

# Introdução aos sistemas operacionais

# Sistema Operacional (SO)

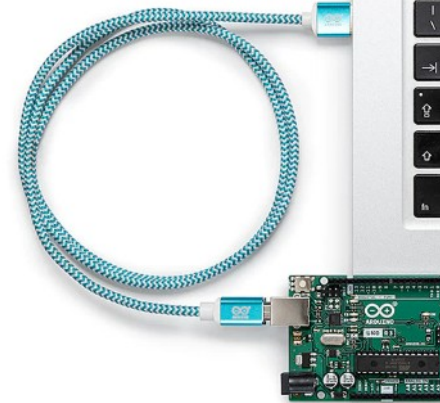
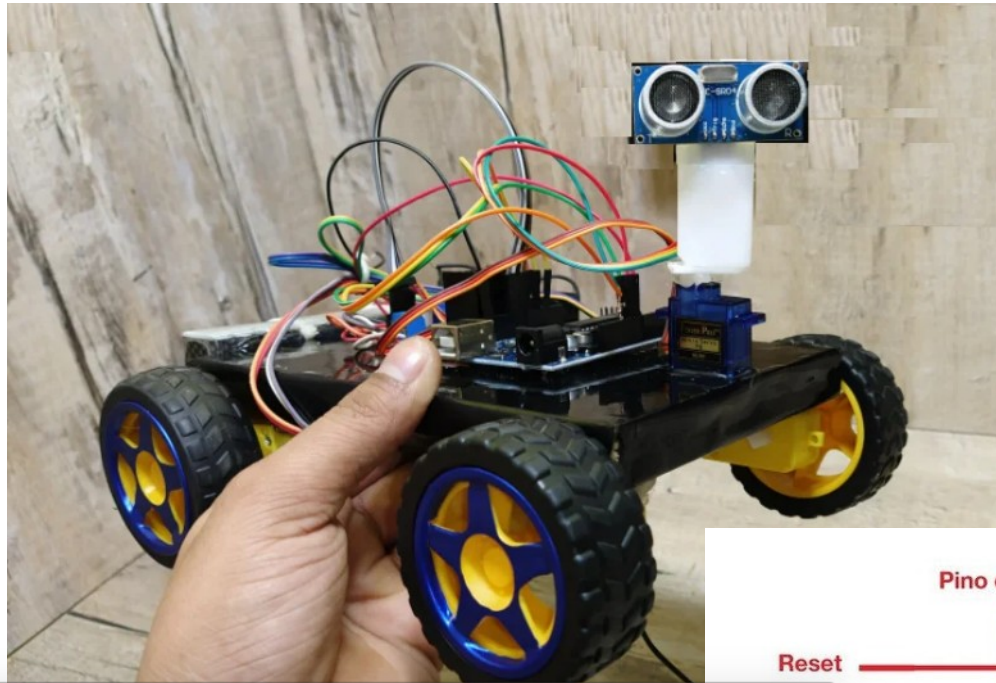
- Controla o computador
- Gerencia recursos de hardware
  - Ex.: teclado, mouse, impressora, disco, etc.
- Interface entre os programas e o hardware
- Conecta o usuário ao hardware
- Quando o computador é ligado o SO é carregado, e está presente até ser desligado



# SO

- Gerencia o computador de forma eficiente e produtiva
- Garante integridade e segurança de dados
- Gerenciador de tarefas: CPU, RAM, Disco, etc.
- Qual o SO do Android, iPhone, do roteador de Internet residencial, e do Arduino?
- O que é o sistema de janelas (interface)?
- Windows procura integrar SO com o sistema de janelas
- \*nix procura separar

# Arduino



Blink | Arduino 1.8.5

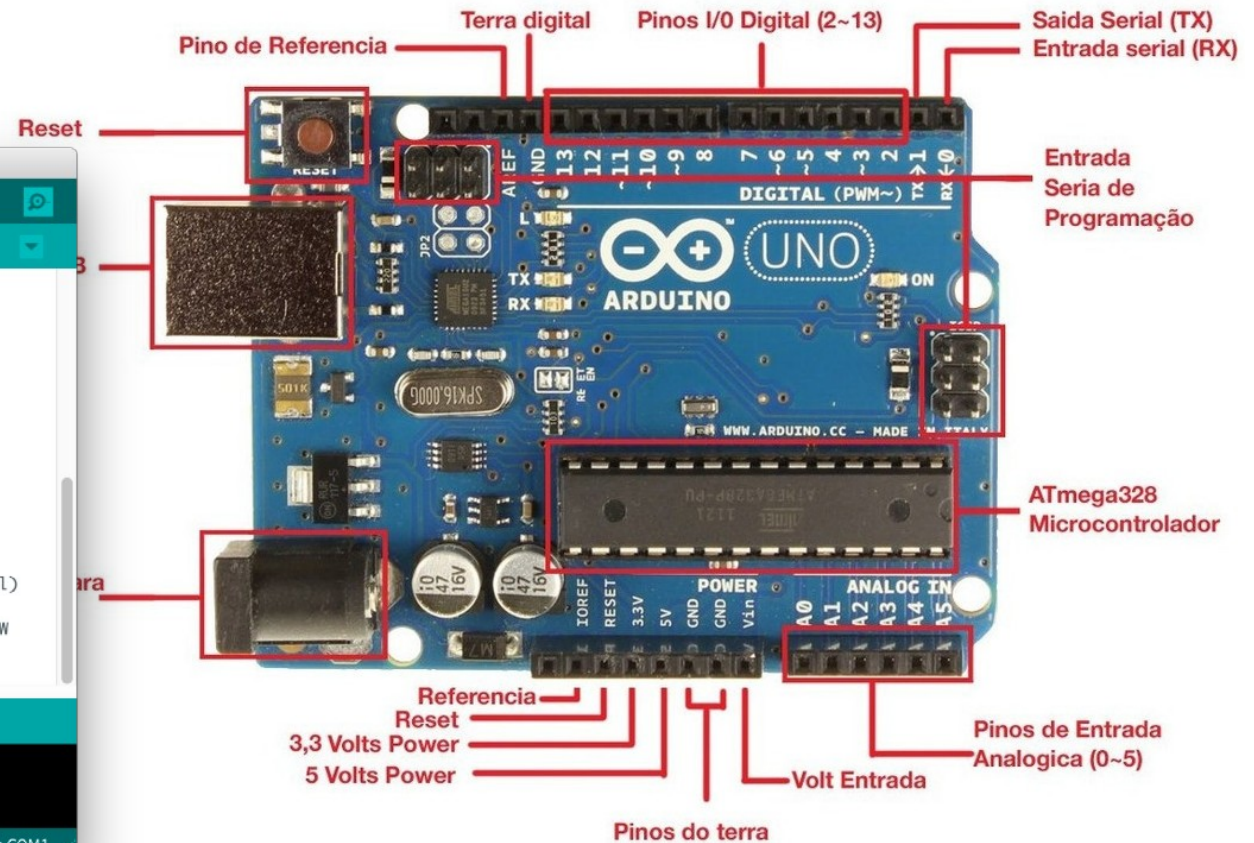
```

Blink §
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
  
```

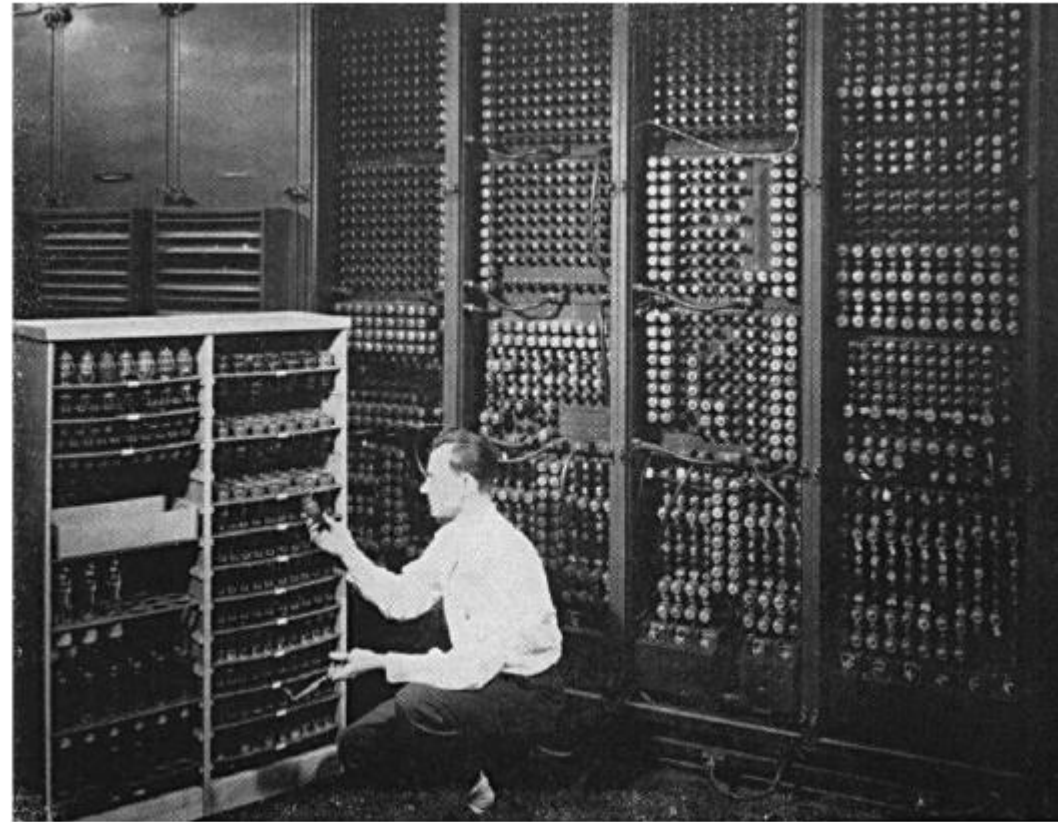
32 Arduino/Genuino Uno on COM1





# Primeira geração (1945-1955)

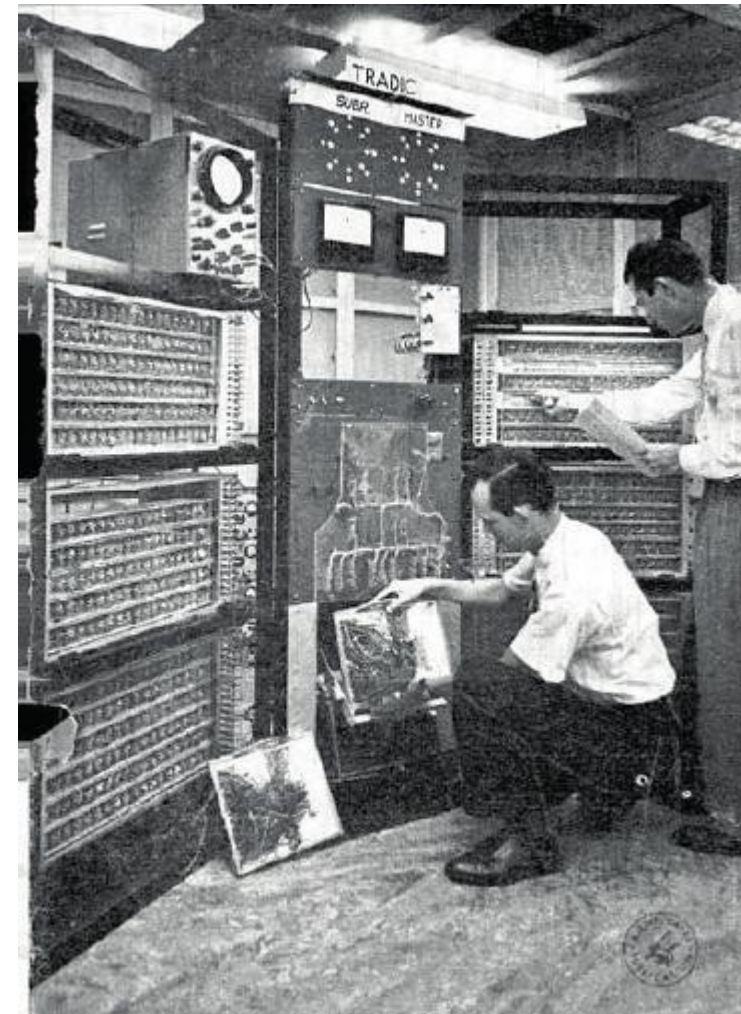
- Os primeiros computadores não possuíam SO
- O programa era carregado diretamente na memória e executado, sem a presença de um SO.
- O programa deveria se preocupar com todos os detalhes de hardware, incluindo eventuais drivers, entrada/saída, etc.



ENIAC (1945)

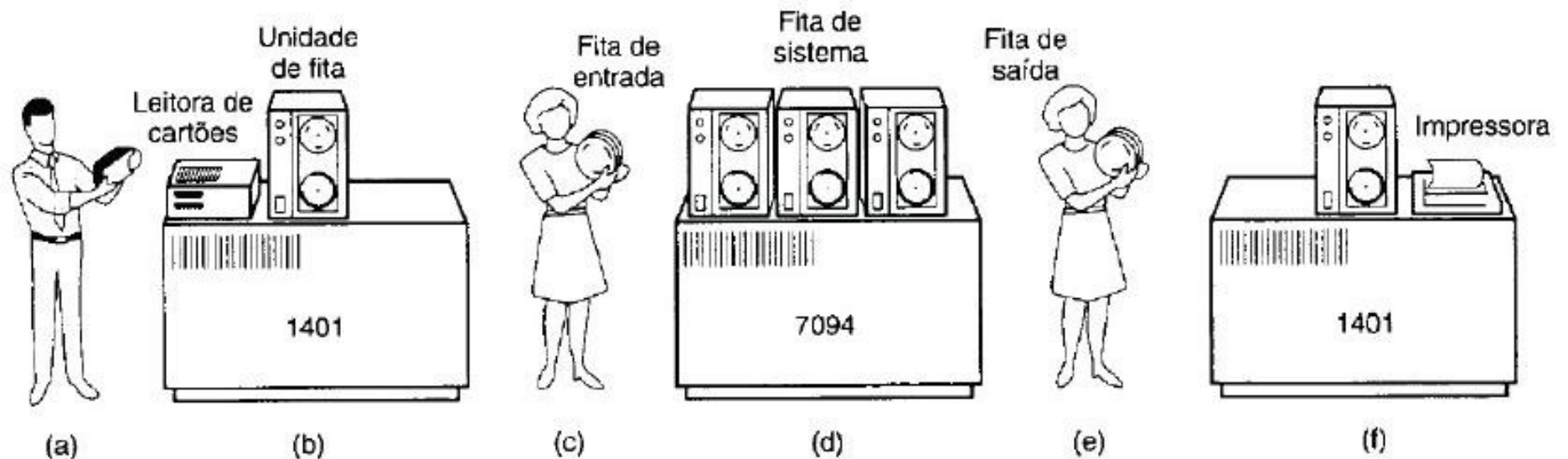
# Segunda geração (1955-1965)

- Possuíam SO
- Chamados de Mainframes
  - Ocupavam salas inteiras
- Programas escritos em papel e depois “digitados” em cartões perfurados
- Demorava até obter o resultado impresso (sem telas/monitores)
- Processamento em lote
  - Vários “jobs” em fita magnética



TRADIC (1954)

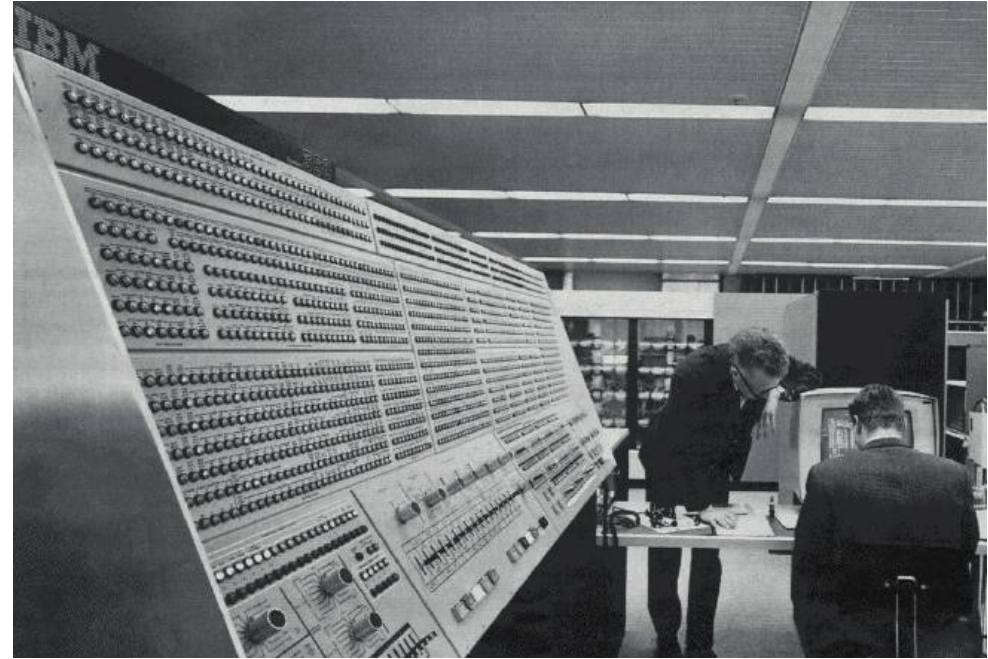
Figura 1.4 | Processamento em lote



- (a) Os programadores trazem os cartões para o 1401.
- (b) O 1401 lê os Jobs em lote na fita.
- (c) O operador leva a fita de entrada para o 7094.
- (d) O 7094 realiza a computação.
- (e) O operador leva a fita de saída para o 1401.
- (f) O 1401 imprime a saída.

# Terceira geração (1965-1980)

- Circuitos integrados
- Padronização de hardware
  - Redução de custo
- Multiprogramação
  - Compartilhamento de memória
- Timesharing
  - Divide o tempo da CPU em intervalos e tarefas
- Multiusuário
  - Multics



IBM System/360 (1965)

- Multiusuário e multitarefa
  - Unix



# Quarta geração (1980 até hoje)

- Computadores pessoais (8 bits)
  - Apple I (1976)
- SOs de 8 bits
  - Basic em ROM fazendo o papel também de SO
  - Apple II DOS
  - CP/M
- SOs de 8/16 bits (IBM PC)
  - MS-DOS
  - Windows
  - \*nix e compatíveis



- Desenvolvimento de SO com recursos de rede
  - Permite conectar a outras máquinas e executar tarefas
- SO distribuídos
- SO para celulares e dispositivos móveis
- Abstração
  - A função do SO como uma máquina estendida é esconder a complexidade do hardware do programador
    - Ex.: drivers de dispositivo

- Gerenciamento de recursos
  - Compartilhamento de memória
  - Dispositivos de E/S para os processos
  - Quem usa qual recurso, evitando conflito
  - Compartilhamento no tempo e no espaço
    - Eficiência
- SO - serviços para os aplicativos dos usuários e ao sistema
  - Carregar um programa na memória e executar
  - Mantém o sistema de arquivos, criação, leitura, escrita
  - Interfaces: impressoras, câmeras, pendrive, etc.
  - Manutenção do estado do sistema

# SO

- Windows
  - Código fechado
  - Licenças
- \*nix (ex.: Linux)
  - Código aberto e livre
- Suporte
  - Pago
  - Comunidade