

## Mãos na massa - estado da próxima rodada

Vamos criar uma nova função para atender o estado `PRONTO_PARA_PROXIMA_RODADA`. Nessa nova função, vamos incrementar uma variável que serve para contar a quantidade de rodadas e toca os LEDs da rodada.

1) Dessa maneira, no início do seu programa, crie uma nova variável chamada `rodada`:

```
int rodada = 0;
```

2) Agora, crie um função `preparaNovaRodada`:

```
void preparaNovaRodada() {  
    rodada++;  
    tocaLedsRodada();  
}
```

3) Implemente a função `tocaLedsRodada()`:

```
void tocaLedsRodada() {  
    for (int i = 0; i < rodada; i++) {  
        piscaLed(sequenciaLuzes[i]);  
    }  
}
```

4) Agora, chame a função `preparaNovaRodada` dentro do `switch` no `case` que atende `PRONTO_PARA_PROXIMA_RODADA`. Na função `loop` altere:

```
void loop() {  
    switch (estadoAtual()) {  
        case PRONTO_PARA_PROXIMA_RODADA:  
            Serial.println("Pronto para a proxima rodada");  
            preparaNovaRodada(); //aqui  
            break;  
        case USUARIO_RESPONDENDO:  
            Serial.println("Usuario respondendo");  
            break;  
        case JOGO_FINALIZADO_SUCESSO:  
            Serial.println("Jogo finalizado com sucesso");  
            break;  
        case JOGO_FINALIZADO_FALHA:  
            Serial.println("Jogo finalizado com falha");  
            break;  
    }  
    delay(MEIO_SEGUNDO);  
}
```

5) Por fim, altere a função `estadoAtual` para verificar se ainda há rodadas:

```
int estadoAtual() {  
    if (rodada < TAMANHO_SEQUENCIA) {  
        return PRONTO_PARA_PROXIMA_RODADA;  
    } else {  
        return JOGO_FINALIZADO_SUCESSO;  
    }  
    return JOGO_FINALIZADO_FALHA;  
}
```

Compile e teste o seu código! Ao rodar, fique de olho no monitor serial.