

02

Interface

Transcrição

[00:00] Repare que no nosso jogo até agora criamos diversas funções. Primeiro, a boas-vindas, que faz vários puts e gets, ou seja, interface do usuário. A função escolhe palavra secreta, que por enquanto faz somente puts, então interface do usuário. Não quer jogar é puts e gets, interface do usuário. Pede um chute, puts e gets, interface do usuário. Tudo isso até agora, exceto joga, é basicamente interface do usuário. Vou pegar todas essas funções e vou arrancar fora. Vou criar outro arquivo e colocar lá dentro. Ele representa a interface do usuário, que é user interface, então ui.rb.

[01:00] No meu código fiquei só com a função joga, que é a função de lógica. Se o gets e o puts é interface com o usuário, por que na função de lógica tem um puts misturado? É um pouco da extração que ainda temos que fazer. Um pouco da interface que tenho com o usuário vou extrair para certas funções. Por exemplo, essa mensagem de você já chutou tal chute. Se isso é interface com o usuário, não vai ficar ali. Ao invés disso, vou chamar a função avisa chute efetuado. E aí passo o chute como parâmetro. Lá no ui.rb vou definir uma função chamada avisa chute efetuado, que recebe o chute.

[02:00] Assim como quando invocamos uma função em Ruby não precisamos colocar os parênteses, quando você define uma função em Ruby também não precisa dos parênteses. Você vai pessoa escrevendo com os dois tipos de sintaxe. Vou misturar nesse exemplo somente sem o parênteses.

[02:20] Também temos ui onde ele encontra ou não encontra letra. Ainda sobrou o avisou que acertou a palavra. Com o errou a mesma coisa.

[03:50] Por fim, quantos pontos você ganhou. O que estou fazendo aqui é avisando o número de pontos que você ganhou até agora. Lá no nosso ui, vou colocar o avisa pontos.

[04:12] Temos o código da nossa função joga que é pura lógica de negócios, e o código de ui, que é pura interface com o usuário. Nesse instante, nossa lógica de negócios chama nossa camada de interface com o usuário que se vira dali para a frente.

[04:32] Se eu tento rodar o jogo, dá erro. Não encontra o dar boas-vindas, porque estamos mandando rodar o forca.rb. Dentro dele não tem o dar boas-vindas. O Ruby vai sair lendo todos os arquivos do seu computador para encontrar todas as funções que existem para executar a função? Não dá. Você tem que falar que quando você está lendo esse arquivo, você requer, necessita do conteúdo que é definido em outro arquivo, require_relative 'ui'. Cuidado que a extensão .rb o próprio Ruby vai colocar para nós. Não precisa colocar.

[05:38] Quando digo isso, estou dizendo para procurar alguém relativo a onde eu estou que tenha esse nome .rb no meu caso específico. Se eu usasse só require ui, ele iria procurar nas bibliotecas do Ruby, seja na padrão ou que eu tenha baixado depois, se essa biblioteca existe. No meu caso, quero procurar no mesmo diretório.

[06:16] Conseguir extrair toda essa parte de ui em um arquivo próprio.