

Resolvendo Problemas

7 Passos para resolver problemas na programação

Labenu_



O que vamos ver hoje?

- 7 passos para aplicar na resolução de desafios de programação
- Solucionando os problemas na prática
- Dicas de Exercícios de Lógica

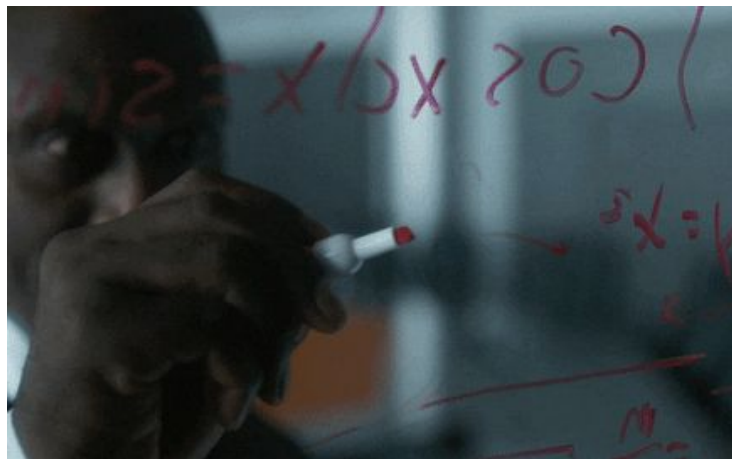


7 Passos Essenciais

Labenu_



Problemas, para que te quero? 🧐



- Como boas pessoas programadoras, **nosso trabalho é solucionar problemas por meio de códigos.**
- Não existe fórmula mágica para resolver todos os desafios e problemas da vida real. Porém a **prática e os passos a seguir**, te nortearão no processo.



7 Passos Fundamentais 🦶



O Desafio

Labenu_



O Desafio



Em Bananinha City, chegou um novo carregamento de vacinas contra gripe. Porém, a distribuição está um caos, por não haver um sistema de gerenciamento das doses.

Sua missão como pessoa desenvolvedora é ajudar a criar um sistema de controle para os futuros pacientes receberem as doses da vacina.

A aplicação deverá cadastrar o nome do(a) paciente, ano de nascimento, nacionalidade e se já tomou ou não a vacina no ano corrente.

Essas informações serão guardadas em uma base de dados para futuras consultas.

Seu primeiro desafio consiste em, dada uma lista dos cidadãos de Bananinha City, selecionar as pessoas que: completam 60 anos ou mais no ano corrente; têm nacionalidade brasileira; e ainda não tomaram a vacina.



7 Passos Essenciais Na Prática

Labenu_



Passo 1

Labenu_



1. Ter Atenção no Enunciado 🦶

- Leia o enunciado com **atenção**
- **Pergunte:** O que o código deve fazer?
- Identifique informações **únicas**, que farão a diferença



1. Ter Atenção no Enunciado 🦶

Em Bananinha City, chegou um novo carregamento de vacinas contra gripe. Porém, a distribuição está um caos, por não haver um sistema de gerenciamento das doses.

Sua missão como pessoa desenvolvedora é ajudar a criar um sistema de controle para os futuros pacientes receberem as doses da vacina.

A aplicação deverá **cadastrar** o nome do(a) **paciente**, ano de nascimento, nacionalidade e se já tomou ou não a vacina no ano corrente.

O que o código deve fazer?

Essas informações serão guardadas em uma base de dados para futuras consultas.

Seu primeiro desafio consiste em, dada uma lista dos cidadãos de Bananinha City, **selecionar as pessoas que: completam 60 anos ou mais no ano corrente; têm nacionalidade brasileira; e ainda não tomaram a vacina.**

Quais informações são únicas?



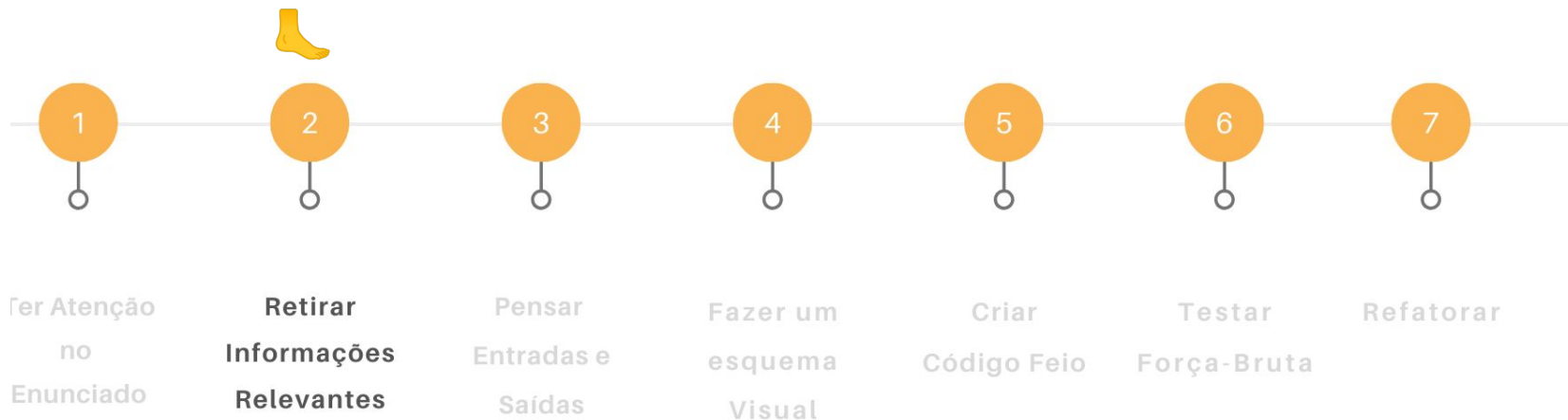
Passo 2

Labenu_



2. Retirar Informações Relevantes 🦶

- Separar **palavras chaves**, como **nome de variáveis**.
- **Verbos** criarão **funções**
- Há alguma **condição** para a **existência** do código?



2. Retirar Informações Relevantes 🦶

Em Bananinha City, chegou um novo carregamento de vacinas contra gripe. Porém, a distribuição está um caos, por não haver um sistema de gerenciamento das doses.

Sua missão como pessoa desenvolvedora é ajudar a criar um sistema de controle para os futuros **pacientes** receberem as doses da vacina.

A aplicação deverá cadastrar o **nome** do(a) paciente, **ano de nascimento**, **nacionalidade** e se já **tomou ou não a vacina** no ano corrente.

Quais são as palavras chaves?

Essas informações serão guardadas em uma base de dados para futuras consultas.

Seu primeiro desafio consiste em, dada uma **lista dos cidadãos** de Bananinha City, **selecionar as pessoas** que: completam **60 anos ou mais** no ano corrente; têm **nacionalidade brasileira**; e ainda **não tomaram a vacina**.

Informações relevantes p/ o código?

Passo 3

Labenu_



3. Pensar nas Entradas e Saídas 🦶

- **Como e quais** serão os **dados** recebidos?
- **Qual/Quais** devem ser as **possíveis respostas**(formato) código?
- **Criar uma função.** Com as entradas e saídas pensadas.



3. Pensar nas Entradas e Saídas 🦶

```
1 [  
2   {  
3     "nome": "Benevides Castro",  
4     "anoDeNascimento": 1950,  
5     "nacionalidade": "brasileira",  
6     "tomouVacina": false  
7   },  
8   {  
9     "nome": "Clara Ruiz",  
10    "anoDeNascimento": 1993,  
11    "nacionalidade": "colombiana",  
12    "tomouVacina": true  
13  },  
14  {  
15    "nome": "Maria Aparecida Luz",  
16    "anoDeNascimento": 1961,  
17    "nacionalidade": "brasileira",  
18    "tomouVacina": false  
19  }  
20 ]  
21
```



```
1 [  
2   {  
3     "nome": "Benevides Castro",  
4     "anoDeNascimento": 1950,  
5     "nacionalidade": "brasileira",  
6     "tomouVacina": false  
7   },  
8   {  
9     "nome": "Maria Aparecida Luz",  
10    "anoDeNascimento": 1961,  
11    "nacionalidade": "brasileira",  
12    "tomouVacina": false  
13  }  
14  
15
```



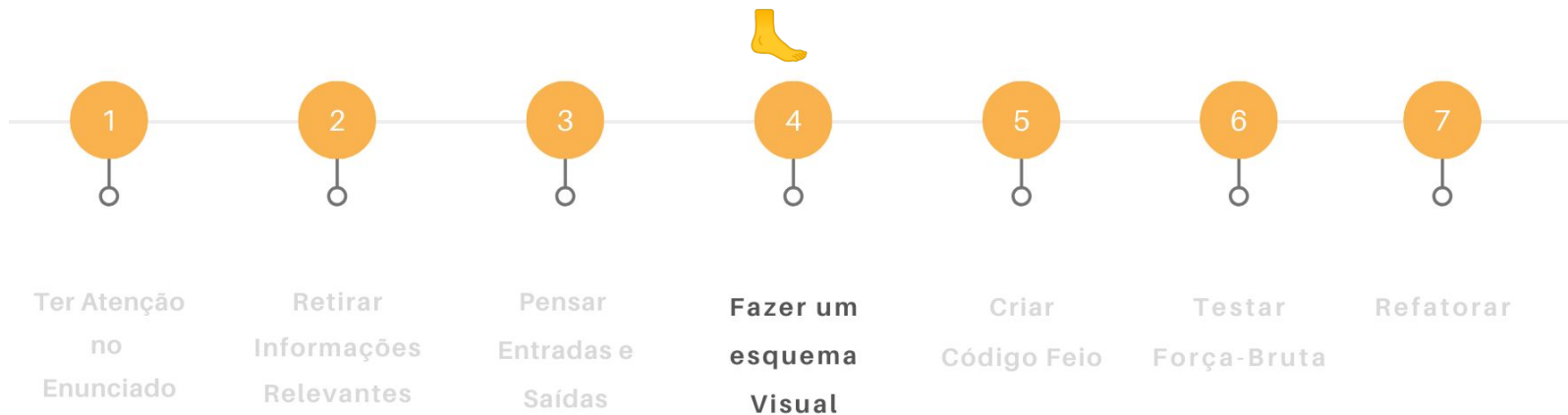
Passo 4

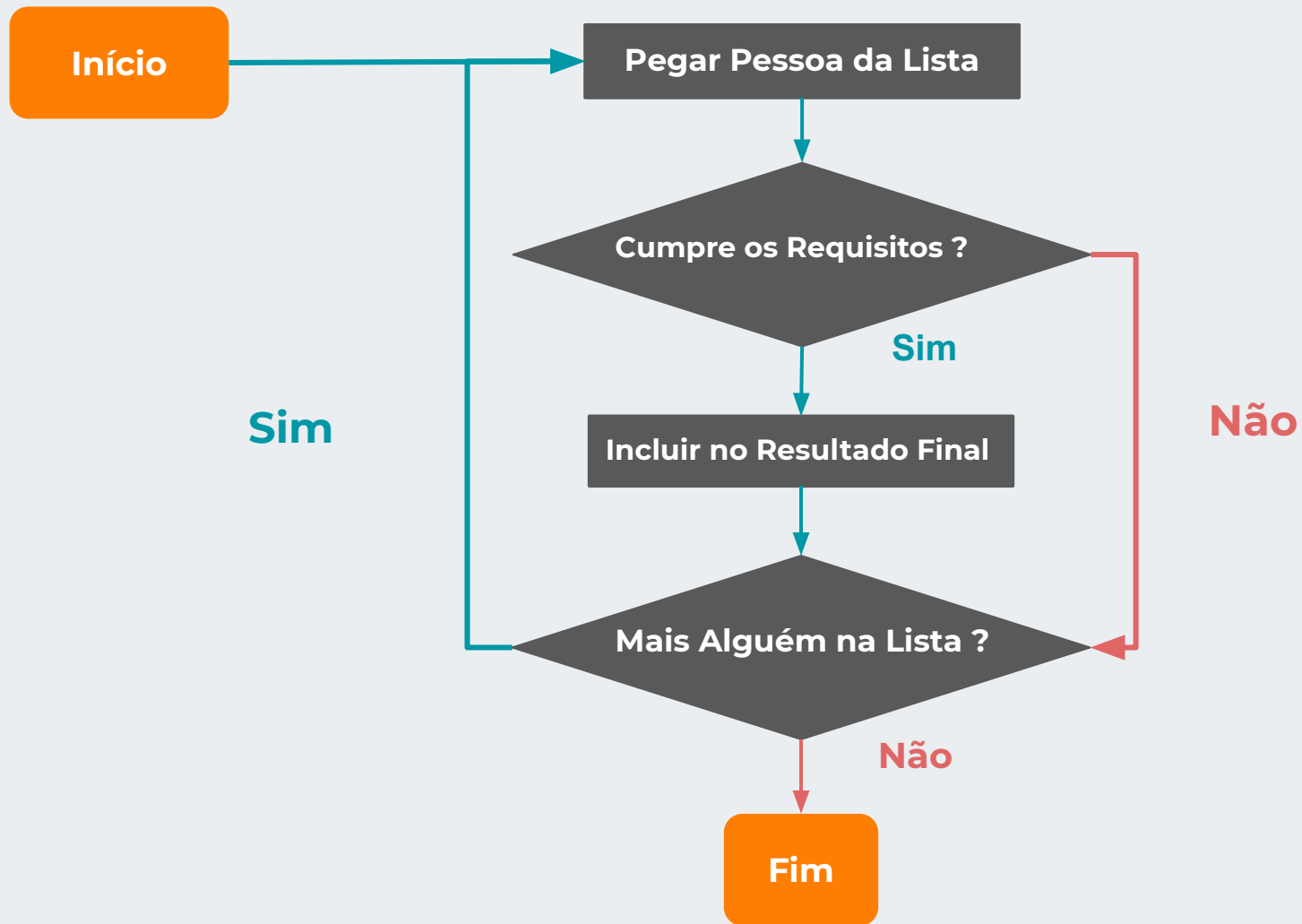
Labenu_



4. Fazer Um Esquema Visual 🦶

- **Crie exemplos** de situações antes
- Monte **diagramas** com a lógica do código
- Pense nos **possíveis cenários** e coloque inputs e outputs





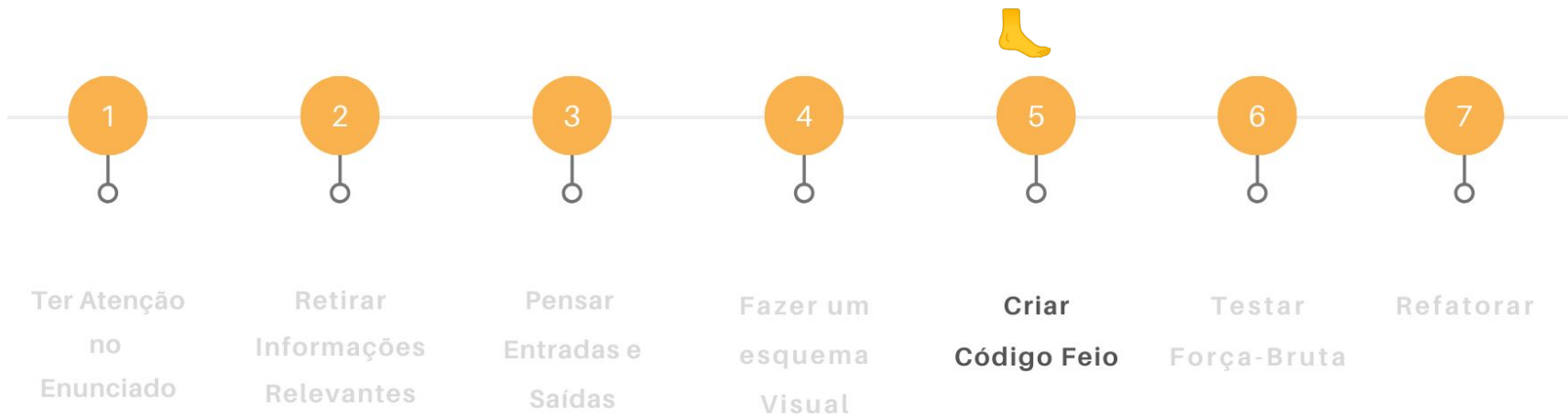
Passo 5

Labenu_



5. Criar Um Código Feio 🦶

- **Não tenha medo!**
Coloque no papel **todas possíveis soluções** pensadas.
- Escolha as **melhores soluções** e ponha em sua IDE.
- **Teste** cada uma das soluções, até obter uma resposta como solicitada no enunciado



5. Criar Um Código Feio 🦶

```
1 function selecionarPessoas(pessoas) {  
2  
3   const pessoasParaVacinar = []  
4  
5   for (let pessoa of pessoas) {  
6     if (  
7       pessoa.nacionalidade === "brasileira" &&  
8       !pessoa.tomouVacina &&  
9       2021 - pessoa.anoDoNascimento >= 60  
10    ){  
11      pessoasParaVacinar.push(pessoa)  
12    }  
13  }  
14  
15  return pessoasParaVacinar  
16 }  
17  
18  
19
```



Passo 6

Labenu_

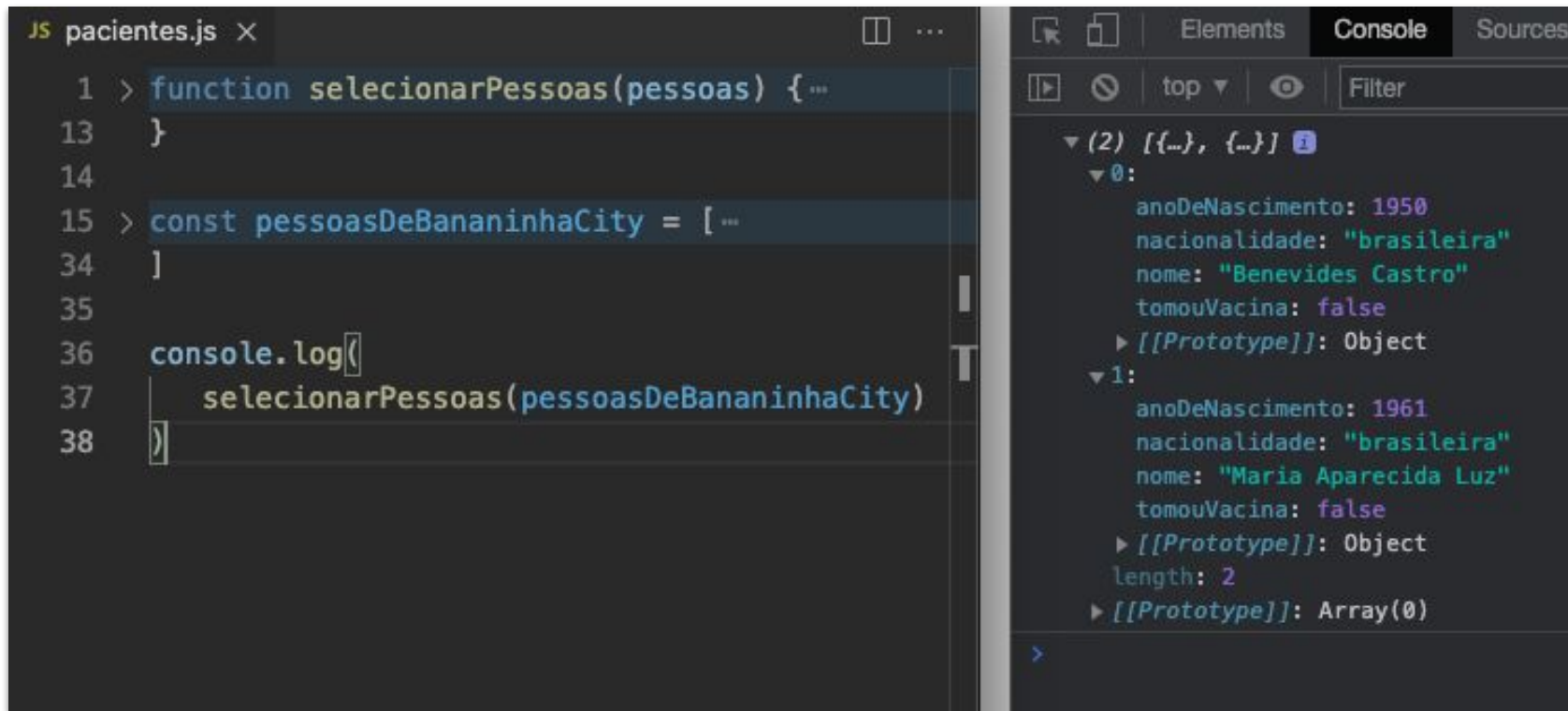


6. Testar: Força-Bruta

- **Solução óbvia** e pouco performática
- Escrever no **nosso idioma** o programa
- **Transcrever** para a linguagem de programação desejada.



6. Testar: Força-Bruta 🦶



The screenshot displays a web browser's developer console with two panels: the Sources panel on the left and the Console panel on the right.

Sources Panel: The file `pacientes.js` is open. The code is as follows:

```
1 > function selecionarPessoas(pessoas) { ...  
13 }  
14  
15 > const pessoasDeBananinhaCity = [...  
34 ]  
35  
36 console.log(  
37     selecionarPessoas(pessoasDeBananinhaCity)  
38 )
```

Console Panel: The console shows the execution of the function, resulting in an array of two objects. The first object is:

```
▼ (2) [{...}, {...}] ⓘ  
  ▼ 0:  
    anoDeNascimento: 1950  
    nacionalidade: "brasileira"  
    nome: "Benevides Castro"  
    tomouVacina: false  
    ▶ [[Prototype]]: Object  
  ▼ 1:  
    anoDeNascimento: 1961  
    nacionalidade: "brasileira"  
    nome: "Maria Aparecida Luz"  
    tomouVacina: false  
    ▶ [[Prototype]]: Object  
length: 2  
▶ [[Prototype]]: Array(0)
```

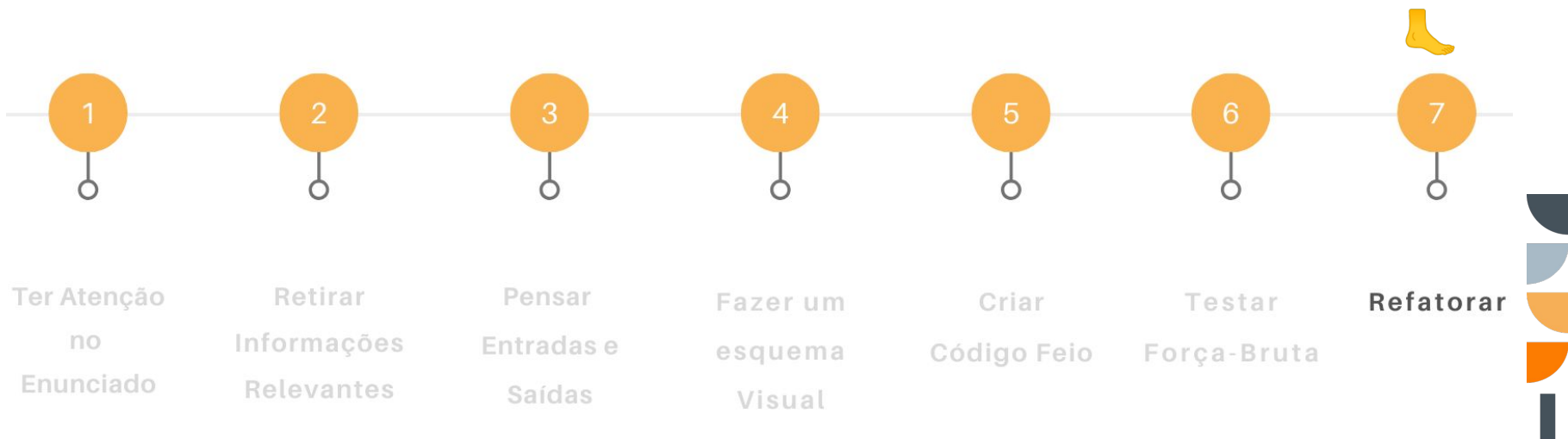
Passo 7

Labenu_



7. Refatorar

- Dê **bons nomes** para as variáveis e funções.
- **Separe as funções** de cada elemento.
- Comece com as **etapas principais.**



7. Refatorar 🦶

```
1 function selecionarPessoas(pessoas) {  
2  
3   const anoCorrente = new Date().getFullYear()  
4  
5   return pessoas.filter(pessoa => {  
6  
7     return (  
8       pessoa.nacionalidade === "brasileira" &&  
9         !pessoa.tomouVacina &&  
10        anoCorrente - pessoa.anoDoNascimento >= 60  
11      )  
12    })  
13  }  
14  
15  
16  
17
```



Dicas de Exercícios de Lógica

Labenu_



Dicas de exercícios

- Necessidade de **prática constante.**
- Melhorar a interpretação e o **raciocínio lógico.**
- Se preparar para **processos seletivos.**



[Português.](#)

HackerRank 

[Inglês](#)



Resumo

Labenu_



Resumo

- Ouça com **atenção**
- Desenhe um **exemplo**
- Comece fazendo um "código de **força bruta**"
- **Otimize** a sua solução
- **Percorra** a solução
- Implemente um **código bonito**
- **Teste** sua resposta





Obrigado(a)!