

# Caderno pedagógico de Programação e Informática

---

## O laboratório do Byte: pensamento computacional, código e maker na sala de aula

*Cinco percursos de aprendizagem baseados em projetos, do desplugado ao Arduino, ancorados em "O Fantástico Laboratório". Da lousa do avô à bancada da turma: ensinar a máquina a sentir.*

---

### Carta ao educador de Computação

Este caderno nasce de uma cena. Um garoto de quinze anos, Percival, encontra no fundo do quintal a máquina inacabada do avô, junta a ela a sucata que a cidade descarta, encaixa no meio de tudo o velho disco rígido do velho Severino, e liga. A máquina não resolve nada: a máquina explode. E da explosão levanta-se, chiando, o Byte: um robô do tamanho de uma criança, feito de carcaça de CPU, lata e parafuso, com um disco rígido girando no lugar do coração e luzes no peito que mudam de cor conforme o que ele sente.

Guarde essa imagem, porque ela é uma aula inteira de computação. O Byte tem uma entrada (o que vê e ouve), um processamento (o que decide), uma memória (o disco do avô), uma saída (a luz que muda de cor). O Byte é hardware e é software. E o Byte só ganha vida quando o programa que faltava, a última peça, não era uma fórmula: era o pathos, a emoção, escrita numa palavra grega que uma menina de oito anos decifrou sozinha. A tese do filme é também a tese da boa aula de programação: a máquina mais poderosa do mundo é a única que ninguém constroi sozinho.

Ensinar computação na Educação Básica não é ensinar a mexer no computador. É ensinar a pensar de um jeito que serve para a vida inteira: quebrar um problema grande em partes pequenas (decomposição), enxergar o que se repete (padrões), separar o que importa do que não importa (abstração) e escrever um passo a passo que qualquer um consiga seguir (algoritmo). Isso se ensina com um computador, com um Arduino, e se ensina também com giz, cartolina e o corpo das crianças no pátio. Este caderno oferece os três caminhos, sempre, porque o pensamento computacional não tem CEP e não pode depender de sala equipada.

Você vai encontrar aqui duas coisas. Primeiro, uma seção robusta de aprofundamento de conteúdo, para você ensinar computação de verdade: pensamento computacional, lógica de programação em blocos e em texto, o Arduino que dá comportamento ao Byte, o hardware por dentro (a CPU, a memória, a entrada e a saída, o disco do avô como coração), a internet e a cidadania digital, e uma breve história da computação com o capítulo brasileiro que quase ninguém conta. Depois, cinco sequências didáticas completas, prontas do primeiro minuto ao produto final, com fichas para copiar, rubricas e caminho desplugado para toda escola, tenha ela vinte computadores ou nenhum.

A porta da edícula, que ficou trancada por décadas, no fim do filme fica escancarada, e lá dentro cabe uma turma inteira discutindo um desenho. Que a sua sala seja essa edícula. Bem-vindo ao laboratório.

---

## Como este caderno funciona

### A base pedagógica

As sequências combinam cinco pilares que a boa educação em computação pede:

- **Pensamento computacional como competência, não como ferramenta:** o objetivo é o jeito de pensar (decompor, reconhecer padrões, abstrair, algoritmizar), que existe com ou sem máquina.
- **Aprendizagem baseada em projetos (ABP):** um desafio real, um produto final, protagonismo de quem aprende.
- **Cultura maker e mão na massa:** aprender construindo, testando, errando e refazendo (o ciclo do inventor, que é também o ciclo do programador: escreve, roda, corrige).
- **Progressão do desplugado ao plugado:** todo conceito nasce sem tela (com o corpo e o papel) e só depois migra para o bloco, para o texto e para a placa. Assim ninguém fica de fora.
- **Desenho Universal para a Aprendizagem (DUA):** múltiplos caminhos de entrada, para que a computação seja de todos.

### O arco de cada percurso

Toda sequência segue o mesmo movimento, herdado do caderno-mãe de sequências didáticas, para você se orientar:

1. **Sensibilização.** Um trecho da obra e uma pergunta acendem a curiosidade.
2. **Investigação.** A turma levanta hipóteses, testa a lógica no papel e no corpo.
3. **Produção.** Mão na massa: o código, o circuito ou o algoritmo começam a existir.
4. **Socialização.** A turma compartilha, apresenta, demonstra o que fez rodar.
5. **Avaliação.** Por evidência (o produto, o código e o percurso), com rubrica e autoavaliação.

### Legenda dos ícones

- 🕒 tempo · 👥 agrupamento · 🎯 produto · ♿ acessibilidade (DUA) · 📄 ficha reproduzível · 💬 fala ou pergunta mediadora do professor · 👁️ fique de olho (o que observar) · 🔌 versão plugada · 📴 versão desplugada · 💾 bloco de código ou algoritmo.

### Matriz-síntese

Nº	Sequência	Eixo de Computação	Faixa	Duração	Produto final
1	O algoritmo do café da Maria (desplugado)	Pensamento Computacional	4º ao 9º	4 aulas	O livro de receitas de algoritmos da turma
2	O Byte ganha comportamento (Scratch)	Pensamento Computacional e Cultura Digital	5º ao 9º	6 aulas	O Byte animado que sente e responde
3	O coração do Byte (hardware e maker)	Mundo Digital	6º ao 9º e EM	5 aulas	O pôster-máquina e o Byte de luz (LED)
4	O LED que sente: dar vida ao Byte com Arduino	Mundo Digital e Pensamento Computacional	8º, 9º e EM	6 aulas	O Byte físico com humor de LED (ou versão desplugada)
5	A rede da Ordem: internet, cidadania e segurança	Cultura Digital	6º ao 9º e EM	5 aulas	O contrato digital e o mapa da internet da turma

Todas dialogam com os **ODS 4 (Educação de qualidade)** e **9 (Indústria, inovação e infraestrutura)**, com o **tema transversal Ciência e Tecnologia**, e podem convergir na feira culminante descrita no fim. As sequências conversam entre si e com o caderno-mãe de sequências didáticas e a plataforma de IA do Byte.

---



---

# Parte I. Aprofundamento de conteúdo: computação de verdade, ancorada no filme

---

*Esta parte é para você, professor. Não é roteiro de aula: é a base de conteúdo, com explicações, exemplos de algoritmo, código em blocos e atividades desplugadas, para você ensinar computação com segurança, mesmo que ela não seja a sua formação de origem. Cada tópico se ancora numa cena da obra, porque o Byte, no fundo, é um curso de computação de pé sobre duas rodas.*

## 1. O pensamento computacional (o jeito de pensar do inventor)

Antes de qualquer linguagem, antes de qualquer computador, existe um jeito de pensar. Chamamos de **pensamento computacional**, e a boa notícia é que ele não pertence à máquina: pertence a quem resolve problemas. Percival usa esse pensamento a história inteira, e a gente pode ensinar seus quatro pilares a partir do que ele faz.

### 1.1. Decomposição: quebrar o problema grande em partes pequenas

Percival não constroi o Byte de uma vez. Ele separa: a estrutura (a carcaça), a fonte de energia (a pedra, a pilha), a memória (o disco do avô), o comportamento (as luzes). Cada parte é um problema menor, resolvido por vez.

**A regra:** um problema grande e assustador vira uma pilha de problemas pequenos e possíveis. Programar é, quase sempre, isto: decompor.

**Exemplo para a sala.** Escovar os dentes parece uma coisa só. Decomponha: pegar a escova, abrir a pasta, pôr a pasta na escova, molhar, escovar de cima, escovar de baixo, cuspir, enxaguar, guardar. Nove partes. Se uma máquina fosse escovar os dentes, ela precisaria de todas, na ordem.

### 1.2. Reconhecimento de padrões: enxergar o que se repete

Maria calcula o tempo do trem com o corpo porque ouviu aquele apito subir a serra dezesseis anos: ela reconheceu um **padrão**. O apito sempre sobe do mesmo jeito, no mesmo ritmo. Reconhecer o padrão é o que permite prever e agir.

**A regra:** onde há repetição, há um padrão, e todo padrão pode virar uma regra que economiza trabalho.

**Exemplo para a sala.** Peça à turma para desenhar uma escada de dez degraus. Depois pergunte: qual é o padrão? "Sobe um, anda um, sobe um, anda um." O padrão descrito uma vez serve para dez degraus ou para mil. Guardar isso é o começo da repetição (o laço), que veremos adiante.

### 1.3. Abstração: separar o que importa do que não importa

O Byte não precisa saber a cor exata da lata de que é feito para funcionar. Precisa saber o essencial: tem entrada, tem energia, tem memória, tem saída. Ignorar os detalhes que não mudam a solução é **abstrair**.

**A regra:** um bom mapa não mostra cada pedra do caminho; mostra as ruas. Abstrair é fazer o mapa certo do problema.

**Exemplo para a sala.** Um mapa do metrô não mostra as curvas reais dos trilhos nem as casas em volta: mostra só as estações e as linhas, porque é só isso que o passageiro precisa. Isso é abstração: guardar o essencial, jogar fora o resto.

#### 1.4. Algoritmo: o passo a passo que qualquer um segue

No fim, tudo vira **algoritmo**: uma sequência finita de passos, clara e sem ambiguidade, que leva de um ponto de partida a um resultado. A receita de bolo é um algoritmo. O caminho de casa até a escola é um algoritmo. O que Alice faz ao decifrar o grego, letra por letra, comparando símbolos, é um algoritmo de decifração.

**A regra:** se você consegue explicar para outra pessoa executar sem você por perto, e ela chega ao mesmo resultado, você tem um algoritmo.

#### Exemplo de algoritmo em português estruturado (pseudocódigo):

```
ALGORITMO Fazer_o_cafe_da_Maria
INÍCIO
  pegar a chaleira
  encher a chaleira com água
  colocar a chaleira no fogo
  ENQUANTO a água não ferver:
    esperar
  desligar o fogo
  colocar o pó no coador
  despejar a água quente devagar
  SE o café estiver fraco:
    passar mais uma vez
  SENÃO:
    servir
FIM
```

Repare que já apareceram, sem susto, os três blocos que sustentam toda a programação: a **sequência** (um passo depois do outro), a **repetição** (o ENQUANTO, esperar até ferver) e a **condição** (o SE o café estiver fraco). É disto que a Parte I trata a seguir.

---

## 2. Lógica de programação: os três blocos que constroem tudo

Todo programa do mundo, do jogo mais complexo ao aplicativo do banco, é feito de três estruturas de controle. Só três. Quem domina as três domina a lógica.

## 2.1. Sequência

Um passo depois do outro, na ordem. A ordem importa: pôr o pó depois da água quente dá outro resultado. O computador é obediente e literal, faz exatamente o que se manda, na ordem em que se manda. Ensinar isso cedo cura muita frustração depois.

## 2.2. Repetição (o laço)

Quando algo se repete, não se escreve dez vezes: escreve-se uma vez e manda-se repetir. Existem dois tipos, e vale nomear os dois:

- **Repetir um número certo de vezes** (laço "repita N vezes"): "pisque a luz do Byte 3 vezes".
- **Repetir enquanto uma condição for verdadeira** (laço "enquanto"): "enquanto a água não ferver, espere".

### Exemplo de laço em pseudocódigo:

```
REPITA 3 VEZES:  
    acender a luz do Byte  
    esperar 1 segundo  
    apagar a luz do Byte  
    esperar 1 segundo
```

## 2.3. Condição (a decisão)

O programa escolhe um caminho conforme uma pergunta de sim ou não. É o **SE / SENÃO**. É aqui que a máquina parece "pensar": ela testa uma condição e decide.

### Exemplo de condição em pseudocódigo, com o Byte sentindo:

```
SE alguém chegar perto do Byte:  
    acender a luz azul (curiosidade)  
SENÃO SE o Byte ficar sozinho por muito tempo:  
    acender a luz roxa (saudade)  
SENÃO:  
    acender a luz verde (calma)
```

## 2.4. Variáveis: a memória do programa

Uma **variável** é uma caixinha com nome que guarda um valor que pode mudar. No filme, o disco do avô é a memória do Byte: guarda coisas. Numa variável a gente guarda um número, um nome, um estado. O nome da caixinha é fixo; o que está dentro pode trocar.

## Exemplo:

```
emocao_do_byte = "calma"
contador_de_amigos = 0

QUANDO um amigo se aproxima:
    contador_de_amigos = contador_de_amigos + 1
    emocao_do_byte = "alegria"
```

Aqui `emocao_do_byte` e `contador_de_amigos` são variáveis. A primeira guarda uma palavra (um texto), a segunda guarda um número que cresce. Toda vez que um amigo chega, o contador aumenta em 1 e a emoção muda. É assim que um programa "lembra" do que aconteceu.

## 2.5. Um bug, e o ato de depurar

Quando a máquina de Percival explode, é o maior **bug** do filme: o programa (a montagem) tinha um erro, e o erro apareceu ao rodar. Programar é conviver com bugs. **Depurar** (em inglês, *debug*) é caçar e corrigir o erro. O nome tem história: nos anos 1940, uma pioneira da computação registrou no caderno de bordo que um inseto (*bug*) preso numa máquina causava falha, e colocou o inseto ao lado da anotação. Ensine a turma a não ter medo do erro: o erro não é o fim do trabalho, é uma pista. O caderno de bordo do inventor (que reaparece nas fichas) é o caderno de bordo do programador.

---

## 3. Programação em blocos com Scratch (o primeiro código de verdade)

O **Scratch**, criado no MIT para crianças, é a porta de entrada mais amável para a programação. Em vez de digitar texto que precisa de sintaxe perfeita, a criança encaixa blocos coloridos como peças de montar, e o programa roda na hora. Não há erro de vírgula, não há tela preta assustadora: há um gato (ou o nosso Byte) que se move quando os blocos se encaixam.

### 3.1. Por que blocos antes de texto

Digitar `print("olá")` e errar um parêntese trava tudo e desanima. Encaixar um bloco "diga olá" não trava: o bloco só encaixa onde faz sentido. Os blocos ensinam a **lógica** (sequência, repetição, condição, variável) sem o peso da **sintaxe**. Quando a lógica está firme, a passagem para o texto (Python, por exemplo) fica natural. A recomendação é começar em blocos e migrar para texto por volta do 8º e 9º ano, ou quando a turma pedir.

### 3.2. Os tipos de bloco (as cores do Scratch)

- **Eventos** (amarelo): quando começa? "Quando a bandeira verde for clicada".

- **Movimento** (azul): mover, girar, ir para.
- **Aparência** (roxo): mudar de cor, dizer algo, mudar de fantasia (trocar a imagem).
- **Controle** (laranja): repetir, se/senão, esperar. São os laços e as condições.
- **Sensores** (azul-claro): tocou em algo? A tecla foi apertada? O mouse chegou perto?
- **Variáveis** (laranja-escuro): criar e mudar a caixinha da memória.

### 3.3. Descrição de um programa em blocos: o Byte que sente

Como este é um caderno de papel, descrevemos o código em blocos em texto, do jeito que ele apareceria encaixado na tela. Copie no quadro e peça à turma para montar no Scratch (ou para encenar, na versão desplugada).

#### Programa "O humor do Byte":

```
[Quando a bandeira verde for clicada]
  definir (emocao) para [calma]
  mudar a cor do Byte para (verde)

[Sempre] (laço infinito, o Byte está sempre "ligado")
  [Se] <o mouse tocar no Byte> [então]
    definir (emocao) para [alegria]
    mudar a cor do Byte para (amarelo)
    tocar o som (pipoca)
  [Senão, se] <a tecla espaço foi apertada> [então]
    definir (emocao) para [susto]
    mudar a cor do Byte para (azul)
  [Senão]
    definir (emocao) para [calma]
    mudar a cor do Byte para (verde)
```

Repare que este código de brinquedo já tem tudo: um evento que inicia, uma variável (`emocao`), um laço infinito (o "sempre", que mantém o Byte vivo, exatamente como o laço principal de um videogame), e uma cadeia de condições (se/senão) que decide a cor conforme o que acontece. É o mesmo Byte do filme: luzes que mudam de cor conforme o que ele sente. A ficção virou especificação de software.

### 3.4. Um passo além: contar os amigos (variável que cresce)

```
[Quando a bandeira verde for clicada]
```

```
  definir (amigos) para (0)
```

```
[Quando eu receber a mensagem (amigo chegou)]
```

```
  mudar (amigos) por (1)
```

```
  dizer (junte a mão: agora somos) por (2) segundos
```

```
  dizer (amigos)
```

Aqui a variável `amigos` começa em zero e cresce de um em um. É a semente da tese do filme dentro do código: o Byte fica mais forte quanto mais mãos se juntam. A última invenção só acende em círculo.

## 4. Introdução ao maker e ao Arduino (dar comportamento ao Byte)

O Scratch mora na tela. O **maker** tira o código da tela e o coloca no mundo: um LED que acende, um motor que gira, um sensor que percebe. A ponte entre o código e o mundo físico, na escola, costuma ser o **Arduino**.

### 4.1. O que é um Arduino

O Arduino é uma pequena placa de circuito, do tamanho de um cartão, com um cérebro simples (um microcontrolador) que a gente programa. Diferente de um computador, ele não tem tela nem teclado: ele conversa com o mundo por seus **pinos**, pequenos furos metálicos onde se ligam fios. Em uns pinos entram informações (um botão foi apertado? está claro ou escuro?); por outros saem comandos (acenda este LED, gire este motor). O Arduino é, em miniatura, o corpo do Byte: entrada, processamento, saída.

Ele é o parente maker do disco do avô e da carcaça de CPU: um coração eletrônico simples e barato que qualquer escola pode ter, e que transforma a estética de robô de sucata do filme em oficina de verdade. A oficina maker que constroi o Byte, prometida fora da tela pelo projeto do filme, é exatamente esta.

### 4.2. A anatomia do gesto maker

Todo projeto de Arduino repete o mesmo ciclo, que é o ciclo do pensamento computacional aplicado à matéria:

1. **Imaginar o comportamento:** "o LED do Byte deve ficar azul quando ele leva um susto".
2. **Montar o circuito:** ligar o LED e um sensor aos pinos certos, com a atenção de sempre (respeitar a polaridade, usar o resistor que protege o LED).
3. **Escrever o código:** dizer à placa o que fazer com o que entra e o que sai.

4. **Carregar e testar:** enviar o código para a placa e ver acontecer. Se não acender, depurar: é fio solto, é código, é o resistor?
5. **Iterar:** melhorar, e anotar cada mudança no caderno de bordo.

### 4.3. O código do Arduino, por dentro (a estrutura setup e loop)

O programa de Arduino (escrito numa linguagem próxima do C) tem sempre duas partes fixas, e vale a pena entender o porquê:

- `setup()` : roda **uma vez**, quando a placa liga. É onde a gente prepara as coisas: "o pino 13 vai servir de saída para o LED".
- `loop()` : roda **para sempre**, repetindo sem parar enquanto houver energia. É o "sempre" do Scratch, é o coração batendo. Enquanto o Byte estiver ligado, o `loop` roda.

#### O menor programa que dá vida ao Byte: a luz que pisca (o "olá, mundo" do maker).

Comentado linha a linha, para você explicar sem medo:

```
// Isto é um comentário. O computador ignora; serve para a gente entender.
// Programa: o coração do Byte pisca, como um pulso.

int pinoDoLed = 13; // guardamos numa variável o número do pino do LED

void setup() {
  pinMode(pinoDoLed, SAIDA); // avisamos que este pino é de SAÍDA (vai mandar sinal)
}

void loop() {
  digitalWrite(pinoDoLed, LIGADO); // acende o LED
  delay(1000); // espera 1000 milissegundos, ou seja, 1 segundo
  digitalWrite(pinoDoLed, DESLIGADO); // apaga o LED
  delay(1000); // espera mais 1 segundo, e o loop recomeça
}
```

(No Arduino real, `SAIDA` se escreve `OUTPUT`, `LIGADO` é `HIGH` e `DESLIGADO` é `LOW`. Deixamos em português na primeira leitura para não assustar; troque pelos termos reais quando a turma estiver pronta. A ideia é idêntica.)

### 4.4. O LED que muda de cor conforme a emoção (o Byte que sente)

O coração deste caderno, e do filme, é um Byte cujas luzes mudam de cor conforme o que ele sente. Isso se faz com um **LED RGB**, um LED que combina vermelho, verde e azul e, misturando as três, produz qualquer cor, exatamente como a tela do celular faz cada pixel. A "emoção" é uma **variável**; a cor é a **saída**; o que muda a emoção é uma **condição** que lê uma **entrada** (um botão,

um sensor de aproximação, a luz do ambiente). Está tudo aqui: entrada, variável, condição, saída. É a Parte I inteira num só circuito.

### **Pseudocódigo do Byte emocional (a lógica, antes do código):**

```
emocao = "calma"

SEMPRE:
  LER o sensor de aproximação
  SE alguém está muito perto:
    emocao = "alegria"
    cor = amarelo
  SENÃO SE está escuro e sozinho:
    emocao = "medo"
    cor = azul
  SENÃO:
    emocao = "calma"
    cor = verde
  ACENDER o LED RGB na cor escolhida
```

A Sequência 4 deste caderno transforma exatamente este pseudocódigo em circuito e em código de Arduino, passo a passo, com uma versão desplugada completa para quem não tem a placa. Ninguém fica de fora do Byte que sente.

---

## **5. O hardware por dentro (o disco do avô como coração)**

Para entender computação, a turma precisa abrir a caixa. O Byte é o melhor professor de arquitetura de computadores que existe, porque ele é um computador com o corpo à mostra: dá para apontar cada parte.

### **5.1. O modelo entrada, processamento, saída (com memória)**

Todo computador, do supercomputador ao celular ao Byte, faz sempre a mesma dança:

- **Entrada (input):** por onde a informação chega. No computador: teclado, mouse, microfone, câmera, sensor. No Byte: o visor com que ele olha, os "ouvidos" com que percebe.
- **Processamento:** onde a decisão acontece, o cérebro, a **CPU** (Unidade Central de Processamento). É quem executa o algoritmo, quem calcula, quem decide qual cor acender.
- **Memória:** onde a informação fica guardada. Há a memória de trabalho, rápida e temporária (a **RAM**, que esquece tudo quando desliga, como um rascunho), e a memória de arquivo, que guarda mesmo desligado (o **disco rígido** ou HD, e hoje o SSD). O disco do avô, que bate no peito do Byte, é essa memória permanente: guarda quem ele é, mesmo apagado. Não é à toa

que os roteiristas puseram o coração ali. A memória é o coração, porque é o que resta quando a luz se apaga.

- **Saída (output):** por onde o resultado sai para o mundo. No computador: a tela, o alto-falante, a impressora. No Byte: as luzes do peito que mudam de cor, os sons que ele inventa, o gesto da antena.

**A frase-síntese para o quadro:** *entra informação, a CPU decide, a memória guarda, sai um resultado.* Todo computador que existe é uma variação disso.

## 5.2. As peças, com o nome de dentro do gabinete

Vale abrir um gabinete velho (ou mostrar fotos) e nomear:

- **Placa-mãe:** a bancada onde tudo se conecta, o quintal onde as peças se falam.
- **CPU (processador):** o cérebro. Faz bilhões de continhas por segundo, uma de cada vez, muito rápido.
- **RAM:** a mesa de trabalho. Quanto maior, mais coisas o computador segura abertas ao mesmo tempo. Some ao desligar.
- **HD ou SSD:** o armário. Guarda arquivos, fotos, o sistema, para sempre (até apagar). O disco do avô é um HD.
- **Fonte de energia:** o coração que bombeia a eletricidade. No Byte, a pedra e a pilha.
- **Placa de vídeo:** quem desenha as imagens complexas na tela (jogos, edição).
- **Portas de entrada e saída:** as bocas por onde entram e saem os cabos (USB, HDMI).

## 5.3. Bit e byte (e por que o robô se chama Byte)

No fundo de tudo, o computador só entende dois estados: ligado ou desligado, 1 ou 0. Cada um desses é um **bit**, a menor migalha de informação. Junte oito bits e você tem um **byte**, que já dá para guardar uma letra. O nome do robô é uma piada de computação: ele é feito de sucata e disco rígido, e se chama Byte, a unidade básica da informação, a menor peça com que se constroi tudo. Um convite delicioso para ensinar o sistema binário: com só dois símbolos, 0 e 1, arranjados em ordem, o computador representa qualquer número, qualquer letra, qualquer foto, qualquer música. Tudo é combinação de acender e apagar, como as luzes no peito do Byte.

---

## 6. Internet, cidadania e segurança digital

O Byte não vive isolado. No universo do filme, o Byte também é o tutor que ensina a garotada a se virar no mundo digital com juízo (é o coração da plataforma de IA do Byte). Computação na Educação Básica inclui, obrigatoriamente, formar um cidadão digital: alguém que usa a rede com autonomia, senso crítico, ética e segurança.

### 6.1. O que é a internet (e o que é a web)

A **internet** é uma rede de redes: milhões de computadores no mundo inteiro, ligados por cabos (muitos deles no fundo do mar) e por sinais, todos falando a mesma "língua" para trocar informação. A **web** (aquilo que a gente abre no navegador) é só uma parte da internet, as páginas que a gente visita. A informação viaja quebrada em pedacinhos (os **pacotes**), que fazem o caminho separados e se remontam na chegada, como uma carta que fosse enviada palavra por palavra e recomposta no destino. Cada máquina na rede tem um endereço (o **IP**), como cada casa tem o seu. Há uma bela rima com a Ordem do Grande Livro do filme: uma rede de guardiões, espalhados pelo mundo e pelos séculos, passando conhecimento de mão em mão. A internet é a Ordem, feita de fios.

## 6.2. Cidadania digital: os direitos e os deveres na rede

Ser cidadão digital é lembrar que do outro lado da tela há sempre uma pessoa. Os pilares para a turma:

- **Respeito:** o que não se diz na cara não se diz no chat. Combater o *cyberbullying* é dever de todos.
- **Pegada digital:** tudo o que a gente posta deixa rastro e pode ficar para sempre. Antes de publicar, perguntar: eu mostraria isto para a minha avó e para a minha futura chefe?
- **Autoria e crédito:** copiar e colar sem citar a fonte é apropriação. Dar crédito é honestidade (e é a lição da própria Ordem, que registra cada mão que escreveu no Livro).
- **Espírito crítico:** nem tudo que circula é verdade. Antes de repassar, checar. A desinformação se espalha porque a gente compartilha sem ler.

## 6.3. Segurança digital: proteger-se e proteger o outro

- **Senhas fortes:** longas, com letras, números e símbolos, e uma para cada lugar importante. A senha é a chave da casa; não se deixa debaixo do tapete nem se empresta.
- **Dados pessoais são um tesouro:** nome completo, endereço, escola, rotina, foto. Não se entrega a desconhecidos, nem a sites, nem a "amigos" que a gente só conhece pela tela. Aqui mora o alerta mais forte do filme: Frida se disfarça de velhinha gentil, ganha a confiança das crianças e as engana. Na internet, nem todo perfil amável é o que diz ser. O golpe do afeto (o **engenheiro social**, o falso amigo) é o mesmo de Frida, só que por tela.
- **Golpes (phishing):** mensagens que fingem ser do banco, da escola, de um prêmio, para roubar senha ou dinheiro. Desconfiar da urgência ("clique agora ou perde"), conferir o remetente, nunca clicar em link duvidoso.
- **Privacidade e a lei:** no Brasil, a Lei Geral de Proteção de Dados (LGPD) protege os dados das pessoas, e com cuidado redobrado os de crianças e adolescentes. Dado de menor não circula sem consentimento de quem responde por ele.

A Sequência 5 deste caderno transforma tudo isto num projeto: a turma escreve o próprio contrato digital e desenha o mapa da internet.

---

## 7. Uma breve história da computação (com o capítulo brasileiro)

Contar de onde vêm as máquinas é dar à turma o sentido de pertencer a uma corrente, a mesma corrente de guardiões que atravessa o filme. A computação não caiu do céu com o celular: é uma história de muitas mãos, ao longo de séculos, cada uma acrescentando um elo, como Percival acrescenta o dele na última cena.

### 7.1. Antes da eletricidade: contar é humano

- **O ábaco** (há milhares de anos, em várias civilizações): as primeiras contas com pecinhas que deslizam. A primeira "máquina" de calcular é de madeira e não tem fio nenhum.
- **Século XVII:** máquinas de engrenagem que somam e subtraem sozinhas.
- **Século XIX, a virada:** um inventor imagina uma máquina programável de engrenagens, e uma matemática escreve o que muitos consideram o **primeiro algoritmo** feito para uma máquina executar. A programação nasce, no papel, antes do computador existir. Guarde este fato para a turma: a primeira pessoa a programar pensou o software antes de haver hardware, como o avô Severino deixa a máquina montada antes de existir quem a terminasse.

### 7.2. O século XX: a máquina ganha corpo

- **Anos 1930 e 1940:** a teoria do que uma máquina pode ou não calcular é formalizada, e as primeiras máquinas eletrônicas gigantes, do tamanho de uma sala, ajudam em cálculos de guerra. É desta época o registro do primeiro *bug* (o inseto na máquina).
- **Anos 1950 a 1970:** as máquinas encolhem. Surge o **transistor** e depois o **chip**, que espremem numa lasquinha de silício o que antes ocupava uma sala. É a mesma lógica do Arduino: o cérebro inteiro num cartão.
- **Anos 1980:** o **computador pessoal** entra em casa. A computação deixa de ser só de governos e empresas.
- **Anos 1990:** a **internet** e a **web** conectam o mundo. É exatamente a época de 1999, quando se passa o filme: um Brasil de crise, mas já sonhando conectado.
- **Século XXI:** o **smartphone** põe na palma da mão mais poder de cálculo do que as salas inteiras dos anos 1940. E a **inteligência artificial** aprende de exemplos, tema da plataforma do Byte e da sequência de IA do caderno-mãe.

### 7.3. A computação no Brasil (o capítulo que quase não se conta)

O Brasil não foi só espectador desta história. Vale a turma saber:

- **Anos 1960 e 1970:** universidades brasileiras montam os primeiros computadores nacionais e formam as primeiras gerações de cientistas da computação do país, com esforço de projeto próprio, não só de compra.
- **A política de informática:** por um período, o país tentou desenvolver uma indústria nacional de computadores, protegendo o mercado para criar tecnologia própria. Foi uma aposta cheia de acertos e de erros, muito debatida, e um bom tema para uma turma mais velha discutir: vale a

pena um país fazer a própria tecnologia, ou é melhor comprar pronta? Não há resposta fácil, e por isso é boa pergunta.

- **Bancos e urnas:** o Brasil se tornou referência mundial em automação bancária e foi um dos primeiros países a informatizar por completo suas eleições, com a **urna eletrônica**, um sistema de computação de escala nacional pensado e operado aqui.
- **Hoje:** o Brasil forma programadores, tem polos de tecnologia, empresas de software reconhecidas e uma comunidade maker viva, das quais a oficina do Byte é uma prima pequena e possível. A mesma tese do filme, de novo: talento não tem CEP. Há gênio da computação no interior, na periferia, em toda escola que abrir a porta da edícula.

**O fecho para a turma:** você, que aprende computação hoje, é o próximo elo da corrente. Como Percival, que no fim não copia só o que leu: vira a página e começa a escrever a parte que ainda não existe, a que só cabe a ele.

---

---

## Parte II. Sequências didáticas (do desplugado ao Arduino)

---

*Cinco percursos completos, prontos para uso, no formato do caderno-mãe: quadro-resumo, pergunta geradora, conexão com a obra, objetivos, BNCC, materiais, percurso aula a aula com falas mediadoras e "fique de olho", diferenciação, opção desplugada, fichas reprodutíveis, produto, rubrica de quatro níveis, autoavaliação e conexões.*

---

# Sequência 1: O algoritmo do café da Maria (desplugado)

<b>Eixo de Computação</b>	Pensamento Computacional (algoritmo, decomposição, sequência, repetição, condição)
<b>Componentes</b>	Computação, Língua Portuguesa, Matemática
<b>Faixa</b>	4º ao 9º do EF (base para todas as outras sequências)
<b>Duração</b>	4 aulas
<b>Agrupamento</b>	duplas e turma
<b>Produto</b>	o livro de receitas de algoritmos da turma
<b>Tema transversal</b>	Ciência e Tecnologia

**Pergunta geradora:** Como se dá uma ordem a uma máquina que faz exatamente o que a gente diz, nem mais nem menos? **Perguntas de apoio:** O que é um passo a passo perfeito? O que acontece se eu pular um passo? Como uma máquina "sabe" a hora de parar ou de escolher?

**Conexão com a obra.** Maria faz o café e manda o irmão para a escola todo dia, sempre na mesma ordem, porque sempre foi ela quem fez isso. E lê o ritmo do trem num instante, calculando o tempo com o corpo, porque reconheceu o padrão em dezesseis anos de apito. A rotina de Maria é um algoritmo, e a leitura do trem é reconhecimento de padrões. Antes de tocar em qualquer computador, a turma vai programar a Maria, o Byte e a si mesma, só com palavras e com o corpo. Esta é a sequência-base: nenhum computador, todo o pensamento.

## Objetivos de aprendizagem.

- **Conceituais:** compreender o que é um algoritmo; distinguir sequência, repetição e condição; entender que a máquina é literal.
- **Procedimentais:** decompor uma tarefa em passos; escrever um algoritmo sem ambiguidade; testar e depurar o próprio algoritmo.
- **Atitudinais:** valorizar a clareza e a precisão; encarar o erro como pista; colaborar na revisão do trabalho do colega.

## BNCC.

- Competências gerais: 1 (conhecimento), 2 (pensamento científico, crítico e criativo), 4 (comunicação), 5 (cultura digital), 9 (empatia e cooperação).
- Computação (eixo Pensamento Computacional: decomposição, algoritmos, reconhecimento de padrões). Língua Portuguesa (o gênero instrucional, o texto de procedimento; clareza e coesão). Matemática (sequências, sucessão de passos, lógica).
- *(Códigos de habilidade por ano na versão da rede.)*

**Materiais.** Cartões ou tiras de papel para escrever passos, o quadro, um objeto simples para "programar" (fazer um sanduíche, desenhar uma figura), as fichas 1A e 1B. Nenhum computador é necessário.

## Percurso

**Aula 1, Sensibilização (a máquina obediente e boba).** ⌚ 50 min · 👥 turma Conte a rotina da Maria e o momento em que ela lê o trem. Depois, o professor vira "robô" e anuncia: "Eu sou uma máquina. Faço exatamente o que vocês mandarem, ao pé da letra. Me ensinem a fazer um sanduíche." A turma dá ordens; o professor executa ao pé da letra, com humor: se disserem só "passe a manteiga", passe a manteiga na mão, no pão fechado, no lugar errado, até virem que faltou detalhe. 💬 "Eu fiz errado? Não, eu fiz exatamente o que vocês disseram. A máquina não adivinha. O que faltou na ordem de vocês?" 👁️ Fique de olho em quem percebe primeiro que a ordem precisa ser completa e na sequência certa: é o nascimento da ideia de algoritmo.

**Aula 2, Investigação (decompor e sequenciar).** ⌚ 50 min · 👥 duplas Apresente a palavra **algoritmo**: um passo a passo claro que qualquer um segue e chega ao mesmo resultado. As duplas escolhem uma tarefa do dia (escovar os dentes, chegar da porta da escola até a sala, o café da Maria) e a **decompõem** em passos numerados na 📄 Ficha 1A. Depois trocam com outra dupla, que executa "como robô", ao pé da letra, e marca onde travou. 💬 "Onde o robô da outra dupla travou? Esse ponto é um bug. Como a gente conserta a receita?"

**Aula 3, Investigação (repetição e condição).** ⌚ 50 min · 👥 turma e duplas Introduza os dois blocos novos com o corpo. **Repetição**: "pisque os olhos 3 vezes", "bata palma enquanto eu levantar a mão" (repita até uma condição). **Condição**: brincadeira do "se/senão": se você está de tênis, levante a mão; *senão*, fique sentado. A turma percebe que a máquina decide olhando uma condição. Reescrevem um algoritmo da Aula 2 usando um "repita" e um "se". 💬 "No café da Maria, o que se repete? O que depende de uma condição (o café ficou fraco ou forte)?" 👁️ Fique de olho em quem confunde "repita 3 vezes" com "repita enquanto": nomear a diferença é o aprendizado.

**Aula 4, Produção e Socialização (o livro de receitas de algoritmos).** ⌚ 50 min · 👥 turma Cada dupla finaliza um algoritmo caprichado (com sequência e, quando couber, repetição e condição) na 📄 Ficha 1B, para virar página do livro coletivo. Um algoritmo é lido em voz alta enquanto um colega o executa como robô diante da turma: se rodar sem travar, "compilou". Montagem do livro de receitas de algoritmos da turma. 💬 "O que faz um algoritmo ser bom? Ele é claro, completo e não deixa a máquina adivinhar."

## Diferenciação

- **4º e 5º:** tarefas do cotidiano, foco na sequência e na clareza; a repetição e a condição entram só como brincadeira de corpo.
- **6º e 7º:** algoritmos com repetição e condição por escrito; introdução do pseudocódigo simples.
- **8º e 9º:** pseudocódigo estruturado (INÍCIO, FIM, ENQUANTO, SE, SENÃO) e um algoritmo com variável (um contador). Ponte direta para a Sequência 2 (Scratch).

## Esta sequência inteira é desplugada.

Não precisa de nenhum equipamento. É a fundação para todas as outras e a garantia de que a computação começa acessível a toda escola. Onde houver computador, ela prepara a turma para programar com a lógica já firme.

### Fichas reproduzíveis

#### Ficha 1A, Decompondo uma tarefa

- A tarefa que vamos programar: \_\_\_\_\_
- Os passos, em ordem (numere): 1. \_\_\_\_\_ 2. \_\_\_\_\_ 3. \_\_\_\_\_ ...
- Onde o robô da outra dupla travou (o bug): \_\_\_\_\_
- Como consertamos: \_\_\_\_\_

#### Ficha 1B, A nossa receita de algoritmo

- Nome do algoritmo: \_\_\_\_\_
- INÍCIO
- Passos (com repetição e condição, se precisar): \_\_\_\_\_
- FIM
- Testamos com um colega-robô? ( ) sim, rodou ( ) travou e corrigimos

### Produto final

O livro (ou mural) de receitas de algoritmos da turma, uma página por dupla.

### Rubrica de avaliação

<b>Critério</b>	<b>1 Em início</b>	<b>2 Em desenvolvimento</b>	<b>3 Adequado</b>	<b>4 Excelente</b>
Decomposição	Não separa em passos	Passos incompletos ou fora de ordem	Passos completos e na ordem	Decompõe com precisão e elegância
Sequência, repetição, condição	Só ordens soltas	Usa sequência	Usa repetição ou condição corretamente	Combina os três com clareza
Depuração	Não percebe o erro	Percebe mas não corrige	Encontra e corrige o bug	Antecipa o erro e testa antes
Clareza da escrita	Ambígua	Precisa de ajuda para entender	Clara e sem ambiguidade	Clara, completa e reaproveitável

### Autoavaliação do estudante

Eu quebrei a tarefa em passos. Eu achei e consertei um bug. Eu sei o que é sequência, repetição e condição. Uma ordem que eu daria melhor da próxima vez: \_\_\_\_\_.

## **Com a família**

Peça em casa uma "receita da família" (de comida, de arrumar a mochila, de qualquer rotina) e escreva como algoritmo, com passos numerados. Toda casa é cheia de algoritmos que ninguém escreveu.

## **Conexões**

É a base de tudo. Liga direto com a Sequência 2 (o mesmo algoritmo vira código em blocos) e com a 4 (o pseudocódigo do Byte emocional vira circuito). Conversa com a Matemática (lógica e sequências) e com a Sequência 3 do caderno-mãe (a Alice decifra o grego com um algoritmo, letra por letra).

---

---

# Sequência 2: O Byte ganha comportamento (Scratch)

<b>Eixo de Computação</b>	Pensamento Computacional e Cultura Digital (programação em blocos)
<b>Componentes</b>	Computação, Arte, Língua Portuguesa
<b>Faixa</b>	5º ao 9º do EF
<b>Duração</b>	6 aulas mais culminância
<b>Agrupamento</b>	duplas (programação em par)
<b>Produto</b>	o Byte animado que sente e responde, no Scratch
<b>Tema transversal</b>	Ciência e Tecnologia

**Pergunta geradora:** Como fazer o Byte, na tela, mudar de cor conforme o que ele "sente"?

**Perguntas de apoio:** O que faz um personagem parecer vivo num programa? Como o programa decide o que fazer? Como o computador "lembra" de uma coisa?

**Conexão com a obra.** As luzes no peito do Byte mudam de cor conforme o que ele sente: azul de susto, tudo aceso de alegria quando prova a pipoca. Isso é uma especificação de software, e a turma vai implementá-la. No Scratch, o Byte da turma vai ganhar exatamente esse comportamento: uma variável de emoção, condições que a mudam, uma cor para cada sentimento. A ficção do filme vira o programa da turma.

## Objetivos de aprendizagem.

- **Conceituais:** compreender eventos, sequência, repetição (laço), condição (se/senão) e variável, num ambiente de blocos.
- **Procedimentais:** construir um programa que responde a entradas; usar variável para guardar um estado; depurar o próprio código.
- **Atitudinais:** programar em par com escuta e revezamento; persistir diante do erro; cuidar da clareza do código.

## BNCC.

- Competências gerais: 1, 2, 4, 5 (cultura digital), 9, 10 (responsabilidade).
- Computação (eixo Pensamento Computacional: algoritmos, depuração; eixo Cultura Digital: criação com tecnologias digitais). Arte (design do personagem, cor e som). Língua Portuguesa (roteiro e narrativa da animação).
- *(Códigos de habilidade por ano na versão da rede.)*

**Materiais.** Computadores ou tablets com Scratch (versão online ou a versão offline, que roda sem internet), o projetor, as fichas 2A e 2B, o algoritmo da Sequência 1 pronto. 🟢 Onde não houver máquinas, ver a opção desplugada (o Scratch de papel).

## Percurso

**Aula 1, Sensibilização e primeiro contato.** ⌚ 50 min · 👥 turma e duplas Exiba a cena das luzes do Byte. Mostre o Scratch no projetor: apresente os tipos de bloco (evento, movimento, aparência, controle, variável) sem teoria longa, encaixando ao vivo três blocos que fazem um personagem se mover e falar. Regra da **programação em par**: dois numa máquina, um dirige (mexe), o outro navega (lê e sugere), e trocam a cada rodada. 💬 "Reparem: eu não escrevi nada, eu encaixei. O bloco só entra onde faz sentido. Errar aqui não trava, é só desencaixar."

**Aula 2, Investigação (sequência e eventos).** ⌚ 50 min · 👥 duplas As duplas fazem o sprite do Byte (desenhado ou uma imagem simples) aparecer, se mover e "dizer olá" quando a bandeira verde é clicada. Reforce que o programa segue a ordem dos blocos: mudar a ordem muda o resultado. Registro do que cada bloco faz na 📄 Ficha 2A. 💬 "O que é o evento que faz o programa começar? E se eu trocar dois blocos de lugar?"

**Aula 3, Produção (o laço: o Byte está sempre ligado).** ⌚ 50 min · 👥 duplas Introduza o bloco **sempre** (laço infinito) e o **repita N vezes**. As duplas fazem o Byte piscar a luz 3 vezes (repita) e depois "ficar vivo", trocando de cor devagar dentro de um "sempre". 💬 "Por que o Byte precisa de um 'sempre'? Porque uma máquina ligada nunca para de checar o mundo. É o coração batendo." 🧠 Fique de olho em quem cria um "sempre" sem uma pausa e trava a tela: ótimo momento para ensinar o `esperar` e a depurar juntos.

**Aula 4, Produção (a condição: o Byte decide).** ⌚ 50 min · 👥 duplas O bloco **se/senão** e os **sensores** (tocou no mouse? tecla apertada?). As duplas fazem o Byte mudar de cor conforme a interação: perto do mouse, amarelo (alegria); tecla espaço, azul (susto); em repouso, verde (calma). É o programa "O humor do Byte" da Parte I. 💬 "Agora o Byte escolhe. Quem decide a cor? A condição. O que ele lê para decidir? O sensor."

**Aula 5, Produção (a variável: o Byte lembra).** ⌚ 50 min · 👥 duplas Crie a variável `emocao` (guarda a palavra do sentimento) e uma variável `amigos` que cresce a cada interação (o contador da Parte I). O Byte mostra na tela quantos amigos já chegaram e fica "mais alegre" quanto mais amigos. 💬 "A variável é a memória do programa, o disco do avô do Byte. O que a gente quer que ele guarde e lembre?" Registro do código na 📄 Ficha 2B.

**Aula 6, Socialização (culminância).** ⌚ 50 min · 👥 turma Feira de Bytes: cada dupla apresenta o seu Byte animado, mostra o código e explica uma decisão que ele toma. Roda final. 💬 "O nosso Byte de tela sente de mentira. O que faria um Byte de verdade sentir? Guardem a pergunta: a Sequência 4 responde com o Arduino."

## Diferenciação

- **5º e 6º:** foco em eventos, sequência e um laço; a condição entra simples (só um "se").
- **7º:** se/senão completo, sensores, uma variável.
- **8º e 9º:** duas ou três variáveis, mensagens entre sprites (o Byte conversa com outro personagem), e o desafio de refatorar o código para ficar mais limpo. Ponte para o texto: mostre o mesmo programinha em Python, lado a lado, para quem quiser ir além.

## **Opção desplugada: o Scratch de papel**

Sem computador, a lógica é idêntica. Faça **blocos de papel** recortados (eventos amarelos, controle laranja, aparência roxos), com velcro ou fita, numa cartolina "palco". Uma criança é o Byte; a dupla monta a sequência de blocos de papel na parede, e a criança-Byte **executa** o que os blocos mandam, na ordem, incluindo repetições e condições ("se o colega chegar perto, mude para o cartão amarelo"). A turma depura apontando onde o Byte-humano travou. Aprende-se exatamente sequência, laço, condição e variável, sem uma única tela.

### **Fichas reproduzíveis**

#### **Ficha 2A, O que cada bloco faz**

- Bloco de evento que usei: \_\_\_\_\_ (o que ele faz: \_\_\_\_\_)
- Bloco de movimento ou aparência: \_\_\_\_\_ (o que ele faz: \_\_\_\_\_)
- A ordem importou? O que mudou quando troquei a ordem: \_\_\_\_\_

#### **Ficha 2B, O código do meu Byte**

- Emoções do meu Byte e a cor de cada uma: \_\_\_\_\_
- A condição que muda a emoção: SE \_\_\_\_\_ ENTÃO cor \_\_\_\_\_
- A variável que ele guarda: \_\_\_\_\_
- Um bug que apareceu e como corrigi: \_\_\_\_\_

### **Produto final**

O Byte animado da dupla no Scratch (ou o Scratch de papel encenado), com pelo menos uma variável, um laço e uma condição, e o registro do código.

### **Rubrica de avaliação**

<b>Critério</b>	<b>1 Em início</b>	<b>2 Em desenvolvimento</b>	<b>3 Adequado</b>	<b>4 Excelente</b>
Lógica (laço e condição)	Só blocos em sequência	Usa laço ou condição com ajuda	Usa laço e condição corretos	Combina laços e condições aninhadas com intenção
Variável	Não usa	Cria mas não usa direito	Usa uma variável com sentido	Usa variável que muda o comportamento de forma expressiva
Depuração	Desiste no erro	Pede que o professor conserte	Encontra e corrige o próprio bug	Isola a causa e explica a correção
Autoria e conexão com a obra	Cópia sem sentido	Byte genérico	Byte que sente, fiel à ideia do filme	Byte expressivo, com narrativa própria e código limpo

### **Autoavaliação do estudante**

Eu fiz o Byte responder a uma condição. Eu usei uma variável. Eu depurei um bug sozinho ou com meu par. Uma coisa que eu quero programar depois: \_\_\_\_\_.

### **Segurança e privacidade (leia antes)**

Se usar o Scratch online, prefira contas de turma administradas pelo professor, nunca contas pessoais das crianças com dados reais. Nada de nome completo, escola ou foto de menor exposto publicamente. A versão offline dispensa internet e é a mais segura.

### **Conexões**

Recebe a lógica da Sequência 1 (o algoritmo desplugado vira blocos) e entrega para a 4 (o mesmo Byte que sente, agora físico com Arduino). Conversa com Arte (design e som do personagem) e com a Sequência 1 do caderno-mãe (a construção maker do Byte).

---

---

# Sequência 3: O coração do Byte (hardware e maker)

<b>Eixo de Computação</b>	Mundo Digital (arquitetura de computadores, hardware)
<b>Componentes</b>	Computação, Ciências da Natureza, Arte
<b>Faixa</b>	6º ao 9º do EF e Ensino Médio
<b>Duração</b>	5 aulas
<b>Agrupamento</b>	grupos de 3 a 4
<b>Produto</b>	o pôster-máquina "por dentro do Byte" e o Byte de luz (LED simples ou de papel)
<b>Tema transversal</b>	Ciência e Tecnologia

**Pergunta geradora:** O que existe por dentro de uma máquina que "pensa", e por que o coração do Byte é um disco rígido? **Perguntas de apoio:** Por onde a informação entra, é processada, é guardada e sai? O que a máquina esquece quando desliga, e o que ela guarda? O que é um bit?

**Conexão com a obra.** O Byte nasce da carcaça de uma CPU, de lata e parafuso, e no centro do peito bate, girando, o disco rígido do avô: onde devia bater um coração, bate a memória. Os roteiristas puseram a memória permanente no lugar do coração, e isso é uma aula inteira de arquitetura de computadores. A turma vai abrir a máquina (de verdade ou em imagem), nomear cada parte e entender por que o disco do avô é, com toda a razão, o coração.

## Objetivos de aprendizagem.

- **Conceituais:** compreender o modelo entrada, processamento, memória, saída; nomear as peças do computador (CPU, RAM, disco, placa-mãe, fonte, portas); distinguir memória temporária de permanente; entender bit e byte.
- **Procedimentais:** identificar as partes num equipamento real ou em imagem; montar um circuito simples de LED; relacionar cada parte do Byte a uma função de computador.
- **Atitudinais:** cuidado ao manusear peças; trabalho em equipe; olhar de reaproveitamento (a sucata que vira máquina).

## BNCC.

- Competências gerais: 1, 2, 4, 5, 9.
- Computação (eixo Mundo Digital: funcionamento dos dispositivos, hardware, representação da informação, bit e byte). Ciências da Natureza (Matéria e energia: eletricidade e circuito simples). Arte (materialidade e projeto do robô de sucata).
- *(Códigos de habilidade por ano na versão da rede.)*

**Materiais.** Um gabinete ou notebook velho para abrir (ou fotos grandes das peças), peças soltas se houver (um HD, um pente de RAM, um processador), cartolina para o pôster, e, para o Byte de luz:

LED, pilha de 3V, fita ou fio; onde houver, resistor. 📌 Sem material elétrico, ver a opção desplugada.

## Percurso

**Aula 1, Sensibilização (abrir o coração).** ⌚ 50 min · 👥 turma Leia o nascimento do Byte, com foco no disco do avô batendo no peito. Depois, abra o gabinete velho diante da turma (ou projete as fotos). Silêncio de museu: é a máquina por dentro. 💬 "Onde vocês acham que fica o cérebro desta máquina? E a memória? Por que será que o filme pôs a memória, o disco, no lugar do coração?" 👁️ Fique de olho nas hipóteses: elas ancoram a aula toda.

**Aula 2, Investigação (entrada, processamento, memória, saída).** ⌚ 50 min · 👥 grupos Apresente o modelo com exemplos do corpo e do computador: os olhos são entrada, o cérebro processa, a lembrança é memória, a fala é saída. No computador: teclado e câmera (entrada), CPU (processamento), RAM e disco (memória), tela e alto-falante (saída). Cada grupo classifica uma lista de peças e de partes do Byte na 📄 Ficha 3A. 💬 "O visor do Byte é entrada ou saída? E a luz do peito? E o disco do avô?"

**Aula 3, Investigação (o que a máquina esquece, o que ela guarda: bit e byte).** ⌚ 50 min · 👥 grupos A diferença entre RAM (rascunho, some ao desligar) e disco (armário, guarda para sempre). Por que o disco do avô é o coração: é o que resta quando a luz apaga. Introduza **bit** (0 ou 1, aceso ou apagado) e **byte** (oito bits, dá uma letra), e a piada do nome do robô. Uma brincadeira de binário: com cartões de "aceso" e "apagado", a turma representa números só ligando e desligando luzes. 💬 "Se a máquina só entende aceso e apagado, como ela guarda uma letra inteira? Uma foto? Uma música? Tudo é combinação de acender e apagar, como as luzes do Byte."

**Aula 4, Produção (o Byte de luz: o menor circuito).** ⌚ 50 min · 👥 grupos Monte o circuito mais simples: um LED, uma pilha, um fio, para acender uma luz no peito de um Byte de papelão. O que é preciso para acender? Um caminho fechado para a corrente. Feche o circuito, abra o circuito. (Com resistor, onde houver, e sempre respeitando a polaridade do LED, a perninha maior no positivo.) 💬 "Acendeu. O que aconteceu quando eu abri o fio? Essa luz é a saída do Byte. É por aqui que, na próxima sequência, ele vai mostrar o que sente." 👁️ Fique de olho em quem inverte a polaridade do LED: é o primeiro "bug de hardware", e um ótimo momento de depuração física.

**Aula 5, Socialização (culminância).** ⌚ 50 min · 👥 grupos e turma Cada grupo finaliza o pôster-máquina "por dentro do Byte" (as partes nomeadas, cada uma ligada à função e à parte correspondente do robô) e mostra o Byte de papelão com a luz acesa. Roda. 💬 "Se vocês fossem consertar um computador que não guarda os arquivos, qual peça olhariam primeiro? E o que o Byte nos ensinou sobre memória e coração?"

## Diferenciação

- **6º e 7º:** foco no modelo entrada, processamento, saída e em nomear as peças; circuito de LED simples.
- **8º e 9º:** memória temporária contra permanente, bit e byte, uma conversão simples de binário; circuito com resistor.

- **Ensino Médio:** aprofundar em como a CPU executa instruções passo a passo (a relação com o algoritmo), e discutir lixo eletrônico e reaproveitamento (a estética de sucata do Byte como ética ambiental).

### **Opção desplugada**

Todo o hardware pode ser ensinado com **fotos grandes** das peças e com peças velhas sem energia, só para manusear e nomear. O circuito de LED, onde faltar material, vira o **jogo do circuito humano**: a turma em roda de mãos dadas é o "fio"; quando alguém solta a mão (abre o circuito), a "corrente" (uma bolinha passada de mão em mão, ou um aperto) para de circular e a "luz apaga". A brincadeira do binário com cartões de aceso e apagado também é 100% desplugada.

### **Fichas reprodutíveis**

#### **Ficha 3A, Anatomia da máquina**

- Peça ou parte do Byte: \_\_\_\_\_
- É entrada, processamento, memória ou saída? \_\_\_\_\_
- A parte do Byte que corresponde: \_\_\_\_\_
- Guarda ao desligar? ( ) sim (permanente) ( ) não (temporária)

#### **Ficha 3B, O nosso circuito**

- O que ligamos: \_\_\_\_\_
- Acendeu? ( ) sim ( ) não. Se não, o que corrigimos: \_\_\_\_\_
- Por que a luz apaga quando abrimos o fio: \_\_\_\_\_

### **Produto final**

O pôster-máquina "por dentro do Byte" e o Byte de papelão com a luz do peito acesa (o circuito real ou o circuito humano, na versão desplugada).

### **Rubrica de avaliação**

<b>Critério</b>	<b>1 Em início</b>	<b>2 Em desenvolvimento</b>	<b>3 Adequado</b>	<b>4 Excelente</b>
Modelo do computador	Não distingue as funções	Distingue com ajuda	Classifica entrada, processamento, memória e saída	Explica o fluxo e transfere a novos aparelhos
Peças e memória	Não nomeia	Nomeia algumas	Nomeia as peças e distingue RAM de disco	Explica por que o disco é o "coração" com precisão
Bit, byte e binário	Não compreende	Repete a definição	Explica bit e byte com exemplo	Converte um número simples em binário

<b>Critério</b>	<b>1 Em início</b>	<b>2 Em desenvolvimento</b>	<b>3 Adequado</b>	<b>4 Excelente</b>
Circuito e trabalho em equipe	Não monta	Monta com muita ajuda	Monta o circuito e divide tarefas	Depura o circuito e coordena o grupo

### **Autoavaliação do estudante**

Eu sei por onde a informação entra e sai. Eu sei o que a máquina esquece ao desligar. Eu montei (ou encenei) um circuito. Uma peça que eu quero entender melhor: \_\_\_\_\_.

### **Com a família**

Peça em casa uma peça de eletrônico velho "com história" (um cabo, um pen drive queimado, uma peça de um aparelho antigo) e o que ela fazia. O Byte é feito da sucata que a casa descarta; o pôster da turma também pode ser.

### **Conexões**

Recebe da Sequência 1 (a lógica) e prepara a 4 (o mesmo LED, agora comandado por código).  
Conversa com Ciências (eletricidade) e com a Sequência 1 do caderno-mãe (a construção do Byte de sucata).

---

---

# Sequência 4: O LED que sente: dar vida ao Byte com Arduino

<b>Eixo de Computação</b>	Mundo Digital e Pensamento Computacional (programação física, maker)
<b>Componentes</b>	Computação, Ciências da Natureza, Arte, Matemática
<b>Faixa</b>	8º e 9º do EF e Ensino Médio (versão desplugada a partir do 6º)
<b>Duração</b>	6 aulas mais culminância
<b>Agrupamento</b>	grupos de 3 a 4 (bancadas maker)
<b>Produto</b>	o Byte físico com humor de LED (ou o Byte desplugado que sente)
<b>Tema transversal</b>	Ciência e Tecnologia

**Pergunta geradora:** Como fazer uma luz de verdade mudar de cor conforme uma "emoção", com código e circuito? **Perguntas de apoio:** Como o código sai da tela e vira ação no mundo? O que a placa lê (entrada) e o que ela comanda (saída)? Como uma condição no código vira uma cor no LED?

**Conexão com a obra.** É o coração do caderno e do filme. O Byte tem luzes no peito que mudam de cor conforme o que ele sente: azul de susto, tudo aceso de alegria. A turma vai construir isso de verdade: um LED (ou LED RGB) que muda de cor conforme uma entrada, comandado por um Arduino. A "emoção" é uma variável, a cor é a saída, o gatilho é uma condição que lê um sensor ou um botão. A oficina maker que constroi o Byte, prometida fora da tela pelo projeto do filme, acontece aqui, na sua sala.

## Objetivos de aprendizagem.

- **Conceituais:** compreender a estrutura setup e loop; relacionar entrada digital, condição e saída; entender o LED RGB como mistura de cores (a mesma do pixel).
- **Procedimentais:** montar um circuito com LED e botão (ou sensor) na protoboard; escrever, carregar e depurar um código de Arduino; iterar a partir de testes.
- **Atitudinais:** rigor e cuidado no circuito (polaridade, resistor); persistência na depuração; colaboração de bancada.

## BNCC.

- Competências gerais: 1, 2, 4, 5, 9, 10.
- Computação (eixo Mundo Digital: programação de dispositivos, entrada e saída; eixo Pensamento Computacional: algoritmos, depuração). Ciências (circuitos, corrente, resistência). Matemática (a mistura RGB como combinação numérica de 0 a 255). Arte (design do Byte).
- *(Códigos de habilidade por ano na versão da rede.)*

**Materiais (versão plugada).** Placas Arduino (uma por bancada), cabos USB, protoboard, LEDs (comuns e, se possível, RGB), resistores, botões, jumpers, um computador com o ambiente do Arduino instalado. Onde possível, um sensor simples (de luz ou de distância). 📱 Sem placas, a versão desplugada abaixo cobre todos os conceitos.

## Percurso

**Aula 1, Sensibilização e o pseudocódigo (a ponte do Scratch para o mundo).** ⌚ 50 min · 👥 turma Retome o Byte do Scratch (Sequência 2) e a cena do filme. Anuncie o salto: "Agora a luz sai da tela." Apresente a placa Arduino e a ideia de entrada e saída pelos pinos. A turma escreve, junta, o **pseudocódigo do Byte emocional** (o da Parte I) no quadro, antes de qualquer fio. Pensa-se a lógica primeiro. 💬 "Antes de montar nada, o que o Byte precisa ler para decidir a cor? O que ele faz com o que leu? Isto é entrada, condição e saída."

**Aula 2, Investigação (setup, loop e a luz que pisca).** ⌚ 50 min · 👥 grupos O menor programa: um LED que pisca (o "olá, mundo" do maker, da Parte I). Cada bancada monta o circuito (LED, resistor, dois pinos) e carrega o código de piscar. Explique `setup` (roda uma vez, prepara) e `loop` (roda para sempre, o coração). 💬 "O `loop` é o 'sempre' do Scratch. Por que uma máquina ligada nunca para de repetir o loop?" 👁️ Fique de olho na polaridade do LED e no resistor: o primeiro bug costuma ser físico. Depurar o circuito é parte da aula.

**Aula 3, Produção (a entrada: o botão que muda tudo).** ⌚ 50 min · 👥 grupos Adicione um **botão** (entrada digital). O código lê o botão: `SE o botão está apertado, acenda; SENÃO, apague`. A condição do código vira ação no mundo. Registro na 📄 Ficha 4A. 💬 "O que a placa lê agora? O botão. Onde no código está a decisão? No 'se'. É a mesma condição do Scratch, agora com um dedo de verdade apertando."

**Aula 4, Produção (o LED RGB: a cor da emoção).** ⌚ 50 min · 👥 grupos Apresente o **LED RGB**: misturando vermelho, verde e azul (cada um de 0 a 255) sai qualquer cor, como cada pixel da tela. A turma define a paleta de emoções do Byte (verde calma, amarelo alegria, azul medo) e programa o LED para acender a cor conforme o botão ou o sensor. 💬 "Amarelo é vermelho mais verde acesos juntos. A cor é matemática: três números de 0 a 255. A emoção virou número, e o número virou luz."

**Aula 5, Produção (o Byte que sente, montagem final).** ⌚ 50 min · 👥 grupos Junte tudo no corpo do Byte (a carcaça de papelão ou sucata da Sequência 3): a variável `emocao`, a condição que a muda conforme a entrada, o LED RGB como peito que muda de cor. Cada bancada dá ao seu Byte um "humor" e o depura até funcionar. Registro do código na 📄 Ficha 4B. 👁️ Fique de olho na iteração: quem testa, ajusta e anota melhora mais rápido. O caderno de bordo do inventor é o caderno de bordo do programador.

**Aula 6, Socialização (culminância).** ⌚ 50 min · 👥 turma e comunidade A oficina do Byte se abre: cada grupo demonstra o seu Byte que sente, mostra o circuito e o código, e explica uma decisão de projeto. Convide outra turma ou as famílias. É a câmara secreta do filme repetida em escala menor na escola. 💬 "O Byte de vocês sente de verdade, ou ele mostra o que a gente programou"

que era para mostrar? Guardem essa pergunta: ela é o começo do debate sobre inteligência artificial."

## Diferenciação

- **8º ano:** LED simples com botão (entrada e saída digital, uma condição). O RGB entra como demonstração.
- **9º ano:** LED RGB com paleta de emoções e duas ou três condições encadeadas.
- **Ensino Médio:** acrescente um sensor real (luz ou distância) como entrada analógica, discuta a mistura RGB de 0 a 255 como número, e abra o debate: um LED que "sente" sente mesmo? O que separa a máquina que responde da pessoa que sente? (Conexão direta com a IA e com a tese do filme.)

### **Opção desplugada: o Byte que sente, sem placa (do 6º ano)**

Todos os conceitos de entrada, condição, variável e saída se ensinam sem Arduino:

- **O Byte humano de cartões de cor.** Uma criança é o Byte, com quatro cartões coloridos na mão (verde, amarelo, azul, vermelho), cada cor uma emoção. Os colegas são os "sensores": um encena "chegar perto", outro "fazer barulho de susto", outro "deixar sozinho". A turma escreve no quadro as **regras** (o programa): "SE alguém chega perto, levante o cartão amarelo; SENÃO, se ficar sozinho, o roxo". A criança-Byte executa as regras conforme os sensores encenam. Isto é, literalmente, o mesmo código, rodado por gente.
- **A variável de papel.** Uma "caixinha da emoção" de cartolina, onde se troca o cartão que está guardado ali dentro conforme as regras: a variável, à vista de todos.
- **Depurar no papel.** Quando a criança-Byte faz a cor "errada", a turma volta às regras escritas e descobre qual condição estava mal escrita. Depuração pura, sem eletrônica. A versão desplugada cobre 100% da lógica; o Arduino, quando houver, só dá corpo elétrico ao que a turma já entendeu.

### **Fichas reproduzíveis**

#### **Ficha 4A, Entrada e saída do meu Byte**

- O que o Byte lê (a entrada): \_\_\_\_\_
- A condição (SE \_\_\_\_\_ ENTÃO \_\_\_\_\_)
- O que o Byte faz (a saída): \_\_\_\_\_
- O primeiro bug (físico ou de código) e como corrigimos: \_\_\_\_\_

#### **Ficha 4B, O código do Byte que sente**

- As emoções e a cor (RGB, se usar) de cada uma: \_\_\_\_\_
- setup (o que preparamos uma vez): \_\_\_\_\_
- loop (o que se repete sempre): \_\_\_\_\_
- Quantas vezes testamos até funcionar: \_\_\_\_\_

## Produto final

O Byte físico que muda a cor da luz conforme uma entrada, com o código e o circuito documentados; ou, na versão desplugada, o Byte humano de cartões executando as regras escritas pela turma, com o registro do "programa".

### Rubrica de avaliação

Critério	1 Em início	2 Em desenvolvimento	3 Adequado	4 Excelente
Circuito e maker	Não monta	Monta com muita ajuda	Monta o circuito correto	Monta, cuida da polaridade e melhora o projeto
Lógica (entrada, condição, saída)	Não relaciona	Relaciona com ajuda	Liga a condição à saída corretamente	Encadeia condições e explica o fluxo com precisão
Depuração	Desiste no erro	Pede ajuda para todo erro	Encontra e corrige o próprio bug	Isola causa (fio ou código) e documenta a correção
Colaboração e documentação	Trabalha isolado	Participa pouco	Divide a bancada e registra	Coordena, itera e deixa o projeto reaproveitável

### Autoavaliação do estudante

Eu liguei uma entrada a uma saída. Eu fiz a luz mudar com uma condição. Eu depurei um bug (de fio ou de código). Um comportamento novo que eu daria ao Byte: \_\_\_\_\_.

### Segurança (leia antes)

Arduino trabalha com tensão baixa (5V), sem risco de choque. Os cuidados são com as peças: não ligar fio direto de um pino ao outro sem componente (curto), respeitar a polaridade do LED, usar o resistor. Ferro de solda, só com o professor e com supervisão. A bancada é maker, então vale a regra do laboratório: cada coisa no seu lugar, e mão que mexe é mão que anota.

### Conexões

É o ápice da progressão: recebe a lógica da Sequência 1, o código da 2, o circuito e o hardware da 3, e junta tudo num Byte físico. Abre para a Sequência 5 (o Byte na rede) e para a Sequência de IA do caderno-mãe (o Byte que "pensa"). Conversa com a Matemática (RGB como número) e com Ciências (circuito).

---

---

# Sequência 5: A rede da Ordem: internet, cidadania e segurança digital

<b>Eixo de Computação</b>	Cultura Digital (internet, cidadania, segurança)
<b>Componentes</b>	Computação, Língua Portuguesa, Geografia
<b>Faixa</b>	6º ao 9º do EF e Ensino Médio
<b>Duração</b>	5 aulas
<b>Agrupamento</b>	duplas e turma
<b>Produto</b>	o contrato digital da turma e o mapa da internet
<b>Tema transversal</b>	Ciência e Tecnologia; Educação em Direitos Humanos

**Pergunta geradora:** Como usar a internet com liberdade, sem cair nas armadilhas de quem se disfarça de amigo? **Perguntas de apoio:** O que é a internet, por onde a informação viaja? Como sei em quem confiar na tela? O que é meu e não se entrega?

**Conexão com a obra.** As crianças pedem socorro a uma velhinha de sorriso manso que as afaga e as esconde, e agradecem a sorte de existir gente boa assim. Era Frida, a vilã, disfarçada, ganhando a confiança delas para chegar ao que queria. É a aula mais importante de segurança digital que um filme pode dar: nem todo perfil gentil na tela é o que diz ser. E a Ordem do Grande Livro, uma rede de guardiões espalhados pelo mundo e pelos séculos, passando conhecimento de mão em mão, é uma bela imagem do que a internet poderia ser no seu melhor: uma rede que une. A turma vai entender a rede e aprender a se proteger dos disfarces de Frida.

## Objetivos de aprendizagem.

- **Conceituais:** compreender o que é a internet e como a informação viaja (pacotes, endereço, rede de redes); distinguir a rede como ferramenta de suas armadilhas.
- **Procedimentais:** identificar sinais de golpe e de perfil falso; criar uma senha forte; checar uma informação antes de repassar.
- **Atitudinais:** agir com respeito e empatia na rede; proteger os próprios dados e os do outro; assumir a autoria e o crédito.

## BNCC.

- Competências gerais: 1, 4, 5 (cultura digital), 7 (argumentação), 9, 10.
- Computação (eixo Cultura Digital: uso ético, crítico e seguro; funcionamento da internet; cidadania digital). Língua Portuguesa (checagem de informação, o gênero contrato ou regras). Geografia (a rede física da internet no mundo, os cabos, o território).
- *(Códigos de habilidade por ano na versão da rede.)*

**Materiais.** Barbante ou lã para o "mapa da internet", cartões de "computador", exemplos impressos de mensagens de golpe (fictícias), o mapa-múndi, as fichas 5A e 5B. 📄 A sequência inteira funciona sem computador; onde houver, o professor pode demonstrar de forma supervisionada.

## Percurso

**Aula 1, Sensibilização (o disfarce de Frida).** ⌚ 50 min · 👥 turma Reconte a cena da velhinha gentil que era Frida. Silêncio, e a pergunta que gela. 💬 "As crianças confiaram e agradeceram a sorte. O que faltou perguntar? O nome dela, quem ela era. Na internet, quantas 'velhinhas gentis' a gente encontra sem saber quem são de verdade?" 👁️ Fique de olho em quem já viveu algo parecido na tela (um pedido estranho, um "amigo" desconhecido): acolha, sem expor.

**Aula 2, Investigação (o mapa da internet, desplugado).** ⌚ 50 min · 👥 turma O **jogo do barbante**: cada estudante é um "computador" com um cartão; passando o barbante de mão em mão, a turma constroi a **rede de redes**. Uma mensagem (um bilhete) é quebrada em pedaços (**pacotes**), cada pedaço faz o caminho pelo barbante e se remonta no destino. Mostre no mapa-múndi que os cabos de verdade cruzam o fundo do mar. 💬 "A internet não é uma nuvem mágica: é fio, é cabo, é máquina. E a mensagem viaja em pedacinhos. Se a Ordem do filme fosse feita de fios, seria isto: uma rede que passa conhecimento de mão em mão."

**Aula 3, Investigação (detetive de disfarces e de golpes).** ⌚ 50 min · 👥 duplas Apresente mensagens fictícias de golpe (o prêmio falso, o banco urgente, o falso amigo pedindo dados) e perfis suspeitos. As duplas viram detetives e apontam os sinais de alerta na 📄 Ficha 5A: urgência exagerada, pedido de dados, remetente estranho, oferta boa demais, o "amigo" que a gente só conhece pela tela. 💬 "Qual é a Frida disfarçada aqui? Como o afago dela (a mensagem gentil, o prêmio) esconde o que ela quer de verdade?"

**Aula 4, Investigação (o que é meu: dados e senha).** ⌚ 50 min · 👥 duplas O que são **dados pessoais** (nome, endereço, escola, rotina, foto) e por que valem um tesouro. A oficina da **senha forte**: longa, com letras, números e símbolos, uma para cada lugar, nunca emprestada. A ideia de **pegada digital** (tudo deixa rastro) e a lei que protege (a LGPD, com cuidado redobrado para menores). 💬 "A senha é a chave da casa. Você deixaria a chave debaixo do tapete? Emprestaria para um desconhecido gentil?"

**Aula 5, Produção e Socialização (o contrato digital da turma).** ⌚ 50 min · 👥 turma A turma escreve e assina o próprio **contrato digital**: as regras de ouro para usar a internet com respeito, crédito, checagem e segurança. Ao lado, o **mapa da internet** feito na Aula 2 vira cartaz. Roda final. 💬 "Se a internet pode ser a rede que une, como a Ordem, quais são as nossas regras para que ela una e não engane?"

## Diferenciação

- **6º e 7º:** foco no reconhecimento de golpes e disfarces, na senha forte e no respeito ao outro; contrato com regras simples.

- **8º e 9º:** aprofundar em como a internet funciona (pacotes, endereço, servidores), pegada digital e a LGPD; o contrato ganha uma cláusula sobre autoria e crédito.
- **Ensino Médio:** desinformação e checagem de fatos, vies dos algoritmos de recomendação, ética do compartilhamento, e o debate sobre privacidade e vigilância.

 **Esta sequência é desplugada na íntegra.**

O jogo do barbante, o detetive de golpes, a oficina da senha e o contrato não precisam de nenhum computador. Onde houver acesso, o professor demonstra de forma supervisionada. A segurança digital se ensina, com força, sem expor nenhuma criança à tela.

 **Fichas reproduzíveis**

**Ficha 5A, Detetive de disfarces**

- A mensagem ou o perfil diz: \_\_\_\_\_
- Sinais de alerta que encontrei: \_\_\_\_\_
- É confiável? ( ) sim ( ) não ( ) na dúvida, não respondo
- O que eu faria: \_\_\_\_\_

**Ficha 5B, A minha senha forte (rascunho, não use uma real aqui)**

- Uma senha fica forte quando ela é: \_\_\_\_\_ (longa? com símbolos?)
- Onde eu nunca guardo a minha senha: \_\_\_\_\_
- Um dado meu que eu nunca entrego na tela: \_\_\_\_\_

 **Produto final**

O contrato digital da turma (as regras de ouro, escritas e assinadas) e o mapa da internet como cartaz.

**Rubrica de avaliação**

<b>Critério</b>	<b>1 Em início</b>	<b>2 Em desenvolvimento</b>	<b>3 Adequado</b>	<b>4 Excelente</b>
Como a internet funciona	Acha que é "mágica"	Explica em parte	Explica rede, pacotes e endereço	Explica com clareza e usa a analogia da Ordem
Segurança (golpes e disfarces)	Confia em tudo	Desconfia às vezes	Identifica os sinais de alerta	Explica o mecanismo do golpe e como agir
Dados e senha	Ignora	Segue com lembrete	Cria senha forte e protege dados	Ensina o colega a se proteger
Cidadania e contrato	Não participa	Participa pouco	Contribui com regras	Lidera pelo exemplo e conecta com direitos

**Autoavaliação do estudante**

Eu sei como a informação viaja na internet. Eu reconheço um golpe ou um perfil falso. Eu sei fazer uma senha forte. Uma regra do nosso contrato que eu levo para a vida: \_\_\_\_\_.

### **Segurança e privacidade (leia antes)**

Nenhuma criança usa conta pessoal nem digita senha real na atividade. As mensagens de golpe são fictícias, criadas pelo professor. O objetivo é formar juízo e autonomia, respeitando a LGPD, nunca expor a criança. Se um estudante relatar uma situação real de risco (assédio, exposição, golpe sofrido), acolha e encaminhe conforme o protocolo da escola.

### **Conexões**

Liga com a Sequência 5 do caderno-mãe (conversar com uma IA com senso crítico) e com a plataforma de IA do Byte, onde o Byte é o tutor de uso seguro da tecnologia. Conversa com a Sequência 2 do caderno-mãe (Frida como personagem, a empatia e o discernimento) e com Geografia (a rede física do mundo).

---

---

# A grande feira: "O Laboratório de Código da Turma"

---

As cinco sequências podem convergir num evento de escola que reúne todos os produtos num só dia, na mesma lógica da feira do caderno-mãe, e podem somar-se a ela:

- **A bancada de algoritmos** exhibe o livro de receitas de algoritmos e desafia os visitantes a serem "robôs" que executam um algoritmo da turma (Seq. 1).
- **A galeria de Bytes de tela** mostra os Byte animados do Scratch, que os visitantes interagem (Seq. 2).
- **O museu da máquina** abre o gabinete e mostra o pôster "por dentro do Byte", com a brincadeira do binário (Seq. 3).
- **A oficina maker** demonstra os Byte físicos que mudam de cor conforme a emoção, com Arduino ou na versão desplugada dos cartões (Seq. 4). É o coração da feira, a câmara secreta do filme repetida na escola.
- **O balcão da rede segura** apresenta o contrato digital e o mapa da internet, e ensina os visitantes a fazer uma senha forte (Seq. 5).

A feira vira a socialização final: as famílias e a comunidade entram no laboratório de código da turma. É a edícula de porta aberta, fora da tela, e a promessa do filme cumprida na sala de aula.

---

# Avaliação no caderno inteiro

---

- **Por evidência, não por prova.** O aprendizado se mostra no produto e no percurso (o algoritmo que roda, o código que funciona, o circuito que acende, o contrato que a turma assina), com foco na colaboração, na depuração e no protagonismo.
  - **O erro como pista.** Em computação, o bug não é fracasso: é informação. A avaliação valoriza quem depura, quem itera, quem anota o que mudou. O caderno de bordo do inventor é o registro de aprendizagem.
  - **Três olhares.** Cada sequência combina a rubrica do professor, a autoavaliação do estudante e a socialização (o olhar dos colegas, de outra turma e da comunidade).
  - **Rubrica em quatro níveis** (em início, em desenvolvimento, adequado, excelente), sempre com descritores, para tornar o critério transparente ao estudante desde o começo.
-

# Acessibilidade e o Desenho Universal para a Aprendizagem (DUA)

---

Cada sequência oferece múltiplos caminhos, para que a computação seja de todos:

- **Múltiplos meios de representação:** todo conceito aparece em mais de um formato (a fala, o corpo no pátio, o cartão colorido, o bloco na tela, o circuito, o pseudocódigo, o código). A cor do Byte, aliás, sempre vem acompanhada de uma palavra (a emoção), para não excluir quem não distingue cores.
- **Múltiplos meios de ação e expressão:** aceite o produto em formatos variados (o algoritmo escrito, o encenado, o código em blocos, o circuito, o pôster, o vídeo curto de demonstração). A versão desplugada de cada sequência garante que ninguém fique de fora por falta de máquina.
- **Múltiplos meios de engajamento:** dê escolha (a emoção do Byte, o algoritmo, o golpe a investigar), trabalhe em duplas e grupos com papéis (a programação em par é DUA na veia), e conecte ao repertório do estudante.

E, como no filme, a acessibilidade nasce com o projeto: o material do universo prevê legenda descritiva, audiodescrição e Libras. Um Byte que sente e comunica por luz, som e gesto é, ele próprio, um convite à comunicação para além da palavra.

---

# Do desplugado ao plugado: como progredir sem equipamento

---

Uma escola sem computador ensina computação com este caderno do começo ao fim. A progressão sugerida:

1. **Sempre comece desplugado.** A lógica (Seq. 1), o hardware por fotos e circuito humano (Seq. 3) e a cidadania digital (Seq. 5) são integralmente desplugados.
  2. **Se houver poucos aparelhos,** rode o Scratch (Seq. 2) em rodízio de duplas, ou use a versão offline num único computador projetado, com a turma dirigindo.
  3. **Se houver um kit de Arduino,** mesmo que um só, faça a Seq. 4 por rodízio de bancadas, e mantenha a versão desplugada dos cartões rodando em paralelo para o resto da turma.
  4. **O conceito nunca depende da máquina.** A máquina dá corpo ao que a turma já entendeu no papel e no corpo. É a mesma lição do Byte: a estrutura já estava montada; faltava a mão certa para dar vida.
-

# Modelo de plano de aula (em branco, para copiar)

---

- Sequência e aula: \_\_\_\_\_
  - Eixo de Computação (Pensamento Computacional / Mundo Digital / Cultura Digital): \_\_\_\_\_
  - Objetivo do dia: \_\_\_\_\_
  - Pergunta mediadora de abertura: \_\_\_\_\_
  - Momentos (tempo): sensibilização \_\_\_ · investigação/produção \_\_\_ · fechamento \_\_\_
  - Agrupamento: \_\_\_\_\_
  - Materiais e ficha: \_\_\_\_\_
  - Plugado ou desplugado (e o plano B se faltar equipamento): \_\_\_\_\_
  - O que observar (evidências, incluindo a depuração): \_\_\_\_\_
  - Acessibilidade (DUA) do dia: \_\_\_\_\_
  - Tarefa ou combinado para casa: \_\_\_\_\_
- 
-

# Back matter

---

## Aprofundamentos para o professor (mini-fichas de conteúdo)

**A1. As três estruturas de controle, num parágrafo cada.** *Sequência:* passos na ordem; a ordem muda o resultado. *Repetição (laço):* repetir sem reescrever, seja um número fixo de vezes ("repita 3"), seja enquanto uma condição durar ("enquanto não ferver"). *Condição:* escolher um caminho conforme uma pergunta de sim ou não ("se... senão..."). Todo software do mundo é combinação destas três com variáveis (as caixinhas de memória).

**A2. Por que ensinar em blocos antes de texto.** Blocos ensinam a **lógica** sem o peso da **sintaxe** (a vírgula, o parêntese, a maiúscula que travam o iniciante). A criança pensa no que quer fazer, não em onde faltou um ponto e vírgula. Firmada a lógica, a passagem para o texto (Python é uma boa primeira linguagem de texto) fica natural, por volta do 8º e 9º ano.

**A3. A estrutura de um programa de Arduino.** Duas partes fixas: `setup()` roda **uma vez** (prepara os pinos); `loop()` roda **para sempre** (o batimento). Entrada pelos pinos (botão, sensor), processamento no microcontrolador, saída pelos pinos (LED, motor). É o modelo entrada, processamento, saída em miniatura, e é o corpo do Byte.

**A4. LED RGB e o pixel.** Um LED RGB tem três luzes (vermelho, verde, azul); regulando a intensidade de cada uma (de 0 a 255), sai qualquer cor. É exatamente como a tela do celular forma cada ponto (pixel). Amarelo é vermelho e verde acesos; branco é os três no máximo; preto é os três apagados. A cor é número: uma bela ponte com a Matemática.

**A5. Bit, byte e binário.** Bit é a menor migalha: 0 ou 1, aceso ou apagado. Oito bits formam um byte, que já guarda uma letra. Com só dois símbolos, em ordem, o computador representa qualquer número, texto, imagem ou som. O robô se chama Byte por isso: a unidade básica de que tudo é feito.

**A6. O ciclo do inventor é o ciclo do programador.** Imaginar, construir, testar, achar o bug, corrigir, repetir. A máquina de Percival explode (o maior bug do filme) e da falha nasce o Byte. Ensine que errar e depurar não é acidente do percurso: é o percurso.

## Glossário de computação (para a turma e para o professor)

- **Algoritmo:** um passo a passo claro e finito que leva a um resultado; a receita.
- **Pensamento computacional:** o jeito de resolver problemas com decomposição, padrões, abstração e algoritmos, com ou sem máquina.
- **Decomposição:** quebrar um problema grande em partes pequenas.
- **Reconhecimento de padrões:** enxergar o que se repete e virar regra.
- **Abstração:** guardar o essencial e ignorar o detalhe que não muda a solução.

- **Sequência:** passos na ordem.
- **Repetição (laço):** repetir sem reescrever (repete N vezes; enquanto durar a condição).
- **Condição:** escolher um caminho conforme sim ou não (se / senão).
- **Variável:** uma caixinha com nome que guarda um valor que pode mudar; a memória do programa.
- **Bug:** um erro no programa; **depurar (debug)** é caçá-lo e corrigi-lo.
- **Programação em blocos:** montar o programa encaixando peças (Scratch), sem digitar sintaxe.
- **Scratch:** ambiente de programação em blocos, criado para crianças.
- **Maker:** a cultura de aprender fazendo, construindo objetos que funcionam.
- **Arduino:** placa simples e barata que se programa para ler entradas e comandar saídas no mundo físico.
- **Microcontrolador:** o pequeno cérebro da placa, que executa o programa.
- **setup e loop:** as duas partes do programa de Arduino (prepara uma vez; repete sempre).
- **Entrada (input):** por onde a informação chega (teclado, botão, sensor).
- **Processamento:** onde a decisão acontece; a **CPU**.
- **Memória:** onde a informação se guarda; **RAM** (temporária, some ao desligar) e **disco/HD/SSD** (permanente).
- **Saída (output):** por onde o resultado sai (tela, luz, som, motor).
- **Bit e byte:** a menor unidade (0 ou 1) e o conjunto de oito bits (uma letra).
- **Binário:** o sistema de dois símbolos (0 e 1) com que o computador representa tudo.
- **LED e LED RGB:** a luzinha; a que mistura vermelho, verde e azul para dar qualquer cor.
- **Circuito:** o caminho fechado por onde a corrente passa e acende a luz.
- **Internet:** a rede de redes que liga computadores no mundo todo.
- **Pacote:** o pedacinho em que a informação viaja pela rede.
- **Cidadania digital:** usar a rede com respeito, ética, crédito e senso crítico.
- **Pegada digital:** o rastro que tudo o que a gente faz na rede deixa.
- **Phishing:** golpe que finge ser de fonte confiável para roubar dados ou dinheiro.
- **Engenharia social:** enganar a pessoa (e não a máquina) para obter acesso ou dados; o disfarce de Frida.
- **LGPD:** a lei brasileira que protege os dados pessoais, com cuidado redobrado para menores.

## Referências e apoios (sem citação fabricada)

- A **Base Nacional Comum Curricular (BNCC)**, do MEC, e a **Normativa da Computação na Educação Básica** (Complemento à BNCC, com os três eixos: Pensamento Computacional, Mundo Digital e Cultura Digital), para o mapeamento fino de objetos de conhecimento e habilidades por ano, na versão de cada rede.
- O ambiente **Scratch** (Scratch Foundation e MIT), inclusive a versão offline, para a programação em blocos sem depender de internet.

- O ecossistema **Arduino** e a documentação oficial da placa e do ambiente de programação, para a Sequência 4.
- O material do movimento de **Computação Desplugada** (Computer Science Unplugged e similares), base das atividades sem computador presentes em todas as sequências.
- O referencial de **Desenho Universal para a Aprendizagem (DUA)**, para a acessibilidade das sequências.
- Guias de **cidadania digital, letramento em mídia e segurança na internet** para a Educação Básica, e o texto da **LGPD** no que toca a dados de crianças e adolescentes, para a Sequência 5.
- Materiais de referência sobre a **história da computação no Brasil** (universidades, política de informática, automação bancária e urna eletrônica), para a Parte I, seção 7.

*(A coordenação pedagógica de cada rede complementa com os documentos curriculares locais e finaliza os códigos de habilidade por ano e componente.)*

---

**Frente educacional do projeto:** filme@ofantasticolaboratorio.com.br · +55 11 98343-3235 · ofantasticolaboratorio.com.br

*O Fantástico Laboratório. A invenção mais poderosa do mundo é a única que ninguém constroi sozinho, e o código, também não.*