

# A4

## Arduino

Desvendando o código

# Como se faz pão de queijo?

## Começamos com uma receita.

### INGREDIENTES

1 xícara de chá de óleo  
1 e 1/2 xícaras de chá de leite  
300g de queijo ralado  
2 colheres de sopa rasa de sal  
500g de polvilho doce  
2 ovos

### MODO DE PREPARO

Misture o polvilho e o sal e reserve.

Ferva o leite e o óleo e em seguida escale o polvilho com o sal.

Deixe esfriar, acrescente os ovos e o queijo.

Quando a massa estiver homogênea, faça bolinhas.

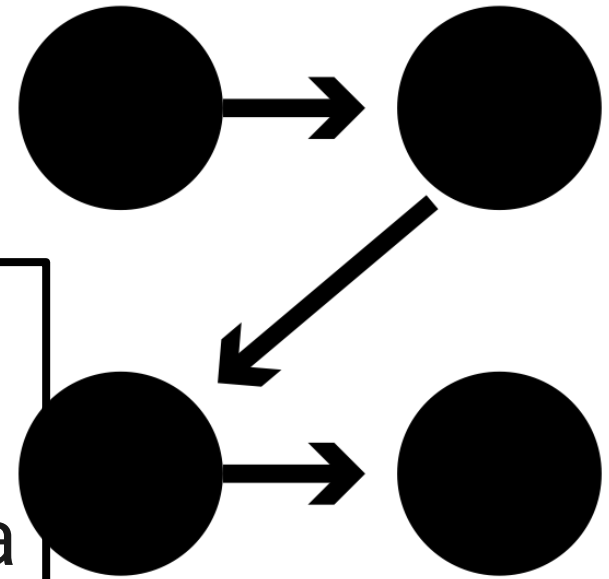
Leve para assar em forno quente até dourar.

É fácil  
identificar o  
texto de uma  
receita. Há dois  
blocos de  
informações  
importantes:  
**INGREDIENTES**  
e **MODOS DE**  
**PREPARO.**

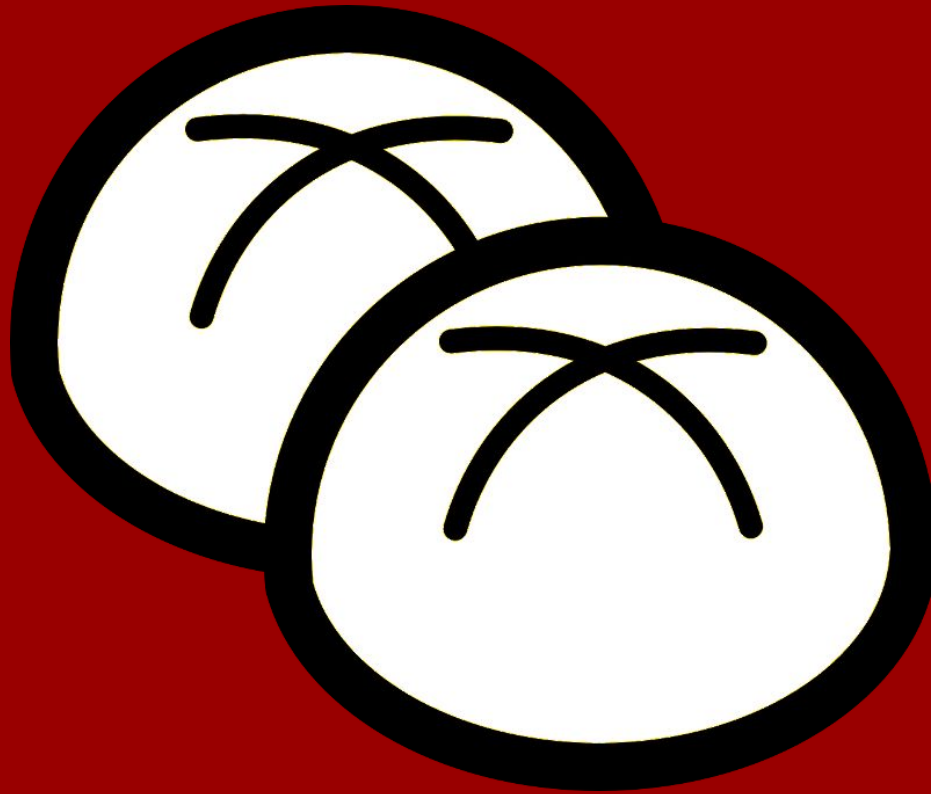


# A lógica no MODO DE PREPARO...

Voltando a receita de Pão de Queijo, faz sentido fazer bolinhas antes de misturar a massa? Não né? Porque a lógica da receita é procedural, uma ação vem após a outra.



**Programar Arduino é como fazer pão de queijo!**



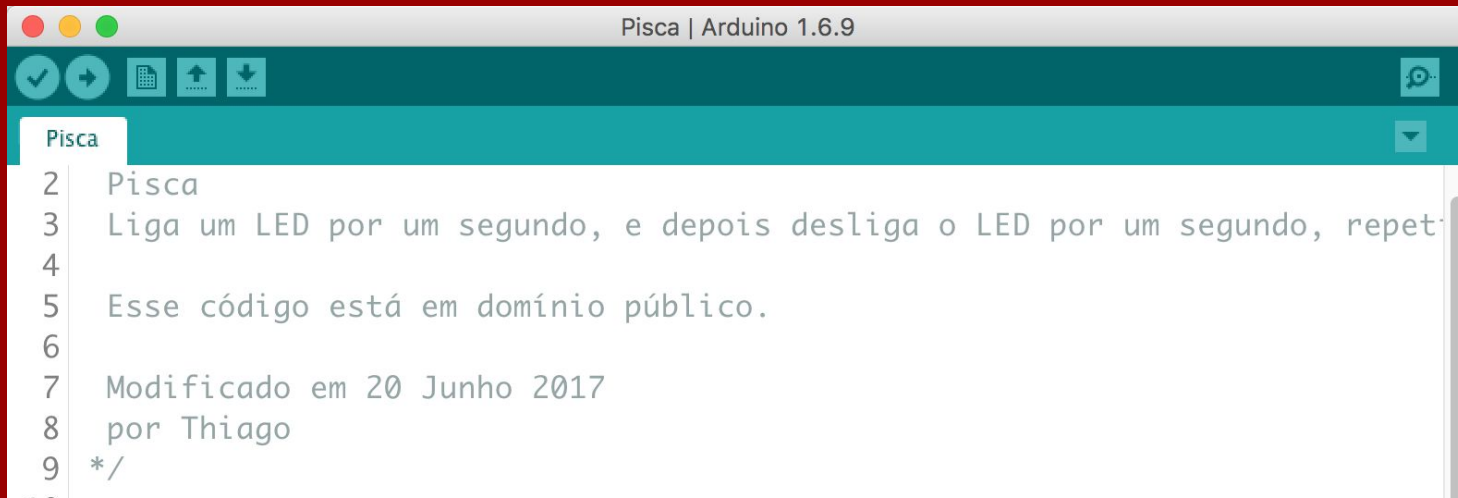
Pisca

```
2  Pisca
3  Liga um LED por um segundo, e depois desliga o LED por um segundo, repet
4
5  Esse código está em domínio público.
6
7  Modificado em 20 Junho 2017
8  por [nome]
9  */
10
11
12 // a função de setup() roda exatamente uma vez quando ligamos o Arduino ou
13 //   apertamos o botão de RESET
14 void setup() {
15   // inicializa o pino 13 como saída digital
16   pinMode(13, OUTPUT);
17 }
18
19 // a função de loop roda repetidamente, para sempre
20 void loop() {
```

**Olhando o código do Blink, quantos blocos de informações existem?**

# Primeira parte do código

Começa com /\*  
Termina com \*/  
Está em cinza.

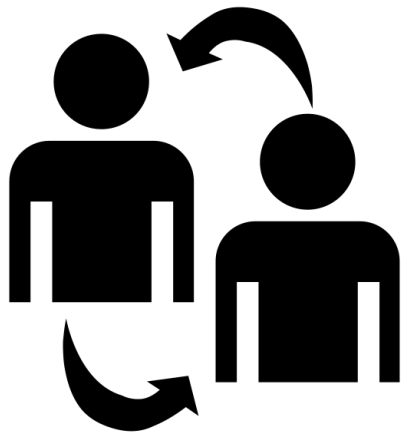


The screenshot shows the Arduino IDE interface with a window titled "Pisca | Arduino 1.6.9". The code editor displays a sketch named "Pisca" with the following content:

```
2  Pisca
3  Liga um LED por um segundo, e depois desliga o LED por um segundo, repet
4
5  Esse código está em domínio público.
6
7  Modificado em 20 Junho 2017
8  por Thiago
9  */
```

The comment block, starting from line 5 and ending at line 9, is highlighted in a light gray color, indicating it is a comment.

É um bloco de **COMENTÁRIO**, com uma breve explicação sobre o código.



Na realidade, COMENTÁRIO é um modo de documentar o código. Serve apenas para humanos. A máquina ignora este bloco de texto.



# setup()

```
// a função de setup() roda exatamente uma vez
//   apertamos o botão de RESET
void setup() {
  // inicializa o pino 13 como saída digital
  pinMode(13, OUTPUT);
}
```

É uma função, um bloco de informação com instruções que são executadas uma única vez para CONFIGURAR o sistema.

# Traduzindo a instrução do setup do Blink

```
pinMode (13, OUTPUT);
```

Significa: O modo da porta em que se encontra o led do exemplo. Há **dois** parâmetros passados na instrução. Um indica a PORTA e o outro o MODO de informação da porta. No caso, o led é uma SAÍDA/OUTPUT de informação.

# loop()

```
// a função de loop roda repetidamente, para sempre
void loop() {
  digitalWrite(13, HIGH);    // liga o LED (HIGH a voltagem positiva)
  delay(1000);               // espera um segundo
  digitalWrite(13, LOW);     // desliga o LED colocando a voltagem para
  delay(1000);               // espera um segundo
}
```

É a segunda função, um bloco de informação com instruções que são executadas REPETIDAMENTE.

# Traduzindo as instruções do loop do Blink

```
digitalWrite(13, HIGH);
```

Significa: Escreva digitalmente na porta em que se encontra o led. Há dois parâmetros passados na instrução. Um indica a PORTA e o outro o ESTADO digital, no caso, HIGH/alto/ligado. Experimente trocar o HIGH por 1 e suba o código novamente no Arduíno. O piscar continua o mesmo, certo?

# Traduzindo as instruções do loop do Blink

```
delay(1000);
```

Significa: Pare o código por um tempo. O tempo no caso da programação do Arduino é medido em milisegundos. Esta instrução pede para parar o código e passa o parâmetro de 1000 milisegundos de espera, o equivalente a 1 segundo.

# Traduzindo as instruções do loop do Blink

```
digitalWrite(13, LOW);
```

Significa: Escreva digitalmente na porta em que se encontra o led. Há dois parâmetros passados na instrução. Um indica a PORTA e o outro o ESTADO digital, no caso, baixo ou desligado. Experimente trocar o LOW por 0 e suba o código novamente no Arduino. O piscar continua o mesmo, certo?

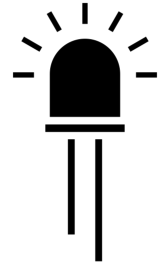
# Traduzindo as instruções do loop do Blink

```
delay(1000);
```

Significa: Pare o código por um tempo. O tempo no caso da programação do Arduino é medido em milisegundos. Esta instrução pede para parar o código e passa o parâmetro de 1000 milisegundos de espera, o equivalente a 1 segundo.

**Está escrito em linguagem de programação exatamente o que o Arduíno está fazendo.**

**O led fica ligado 1s e desligado 1s.**



**Experimente mudar os valores do delay() para 500, 100 ou mesmo 2000 e veja o que acontece.**





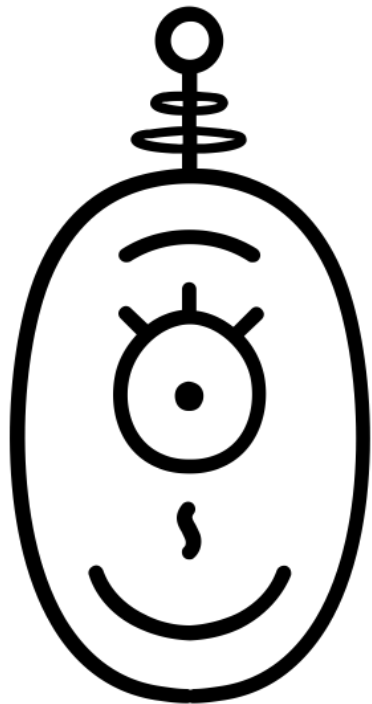
# Segundo comentário

```
// a função de setup() roda exatamente uma vez
//   apertamos o botão de RESET
void setup() {
  // inicializa o pino 13 como saída digital
  pinMode(13, OUTPUT);
}
```

Notaram certas linhas de comentário cinzas ao longo do código? Ao invés de bloco é possível também fazer comentários usando duas barras.

// legal isso de programar né?

Pronto, você está preparado para o  
próximo passo!



**A5**