Organização e Montagem de Computadores

Livro 1 — versão 0.1

João Paulo Ferreira Guimarães

4 de agosto de 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

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.

Figura pendente

linha do tempo dos dispositivos de cálculo — ábaco, ossos de Napier, régua de cálculo, calculadora mecânica Curta

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.

Analogia. A diferença entre uma calculadora comum e um computador com programa armazenado pode ser ilustrada pela comparação entre uma calculadora de bolso e uma planilha eletrônica: na calculadora, cada operação deve ser inserida manualmente pelo usuário; na planilha, uma fórmula é armazenada uma única vez e reaplicada automaticamente sempre que os dados de entrada mudam.

O conceito de programa armazenado é a base da **arquitetura de von Neumann**, tratada em profundidade no Capítulo 5.

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.

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 so-

mente 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).

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.

Analogia. Esse procedimento é equivalente a testar o pneu de um carro com defeito trocando-o pelo pneu de um carro em pleno funcionamento: se o carro com defeito continuar apresentando o mesmo problema, conclui-se que a peça testada não era a causa, e outro componente deve ser investigado.

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

Ano	Declaração	Autor/fonte
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.
- **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 pendente

foto do IBM PC original de 1981

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 4 e na disciplina de Manutenção de Computadores).
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; 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.

Analogia. O funcionamento dessa hierarquia pode ser comparado ao trabalho de um padeiro: o armazenamento secundário corresponde à despensa, onde os ingredientes ficam guardados por longos períodos; a memória RAM corresponde à mesa de trabalho, onde os ingredientes necessários no momento são reunidos; os registradores e a memória cache correspondem às próprias mãos do padeiro, mantendo o essencial em acesso imediato.

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 4, §4.1), 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 4, §4.10, trata do chipset e da interconexão desses componentes). Praticamente todo smartphone, tablet e Raspberry Pi é organizado em torno de um SoC.

Analogia. A diferença entre esses processadores é comparável à de uma cozinha profissional: a CPU é o cozinheiro generalista, capaz de preparar qualquer prato do cardápio, ainda que sem a velocidade de um especialista; a GPU é o cozinheiro que só faz uma tarefa (por exemplo, grelhar) mas prepara centenas de porções simultaneamente; o DSP é o cozinheiro que só sabe temperar, mas o faz de forma extremamente rápida e consistente; a NPU é o cozinheiro treinado especificamente para reconhecer, por experiência acumulada, qual tempero combina com qual prato. O SoC, nessa analogia, não é mais um cozinheiro — é a decisão de montar a cozinha inteira dentro de um único módulo compacto, em vez de espalhar fogão, geladeira e bancada em cômodos separados.

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 4, §4.6)	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 2, §2.7)	Módulos substituíveis ou soldados, conforme o modelo	Soldada/empacotada junto ao SoC	Módulos ECC (ver adiante), muitos slots, capacidade na casa de centenas de GB a TB
Alimentação	Fonte ATX interna (Capítulo 4, §4.1)	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), e placas-mãe capazes de hospedar mais de um processador físico simultaneamente.

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 [7] — aponta **duas** causas físicas bem estabelecidas, ambas formas de radiação ionizante, hoje normatizadas por um padrão da indústria de testes de memória [8]:

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 (Capítulo 2, §2.3) ou de um flip-flop de SRAM (Capítulo 2, §2.2).
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 (Capítulo 2, §2.10 e §2.14). 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 livro de Manutenção de Computadores trata da litografia em profundidade) 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.

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 2), sistema operacional e instalação (Capítulo 3), hardware físico e montagem (Capítulo 4) e arquitetura de processadores (Capítulo 5).

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. 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.
8. 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.

2 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.

2.1 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.

Figura pendente

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

2.2 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 2.3), 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 2.8), 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.

Analogia. A memória estática funciona como um objeto colocado sobre uma mesa: uma vez posicionado, ele permanece ali sem exigir nenhuma ação adicional para se manter no lugar.

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 2.8).

2.3 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.

Analogia. Guardar um bit num capacitor é como guardar um "1 lógico" na forma de água dentro de um copo: se o copo for deixado exposto, a água evapora lentamente e, depois de algum tempo, o "combinado" original (copo cheio = 1) deixa de valer. Para que o dado se mantenha, é preciso completar o copo periodicamente.

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.

Analogia. A relação entre densidade de armazenamento e ocupação de espaço é comparável à densidade populacional: assim como o litoral brasileiro concentra mais pessoas por quilômetro quadrado do que o interior, uma tecnologia de memória mais densa concentra mais bits por milímetro quadrado de chip — reduzindo o custo de produção por unidade de dado armazenado.

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.

2.4 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

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].

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.

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 2.6).

Figura pendente

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

2.5 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.

2.6 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

2.7 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.

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.

2.8 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.

Analogia. A operação de cache é equivalente a um ajudante de obra que, ao perceber que o pedreiro está prestes a precisar de martelo e pregos, já traz os dois de uma vez e os deixa próximos, em vez de esperar cada pedido individual e ir buscar um item de cada vez.

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 2.2), 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

2.9 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 2.11.

2.10 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.

Analogia. O funcionamento é equivalente a uma fechadura eletromagnética de porta: enquanto há corrente, o eletroímã segura a porta fechada; ao

cortar a energia, o campo desaparece e a porta libera. Da mesma forma, a cabeça do HD usa corrente elétrica para gerar (ou não) o campo magnético que orienta cada bit no prato.

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.

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 2.12), 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.

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

2.11 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 3.

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 3).

Analogia. Quando um arquivo é "apagado", ele geralmente não é sobrescrito de imediato — apenas seus clusters são marcados como disponíveis nos metadados, de forma equivalente a um hotel liberando um quarto após o check-out: o quarto está disponível para o próximo hóspede, mas um objeto esquecido pelo hóspede anterior (como um carregador de celular) ainda está fisicamente lá até que alguém o remova ou o quarto seja reocupado. É justamente por isso que existem técnicas de recuperação de arquivos apagados: enquanto os clusters correspondentes não forem sobrescritos por um novo dado, a informação original ainda pode ser resgatada.

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 anunciado com "16 GB" aparece, portanto, com uma capacidade ligeiramente menor quando visualizado pelo sistema operacional.

2.12 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*).

Analogia. A porta flutuante é isolada eletricamente por duas camadas finíssimas de óxido, formando uma espécie de sanduíche: assim como o recheio de um hambúrguer fica isolado entre dois pães, os elétrons armazenados na porta flutuante ficam retidos entre as duas camadas isolantes de óxido.

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 2.9), 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.

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 2.8.

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 4), 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

2.13 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 superfície 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].

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.

2.14 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

---

2.15 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), e as três grandes famílias de armazenamento secundário — magnética, flash e ótica —, cada uma com vantagens e fragilidades próprias, culminando na discussão de estratégias de backup e redundância. Esses conceitos formam a base necessária para o Capítulo 3, 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 2.11 — são, na prática, a camada de software que organiza e dá sentido à memória secundária estudada neste capítulo.

2.16 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-`

3 Sistema operacional, sistemas de arquivo e instalação

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; o procedimento de inicialização do computador, do POST ao carregamento do sistema operacional; e o processo completo de instalação de um sistema operacional, incluindo backup, particionamento avançado, instalação em dual boot e instalação de drivers.

3.1 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.

Analogia. Uma calçada não é a rua nem é a casa: é o elemento que fica entre as duas, permitindo que se passe de um espaço para o outro. Da mesma forma, o sistema operacional não é o hardware nem é o aplicativo que o usuário está utilizando — ele é a camada que está entre os dois, viabilizando a comunicação.

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

3.1.1 Abstração e plataforma

O sistema operacional cria uma **abstração** sobre o hardware: 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 — por exemplo, para emitir um som ou desenhar uma janela — e é o sistema operacional quem trata da comunicação efetiva com aquele hardware específico.

Esse ambiente para o qual um software é desenvolvido é chamado de **plataforma**. Um programa feito para a plataforma Windows não roda nativamente em outra plataforma, a menos que exista uma versão da ferramenta usada (por exemplo, um banco de dados) disponível também para essa outra plataforma — 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: a possibilidade de executar **múltiplos programas ao mesmo tempo**, alternando (fazendo o chaveamento) entre eles os recursos de hardware disponíveis.

3.1.2 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.

3.2 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**.

Analogia. O cluster pode ser entendido como uma gaveta: o sistema operacional não guarda dados em posições arbitrárias do disco, mas em unidades fixas — as gavetas —, cada uma com um tamanho definido no momento da formatação.

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

3.2.1 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.

3.2.2 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

Cenário	Cluster	Arquivo	Efeito
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.

3.3 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.

3.3.1 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 (Seção 3.9).

Figura pendente

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

---

3.4 Partições: divisão lógica do disco

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

Analogia. A fronteira entre Brasil e Paraguai não corresponde a nenhum traço físico no terreno; é uma convenção entre duas nações. Da mesma forma, um único disco pode ser "dividido" em duas ou mais partições sem que exista qualquer separação física real — apenas um combinado, registrado em metadados, de que até determinado ponto o espaço pertence a uma partição e, a partir dali, a outra.

Analogia. Um único armário pode ter suas gavetas divididas por convenção entre dois usuários — "estas gavetas são suas, estas são minhas" — sem que o armário seja fisicamente serrado ao meio. O mesmo raciocínio se aplica a um disco dividido em partições.

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.

3.4.1 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 na Seção 3.11.

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

3.5 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**.

Analogia. A tabela de partições funciona como a capa e o sumário de um livro: reúne, num espaço fixo no início do disco, a informação de "para qual capítulo (partição) ir e onde ele começa".

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

Analogia. A relação entre MBR e GPT é comparável à relação entre um aparelho de ar-condicionado de janela, antigo e volumoso, e um aparelho split moderno: ambos cumprem a mesma função básica, mas o mais recente resolve limitações técnicas do mais antigo.

3.5.1 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.

3.5.2 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.

Analogia. A retrocompatibilidade do GPT em relação ao MBR é equivalente à de um console de videogame mais recente que ainda executa jogos de gerações anteriores — como o PlayStation 5, capaz de rodar a maior parte dos jogos de PlayStation 4, ou o Nintendo Switch 2, compatível com os cartuchos do Switch original. O console mais antigo, por outro lado, não consegue executar jogos feitos exclusivamente para o mais novo.

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

3.6 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.

3.6.1 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.

Analogia. O POST equivale à triagem de um pronto-socorro: um exame rápido que determina se o paciente (o hardware) está em condições minimamente operacionais antes de qualquer procedimento seguinte.

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).

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.

3.6.2 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.

Analogia. Se o computador fosse comparado a um jogo, o Setup seria o menu de configurações desse jogo.

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.

3.6.3 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 memória apresentado no Capítulo 1 (Seção 1.10) e aprofundado no Capítulo 2.

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 3.6.2). 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

3.6.4 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 (Seção 3.9) —, 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.

3.7 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.

Esse processo exige, ele próprio, criar uma tabela de partições e um sistema de arquivo válidos no pendrive, copiando e organizando os arquivos de instalação de acordo com esse esquema.

Uma ferramenta recomendada para essa tarefa é o **Rufus**, software gratuito, de código aberto e leve, com interface gráfica simples para essa finalidade [6]. Ao gravar uma mídia de instalação com o Rufus, é preciso escolher dois parâmetros centrais:

- O **esquema de partição**: MBR ou GPT.
- O **sistema de destino**: BIOS (*legacy*) ou UEFI.

Como o UEFI é retrocompatível com o BIOS (Seção 3.5.2), 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

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.

Figura pendente

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

3.8 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 (Seção 3.6.3) e direcioná-lo para a mídia de instalação (Seção 3.7), 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 (Seção 3.3) 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 3.8.2).

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.

3.8.1 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 — tarefa aprofundada na disciplina de Manutenção de Computadores.

3.8.2 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 (Seção 3.6.3). 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”

3.9 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 na Seção 3.6.4: 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)

3.10 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 2. O dimensionamento da partição swap é função da quantidade de RAM instalada na máquina — não um valor fixo independente dela [7]; 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 /

3.11 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 na Seção 3.4.1.

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 na Seção 3.5.1) sem preservar um gerenciador de boot alternativo ali presente [8]. 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

3.12 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

---

3.13 Atualização do sistema operacional e do firmware

3.13.1 Atualização do sistema operacional

Um sistema operacional instalado (Seção 3.8) 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 3.11, a propósito do GRUB). Do ponto de vista do usuário, a rotina de manutenção preventiva de software (aprofundada na disciplina de Manutenção de Computadores) inclui manter essas atualizações em dia — a maior parte das infecções por malware (Seção 3.8.1) 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 3.12) — 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 3.8.1), da mesma forma que uma reinstalação completa.

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

A Seção 3.6 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 2, §2.12): 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 2 (§2.12) — 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, Seção 3.6.2) 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 4, §4.6, 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 4, §4.3). 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 do livro de Manutenção de Computadores) e nunca deve ser interrompida manualmente.

3.13.3 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 4, §4.3.4), 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 4, §4.10.2) 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 2, §2.12). 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.

3.13.4 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 3.13.3) 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

---

3.14 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 2 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, o procedimento completo de inicialização do computador (POST, Setup/BIOS e boot) e 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, e a redundância de hardware que o BIOS duplo oferece contra esse risco. Esses procedimentos dependem diretamente dos componentes físicos tratados no Capítulo 4 — a placa-mãe, seus barramentos e os próprios dispositivos de armazenamento — e do firmware gravado nesses componentes, cuja relação com a arquitetura de processadores será aprofundada no Capítulo 5.

3.15 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`.
6. RUFUS. Página oficial. Disponível em: `https://rufus.ie/`; repositório de código-fonte: `https://github.com/pbatard/rufus`.
7. 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`.
8. UBUNTU COMMUNITY HELP WIKI. “WindowsDualBoot.” Disponível em: `https://help.ubuntu.com/community/WindowsDualBoot`.

4 Hardware físico e montagem

Neste capítulo você vai estudar os componentes físicos de um computador desktop e o procedimento de desmontagem e remontagem do gabinete, o funcionamento do POST sob a ótica dos componentes de hardware que ele verifica, o papel da bateria CMOS e do painel frontal, os critérios de compatibilidade entre processador e placa-mãe (fabricante, soquete e geração), e os sistemas de refrigeração a ar e a líquido, incluindo a função física da pasta térmica e o procedimento de troca de CPU.

4.1 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
Processador (CPU)	Executa as instruções dos programas	Sim

Submódulo	Função	Essencial para ligar?
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)

Analogia. O princípio de modularidade que rege essa tabela é o mesmo de um ônibus, cujos submódulos — motor, parte elétrica, pneus, suspensão, bancos — podem ser removidos e substituídos individualmente: a falha de um deles interrompe o funcionamento do todo, mas a substituição pelo módulo equivalente o restaura. O mesmo vale para um liquidificador doméstico: ele precisa de energia, do copo e da tampa para funcionar com segurança; a ausência da tampa não impede o funcionamento, mas é dispensável apenas nesse sentido restrito — o correto, ao perder um desses módulos, é substituí-lo, e não descartar o aparelho inteiro.

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

4.1.1 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.

4.2 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 4.1.1.

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 4.6 a 4.8).

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

4.2.1 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 4.1 e 4.6) —, uma placa de expansão (GPU, placa de captura, placa de rede adicional) se conecta a um dos slots PCIe descritos na Seção 4.10.4, 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 livro de Manutenção de Computadores, Capítulo 2, §2.10).

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 4.2), só se encaixa fisicamente numa única orientação.

4.3 O POST sob a ótica dos componentes de hardware

O Capítulo 3 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.

4.3.1 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.
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.

Analogia. O POST pode ser comparado a um exame de sangue: assim como um médico infere, a partir de indicadores como a contagem de glóbulos brancos, se há uma infecção em curso, o técnico infere, a partir do resultado do POST, se os componentes vitais do computador estão operacionais. É por isso que o ato de o computador exibir a tela inicial do POST é popularmente chamado de "**dar tela**" ou "*dar post*".

4.3.2 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 na Seção 4.5): dispensável ao POST, embora seja o meio usual de acionar a energização do sistema.
- **Bateria CMOS** (tratada na Seção 4.4): não impede o POST, mas afeta a retenção de configurações.

4.3.3 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.

4.3.4 Metodologia de diagnóstico

A ausência de POST indica, necessariamente, uma falha em um dos quatro componentes vitais listados na Seção 4.3.1 — nunca em memória secundária, periféricos ou rede. O procedimento de diagnóstico sistemático (a ser aprofundado na disciplina de Manutenção de Computadores) 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

4.4 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

4.5 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.

4.5.1 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.

4.5.2 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

4.6 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 5.

4.6.1 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].

Analogia. A abordagem da Intel para soquetes pode ser comparada a um cenário hipotético em que cada geração de um smartphone tivesse seu próprio carregador, incompatível com o carregador da geração anterior.

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.

4.6.2 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

4.6.3 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 5.

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

4.7.1 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.

Analogia. O funcionamento do dissipador pode ser comparado ao ato de espalhar um alimento quente (como um brigadeiro) em um prato: ao aumentar a área de contato do alimento com o ar, a troca de calor é acelerada. O mesmo princípio — maior área de superfície, maior taxa de troca térmica — está por trás das aletas de um dissipador: a mesma massa de material metálico, com muito mais superfície exposta ao ar, dissipa calor de forma muito mais eficiente do que um bloco compacto.

4.7.2 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.

Analogia. A mesma lógica está presente na refrigeração de motores automotivos: o Fusca e seus derivados (como a Kombi e os buggies) utilizavam refrigeração a ar, o que explica seu som característico de motor; carros modernos, por gerarem mais calor e exigirem dissipação mais eficiente, utilizam refrigeração líquida com radiador.

4.7.3 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)

4.8 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 4.7.2, 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.

4.8.1 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 4.6.2) 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 proces-

sador 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

---

4.9 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 4.1), como ela se relaciona com o soquete do processador (Seção 4.6). Esta seção trata da placa-mãe como objeto físico: seu tamanho padronizado e os pequenos componentes de configuração manual que ela expõe.

4.9.1 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 2 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

Fator de forma	Dimensões aproximadas [5]	Slots de expansão (observação de mercado, não parte da especificação)	Uso típico
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 (Seção 4.10.4), 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

4.9.2 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 (Seção 4.10.4).

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 (Seção 4.10.4, adiante), 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 (Seção 4.10.2), 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.

4.9.3 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. A Seção 4.4 apresentou 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 4.5) e dos conectores de alimentação do processador (livro de Manutenção de Computadores, Capítulo 2, §2.11).

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 (Seção 4.10.5) — 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

4.10 Barramentos: do PCI ao PCIe

4.10.1 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 4.10.4, 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 4.10.5.

Figura pendente

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

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

O **chipset** — já mencionado no livro de Manutenção de Computadores (Capítulo 2, §2.11) 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) [6], 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 4.10.5), áudio, rede, o chip de BIOS/UEFI (Capítulo 3, §3.6) e os demais slots PCI/PCIe de expansão.

Analogia. A relação entre ponte norte e ponte sul é comparável à hierarquia de uma empresa de logística: a ponte norte é o centro de distribuição regional, conectado por uma rodovia expressa diretamente à fábrica (o processador) e capaz de atender rapidamente os poucos clientes de maior volume (memória RAM, vídeo); a ponte sul é o centro de distribuição local, que recebe carga do centro regional e a redistribui para o grande número de pequenos destinos (USB, SATA, áudio, rede) — cada um exigindo menos velocidade individual, mas em maior quantidade.

4.10.3 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 2 trata a RAM em profundidade) e, em processadores mais recentes, um conjunto próprio de **pistas PCIe** (Seção 4.10.4, adiante) dedicadas à placa de vídeo e ao armazenamento NVMe (Seção 4.10.5).

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 [7]. 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 5) 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)

4.10.4 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 [8]:

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 4.10.1), cada slot PCIe estabelece uma conexão **ponto a ponto** exclusiva com o controlador — não é, tecnicamente, um "barramento" no sentido estrito da Seção 4.10.1, 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 4.10.5) 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 [9], mantendo o mesmo princípio físico de conexão — um padrão de evolução comparável ao da família DDR de memória (Capítulo 2, §2.4).

Figura pendente

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

4.10.5 SATA e NVMe: interfaces de armazenamento

A Seção 4.1.1 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 4.10.1, da migração geral de interfaces paralelas para seriais. O SATA usa seu próprio controlador (parte da ponte sul, Seção 4.10.2) e seu próprio protocolo de comunicação, projetado nos anos 2000 [10] tendo o HD mecânico (Capítulo 2, §2.10) 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 2, §2.12), 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 4.10.3) — 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 (Seção 4.1.1) — 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) [11], 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.

4.11 Entrada, saída e periféricos

4.11.1 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 3, §3.1.1).

4.11.2 Estudo de caso: CPU e memória secundária

A comunicação entre a CPU e um HD ou SSD (Capítulo 2) passa por uma controladora que desempenha quatro funções:

1. **Conexão física** — o conector elétrico propriamente dito (SATA ou NVMe, Seção 4.10.5).
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 2, §2.10).
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.

4.11.3 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 3, §3.6); atualmente, a pilha é mais sofisticada e passa por três camadas, de baixo para cima: o **chipset** (Seção 4.10.2), que recebe o sinal elétrico bruto do dispositivo; o **driver** (Capítulo 3, §3.12), 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.

4.11.4 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 [12]:

- **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.

Analogia. A diferença entre esses três modos é comparável a esperar uma encomenda em casa: a espera ocupada é ficar parado olhando pela janela sem fazer mais nada; o *polling* é ir até a janela a cada dez minutos, entre outras tarefas domésticas, para checar se a entrega chegou; a comunicação por interrupção é a campainha tocando exatamente quando o entregador chega, liberando a pessoa para fazer qualquer outra coisa no meio tempo; o DMA é contratar um porteiro para receber a encomenda inteira e só avisar quando ela já estiver guardada dentro de casa, sem exigir qualquer atenção da pessoa durante o processo de entrega em si.

4.11.5 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 [13]:

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 4.11.1: 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)

4.11.6 Especificação de periféricos

Assim como processador, memória e armazenamento têm critérios objetivos de especificação (Capítulo 2 e Capítulo 4, §4.6.3), cada família de periférico tem seu próprio conjunto de critérios relevantes — geralmente ligados à natureza do dado que aquele dispositivo transmite (Seção 4.11.1):

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 (Capítulo 2, do livro de Manutenção de Computadores, trata potência elétrica em profundidade)

Nota prática. Como qualquer especificação de hardware (Capítulo 4, §4.6.3), 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.

4.12 Síntese do capítulo

Este capítulo tratou dos componentes físicos de um computador desktop, do procedimento de desmontagem e remontagem do gabinete, e retomou o POST — apresentado no Capítulo 3 do ponto de vista do software — sob a ótica dos quatro componentes de hardware vitais à sua execução: fonte, CPU, RAM e placa-mãe. Foram tratados também o papel da bateria CMOS na retenção de configurações, o funcionamento elétrico do botão liga/desliga e do painel frontal, os critérios de compatibilidade entre processador e placa-mãe (fabricante, soquete e geração), e os 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 — e como via de comunicação: a hierarquia de barramentos que liga processador, memória, armazenamento e periféricos, 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, além dos princípios de entrada e saída que regem a comunicação com teclado, mouse e demais periféricos. 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 5.

4.13 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`.
6. 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).

7. 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.
8. PCI-SIG. Especificações oficiais PCI/PCI Express. Disponível em: https://pcisig.com.
9. PCI-SIG. Especificações oficiais PCI Express (PCIe 3.0/4.0/5.0). Disponível em: https://pcisig.com.
10. 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.
11. SATA-IO. Especificação SATA. Disponível em: https://www.sata-io.org.
12. 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).
13. 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).

5 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.

5.1 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.

Analogia. A relação entre arquitetura e organização é equivalente à relação entre a especificação de um motor e o carro que o utiliza: quando se diz que um carro é “1.0” ou “2.0 turbo”, essa característica não pertence ao carro em si, mas ao motor que ele carrega. Da mesma forma, quando se diz que um computador “tem arquitetura x86”, essa característica não é do computador, mas do processador nele instalado — o computador apenas herda essa classificação por consequência.

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

5.2 A arquitetura de von Neumann formalizada

O Capítulo 1 introduziu o conceito de **programa armazenado**, formalizado por escrito no relatório de 1946 assinado por Arthur Burks, Herman Goldstine e John von Neumann [2]: a ideia de que instruções e dados residem na mesma memória do computador, permitindo que o programa seja executado de forma autônoma, sem intervenção humana a cada etapa. O nome consagrado do conceito — "arquitetura de von Neumann" — é o adotado neste livro, mas, como já observado no Capítulo 1, a formalização foi um trabalho conjunto dos três autores.

Esse princípio tem uma consequência que 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 5.1) — 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

5.3 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 [3] (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.

Analogia. O mesmo princípio rege o mercado de videogames: um PlayStation 5 executa jogos de PlayStation 4, e um Nintendo Switch 2 executa jogos do Switch original — mas nenhum dos dois executa, nativamente, um

cartucho de Super Nintendo, por pertencer a uma família de hardware totalmente diferente. Retrocompatibilidade existe dentro de uma mesma linhagem de arquitetura, não entre arquiteturas distintas.

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
80386	Lançado em 1985 [4]; 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 [5]
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 [6]

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 [7]. 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

5.4 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.

Analogia. Esse princípio pode ser ilustrado pela numeração de apartamentos em um prédio. Se o sistema de numeração usar apenas um dígito decimal, existem 10 endereços possíveis (0 a 9). Com dois dígitos, existem 100 endereços possíveis (0 a 99). Quanto mais dígitos disponíveis, mais endereços podem ser diferenciados — e, uma vez esgotada a quantidade de dígitos, não é possível criar um novo endereço, mesmo que fisicamente haja espaço para mais um apartamento. O mesmo ocorre com bits: cada bit adicional dobra a quantidade de endereços de memória que o processador consegue distinguir.

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** [8].

Uma consequência prática dessa relação de subconjunto (x86 ⊂ 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

5.5 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 família 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 [7]. 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 [9].

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.

Analogia. O conjunto de instruções de um processador pode ser comparado à língua falada por um profissional: um engenheiro fluente apenas em alemão, por mais competente que seja, não conseguirá interpretar um manual técnico escrito em árabe — não por incapacidade técnica, mas porque não conhece aquele "idioma". Da mesma forma, um sistema operacional desenvolvido para a arquitetura x86/x64, como o Windows, não é executado em um processador ARM ou AVR: as instruções que o compõem simplesmente não existem naquele processador. A esse ambiente de compatibilidade entre software e arquitetura dá-se o nome de **plataforma**.

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)

5.6 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 [7])	Historicamente mais baixo (idem)

Característica	CISC	RISC
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.

Analogia. O mesmo raciocínio lógico que afirma "toda vaca é um mamífero, mas nem todo mamífero é uma vaca" aplica-se aqui: todo processador x64 é CISC, mas nem todo processador CISC é x64; todo processador ARM é RISC, mas nem todo processador RISC é ARM (o AVR, por exemplo, também é RISC, mas não é ARM).

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.

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

A distinção entre arquitetura e organização (Seção 5.1) 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 [10].

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

5.8 Revisão integrada: firmware, particionamento e diagnóstico

Esta seção consolida, de forma breve, conceitos tratados em capítulos anteriores deste livro, retomados aqui como revisão integrada antes do fechamento do curso.

Bateria da placa-mãe (CMOS). A placa-mãe possui uma pequena memória volátil dedicada a armazenar as configurações do firmware — data e hora do sistema, parâmetros de inicialização e demais configurações definidas pelo *setup* da BIOS/UEFI (ver Capítulo 4). Por ser uma memória volátil, essa informação depende de energia contínua para não ser perdida; a bateria da placa-mãe existe justamente para manter essa memória energizada mesmo com o computador desligado da tomada.

Tabelas de partição: MBR e GPT. MBR (*Master Boot Record*) é o esquema de particionamento mais antigo, limitado a quatro partições primárias (contornável por meio de uma partição estendida contendo partições lógicas) e a discos de até aproximadamente 2 TB [11]. GPT (*GUID Partition Table*) é o esquema mais recente, sem essa limitação prática de tamanho de disco, com suporte a um número muito maior de partições, rótulos (*labels*) nomeáveis e uma cópia de segurança da própria tabela de partição em outra região do disco. O uso de GPT depende de suporte do firmware da placa-mãe: computadores com BIOS tradicional (não UEFI) não têm suporte a GPT; computadores com UEFI podem trabalhar tanto com GPT quanto, por retrocompatibilidade, com MBR.

Característica	MBR	GPT
Idade	Mais antigo	Mais recente
Partições primárias	Até 4 (ou partição estendida com lógicas)	Muito superior a 4
Tamanho máximo de disco	~2 TB	Sem essa limitação prática
Backup da tabela de partição	Não	Sim
Requisito de firmware	BIOS ou UEFI	Exclusivamente UEFI

BIOS, UEFI e Setup. BIOS (*Basic Input/Output System*, ver Capítulo 1) designa, em sentido amplo, o conjunto de softwares de firmware presentes

na placa-mãe — entre eles o programa de *setup* (interface de configuração), o POST (rotina de autoteste na inicialização) e utilitários de diagnóstico de memória. UEFI (*Unified Extensible Firmware Interface*) é a evolução moderna dessa camada de firmware, sucedendo a BIOS tradicional [12]; a relação entre BIOS e UEFI é análoga à relação entre MBR e GPT — a mais nova sucede a mais antiga, mantendo retrocompatibilidade.

GRUB e inicialização com múltiplos sistemas operacionais. Um *bootloader* é o software responsável por indicar ao processador qual região do disco deve ser carregada para dar início à execução de um sistema operacional — necessário porque o processador, conforme discutido na Seção 5.2, apenas executa instruções, sem qualquer capacidade de decidir autonomamente o que carregar. O GRUB é um bootloader típico de sistemas Linux, usado em configurações de inicialização dupla (*dual boot*). Recomenda-se instalar o Windows antes do Linux num mesmo disco porque o instalador do Windows, por padrão, sobrescreve a região de inicialização do disco sem considerar a coexistência com outro sistema operacional; o Linux, por filosofia de projeto, já assume a possibilidade de múltiplos sistemas operacionais e instala um bootloader capaz de gerenciar essa escolha.

Live CD / Live USB. Denomina-se *live CD* (ou *live USB*, por extensão histórica do termo) um sistema operacional capaz de ser executado inteiramente a partir da memória secundária removível, sem necessidade de instalação. Esse recurso é uma ferramenta de diagnóstico relevante para a disciplina de manutenção: ao inicializar um computador a partir de um live USB e testar componentes (por exemplo, conectividade Wi-Fi) fora do sistema operacional instalado, é possível isolar se um problema relatado é de hardware ou de software — aplicando o mesmo princípio de substituição de módulo apresentado no Capítulo 1 (Seção 1.6.2), mas por meio de software.

Funcionamento mínimo sem memória secundária. Retomando o modelo de hierarquia de memória do Capítulo 1: um computador do tipo desktop necessita de memória RAM presente e operacional para executar qualquer programa; sem memória primária, nenhuma instrução pode ser processada. A memória secundária (HD/SSD), por outro lado, não é estritamente necessária para que o computador ligue e opere — é possível, por exemplo, executar um sistema por meio de um live USB, sem qualquer armazenamento interno.

5.9 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 ar-

quitetura ARM.

Com este capítulo encerra-se a jornada proposta por este livro: partindo da definição técnica de computador e de sua evolução histórica (Capítulo 1), passando pela hierarquia de memória (Capítulo 2), pela instalação e configuração de sistemas operacionais (Capítulo 3) e pela montagem e manutenção do hardware físico (Capítulo 4), até chegar aos fundamentos teóricos de arquitetura de processadores que sustentam tudo o que foi estudado nos capítulos anteriores.

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. No semestre seguinte, o curso de Manutenção de Computadores (2026.1), com a mesma turma, dará continuidade a este conteúdo com foco em diagnóstico e reparo: aplicando, na prática cotidiana da bancada técnica, os fundamentos teóricos de hardware, memória, sistema operacional e arquitetura consolidados ao longo deste livro.

5.10 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. 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; sobre a disputa de crédito com Eckert e Mauchly: COMPUTER HISTORY MUSEUM. “The Neverending Quest for ‘Firsts’.” Disponível em: `https://computerhistory.org/blog/the-neverending-quest-for-firsts/`.
3. 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`.
4. STALLINGS, William. *Arquitetura e Organização de Computadores*. 10. ed. São Paulo: Pearson Education do Brasil, 2018.
5. TEDIUM. “Why Intel Couldn’t Trademark Numbers Anymore.” Disponível em: `https://tedium.co/2017/05/18/intel-386-486-trademark-battle`
6. 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`.

7. 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.
8. WIKIPEDIA. “x86-64.” Disponível em: https://en.wikipedia.org/wiki/X86-64.
9. 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.
10. ASAHI LINUX. Página “About” do projeto. Disponível em: https://asahilinux.org/about/.
11. MICROSOFT. “Windows and GPT FAQ.” Disponível em: https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-and (edição/data exata a confirmar pelo autor).
12. UEFI FORUM. Especificação oficial. Disponível em: https://uefi.org/specifications (edição específica a confirmar pelo autor).