

Separações

Transcrição

[00:00] Agora que organizamos bem nosso código, separamos camadas e um tipo de camada, não só mais naquele nível bem básico de função, agora estamos separando em arquivos um pouco mais distintos, de acordo com a semântica e significado das nossas funções, o que queremos fazer é implementar a funcionalidade de imprimir a palavra secreta mascarada. Isso é, as letras que você já escolheu eu mostro e as que você ainda não conhece eu mostro como underline.

[00:35] Se eu tenho a palavra programador e você já chutou a letra O, vou mostrar onde a letra está. Vamos pensar que nem criança, como ela implementaria isso. Dados os chutes que já foram feitos e a palavra secreta. Eu vou ter que ir para cada uma das letras da palavra secreta. Se a letra já foi chutada, se ela está dentro dos chutes, então imprime a letra. Senão, imprime underline.

[01:30] Para fazer isso, posso dizer que para cada uma das letras da palavra secreta, quero passar por todas as letras. Eu podia fazer um for com contador, um caractere que vai de um em um. Não é o que eu quero. Quero pegar a palavra secreta, que é uma string, transformar numa array de chars para cada letra lá dentro. Para cada letra, se a letra já foi chutada, ou seja, se os chutes incluem essa letra, imprima a letra. Senão, imprime underline. E tenho meu algoritmo. Essa função trabalha tanto a lógica para extrair a palavra mascarada quanto imprimir a palavra mascarada.

[02:28] Nós misturamos lógica com interface do usuário. Como posso separar os dois? Fazendo espaços distintos. Acabei de citar que o primeiro passo é calcular a palavra mascarada, e o segundo imprimir a palavra mascarada. Não precisa fazer isso ao mesmo tempo. Às vezes posso e otimizó a memória, mas não é nosso caso. Posso simplesmente calcular a palavra inteira e imprimir de uma vez só.

[03:00] Eu vou criar uma função de lógica no `forca.rb` que calcula a palavra mascarada. Vou copiar o código que tínhamos escrito antes e vou definir palavra mascarada e coloco o código dentro. Esse código precisa tanto da palavra secreta quanto dos chutes. Então, palavra secreta e chutes são argumentos parâmetros para minha função. Para cada letra dentro da palavra secreta, se inclui a letra, quero criar uma string que eu fosse concatenando cada uma das letras ou underlines. Eu vou ter uma máscara que começa vazia. Para cada letra que quero adicionar lá dentro, falo para máscara colocar no final da string, concateno a string esse caractere.

[04:11] No final, devolvemos a máscara. E quando podemos usar essa palavra mascarada? Antes de pedir um chute. Se eu vou pedir um chute para você, preciso falar o que você sabe até agora. Temos isso na nossa lógica.

[04:42] Agora, quando pedimos um chute válido passamos a máscara para a frente. Quando pedimos um chute válido, estamos recebendo a máscara. No cabeçalho, vamos imprimir a máscara, no nosso `ui.rb`.

[05:42] Implementamos a funcionalidade que queríamos de mostrar a palavra mascarada para o nosso usuário.