

Letras

Transcrição

[00:00] Dada a estrutura lógica que uma criança usa para contar quantas vezes uma letra aparece ou se não aparece, vamos escrever. Aí traduzimos para linguagem de programação. Dada a palavra e letra que estou procurando, para cada letra na palavra secreta vamos comparar se a letra for igual à que estou procurando, vou somar um no contador, que começa em zero. Se essa letra for diferente, não faz nada, mas vai para o próximo.

[01:30] Quando terminei a palavra inteira, consigo verificar se o número de vezes que encontrei a letra for diferente de zero, quer dizer que encontrei a letra. Senão, se ela é igual a zero, não encontrei a letra. Quer dizer que você errou.

[02:06] Vamos implementar isso em Ruby. A letra procurada é o chute na posição zero. O contador é o total encontrado, que é zero, para cada letra na palavra secreta. Se eu quero passar por cada letra na palavra secreta, então vou com i de zero até palavra secreta size menos um. Se essa letra for igual à letra que estou procurando, encontrei e aumento o total.

[03:30] No final, se o número de vezes que encontrei a letra for diferente de zero, quer dizer que encontrei a letra. Senão, letra não encontrada. Se eu não encontrei, somo no erro.

[04:25] Rodando meu jogo, tudo certo. É assim que resolvo o problema. Escrevo no papel, descrevo o algoritmo e escrevo no programa. Se estiver difícil de transcrever para o programa, quebre em palavras simples, ideias simples, você vai ver os ifs e fors aparecendo.

[05:26] Mas o código ainda está bem feio. Vamos ter que refatorar.