

# Introdução à Computação

Rosane Minghim e Guilherme P. Telles

9 de Agosto de 2012

Não é difícil perceber que os computadores digitais têm uma grande influência na nossa sociedade. Por um lado, facilitam a organização das cidades, proporcionam conforto, lazer e alavancam descobertas e avanços científicos que melhoram a vida das pessoas. Por outro, fazem aumentar as diferenças sociais e causam a substituição do homem em vários tipos de trabalhos.

Este livro pretende ensinar como programar computadores. Programar um computador significa fazer com que ele realize uma determinada tarefa para nós. Este livro não avança na discussão sobre os impactos da tecnologia nas nossas sociedades, nem sobre as maneiras boas e ruins de usarmos a tecnologia e os computadores. No entanto, somos otimistas e acreditamos que um computador é uma ferramenta que pode ser usado para finalidades justas e por isso este livro existe.

Algumas siglas ou palavras são mantidas em inglês, como CPU, *hardware*, *software* e etc, porque estão incorporadas ao vocabulário técnico da computação.

# Capítulo 1

## Introdução

Nos primeiros tempos, os computadores eram grandes fisicamente, apresentavam defeitos em altas taxas e a forma de programá-los era estabelecendo conexões físicas entre conectores, usando cabos e interruptores. O papel do programador era muito próximo do construtor do computador, isto é, para programar um computador era obrigatório conhecê-lo em seus mínimos detalhes. À medida que os anos se passaram, os computadores diminuíram de tamanho, tornaram-se mais robustos e mais rápidos. Paralelamente ao desenvolvimento do *hardware* (o conjunto de dispositivos físicos que compõe um computador), surgiram vários tipos de linguagens de programação, que deixaram de ser físicas e passaram a ser textuais. Assim como uma língua que as pessoas de uma nação entendem, uma linguagem de programação é uma convenção sintática usada para compor programas, que são as instruções que o computador vai seguir (o conjunto de programas usados em um computador é chamado de *software*).

Programar um computador é uma tarefa que pode ser desenvolvida em diferentes níveis (mais próxima ou mais distante do hardware), e em diferentes linguagens. Algumas linguagens de programação estão bem próximas do hardware, como as linguagens de montagem (do inglês *assembly languages*). Para programar usando esse tipo de linguagens é preciso conhecer o hardware em detalhes. Outras linguagens, como a Prolog, praticamente ignoram a existência do computador: o programa estabelece regras e conceitos relacionados ao problema que se deseja solucionar, sem se preocupar com que operações o computador tem que realizar para isso.

Este livro pretende ajudar a programar computadores em uma **linguagem estruturada**. Exemplos dessas linguagens são C, Pascal e Modula2. Uma linguagem desse tipo permite a construção de programas que têm uma estrutura bastante relacionada à forma como um computador digital é organizado, mas sem no prender-se a detalhes. Essa distância moderada entre

as linguagens estruturadas e o hardware em geral é responsável por um bom desempenho dos programas<sup>1</sup>. A estrutura exigida pelas linguagens estruturadas em geral é responsável por uma boa legibilidade e compreensibilidade dos programas<sup>2</sup>. As linguagens estruturadas são bastante usadas no desenvolvimento de vários tipos de sistemas de software.

Ao longo dos vários capítulos que compõem este livro vamos introduzir os elementos principais das linguagens estruturadas. Assim, esperamos que quem tomar as lições que oferecemos aqui possa aprender os conceitos que permitirão programar em várias linguagens com pequeno esforço adicional.

Este capítulo é uma preparação. Na próxima seção vamos olhar rapidamente para a história dos computadores, ressaltando seus marcos principais. Em seguida, vamos olhar para um modelo simplificado de computador que vai permitir entender os computadores em um nível detalhado. Depois vamos considerar os compiladores e as formas através das quais um programa escrito em uma linguagem de programação é compreendido pelo computador.

A apresentação dos tópicos nas seções seguintes é a menor possível para que o leitor consiga aproveitar e entender melhor o resto do livro. Explicações mais abrangentes e profundas sobre cada um dos tópicos podem ser encontradas nas obras listadas em cada seção.

## 1.1 História do Computador

As raízes da computação estão ligadas ao desenvolvimento da lógica e da matemática ao longo dos séculos. Ao mesmo tempo em que a matemática desenvolvia-se, surgiam os instrumentos para automatizar os cálculos, isto é, a evolução conceitual foi seguida pela evolução tecnológica. Nesta seção destacamos os principais marcos desse desenvolvimento, citando os homens e máquinas que se destacaram.

### 1.1.1 A ciência da computação

Começando com a lógica proposicional de Aristóteles, várias contribuições para o desenvolvimento da lógica e da álgebra que levaram ao surgimento da computação foram dadas por matemáticos como Crisepo de Soles, Euclides, Diophantes, Al-kharazmi, Lulio, Galileo Galilei, Descartes e Fermat.

---

<sup>1</sup>A eficiência de um programa também está ligada à forma de construção do programa, não apenas à linguagem escolhida.

<sup>2</sup>Novamente, tais características também estão ligadas à forma de construção do programa, não apenas à linguagem escolhida.

De interesse especial para a computação foram os trabalhos de Boole, que em 1847 fundou a lógica simbólica e propôs um sistema de dois estados (0 e 1) para quantificar a lógica (álgebra Booleana), e de Claude Shannon, que em 1937 mostrou que a álgebra Booleana pode ser usada para descrever a operação de circuitos chaveados e que tais circuitos são capazes de executar operações lógicas, aritméticas e de memória.

A ciência da computação é a ciência de resolver problemas com o auxílio do computador. Pode-se dizer que a ciência da computação nasceu do trabalho de Alan Turing, que em 1936 apresentou um modelo formal do que é “realizar um cômputo”. Tal modelo era de interesse dos matemáticos à época, dentre os quais destacaram-se David Hilbert e Kurt Gödel, pois relaciona-se com a formalização da própria matemática. Contemporâneos de Turing, Alonso Church, Stephen C. Kleene e Emil L. Post contribuíram nessa mesma década para estabelecer os limites teóricos do que é possível computar com máquinas equivalentes aos computadores digitais atuais.

A ciência da computação desenvolveu-se e diversificou-se em vários ramos, em parte influenciada pela matemática, em parte influenciada pelos avanços tecnológicos. Dentre esses ramos, podemos citar, teoria da computação, bancos de dados, computação gráfica, inteligência artificial e a engenharia de software. As questões que envolvem a programação de computadores e a construção de sistemas são tratadas pela engenharia de software.

### 1.1.2 Dispositivos para calcular

Os dispositivos para cálculo podem ser classificados em analógicos ou digitais. Os instrumentos analógicos representam os números por quantidades físicas, como deslocamento linear ou angular, voltagem ou corrente elétrica, e as operações matemáticas são implementadas através de relações físicas ou geométricas equivalentes a elas. Os dispositivos digitais representam os números diretamente como dígitos e as operações matemáticas sobre eles são implementadas por funções desses números.

Os dispositivos mais antigos de que temos notícia para auxiliar o cálculo são os ábacos, que já existiam no século V a.C. Os ábacos continuaram a evoluir ao longo dos séculos e podem ser usados para somar, subtrair, multiplicar, dividir, calcular raízes quadradas e cúbicas. Os ábacos não fazem os cálculos, apenas ajudam a pessoa que calcula a manter controle dos números, dos vai-um e assim por diante.

A primeira calculadora mecânica de que tem-se notícia foi projetada por Leonardo da Vinci, em torno do ano 1500<sup>3</sup>. Em 1623, Wilhelm Schickard

---

<sup>3</sup>Em 1621, William Oughtred inventou a régua de cálculo, que não é considerada uma

projetou uma máquina de calcular que fazia as quatro operações aritméticas básicas. Blaise Pascal construiu uma máquina de somar em 1642. Em 1671, Gottfried Wilhelm von Leibniz inventou uma máquina que era capaz de somar, subtrair, multiplicar, dividir e calcular raízes quadradas.

Em 1823, Charles Babbage começou a construir a *máquina de diferenças*, que tinha o objetivo de calcular tabelas de funções matemáticas automaticamente e com precisão. Essa máquina deveria ser movida a vapor, mas por conta de limitações técnicas, ela não foi construída por Babbage. Em 1839, Babbage propôs a *máquina analítica*, um dispositivo revolucionário e que já agregava vários conceitos que estão presentes nos computadores modernos, mas que também não chegou a ser construída. A idéia é que a máquina analítica fosse capaz de executar qualquer tipo de cálculo, seguindo tipos diferentes de instruções. Ele imaginou que precisaria de um novo tipo de linguagem para programar sua máquina. Também imaginou a transferência de controle condicional, que permitiria à máquina executar instruções em qualquer ordem e não apenas em ordem estritamente seqüencial (dando origem a conceitos como subrotinas, loops e saltos) e a capacidade de que a máquina alterasse seu próprio programa. A entrada de dados dessa máquina deveria ser através de cartões perfurados, que representariam padrões algébricos. A idéia de usar os cartões foi inspirada pelo tear de Jacquard, que usava cartões perfurados para automatizar a tecelagem de padrões de desenho. Exceto por Augusta Ada Lovelace, que era notória matemática e auxiliou Babbage em vários aspectos relacionados à programação da máquina analítica, talvez ninguém mais tenha tido capacidade para compreender completamente e vislumbrar o alcance dessas novas idéias naquele momento.

Outras máquinas surgiram nos anos seguintes, dentre elas a de Thomas de Colmar, que era capaz de somar, subtrair, multiplicar e dividir, e em 1840 inaugurou a manufatura de máquinas de calcular. No fim do século XIX as máquinas de calcular já eram capazes de armazenar resultados parciais e reutilizá-los em novos cálculos, e de imprimir os resultados. Máquinas derivadas dessas precursoras continuaram a ser produzidas até a década de 1970, quando foram substituídas pelas calculadoras eletrônicas.

Um desenvolvimento importante do século XIX foi a introdução dos cartões perfurados por Herman Hollerith no censo de 1890, também inspirado pelo tear de Jacquard. A máquina eletromecânica construída por ele era capaz de (i) ler um conjunto de 50 a 220 cartões perfurados por minuto automaticamente, cada cartão contendo em torno de 80 dígitos decimais, (ii) somar, multiplicar ou ordenar números e (iii) perfurar cartões com resultados. A tecnologia de cartões perfurados foi muito usada em computadores.

---

calculadora mecânica e sim um instrumento de apoio aos cálculos, como o ábaco.

Até a década de 1930, vários dispositivos analógicos, mecânicos e eletromecânicos foram construídos ou projetados para finalidades específicas, como o cálculo de integrais, a resolução de sistemas de equações diferenciais e a previsões dos tempos das marés.

A partir de meados da década de 1930, as idéias de Babbage começaram a ser implementadas. Os computadores de uso geral que surgiram foram classificados em gerações, de acordo com a tecnologia que utilizavam.

### **Geração 0 – Eletromecânicos (dos anos 1930 aos anos 1950)**

Na Alemanha, Konrad Zuse desenvolveu seus computadores Z1, Z2, Z3 e Z4, entre 1936 e 1945. Os computadores de Zuse eram eletromecânicos, funcionavam com relés e usavam aritmética binária. Por conta da guerra, o trabalho de Zuse permaneceu obscuro. Na década de 1950 Zuse fundou uma empresa de computadores que mais tarde foi incorporada pela Siemens.

Paralelamente, nos Estados Unidos também foram desenvolvidas máquinas eletromecânicas baseadas em relés para calcular. O Modelo I de George Stibitz e S.B. Willians foi usado para automatizar cálculos na companhia telefônica Bell. Em 1943 ficou pronto o Mark I, um projeto de Howard Aiken construído pela Marinha, pela Universidade de Harvard e pela IBM.

### **1a Geração – Válvulas (dos anos 1940 aos anos 1960)**

O primeiro computador eletrônico de uso geral foi o ENIAC, projetado por John William Mauchly e J. Presper Eckert Jr. O ENIAC usava válvulas ao invés de relés. Era mais silencioso, mas imenso e consumia energia suficiente para iluminar uma pequena cidade.

Como efeito do fato de que o ENIAC era programado fisicamente, através de fios e interruptores, Mauchly e Eckert discutiram a criação de um computador com programa armazenado, isto é, um computador em que uma memória que permitisse leitura e escrita seria usada para armazenar os dados e o programa. Dessa forma o processamento poderia seguir seguir caminhos alternativos, com base no resultado de cálculos anteriores. A partir dessa idéia, em 1944 eles propuseram a construção de um novo computador, o EDVAC.

Em 1944, John von Neumann fazia parte do projeto Manhattan quando soube do ENIAC. Provavelmente por antever as possibilidades de um computador como aquele aplicado aos cálculos de armas atômicas, passou a fazer parte da equipe de projeto de ENIAC e do EDVAC, como consultor. Em 1945 ele escreveu um relatório intitulado “First Draft of a report to the EDVAC”,

que apresentava os elementos básicos de um computador com programa armazenado:

1. Uma memória para escrita e leitura, com posições endereçáveis individualmente e em qualquer ordem e que armazenasse instruções e dados.
2. Uma unidade que fosse capaz de realizar tanto operações aritméticas como operações lógicas.
3. Uma unidade de controle que pudesse interpretar uma instrução armazenada na memória e selecionar um caminho de execução diferente com base nos resultados de operações anteriores.

Assim como na máquina analítica de Babbage, von Neumann previu a possibilidade de que o computador fosse capaz de modificar seus próprios programas. Praticamente todos os computadores daquela época em diante seguem esse modelo de organização, que ficou conhecido como “arquitetura de von Neumann”, apesar de muitas idéias para a arquitetura terem sido anteriores à participação dele no projeto de computadores. No entanto acredita-se que von Neumann era o único com genialidade suficiente para sintetizar e apresentar os resultados daquela maneira, àquela época.

Antes que o EDVAC funcionasse completamente, em 1951, o primeiro computador disponível comercialmente ficou pronto. Era o UNIVAC I, baseado no projeto do próprio EDVAC.

## **2a Geração – Transistores (dos anos 1960 ao início dos anos 1970)**

Nessa fase do desenvolvimento dos computadores houve sensíveis reduções do seu tamanho e aumentos da sua confiabilidade. Um transistor substituiu em torno de 40 válvulas, além de ser mais barato e de dissipar pouquíssimo calor.

Um representante dessa geração foi o CDC 1604, construído por Seymour Cray.

## **3a Geração – Circuitos Integrados (do fim dos anos 1960 até os anos 1970)**

Nessa geração, os computadores usavam circuitos integrados, que continham milhares de transistores em um único chip. Além de melhorar o desempenho, os computadores ficaram ainda menores, do tamanho de geladeiras.



#### 4a Geração – Microprocessadores (desde meados dos anos 1970)

O aumento da capacidade de integração permitiu colocar milhões de transistores em um único chip. Em 1971, Ted Hoff da Intel instruiu um chip que continha todas as unidades lógicas, aritméticas e de controle de um computador.

A conseqüência deste avanço foi a popularização dos computadores de uso pessoal e a expansão acelerada do uso de microprocessadores para fins especializados, como em máquinas, carros e equipamentos eletrônicos.

#### Novos Avanços

Computadores paralelos in a box

Redes de alta velocidade

Clusters

As linguagens de programação também evoluíram. Nos primeiros computadores a programação era feita através de fios e interruptores, através dos quais o programador inseria seqüências de zeros e uns que representavam as instruções e os dados. Muito rapidamente percebeu-se que essa forma de programação diretamente em código de máquina era ineficiente e difícil de compreender ou modificar. Desde esse momento, vários tipos de linguagens foram surgindo gradativamente, na tentativa de afastar o programador do computador físico e aproximá-lo do problema que deve ser resolvido. Surgiram várias linguagens desde então, de certa forma acompanhando o desenvolvimento dos computadores e de novas tecnologias.

## 1.2 Números Binários

Todo número é representado em uma base. Na base 10, usamos dez dígitos diferentes para representar um número: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.

Na base 2 usamos apenas dois dígitos para representar um número: 0 e 1. Um computador representa esses dois dígitos como ausência e presença de eletricidade. Um computador é construído para armazenar e operar ausência e presença de eletricidade que representam números.

## 1.3 Organização de Computadores

Nesta seção apresentamos um modelo de computador digital genérico. Embora a linguagem Pascal não tenha comandos para controlar estes componentes diretamente, este conhecimento será útil para compreender melhor

várias estruturas e comandos das linguagens estruturadas, como declarações de variáveis, decisões e repetições.

Um computador é um conjunto de componentes elétricos, eletrônicos e mecânicos que são usados para armazenar e operar ausência e presença de eletricidade<sup>4</sup>, campo magnético ou luz. Ausência e presença de eletricidade, campo magnético e luz representam os dígitos binários 0 e 1. Um dígito binário é chamado de **bit**. Sequências de dígitos binários são usadas para compor números binários. Números decimais, letras, palavras, imagens, som e vários outros tipos de dados e informações são representados como conjuntos de números binários. Um computador digital pode ser visto, de forma simplificada, como uma máquina que realiza operações aritméticas e lógicas sobre números binários de forma coordenada e bem rápida.

Os principais componentes de um computador digital contemporâneo são:

1. A memória RAM: A memória RAM armazena dados binários enquanto o computador enquanto há energia elétrica. Os números são armazenados na memória e recuperados dela individualmente, através do seu endereço. Uma memória RAM pode ser vista como uma tabela:

Endereço	Número Armazenado
00001	1101101001001010
00010	0101010110001010
00011	0101010101011010
00100	1001010101001100
00101	0111010011101001
⋮	⋮
11111	1011101101011010

Cada linha da tabela tem um número que a identifica univocamente e um dado. Esse número é o endereço do dado. Cada dado armazenado na memória tem um tamanho, que é o número de bits que o compõem. No exemplo acima, cada dado tem 16 bits. Cada endereço tem 5 bits. A memória tem um tamanho limitado, que é o número de dados que ela consegue armazenar. No nosso exemplo, a memória tem  $2^5 = 32$  posições.

A memória RAM armazena informações de interesse de quem usa o computador, como por exemplo os números em uma planilha de custos para a construção de uma casa. A memória também armazena os programas que manipulam os dados, isto é, as instruções que indicam o que

<sup>4</sup>Na verdade, níveis alto e baixo de tensão elétrica.

fazer com os dados, por exemplo, a planilha eletrônica. Os programas também são representadas por números binários.

O significado de cada dado armazenado na memória depende de quem o usa. Um mesmo número binário pode representar uma instrução, uma letra, um número inteiro positivo ou negativo, um número em ponto flutuante e assim por diante.

Um **programa** então é uma seqüência de posições na memória, em que cada posição possui uma instrução, e que tem uma instrução de término de execução no seu final.

2. O processador (ou CPU): O processador é o componente que executa um programa que está armazenado na memória. Os principais componentes da CPU são a unidade lógica e aritmética e os registradores. A unidade lógica e aritmética executa operações aritméticas e lógicas. Cada registrador é um tipo de memória que pode armazenar um dado.

As instruções que compõem um programa podem fazer, dentre outras coisas, com que a CPU transfira dados da memória para os registradores ou vice-versa, faça alguma operação lógica ou aritmética envolvendo os dados nos registradores, transfira dados do controlador de entrada-e-saída para a memória ou vice-versa.

Um programa é executado em seqüência pela CPU, uma instrução por vez. Algumas instruções alteram o fluxo de execução, fazendo com que uma parte do programa seja executada apenas se uma certa condição for verificada (por exemplo, se o número armazenado em uma certa posição da memória é igual a zero) ou repetindo uma parte do programa um certo número de vezes (por exemplo, multiplicando por dois um conjunto de números armazenados na memória).

3. O controlador de entrada e saída: O controlador de entrada e saída se comunica e troca dados com outros dispositivos que chamamos de periféricos. Exemplos de periféricos são unidades de disco, fita, CD e DVD, teclado, mouse e impressoras.
4. O barramento: O barramento é o canal utilizado para o fluxo de dados entre a memória, a CPU e os demais componentes do computador. O barramento possui um tamanho, isto é, o número de bits que podem ser transferidos de uma única vez.

O barramento é utilizado tanto para transferir dados quanto endereços. Por exemplo, quando a CPU quer que o dado armazenado na posição X da memória seja copiado no registrador R1, primeiro o processador

envia o endereço  $X$  pelo barramento. Depois a memória copia o dado armazenado na posição  $X$  no barramento, e o registrador R1 armazena o dado.

## 1.4 Sistemas Operacionais

Um sistema operacional é um sistema de software que tem duas funções principais:

1. Esconde as particularidades de dispositivos como memórias, unidades de disco, unidades de fita, placas de rede, impressoras e outros, dando ao programador uma forma simplificada e padronizada para usar tais dispositivos.

Por exemplo, para ler dados ou gravar em um disco, basta conhecer o nome de um **arquivo**, que é a estrutura relacionada a uma região delimitada do disco, e usar um conjunto de operações simples que o sistema operacional oferece para ler os dados do arquivo. Não importam o tipo, a marca ou o modelo do disco, como ele funciona, nem onde os dados estão, as operações para ler e gravar no arquivo serão sempre as mesmas.

2. Gerencia a memória, a CPU (ou as CPUs se houver mais de uma) e os periféricos de tal forma que mais de um programa possa ser executado ao mesmo tempo ou um programa paralelo possa ser executado em mais de uma CPU. Se vários programas são executados ao mesmo tempo, o sistema operacional faz um revezamento entre eles, deixando que algumas instruções de cada um sejam executadas de forma intercalada, dando a sensação de que eles são executados ao mesmo tempo. O sistema operacional também tem a função de separar os programas de tal forma que eles não interfiram uns com os outros.

Exemplos de sistemas operacionais são Linux, Windows, MS-DOS, AIX, dentre outros.

## 1.5 Arquivos

Dispositivos de armazenamento secundário permitem o armazenamento e a recuperação de dados digitais de forma não-volátil, isto é, que se mantém quando a energia é desligada. Dispositivos de armazenamento secundário típicos são discos magnéticos (HD, disquete, zip), discos ópticos (CD, DVD,

MD), fitas magnéticas (DAT, DLT) e cartões de memória (flash, SD). Cada um desses dispositivos é diferente dos demais no funcionamento, na forma de armazenar os dados, na velocidade de acesso, na capacidade de armazenamento e outros.

Um arquivo é um mecanismo que esconde os detalhes do dispositivo e da organização dos dados no dispositivo. Usando arquivos, os dados são acessados da mesma forma, independentemente do tipo do dispositivo. Além disso, o arquivo associa um nome aos dados, o que nos livra de saber em que parte do dispositivo eles estão.

## 1.6 Compiladores e Interpretadores

De maneira geral, um compilador é um programa que traduz um programa escrito em uma linguagem em um programa equivalente em outra linguagem. Normalmente usamos a palavra de maneira mais específica e chamamos de compilador o programa que traduz um programa escrito em uma linguagem de programação (chamado de código-fonte) em um programa que pode ser executado por um computador (chamado de código-objeto).

Um interpretador é um programa lê um programa escrito em uma linguagem de programação e executam o programa, sem gerar código-objeto explicitamente. Este tipo de tradução é usado em linguagens de comandos (*scripts*) ou em linguagens que permitem construções cujo tamanho (porção da memória que ocupa) não pode ser determinado antecipadamente.

Em muitos casos os compiladores estão acoplados a ambientes de desenvolvimento que incluem outras ferramentas. Exemplos de tais ferramentas são aquelas para detecção de erros (*debuggers*), para medir o tempo gasto pelo programa (*profilers*), para construção de diagramas e para o desenvolvimento cooperativo de programas.