

01

## Os estados do jogo

### Transcrição

Primeiro, vamos retirar do circuito o potenciômetro, assim como desconectamos os fios terra e da força.

Vamos acessar o Draw io, um programa que nos auxilia através do recurso visual a planejar nosso projeto. Já temos cada peça necessária para o circuito, mas um segredo também é pensar no relacionamento dos componentes entre si! Esse programa nos ajuda de uma maneira fácil a chegar no resultado pretendido.

O primeiro passo é entender que o jogo passa por diversos estados. E o `draw.io` nos ajuda a visualizar a sequência desses acontecimentos. Vamos lá?!

1. O primeiro estado é facilmente identificável, o `SETUP`, ou, estado inicial. Portanto, criamos uma caixa equivalente para ele. Nesse estado utilizamos uma função que nos ajuda a iniciar portas, o jogo, etc.
2. Uma vez que o `SETUP` estiver pronto, podemos começar a jogar. Dessa maneira, estamos prontos para iniciar uma próxima rodada e é exatamente isso que vamos colocar na segunda caixa: `PRONTO_PARA_PROX_RODADA`
3. Depois dessa rodada ele executa a função que mostra as luzes. Após a primeira sequência de luzes respondida, ocorrerá uma segunda, terceira, até que o usuário acerte todas. Nesse estado é bastante claro que o usuário está respondendo, por isso, colocamos uma terceira caixa `USUARIO_RESPONDENDO`.
4. Podemos acrescentar setas que indicam o caminho do projeto. Portanto, vamos do `SETUP` para o `PRONTO_PARA_PROX_RODADA` e, por fim, para o `USUARIO_RESPONDENDO`.
5. O próximo estado é `FIM_DO_JOGO`. Assim, ligamos a seta do `USUARIO_RESPONDENDO` a essa última caixa.

Por enquanto temos o seguinte:



Mas, o projeto real não é tão perfeito! Esse esquema indica que nós temos apenas uma rodada e, na verdade, é um pouco mais complicado do que isso! Por exemplo, só é fim de jogo depois que o usuário finaliza todas as etapas, então, podemos acrescentar antes de `FIM_DO_JOGO` um comentário dizendo que "Todas as Luzes respondidas corretamente" indicam o fim do jogo.

Se as luzes não foram respondidas corretamente, recomeçam os processos, dessa maneira, nós inserimos uma seta que leva de volta de `USUARIO_RESPONDENDO` para `PRONTO_PARA_PROX_RODADA` e inserimos o comentário `Número de Luzes Respondidas < Tamanho Sequencia`. Assim, se o número de luzes for menor que a sequência, então, somos direcionados para o `PRONTO_PARA_PROXIMA_RODADA` que deve possuir uma condição marcada pelo comentário: `Aumentar número de luzes`.

Esse é o fluxo do jogo até que o usuário consiga finalizar! Mas, não temos só uma maneira de finalização! Ele pode acabar de duas maneiras: `FIM_DO_JOGO_SUCESSO` e `FIM_DE_JOGO_COM_FALHA`. Nessa segunda maneira nós acrescentamos um comentário, Usuário errou .

Com isso, temo um diagrama bastante fiel do que acontece no jogo:

