet caiu; monitor quebrou; servidor da secretaria parou no dia da matrícula

Exemplo. Um servidor da secretaria escolar cair justamente no dia de matrícula tem impacto alto (afeta um processo crítico e um grande volume de pessoas) e urgência alta (o processo tem prazo), resultando em prioridade crítica: todo o trabalho em andamento é interrompido para atender essa demanda primeiro. Já a ausência do papel de parede padrão em um computador é um problema de impacto e urgência baixos, que pode aguardar.

Quando não existe um sistema formal de chamados — como ocorre em equipes pequenas de suporte —, a definição de prioridade cabe ao critério do supervisor responsável, que aplica a política da organização mesmo sem registrá-la formalmente em um sistema. Um técnico que ingressa em uma empresa recebe, na fase de treinamento, a orientação sobre qual política de priorização deve seguir.

Figura pendente

matriz impacto x urgência com quadrantes de prioridade destacados

14.4 Níveis de suporte e o custo do atendimento

Uma vez aberto, o chamado (também chamado de **ticket** — o nome vem do francês antigo *estiquet* (“nota anexada”), a mesma raiz de “etiqueta” em português [8]) segue por níveis de suporte sucessivos, até ser resolvido. A cada nível não resolvido, o problema é **escalado** para o nível seguinte.

Nível	Descrição	Modo típico de atendimento
Nível 0	Autoatendimento: o próprio usuário resolve o problema com apoio de um guia ou tutorial fornecido pelo sistema, sem necessidade de abrir chamado	Tutoriais, guias de configuração, e cada vez mais chatbots baseados em IA generativa como camada intermediária
Nível 1	Triagem e suporte básico	Telefone, WhatsApp, acesso remoto
Nível 2	Suporte técnico local	Visita física ao local, diagnóstico dedicado
Nível 3	Especialistas ou fabricantes do equipamento	Escalonamento para equipes especializadas externas

Modelo de níveis de suporte (0 a 3) [9]; a nomenclatura e os limites exatos entre níveis variam conforme a fonte consultada.

A escalada de um nível para outro não é gratuita: cada nível tem um custo mais alto do que o anterior. O custo de um técnico que se desloca fisicamente é maior do que o de um atendente de central telefônica; a esse custo de mão de obra somam-se custos de equipamento, transporte e logística. Por isso, o técnico de informática deve buscar, sempre que possível, resolver o problema no nível mais baixo em que a solução for viável — o que reforça a importância do acesso remoto como ferramenta de contenção de custo (Seção 14.5).

Exemplo real de escalonamento. A operadora de internet que enfrentou uma queda prolongada e generalizada em sua qualidade de serviço (o mesmo caso citado na Seção 14.1) precisou recorrer ao Nível 3 — especialistas e fabricantes de equipamento de infraestrutura [2] — porque o problema não se resolveu em um dia, nem em uma semana, evidenciando uma falha estrutural que os níveis anteriores de suporte não tinham capacidade de resolver sozinhos.

Figura pendente

diagrama de escalonamento nível 0 → nível 1 → nível 2 → nível 3, com custo crescente indicado

14.5 Acesso remoto: ferramentas e uso ético

Grande parte dos problemas resolvidos nos níveis 1 e 2 de suporte envolve apenas configuração de software, e não troca física de peças. Nesses casos, o **acesso remoto** permite ao técnico operar o computador do usuário a distância, evitando o custo de deslocamento. Entre as ferramentas mais usadas estão o SSH (acesso remoto via terminal, tipicamente para servidores e equipamentos de rede), o TeamViewer, o AnyDesk e o RustDesk, além da própria funcionalidade de Área de Trabalho Remota do Windows.

O acesso físico à máquina continua sendo necessário apenas quando o problema exige a substituição de um componente de hardware — tudo o que envolve exclusivamente configuração de software pode, em princípio, ser resolvido remotamente.

Exemplo. Um problema doméstico comum — um arquivo PDF que passou a abrir automaticamente no Word em vez do leitor de PDF, por associação de tipo de arquivo alterada — foi resolvido a distância por meio do AnyDesk: o técnico recebeu um código de acesso gerado no computador remoto e, a partir dele, assumiu o controle de teclado e mouse da máquina até corrigir a associação de arquivo.

Por envolver o controle total do computador de outra pessoa — muitas vezes com acesso a e-mails, mensagens, fotos e documentos pessoais —, o acesso remoto exige a observância de princípios éticos claros:

- **Consentimento explícito.** O acesso remoto só deve começar depois que o usuário autoriza expressamente o início da sessão.
- **Transparência durante a operação.** O técnico deve narrar o que está fazendo a cada passo (por exemplo, “vou abrir a configuração de rede”), para que o usuário se sinta seguro de que nada além do necessário está sendo acessado.
- **Não acessar dados pessoais.** Arquivos e mensagens pessoais não devem ser abertos, exceto quando estritamente necessários à resolução do problema relatado.
- **Encerramento comunicado e registrado.** Ao final, o técnico deve informar explicitamente que a sessão está sendo encerrada, e registrar o que foi feito, por quem, quando e onde. Esse registro protege tanto o usuário quanto o técnico: sem ele, um uso indevido do computador feito pelo próprio usuário depois do atendimento poderia ser injustamente atribuído ao técnico que teve acesso remoto anteriormente.

Esses princípios se relacionam diretamente com a **Lei Geral de Proteção de Dados (LGPD, Lei nº 13.709/2018)** [10]: dados pessoais de terceiros

não podem ser tratados sem o devido consentimento, o que se aplica tanto ao conteúdo acessado durante uma sessão remota quanto aos registros gerados por ela.

Figura pendente

tela de autenticação de uma ferramenta de acesso remoto, com destaque para o código de sessão e o aviso de consentimento

14.6 Sistemas de chamados na prática

O registro e o acompanhamento de chamados são feitos por sistemas de *ticketing*. O **GLPI** [11], por exemplo, é um software livre amplamente usado para essa finalidade, que organiza os chamados pendentes e os já designados a um responsável, funcionando como um painel supervisório do trabalho da equipe de suporte.

14.6.1 O SUAP como central de serviços do IFRN

No IFRN, a abertura de chamados é feita pelo módulo de serviços do SUAP. O usuário escolhe primeiro a **área** do serviço (Administração, Comunicação, EAD, Ensino, Gestão de Pessoas, Manutenção Predial, Pesquisa ou Tecnologia da Informação) [12] e, dentro dela, uma **subcategoria** cada vez mais específica — por exemplo, dentro de Tecnologia da Informação: Redes e Internet, Data Center, Equipamentos, Segurança da Informação, Sistemas e Aplicativos, entre outras.

Exemplo de simulação de abertura de chamado. Para um problema de sinal de Wi-Fi instável em um laboratório, o caminho na árvore de categorias é: Tecnologia da Informação → Redes e Internet → Rede sem fio, onde o sistema oferece três opções específicas: informar problema de acesso à rede sem fio, solicitar acesso à rede corporativa, ou solicitar expansão de rede sem fio. Ao abrir o chamado, o usuário preenche uma descrição do problema (por exemplo, “verificar viabilidade de expansão da rede sem fio nos laboratórios de manutenção”), pode adicionar outros interessados (como o coordenador do setor afetado, que passa a acompanhar as atualizações do chamado), anexar evidências (como um print do sinal fraco) e, dependendo da categoria, escolher entre atendimento pela equipe de TI local ou pela equipe de TI remota — alguns serviços, como VPN, só são resolvidos por esta última. Uma vez aberto, o chamado entra na fila de atendimento com um prazo de resposta esperado, tipicamente entre 48 e 120 horas conforme a categoria [13].

Um mesmo caso é útil para revisar a categorização apresentada na Seção 14.2: um sinal de Wi-Fi instável (mas não totalmente indisponível) não interrompe o uso, então não é um incidente; também não é um evento, porque não se trata de um sinal detectado por infraestrutura de monitoramento; é

uma **solicitação**, e por não interromper o fluxo de ninguém, recebe prioridade baixa.

Antes de permitir a abertura de um chamado, o próprio SUAP oferece, quando disponível, um guia de solução (por exemplo, um tutorial de configuração de rede) — uma tentativa de resolver o problema em Nível 0, sem necessidade de acionar a equipe de TI.

Figura pendente

captura de tela do SUAP — árvore de categorias de serviço até “Rede sem fio”] [IMAGEM: captura de tela do SUAP — formulário de abertura de chamado preenchido, com campos de descrição, interessados e anexo

14.6.2 O mesmo modelo no desenvolvimento de software

A lógica de chamados categorizados, priorizados e distribuídos entre responsáveis não é exclusiva da manutenção de hardware. No desenvolvimento de software, o equivalente ao chamado é a **issue**, registrada em plataformas de controle de versão (como o Git). Em equipes que seguem a metodologia Scrum, as tarefas são organizadas dentro de um intervalo de tempo fixo (*sprint*) — mas quem as distribui não é o *Scrum Master*: o time de desenvolvimento é **auto-organizado**, e nem mesmo o Scrum Master tem autoridade para dizer ao time como realizar seu trabalho [14]. O papel do Scrum Master é dar suporte ao time e servir de ponte de comunicação com quem está fora dele; quem mais se aproxima de “distribuir” prioridades é o *Product Owner*, ao ordenar o Backlog do Produto — e, mesmo assim, sem atribuir tarefas individualmente. Outra abordagem comum é o método **Kanban** [15], no qual cada tarefa avança por colunas visuais que tornam o fluxo de trabalho visível a toda a equipe simultaneamente — por exemplo, do *backlog* (tarefas pendentes) para “em resolução”, depois “teste”, “avaliação” e, por fim, “finalizado”. O Kanban, como método, define princípios (visualizar o fluxo, limitar trabalho em andamento) e não uma lista fixa de nomes de coluna — o número e os nomes das colunas variam de equipe para equipe.

Vale registrar uma ressalva prática sobre sistemas de chamados em geral: a exigência de registrar formalmente cada atendimento pode gerar uma burocracia que, paradoxalmente, reduz a produtividade — por exemplo, quando um técnico já resolveu informalmente um problema simples e precisa, ainda assim, abrir um chamado só para que a solução conste no sistema. Esse é um custo real da formalização do atendimento, que deve ser equilibrado com os benefícios de rastreabilidade e priorização que o sistema oferece.

14.7 Troubleshooting: problemas comuns e possíveis soluções

Encerrado o percurso teórico do livro, esta seção consolida, em formato de referência rápida, os problemas mais comumente relatados por um usuário e o roteiro de hipóteses (Seção 12.4) que um técnico deveria seguir para diagnosticá-los — sem repetir a explicação de cada mecanismo, já detalhada no capítulo correspondente.

14.7.1 Computador não liga

Hipótese, em ordem de custo de verificação	Onde aprofundar
Tomada, régua ou disjuntor sem energia	Apêndice A, §A.3
Cabo de força ou fonte com defeito (teste do jumper PS_ON/COM)	Capítulo 11, §11.2.2
Botão do painel frontal ou sua fiação	Capítulo 8, §8.3.2
Mau contato de memória RAM	Capítulo 11, §11.4.3

14.7.2 Liga, mas não dá POST (sem imagem, com ou sem bipes)

Hipótese	Onde aprofundar
Memória RAM ausente, com defeito ou mal encaixada	Capítulo 9, §9.1.1 e §9.1.3
Fonte energiza, mas não sustenta carga real	Capítulo 11, §11.2.2 (nota de atenção)
Speaker de aviso desconectado ou invertido (falso negativo de “sem bipe”)	Capítulo 11, §11.4.3
CPU incompatível com a placa-mãe (soquete/geração)	Capítulo 8, §8.4.1

14.7.3 Trava, reinicia sozinho ou perde desempenho sob carga

Hipótese	Onde aprofundar
Thermal throttling por refrigeração insuficiente ou pasta térmica ressecada	Capítulo 4, §4.4; Capítulo 8, §8.6
Fonte incapaz de sustentar o pico de potência da configuração	Capítulo 13, §13.4
Memória RAM configurada acima da frequência suportada pelo conjunto placa-mãe/processador	Capítulo 5, §5.5
Acúmulo de poeira no dissipador (mais comum em notebooks)	Capítulo 13, §13.5.2

14.7.4 Sem som, Wi-Fi, Bluetooth ou touchpad

Hipótese	Onde aprofundar
Driver ausente ou desatualizado (causa mais provável)	Capítulo 7, §7.6
Isolar hardware vs. software testando em um Live USB	Capítulo 7, §7.3
Dispositivo fisicamente desabilitado no firmware	Capítulo 6, §6.6.2

14.7.5 Computador lento no uso geral

Hipótese	Onde aprofundar
Pouca memória RAM disponível, sistema recorrendo a memória virtual	Capítulo 5, §5.9
Malware em execução em segundo plano	Capítulo 7, §7.8
Armazenamento mecânico (HD) como gargalo do sistema	Capítulo 5, §5.10; Capítulo 13, §13.4
Fragmentação de disco (sistemas com HD)	Capítulo 6, §6.2.2

14.7.6 Perda de dados ou suspeita de arquivo corrompido/apagado

Hipótese	Onde aprofundar
Arquivo apagado, mas cluster ainda não sobrescrito (recuperável)	Capítulo 6, §6.3.1
Backup ausente ou desatualizado no momento da falha	Seção 12.3; Capítulo 7, §7.2.1
Mídia de armazenamento com falha física iminente	Capítulo 5, §5.10 e §5.12

Como usar esta tabela. Cada linha é ordenada, sempre que possível, da hipótese mais barata de testar para a mais cara — o mesmo princípio de priorização apresentado na Seção 12.4. Esta seção não substitui o raciocínio de diagnóstico: é um ponto de partida para gerar hipóteses, não uma lista de respostas prontas a aplicar sem investigação.

14.8 Síntese do capítulo

Este capítulo apresentou a organização do suporte técnico: a central de serviços como ponto único de entrada das demandas, a categorização em evento, incidente e solicitação, a priorização por impacto e urgência, o escalonamento por níveis de suporte com custo crescente, o uso ético do acesso remoto e o funcionamento prático de um sistema de chamados institucional, o SUAP. Fechou com uma referência rápida de troubleshooting, remetendo os problemas mais comuns relatados por usuários de volta aos mecanismos explicados ao longo dos quatro capítulos.

Com isso se encerra a formação teórica do curso. O percurso começou pela definição do computador e pelo princípio de modularidade que estrutura todo diagnóstico técnico (Capítulo 1); avançou para a base elétrica do funcionamento de um computador, com a fonte de alimentação como ponto de partida de qualquer diagnóstico de hardware (Capítulo 11); formalizou a própria manutenção como disciplina, com seu raciocínio de diagnóstico por hipóteses (Capítulo 12); tratou do processador e dos critérios objetivos de avaliação de desempenho por meio de benchmarks (Capítulo 13); e fecha, neste capítulo, com a dimensão de atendimento da manutenção — como organizar, priorizar e conduzir eticamente o suporte ao usuário. Resta ao curso a etapa final, prática: a apresentação dos trabalhos de dimensionamento de computadores para perfis de uso específicos, na qual os conceitos estudados ao longo destes catorze capítulos são aplicados a um caso concreto de especificação de hardware.

14.9 Referências

1. IFRN. “Suap.” Portal IFRN — Tecnologia da Informação. Disponível em: https://portal.ifrn.edu.br/institucional/tecnologia-da-informaca servicos/suap/.
2. VIACERTANATAL. “Instabilidade na Alares gera reclamações e leva clientes a buscar cancelamento em Natal.” 2026. Disponível em: https://www.viacertanatal.com.br/2026/07/instabilidade-na-alares-gera.html; BLOG DO BG. “Clientes da Alares relatam problemas nos serviços de internet há semanas e comemoram chegada da fiscalização do Procon à empresa.” 2026. Disponível em: https://www.blogdobg.com.br/video-clientes-da-alares-relatam-problemas-nos-servicos-de-i (Caso a confirmar pelo autor.)
3. PEOPLECERT. “ITIL (Version 5) explained.” 2026. Disponível em: https://www.peoplecert.org/news-and-announcements/itil-version-5-ex ADVISEDSKILLS. “ITIL (Version 5) in 2026: What is confirmed, what is available, and what comes next.” Disponível em: https://www.advisedskills.com/blog/it-service-management/itil-version-5-in-2026-what-is-confi
4. PORTAL GSTI. “Gerenciamento de Eventos x Gerenciamento de Incidentes da ITIL.” Disponível em: https://www.portalgsti.com.br/2013/01/gerenciamento-de-eventos-x-gerenciamento-de-incidentes-da-i.html.
5. AXELOS/PEOPLECERT. *ITIL 4 Glossary of Terms*. Fonte secundária em português: HDI BRASIL. “Incidentes ou Requisições: como classificar problemas de desempenho de serviços.” Disponível em: https://hdibrasil.com.br/conteudo/incidentes-ou-requisicoes-como-classifica
6. AXELOS/PEOPLECERT. *ITIL 4 Glossary of Terms*.
7. INVGATE. “ITIL® Priority Matrix: How to Build And Use It in Your Service Desk.” Disponível em: https://blog.invgate.com/itil-priority-matrix
8. ONLINE ETYMOLOGY DICTIONARY. “Ticket.” Disponível em: https://www.etymonline.com/word/ticket.
9. TOPDESK. “Understanding service desk tiers: A guide to Tier 0 through Tier 3.” Disponível em: https://www.topdesk.com/en/blog/service-desk-tiers-explained/; INVGATE. “IT Support Levels Explained From Tier 0 to Tier 4.” Disponível em: https://blog.invgate.com/the-5-levels-of-it-support.
10. BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). Disponível em: https://www2.camara.leg.br/legin/fed/lei/2018/lei-13709-14-agosto-2018-787077-publicacaoori.html.
11. GLPI PROJECT. Página oficial. Disponível em: https://glpi-project.org/.
12. Captura de tela do próprio SUAP (evidência primária); IFRN. “Como abrir chamado no Suap?” Disponível em: https://portal.ifrn.edu.br/institucional/tecnologia-da-informacao/tutoriais/suap/como-abrir (Estabilidade da lista de áreas/subcategorias a confirmar pelo autor na data de publicação.)

13. Documentação interna da DIGTI/IFRN sobre SLA por categoria de serviço — não localizada publicamente nesta revisão; faixa a confirmar pelo autor com documentação interna ou captura de tela antes de publicar como fato geral do módulo de TI do SUAP.
14. Base local: MARÇULA, Marcelo; BENINI FILHO, Pio Armando. *Informática: Conceitos e Aplicações*; SCRUM.ORG. *The Scrum Guide*. Disponível em: `https://scrumguides.org/scrum-guide.html`.
15. KANBAN UNIVERSITY / David J. Anderson, Método Kanban; TARGET TEAL. “Método Kanban: um guia (quase) completo.” Disponível em: `https://targetteal.com/blog/metodo-kanban/`.

15 Apêndice A — Fundamentos de Eletricidade

Este apêndice reúne as grandezas elétricas fundamentais (tensão, corrente, resistência e potência), os princípios de proteção e aterramento de uma instalação elétrica residencial, e o funcionamento de transformadores — a base física sobre a qual se apoia o Capítulo 11, que trata da fonte de alimentação do computador propriamente dita.

15.1 A.1 Grandezas elétricas fundamentais

15.1.1 A.1.1 Diferença de potencial (tensão)

Toda análise elétrica de um computador começa na tomada, e toda tomada é, fisicamente, uma **diferença de potencial elétrico** (DDP). O conceito de potencial elétrico pode ser compreendido por analogia com o potencial gravitacional.

O mesmo raciocínio se aplica à eletricidade: a diferença de potencial elétrico só existe entre dois pontos. É por essa razão que toda tomada elétrica possui, no mínimo, dois furos — um representando cada um dos dois pontos entre os quais existe a diferença de potencial. A grandeza que quantifica essa diferença de potencial é chamada de **tensão** ou **voltagem**, medida em **volts** (V).

Figura pendente

diagrama dos pontos A, B e C em mesas empilhadas, com setas indicando as diferenças de potencial gravitacional entre cada par

Quando um caminho condutor é fornecido entre dois pontos de potenciais elétricos diferentes, os elétrons se deslocam da região de maior potencial para a de menor potencial — exatamente como o lápis cai da mesa para o chão, transformando energia potencial gravitacional em energia cinética. Esse movimento de elétrons pode ser aproveitado para realizar trabalho: aquecer uma resistência, acender um LED, ou acionar os transistores que formam os circuitos lógicos de um computador.

Exemplo. Um dispositivo eletrônico é projetado para operar dentro de uma faixa de tensão específica — assim como uma roda-d'água é projetada

para funcionar sob uma queda d'água de determinada altura. Uma roda-d'água dimensionada para uma queda de 2 metros não resiste a uma queda de 20 metros: a estrutura se rompe porque não foi projetada para aquela energia. Da mesma forma, um equipamento projetado para operar em 110 V, ao ser ligado em 220 V, recebe o dobro da tensão para a qual foi dimensionado — o que, pela Lei de Ohm (Seção A.1.3), provoca uma corrente elétrica também maior do que a suportada pelos seus componentes, danificando-os.

15.1.2 A.1.2 Corrente elétrica

Corrente elétrica é o fluxo ordenado de elétrons através de um condutor, medido em **ampere** (A). Historicamente, antes de se compreender que a carga móvel era o elétron (de carga negativa), adotou-se a convenção de que a corrente flui do polo positivo para o polo negativo — o chamado **sentido convencional** da corrente. O **sentido real** do movimento dos elétrons é o oposto: do polo negativo (região com excesso de elétrons) para o polo positivo (região com falta de elétrons). Essa convenção histórica permanece em uso na análise de circuitos até hoje.

15.1.3 A.1.3 Lei de Ohm e resistência elétrica

A relação entre tensão, corrente e resistência é dada pela **primeira Lei de Ohm**:

$$V = R \times I$$

onde V é a tensão (volts), R é a **resistência elétrica** (ohms, Ω) e I é a corrente (amperes). A resistência mede a oposição de um material à passagem de corrente: materiais condutores (como cobre) têm resistência muito baixa; materiais isolantes (como borracha ou ar) têm resistência muito alta.

Exemplo. Em um circuito simples formado por uma fonte de tensão, um resistor e um LED em série, aumentar a tensão da fonte, mantendo a mesma resistência, provoca um aumento proporcional na corrente — e um LED submetido a maior corrente brilha com maior intensidade. Tensão e corrente são, nesse circuito, grandezas diretamente proporcionais: quando uma sobe, a outra sobe na mesma proporção.

Essa relação também explica por que um **curto-circuito** é perigoso. Um curto-circuito ocorre quando um caminho de resistência próxima de zero é criado entre dois pontos de potenciais diferentes — por exemplo, um pequeno objeto condutor (um clipe de papel, um grampo metálico) tocando acidentalmente os dois pinos de uma tomada. Como a corrente é inversamente proporcional à resistência, uma resistência próxima de zero produz uma corrente extremamente alta — teoricamente, o máximo que a fonte for capaz de fornecer. Esse pico de corrente é o que provoca aquecimento excessivo dos condutores, podendo causar incêndio se não houver um dispositivo de proteção interrompendo o circuito (ver Seção A.3.1).

Figura pendente

circuito simples fonte–resistor–LED com seta indicando corrente e legenda “V = R × I”

15.1.4 A.1.4 Corrente contínua e corrente alternada

A corrente elétrica pode ser **contínua** (CC ou, do inglês, DC) ou **alternada** (CA ou AC).

Na **corrente contínua**, os elétrons se deslocam sempre no mesmo sentido, produzida por uma fonte de tensão fixa ao longo do tempo — como uma pilha, cuja diferença de potencial entre os polos permanece constante em, por exemplo, 1,5 V. Representando essa tensão em um gráfico ao longo do tempo, obtém-se uma linha reta e constante.

Na **corrente alternada**, a tensão da fonte varia periodicamente entre valores positivos e negativos, seguindo um comportamento **senoidal**: a tensão sobe até um valor máximo, desce até um valor mínimo (negativo) e repete esse ciclo indefinidamente. Como consequência, os elétrons não se deslocam sempre no mesmo sentido — eles oscilam, ora em uma direção, ora na direção oposta, tantas vezes por segundo quanto for a **frequência** da onda. A tomada residencial no Rio Grande do Norte fornece uma tensão alternada de 220 V (valor eficaz, ou RMS) a 60 Hz [1] — ou seja, o ciclo de oscilação se repete 60 vezes por segundo.

Exemplo. O funcionamento de um chuveiro elétrico não depende do sentido em que os elétrons se movem: o simples ir e vir do elétron, induzido pela tensão alternada, já é suficiente para aquecer a resistência do chuveiro e, consequentemente, a água. O trabalho realizado (aquecimento) não exige que a corrente seja contínua.

A razão pela qual a energia elétrica é gerada, transportada e distribuída na forma alternada — e não contínua — está diretamente ligada à minimização de perdas no transporte, tratada em detalhe na Seção A.2.

15.2 A.2 Potência elétrica e efeito Joule

Potência elétrica é a capacidade de um dispositivo realizar trabalho por unidade de tempo, medida em **watts** (W). É calculada como o produto entre tensão e corrente:

$$P = V \times I$$

Combinando essa equação com a Lei de Ohm ($V = R \times I$), obtém-se uma segunda forma equivalente:

$$P = R \times I^2$$

Essa segunda forma é especialmente importante porque descreve a **potência dissipada por efeito Joule** — o calor gerado pela passagem de corrente através de um condutor com resistência. Quanto maior a corrente que passa por um condutor, maior é a potência perdida em forma de calor, e essa perda cresce com o **quadrado** da corrente.

Exemplo. Um chuveiro elétrico ligado em 220 V, puxando uma corrente da ordem de 5 A, tem potência de $220 \times 5 = 1100$ W. Um carregador de celular, ligado na mesma tomada de 220 V mas puxando apenas cerca de 0,01 A, tem potência de $220 \times 0{,}01 = 2{,}2$ W. Ambos os dispositivos estão sujeitos à mesma tensão da tomada; o que os diferencia é a corrente que cada um demanda — e essa diferença de corrente é o que torna o chuveiro elétrico centenas de vezes mais potente que o carregador.

15.2.1 Por que a energia é transportada em alta tensão

A rede elétrica gera energia em uma determinada tensão e, antes de transportá-la por longas distâncias até os centros consumidores, eleva essa tensão para valores muito mais altos (por meio de transformadores, tratados na Seção A.5), reduzindo-a de volta a 220 V apenas próximo ao ponto de consumo. Essa escolha de projeto decorre diretamente da equação $P = R \times I^2$: para transportar uma mesma potência, é possível usar uma tensão alta com corrente baixa, ou uma tensão baixa com corrente alta. Como a perda por efeito Joule cresce com o quadrado da corrente, transportar a energia com uma corrente menor (elevando a tensão) reduz drasticamente as perdas ao longo dos cabos de transmissão — o que explica também o zumbido característico dos transformadores de poste, resultado da conversão eletromagnética de tensão em andamento.

Figura pendente

diagrama simplificado da rede elétrica — geração, elevação de tensão para transporte, redução de tensão em subestações e no poste, chegada em 220 V à residência

---

15.3 A.3 Proteção da instalação elétrica

Normas aplicáveis. O conteúdo desta seção e da Seção A.4 (disjuntores, dimensionamento de condutores, aterramento) segue as prescrições das normas técnicas brasileiras de instalações elétricas de baixa tensão e de segurança em instalações elétricas [2].

15.3.1 A.3.1 Disjuntores

Um **disjuntor** é um dispositivo eletromecânico que interrompe automaticamente a passagem de corrente elétrica quando ela ultrapassa um determinado limite. Seu funcionamento se baseia no fato de que toda corrente elétrica alternada produz um campo magnético proporcional à sua intensidade: dentro do disjuntor existe um sensor sensível a esse campo magnético; quando a corrente excede o limite para o qual o disjuntor foi dimensionado (por exemplo, 10 A), o campo magnético resultante aciona um mecanismo de mola que desarma o disjuntor, desconectando mecanicamente o circuito — produzindo o característico estalo audível.

Uma instalação elétrica é organizada hierarquicamente: da rede pública, a energia chega a um **quadro geral**, de onde é distribuída para um ou mais **quadros de distribuição**, cada um atendendo a um setor específico da edificação (um andar, uma sala). Cada circuito de um quadro de distribuição — iluminação, tomadas, ar-condicionado — é protegido por seu próprio disjuntor, de forma que uma falha em um circuito não derrube os demais. Esse isolamento por circuito é análogo ao encapsulamento em programação orientada a objetos: cada objeto (aqui, cada sala ou circuito) mantém seus próprios atributos e falhas isoladas do restante do sistema.

A instalação pode ser **monofásica** (uma única fase de 220 V mais um neutro de referência) ou **trifásica** (três fases de 220 V, cada uma referenciada ao mesmo neutro, usadas para distribuir cargas maiores entre três circuitos independentes).

Figura pendente

hierarquia poste → quadro geral → quadros de distribuição → disjuntores individuais por circuito

15.3.2 A.3.2 Dimensionamento de condutores

Cada condutor (fio) elétrico suporta uma corrente máxima, determinada pela sua **bitola** (área de seção transversal). Quanto maior a bitola do fio, maior a corrente que ele pode conduzir sem aquecer excessivamente.

O disjuntor de um circuito deve ser dimensionado de acordo com a capacidade do fio que ele protege: se um fio suporta no máximo 10 A, o disjuntor daquele circuito deve desarmar antes que essa corrente seja ultrapassada. Uma causa comum de disjuntores desarmando repetidamente é a **sobrecarga**: ligar, em um mesmo circuito, equipamentos cuja soma de potências excede a capacidade projetada para aquele fio e aquele disjuntor — por exemplo, uma impressora de alto consumo e uma pistola de cola quente compartilhando a mesma tomada. Quando essa sobrecarga não é interrompida por um disjuntor subdimensionado (ou por um disjuntor trocado por um de corrente nominal mais alta sem trocar o fio correspondente), o próprio condutor pode superaquecer e provocar um curto-circuito ou incêndio.

15.4 A.4 Aterramento e segurança elétrica

15.4.1 A.4.1 O planeta Terra como referência de potencial

O terceiro pino presente na maioria das tomadas modernas corresponde ao **aterramento** (terra). O planeta Terra, por sua imensa massa e extensão, funciona como uma fonte praticamente infinita de cargas elétricas: ele pode absorver um excesso de cargas de qualquer ponto conectado a ele, ou fornecer cargas a um ponto que esteja com déficit, sempre tendendo ao equilíbrio elétrico.

O **ar** é naturalmente isolante — seus átomos mantêm os elétrons presos com energia suficiente para impedir a condução em condições normais — o que explica por que é seguro aproximar a mão de um fio energizado sem tocá-lo diretamente. Quando, porém, a diferença de potencial entre dois pontos se torna extrema (como entre uma nuvem carregada e o solo), a rigidez elétrica do ar é rompida, e ele passa a conduzir: esse fenômeno é o **raio**. O trajeto irregular de um raio decorre do fato de que a descarga segue o caminho de menor resistência entre as moléculas de ar naquele instante. Estruturas pontiagudas, como para-raios, favorecem a descarga por um efeito conhecido como **poder das pontas**: a geometria fina concentra o campo elétrico, oferecendo um caminho de menor resistência para a corrente até o fio terra.

Figura pendente

ilustração do trajeto irregular de um raio e de um para-raios conduzindo a descarga até o fio terra

Exemplo. Um carregador ou notebook com gabinete metálico, ligado a um cabo de alimentação de apenas dois pinos (sem aterramento), pode transmitir ao usuário uma leve sensação de formigamento ao ser tocado enquanto carrega. Essa sensação ocorre porque um pequeno vazamento de corrente se acumula no chassi metálico e, na ausência de um caminho de aterramento de baixa resistência, o próprio corpo do usuário se torna o caminho disponível para o escoamento dessas cargas até a terra.

15.4.2 A.4.2 Instalação de aterramento e o papel da concessionária

Em uma instalação elétrica residencial, os condutores de **fase** e **neutro** são fornecidos pela concessionária de energia (no Rio Grande do Norte, a Cosern): problemas nesses dois condutores são de responsabilidade dela. O condutor de **terra**, por outro lado, é de responsabilidade da instalação elétrica local — tipicamente executado com hastes metálicas cravadas no solo, muitas vezes aproveitando o próprio ferro da fundação da edificação,

conectadas por fio a uma resistência de aterramento baixa o suficiente para escoar cargas com eficiência.

Como a tomada é uma diferença de potencial alternada, não existe, tecnicamente, um lado “certo” e um lado “errado” entre fase e neutro do ponto de vista da física básica — mas, na prática, muitos equipamentos e normas de instalação definem um lado correto para cada função, e essa correspondência deve ser verificada com instrumentos apropriados, nunca presumida.

Atenção. Um erro de instalação — por exemplo, o condutor de fase sendo conectado, por engano, ao ponto que deveria ser o terra — pode fazer com que toda a fiação de aterramento de uma edificação fique energizada em 220 V. Nessa condição, qualquer equipamento aterrado (como uma geladeira) torna-se, ele próprio, uma fonte de choque ao ser tocado. Por essa razão, um técnico nunca deve presumir que uma instalação elétrica está correta: a tomada deve ser testada com um multímetro antes de qualquer intervenção, identificando corretamente fase, neutro e terra.

15.4.3 A.4.3 Choque elétrico e o chuveiro elétrico

O choque elétrico é perigoso porque a corrente elétrica que atravessa o corpo humano pode interferir nos impulsos nervosos que controlam a musculatura — inclusive o músculo cardíaco. Uma corrente elétrica no momento errado pode desorganizar o ritmo cardíaco, causando uma disfunção potencialmente fatal.

Atenção. Antes de qualquer manuseio de fiação elétrica, energia deve ser desligada na fonte (disjuntor) e, sempre que possível, confirmada como desligada com um instrumento de medição. Nunca se deve presumir que um circuito está desenergizado apenas porque um interruptor foi acionado.

O **chuveiro elétrico** — uma invenção brasileira [3] — costuma causar estranhamento em visitantes de países onde o aquecimento de água é feito por gás ou por reservatórios térmicos (*boilers*), a ponto de gerar receio em relação ao seu uso. Esse receio é, tecnicamente, infundado: o chuveiro elétrico não estabelece contato elétrico com a água. A corrente elétrica atravessa apenas a resistência interna do aparelho, que se aquece por efeito Joule; a água, ao passar por essa resistência já aquecida, absorve o calor por condução térmica, sem que corrente elétrica normalmente flua através dela. Um choque em um chuveiro elétrico só ocorre diante de uma falha grave de isolamento ou de instalação — não é uma característica intrínseca do funcionamento do equipamento.

15.5 A.5 Transformadores e transporte de energia

Um **transformador** é o dispositivo responsável por elevar ou reduzir uma tensão alternada, sendo o elemento central tanto no transporte de energia

em alta tensão (Seção A.2) quanto no primeiro estágio de uma fonte de alimentação linear (Seção A.6).

Seu princípio de funcionamento se baseia na **Lei de Faraday**: uma corrente elétrica variável produz um campo magnético; um campo magnético variável, por sua vez, induz uma corrente elétrica em um condutor próximo. Um transformador é fisicamente constituído por dois enrolamentos de fio (bobinas) — o primário e o secundário — em torno de um núcleo ferromagnético comum, sem contato elétrico direto entre eles. A corrente alternada que circula pela bobina primária gera um campo magnético variável no núcleo, que induz uma corrente na bobina secundária.

A relação entre a tensão de entrada e a tensão de saída é determinada pela razão entre o **número de espiras** (voltas do fio) em cada bobina: se o secundário possui mais espiras do que o primário, a tensão é elevada e a corrente correspondentemente reduzida (para conservar a potência); se possui menos espiras, a tensão é reduzida e a corrente aumentada.

Como a energia não se perde nessa conversão (idealmente), o produto tensão × corrente se mantém aproximadamente constante entre primário e secundário: elevar a tensão implica reduzir a corrente na mesma proporção, e vice-versa. É exatamente essa propriedade que permite elevar a tensão da rede elétrica para transporte de longa distância (reduzindo a corrente e, com ela, as perdas por efeito Joule) e depois reduzi-la de volta a 220 V para uso residencial — passando antes por transformadores de subestação e, por fim, pelos transformadores de poste que abaixam a tensão para o nível final entregue às residências.

Como o funcionamento do transformador depende de variação de campo magnético, ele **exige corrente alternada** para operar: não é possível usar um transformador diretamente sobre uma tensão contínua constante, pois esta não gera a variação de campo magnético necessária à indução.

Figura pendente

diagrama de um transformador com bobina primária e secundária em torno de um núcleo ferromagnético, indicando número de espiras e tensões de entrada/saída

---

15.6 A.6 Da corrente alternada à contínua: fontes lineares

Todos os componentes internos de um computador — processador, memória, circuitos lógicos — operam com **corrente contínua**. A função central de qualquer fonte de alimentação é, portanto, converter a corrente alternada da tomada em corrente contínua nos níveis de tensão exigidos por cada componente.

O modelo mais simples de conversão é a **fonte linear**, construída por meio de quatro estágios em sequência:

1. **Transformador** — reduz a tensão alternada de entrada (por exemplo, 220 V) para uma tensão alternada de menor amplitude (por exemplo, 10 V), conforme descrito na Seção A.5.
2. **Retificador** (ponte de diodos) — inverte a parcela negativa da onda alternada para o lado positivo, já que um diodo permite a passagem de corrente em apenas um sentido. Um único diodo eliminaria completamente a metade negativa da onda; uma ponte de diodos "rebate" essa metade negativa para cima, produzindo uma onda inteiramente positiva, ainda que pulsante.
3. **Filtro capacitivo** — um capacitor de grande capacitância, ligado em paralelo com a carga, suaviza as oscilações (o chamado *ripple*) da onda retificada, aproximando-a de uma tensão contínua estável.
4. **Regulador de tensão** — ajusta e estabiliza a tensão final na saída.

Fontes lineares são robustas e relativamente simples de projetar, mas apresentam uma desvantagem física significativa: para uma mesma potência, seus componentes (especialmente o transformador) são consideravelmente maiores e mais pesados do que os de uma fonte chaveada equivalente. Por essa razão, fontes lineares praticamente não são mais utilizadas em computadores modernos, restringindo-se a aplicações específicas, como equipamentos de áudio de alta fidelidade, em que a característica do circuito linear é tecnicamente desejável.

Figura pendente

diagrama de blocos de uma fonte linear — transformador → ponte de diodos → filtro capacitivo → regulador — com a forma de onda em cada estágio

---

15.7 Síntese do apêndice

Este apêndice apresentou as grandezas elétricas fundamentais — tensão, corrente, resistência e potência —, os princípios de proteção por disjuntores e dimensionamento de condutores, o aterramento e a segurança contra choque elétrico, e o funcionamento de transformadores e circuitos retificadores até a fonte linear. Esses fundamentos sustentam diretamente o Capítulo 11, que trata da fonte chaveada moderna — a fonte ATX — e de sua arquitetura, seus sinais de controle e sua metodologia de dimensionamento e diagnóstico.

15.8 Referências

1. NEOENERGIA COSERN. “Normas Técnicas — Padrão de Entrada de Energia.” Disponível em: https://www.neoenergia.com/web/rn/normas-tecni
2. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 5410: Instalações elétricas de baixa tensão*. ABNT. (Edição a confirmar pelo autor.) BRASIL. Ministério do Trabalho e Emprego. *NR-10: Segurança em Instalações e Serviços em Eletricidade*. (Edição a confirmar pelo autor.)
3. Engenharia 360. “O inventor brasileiro do chuveiro elétrico.” Disponível em: https://engenharia360.com/o-inventor-brasileiro-do-chuveiro-e WIKIPÉDIA. “Chuveiro elétrico.” Disponível em: https://pt.wikipedia.org/wiki/Chuveiro_el%C3%A9trico.

Texto de contracapa pendente

Definir o texto de apresentação/sinopse do livro (metadado `quarta-capa` no YAML) para a quarta capa.