

Mãos na massa: Praticando o Software

Sabemos que uma boa prática é manter o código limpo e organizado. O objetivo nesse exercício é dar uma pequena organizada no método `setup` e criar algumas funções, constantes e variáveis que vão nos ajudar no futuro.

Vamos lá:

1) Primeiro, extraia as diversas chamadas de `pinMode()` do método `setup` para uma função separada. Dê um nome semântico a ela, por exemplo, `iniciaPortas()` :

```
void iniciaPortas(){
    pinMode(LED_VERMELHO, OUTPUT);
    pinMode(LED_VERDE, OUTPUT);
    pinMode(LED_AMARELO, OUTPUT);
    pinMode(LED_AZUL, OUTPUT);
}
```

2) Não esqueça de chamar a nova função no método `setup()` :

```
void setup(){
    Serial.begin(9600);
    iniciaPortas();
}
```

3) Em seguida, crie uma nova constante, a `TAMANHO_SEQUENCIA` , com valor 4. Ela servirá para guardar o tamanho das nossas sequências:

```
// Constantes anteriores omitidos

// Nova Constante
#define TAMANHO_SEQUENCIA 4
```

4) Agora, crie uma variável global, que seja uma array de `int` e com o tamanho da nossa constante recém criada, dê o nome de `sequenciaLuzes` :

```
// Constantes omitidos

/// Nosso novo array, usando o constante TAMANHO_SEQUENCIA
int sequenciaLuzes[TAMANHO_SEQUENCIA];
```

5) Crie uma função `iniciaJogo()` , onde você deve inicializar cada posição do nosso array com um LED diferente:

```
void iniciaJogo(){
    sequenciaLuzes[0] = LED_AZUL;
    sequenciaLuzes[1] = LED_VERDE;
    sequenciaLuzes[2] = LED_VERMELHO;
```

```
    sequenciaLuzes[3] = LED_AMARELO;  
}
```

6) Não se esqueça de chamar a função `iniciaJogo()` dentro da função `setup()` :

```
void setup(){  
    Serial.begin(9600);  
    iniciaPortas();  
    iniciaJogo();  
}
```

7) Para testar isso tudo, podemos executar o seguinte `for` no nosso método `loop()` :

```
void loop(){  
    for(int indice = 0; indice < TAMANHO_SEQUENCIA ; indice++){  
        piscaLed(sequenciaLuzes[indice]);  
    }  
}
```

Se tudo ocorreu bem, veremos os LEDs piscar na sequência que definimos em `iniciaJogo()` !