Passo-a-passo do algoritmo busca binária – usando o teste de mesa.

```
Algoritmo: busca-binária
  Seja V um vetor ordenado;
  Seja E um elemento que irá buscar no vetor;
  Seja B o índice mais baixo do vetor;
  Seja A o índice mais alto do vetor;

  enquanto B <= A faça:
      meio = (B + A) / 2
      chute = V[meio]
      se chute == E então: retorne meio
      se chute > E então: A = meio - 1
      senão: B = meio + 1 # se for menor.
 retorne -1
```

Exemplo 1

Seja **vetor** = `[10,11,12,13,14,15,16]` um vetor ordenado

Seja **valor** = 12, o valor a ser procurado no vetor

Seja **baixo** = 0 (inicialmente o menor índice no vetor)

Seja **alto** = 6 (inicialmente o maior índice no vetor) ou seja **`tamanho`**`(vetor) -1 = 7 - 1 => 6`

Começa o loop aqui:

Dentro do **loop** é sempre feito o cálculo da variável **meio** e da variável **chute**

1) **Meio =** (baixo + alto) / 2
   O meio sempre será a metade do vetor, se o número der decimal, arredonda para baixo.
2) **Chute** = vetor[meio]
   O chute é o valor no vetor que está no índice **meio**
   Próximo passo é comparar:
3) Se o **`chute == valor`** então achou o valor dentro do vetor no índice **meio**
   retorne **meio** e encerre o loop.
4) Se o **`chute > valor`** então atualiza a variável **alto** para **alto = meio -1**
   Retorna para o passo 1)
5) Se o **`chute < valor`** então atualize a variável **baixo** para baixo = **meio + 1**
6) **Retorna para o passo 1**

passo	instrução	baixo	alto	Meio	chute
-	-	-	-	-	-
	baixo = 0	0			
	Alto = tamanho(vetor)-1		6		
Loop 1	Meio = (baixo + alto)/2			3	
	Chute = vetor[meio]				13
	Chute == valor? Não... é maior				
	alto = meio -1		2		
Loop 2	Meio = (baixo+alto)/2			1	
	Chute = vetor[meio]				11
	Chute == valor? Não.. é menor				
	Baixo = meio +1	2			
Loop 3	Meio = (baixo + alto) / 2			2	
	Chute = vetor[meio]				12
	Chute == valor? É... retorne meio				

Quantidade de passos no pior caso: O(log n) = O(log 7)

Exemplo 2

(siga o algoritmo da primeira página)

Seja **vetor** = `[100, 102, 103, 107, 200, 450, 500, 502, 505, 600]` um vetor ordenado

Seja **valor** = 500, o valor a ser procurado no vetor

Seja **baixo** = 0 (inicialmente o menor índice no vetor)

Seja **alto** = 9 (inicialmente o maior índice no vetor) ou seja **tamanho**(vetor) -1 = 10-1 => 9

passo	instrução	baixo	alto	Meio	chute
-	-	-	-	-	-
	baixo = 0	0			
	Alto = tamanho(vetor)-1		9		
Loop 1	Meio = (baixo + alto)/2			4	
	Chute = vetor[meio]				200
	Chute == valor? Não... é menor				
	baixo = meio +1	5			
Loop 2	Meio = (baixo+alto)/2			7	
	Chute = vetor[meio]				502
	Chute == valor? Não.. é maior				
	alto = meio -1		6		
Loop 3	Meio = (baixo + alto) / 2			5	
	Chute = vetor[meio]				450
	Chute == valor? Não ... é menor				
	baixo = meio +1	6			
Loop 4	Meio = (baixo+alto)/2			6	
	Chute = vetor[meio]				500
	Chute == valor? É. Retorne meio.				

Quantidade de passos no pior caso: O(log n) = O(log 10)

Exercício 3

(siga o algoritmo da primeira página)

Seja **vetor** = [201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250] um vetor ordenado.

Seja **valor** = 225, o valor a ser procurado no vetor

Seja **baixo** = 0 (inicialmente o menor índice no vetor)

Seja **alto** = 49 (inicialmente o maior índice no vetor) ou seja **tamanho**(vetor) -1 = 50-1 => 49

passo	instrução	baixo	alto	Meio	chute
-	-	-	-	-	-
	baixo = 0	0			
	Alto = tamanho(vetor)-1		49		
Loop 1	Meio = (baixo + alto)/2			24	
	Chute = vetor[meio]				225
	Chute == valor? É. Retorne meio.				

Quantidade de passos no **pior caso**: O(log n) = O(log 50) => 6

Exercício 4

(siga o algoritmo do exercício 1)

Seja **vetor** = [201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250] um vetor ordenado.

Seja **valor** = 224, o valor a ser procurado no vetor

Seja **baixo** = 0 (inicialmente o menor índice no vetor)

Seja **alto** = 49 (inicialmente o maior índice no vetor) ou seja **tamanho**(vetor) -1 = 50-1 => 49

passo	instrução	baixo	alto	Meio	chute
-	-	-	-	-	-
	baixo = 0	0			
	Alto = tamanho(vetor)-1		49		
Loop 1	Meio = (baixo + alto)/2			24	
	Chute = vetor[meio]				225
	Chute == valor? Não.. é maior				
	alto = meio -1		23		
Loop 2	Meio = (baixo + alto)/2			11	
	Chute = vetor[meio]				212
	Chute == valor? Não.. é menor				
	baixo = meio + 1	12			
Loop 3	Meio = (baixo + alto)/2			17	
	Chute = vetor[meio]				218
	Chute == valor? Não.. é menor				
	baixo = meio + 1	18			
Loop 4	Meio = (baixo + alto)/2			20	
	Chute = vetor[meio]				221
	Chute == valor? Não.. é menor				
	baixo = meio + 1	21			
Loop 5	Meio = (baixo + alto)/2			22	
	Chute = vetor[meio]				223
	Chute == valor? Não.. é menor				
	baixo = meio + 1	23			
Loop 6	Meio = (baixo + alto)/2			23	
	Chute = vetor[meio]				224
	Chute == valor? Não.. é menor				
	Chute == valor? É. Retorne meio.				

Quantidade de passos no **pior caso**: O(log n) = O(log 50) => 6