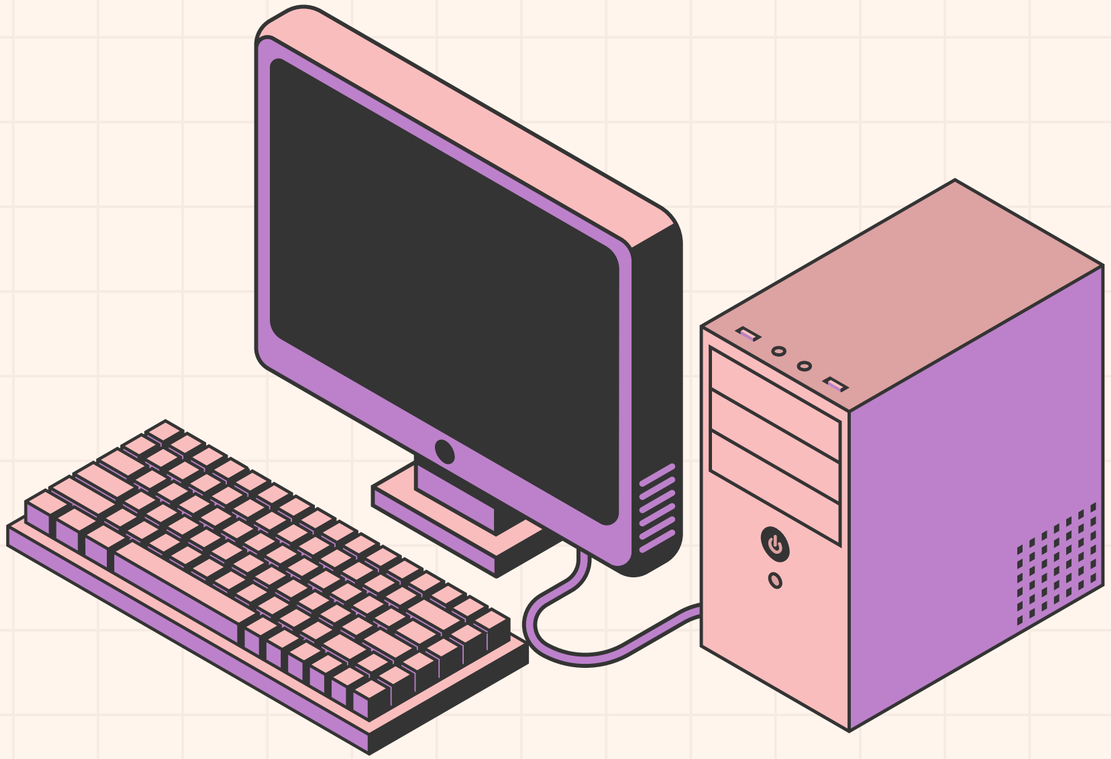
Firjan SENAI

TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

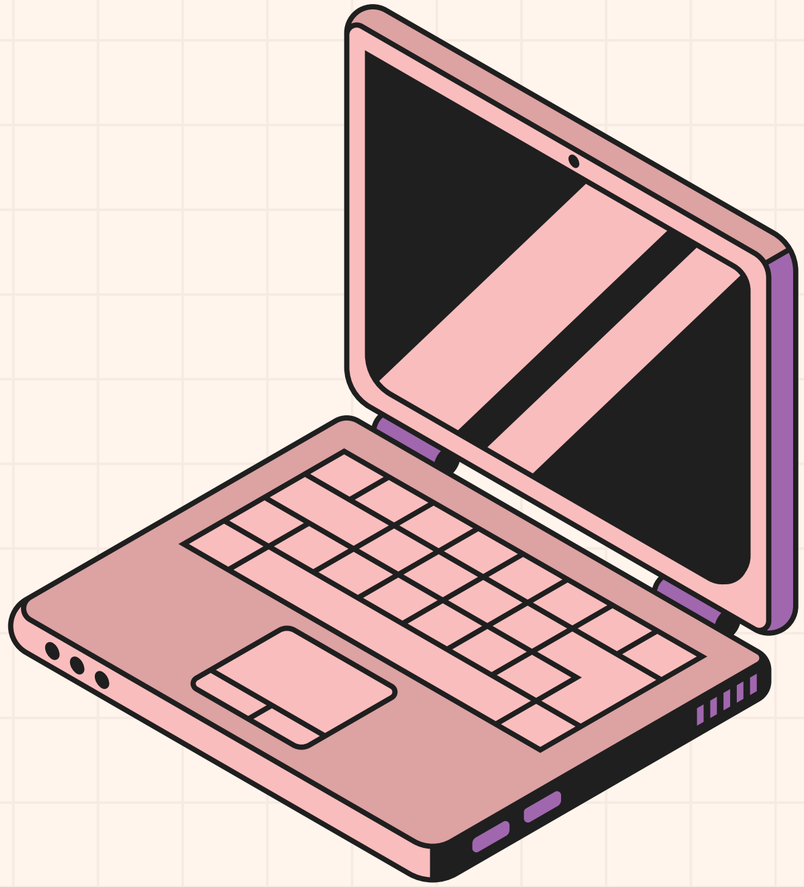
Professor Marco Aurelio



Firjan SENAI

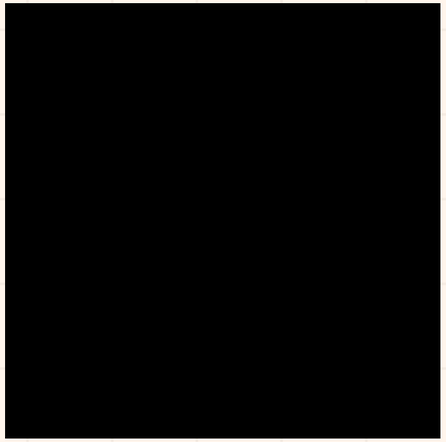
LÓGICA DE PROGRAMAÇÃO

- Tópico 01: Binários
- Tópico 02: Algoritmos e eficiência
- Tópico 03: Pseudocódigo



Firjan SENAI

INPUT
ENTRADA



OUTPUT
SAÍDA

BINÁRIOS

- Um computador, no nível mais baixo, armazena dados em binário, um sistema numérico em que existem apenas dois dígitos, 0 e 1.
- Quando aprendemos a contar, podemos ter usado um dedo para representar uma coisa. Esse sistema é chamado de unário. Quando aprendemos a escrever números com os dígitos de 0 a 9, aprendemos a usar o sistema decimal.
- Por exemplo, sabemos que o seguinte representa cento e vinte e três:

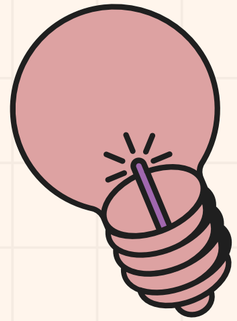
1 2 3

- O "3" está na coluna das unidades, o "2" está na coluna das dezenas e o "1" está na coluna das centenas.

- Portanto, "123" é 100 × 1 + 10 × 2 + 1 × 3 = 100 + 20 + 3 = 123.
- Cada lugar para um dígito representa uma potência de dez, pois há dez dígitos possíveis para cada lugar.

Firjan SENAI

BINÁRIOS



- Em binário, com apenas dois dígitos, temos potências de dois para cada valor de lugar:

4 2 1
0 0 0

- Isso ainda seria igual a 0.
- Agora, se alterarmos o valor binário para, digamos, "0 1 1", o valor decimal será 3.

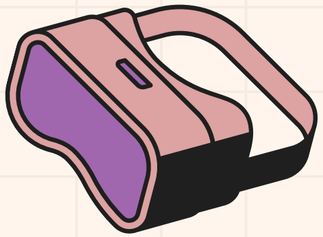
4 2 1
0 1 1

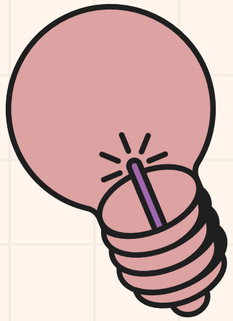
- Se quiséssemos representar 8, precisaríamos de outro dígito:

8 4 2 1
1 0 0 0

- E o binário faz sentido para computadores porque nós os alimentamos com eletricidade, que pode estar ligada ou desligada, então cada bit só precisa estar ligado ou desligado. Em um computador, há milhões ou bilhões de interruptores chamados transistores que podem armazenar eletricidade e representar um bit como "ligado" ou "desligado".
- Com bits ou dígitos binários suficientes, os computadores podem contar até qualquer número.

Firjan SENAI

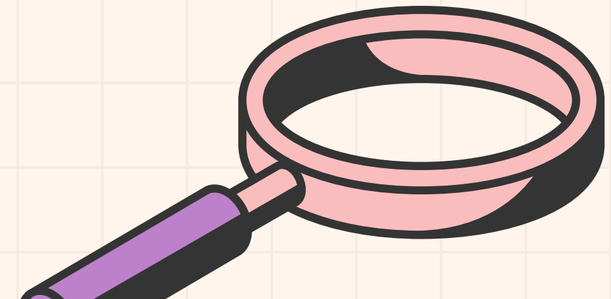




REPRESENTAÇÃO DOS DADOS

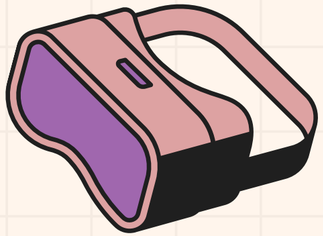
- Para representar letras, tudo o que precisamos fazer é decidir como os números são mapeados para as letras. Alguns humanos, muitos anos atrás, decidiram coletivamente usar um mapeamento padrão chamado ASCII. A letra "A", por exemplo, é o número 65, e "B" é 66, e assim por diante. O mapeamento também inclui pontuação e outros símbolos. Outros caracteres, como letras com acentos e emojis, fazem parte de um padrão chamado Unicode que usa mais bits do que ASCII para acomodar todos esses caracteres.

    - Quando recebemos um emoji, nosso computador está na verdade apenas recebendo um número decimal como "128514" ("1111101100000010" em binário, se você puder ler isso mais facilmente) que ele então mapeia para a imagem do emoji.

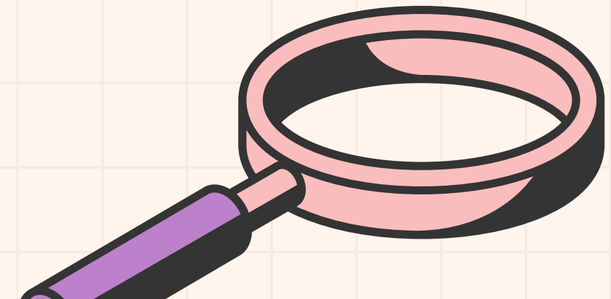


Firjan SENAI

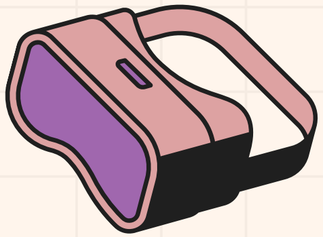
ASCII TABLE

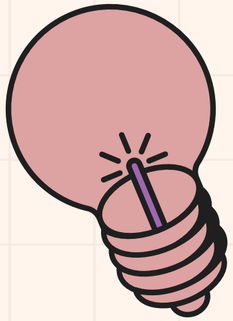


Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]



Firjan SENAI



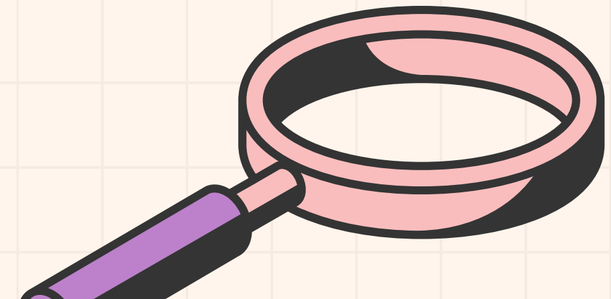


REPRESENTAÇÃO DOS DADOS

Uma imagem também é composta de muitos pontos quadrados menores, ou pixels, cada um dos quais pode ser representado em binário com um sistema chamado RGB, com valores para luz vermelha, verde e azul em cada pixel. Ao misturar diferentes quantidades de cada cor, podemos representar milhões de cores:

72	73	33

Os valores vermelho, verde e azul são combinados para obter uma cor amarelo claro:



REPRESENTAÇÃO DOS DADOS

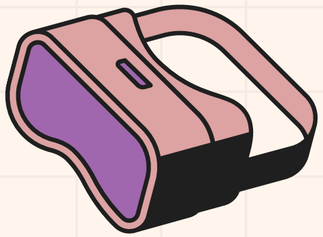
Podemos ver isso em um emoji se aumentarmos o zoom o suficiente:

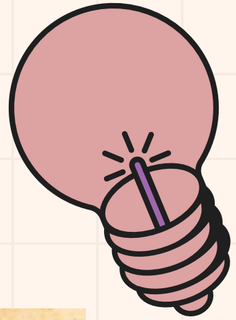
Firjan SENAI

Firjan SENAI

CÓDIGO

Firjan SENAI

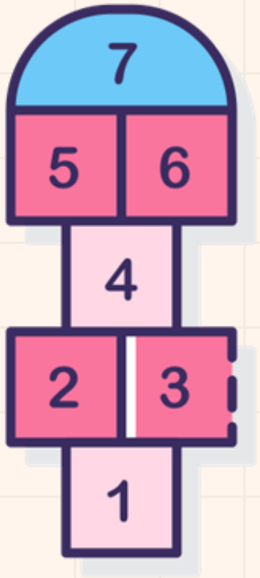


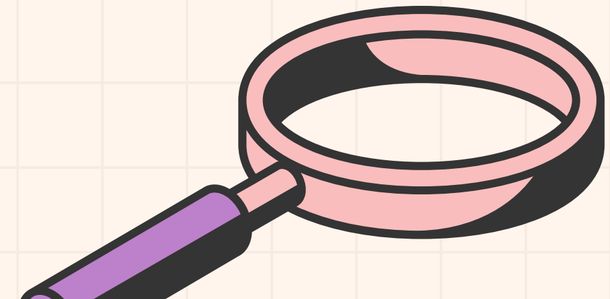


ALGORITMOS

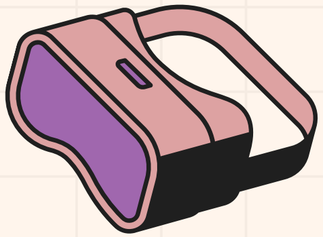
Sequência finita de passos que levam à execução de umatarefa

Algo muito comum no nosso dia a dia, sendo de TI ou não

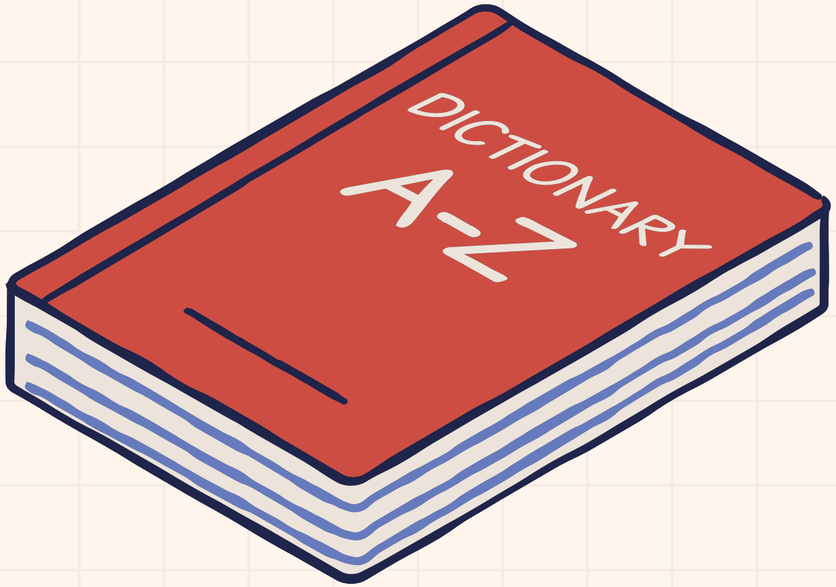


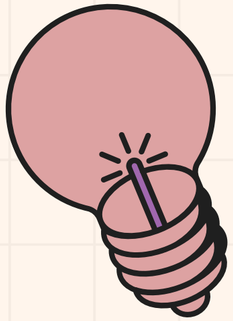


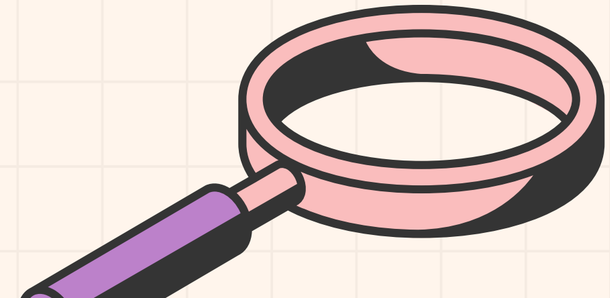
Firjan SENAI

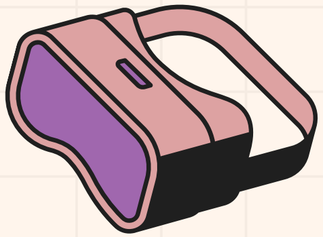


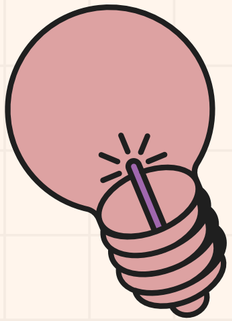
ALGORITMO DE BUSCA











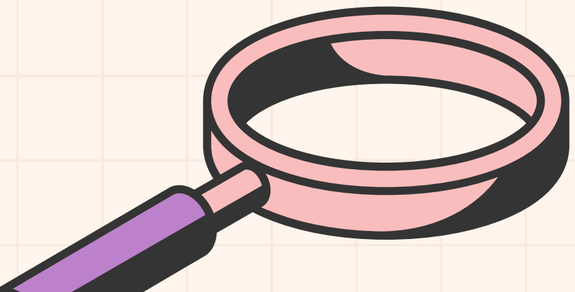
ALGORITMOS E DESEMPENHO

Digamos que queremos encontrar o significado da palavra **Jogo** no dicionário.

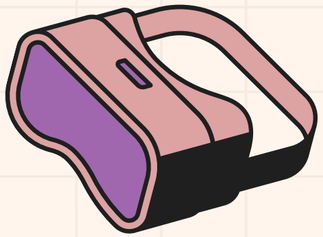
- Podemos começar folheando o livro uma página de cada vez até encontrarmos Jogo ou chegarmos ao fim do livro.

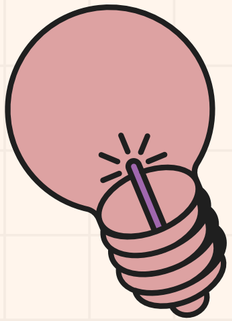
- Podemos também folhear duas páginas por vez, mas se formos longe demais, teremos de saber voltar uma página.

- Mas uma forma ainda mais eficiente seria abrir a lista telefônica ao meio, decidir se Jogo estará na metade esquerda ou direita do livro (já que o livro está em ordem alfabética) e descartar imediatamente metade do problema. Podemos repetir isso, dividindo o problema pela metade sempre. Com 1024 páginas para começar, precisaríamos de apenas 10 passos dividindo pela metade antes de sobrar apenas uma página para verificarmos.

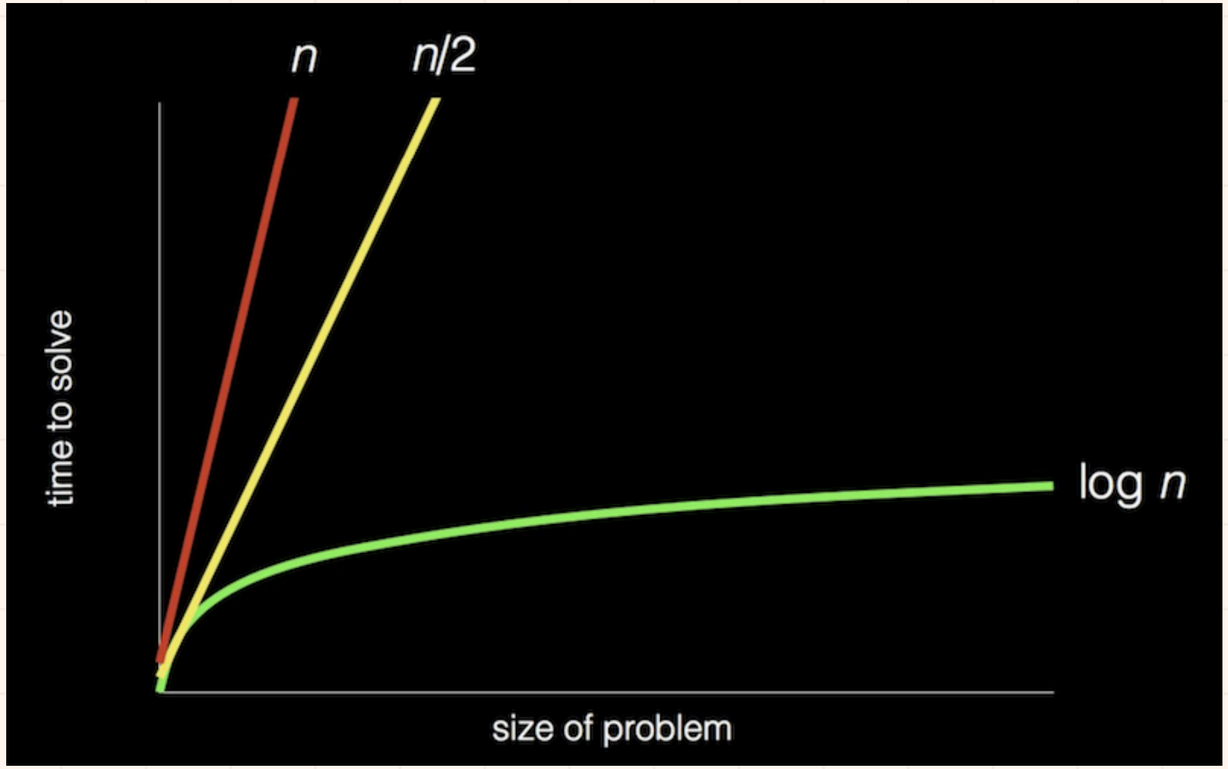


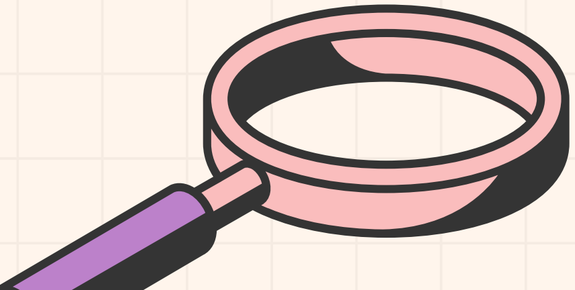
Firjan SENAI



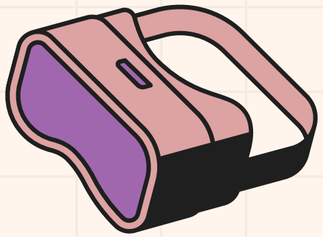


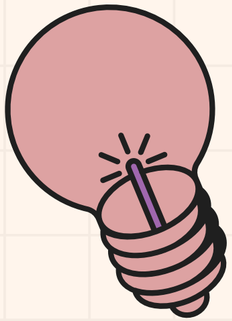
ALGORITMOS E DESEMPENHO





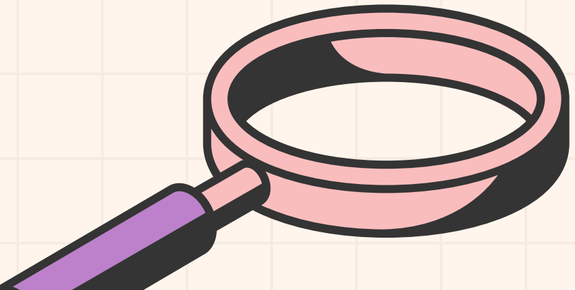
Firjan SENAI





ALGORITMOS E DESEMPENHO

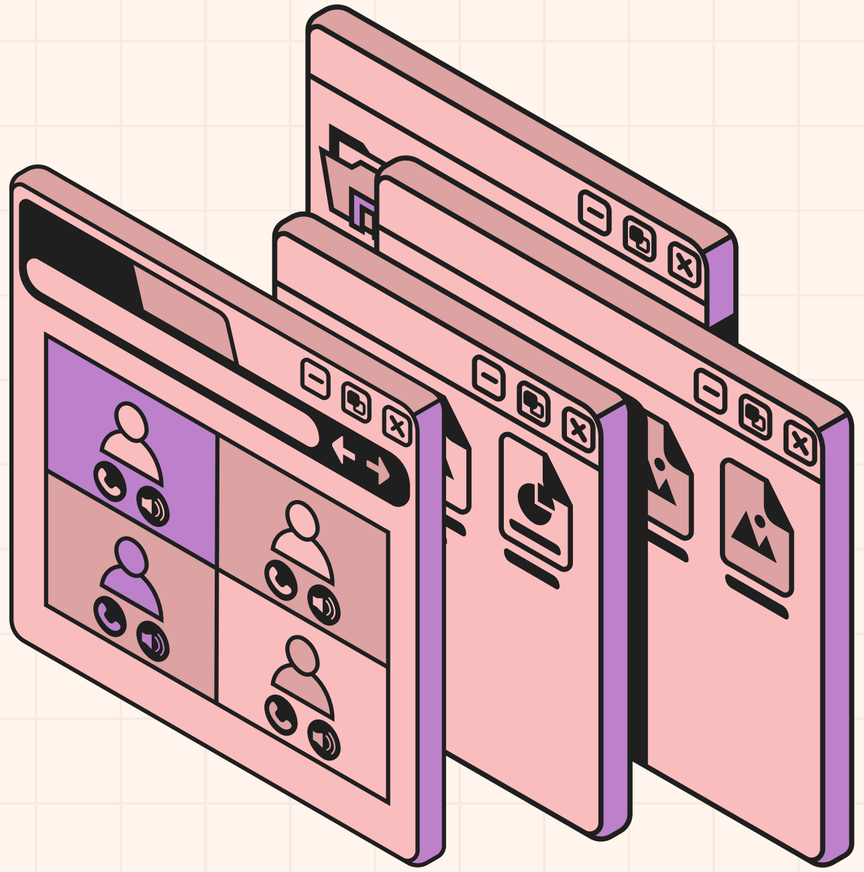
- A nossa primeira solução, uma página por vez, é como a linha vermelha: nosso tempo para resolver aumenta linearmente conforme o tamanho do problema aumenta.
- A segunda solução, duas páginas por vez, é como a linha amarela: a inclinação é menor, mas ainda linear.
- Nossa solução final é como a linha verde: logarítmica, pois nosso tempo para resolver aumenta cada vez mais devagar conforme o tamanho do problema aumenta. Ou seja, se a dicionário passasse de 1000 para 2000 páginas, precisaríamos de mais um passo para encontrar Jogo. Se o tamanho dobrasse novamente de 2000 para 4000 páginas, ainda precisaríamos de apenas mais um passo.



PSEUDOCÓDIGO

Podemos escrever pseudocódigo, uma sintaxe informal que é apenas uma versão mais específica de português (ou outra língua humana) que representa nosso algoritmo:

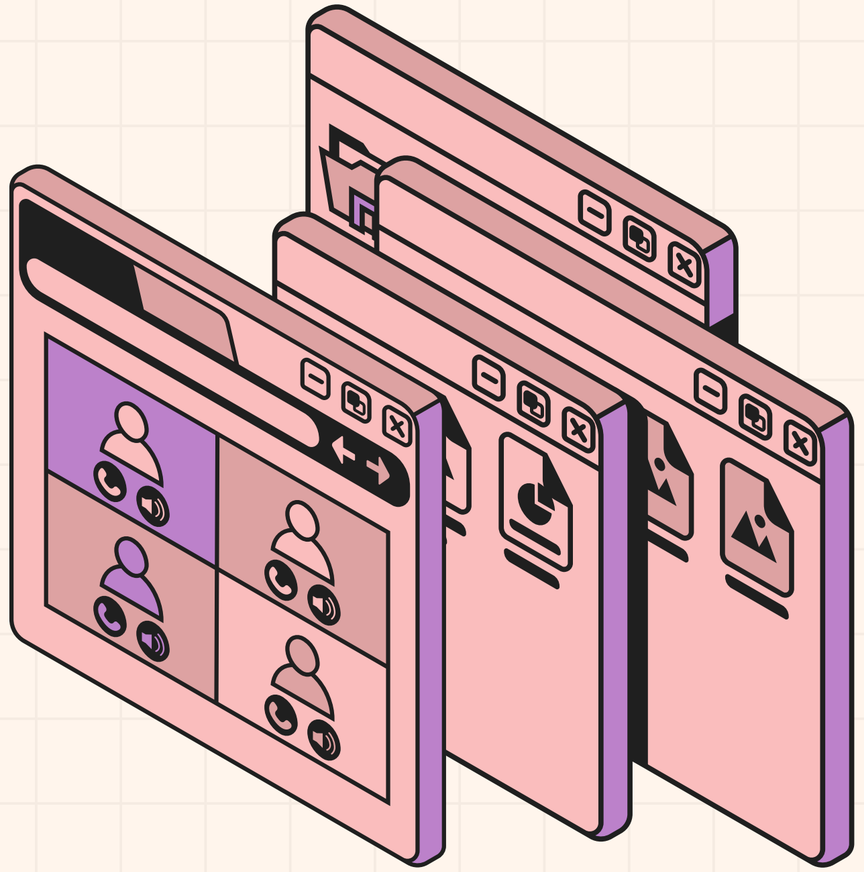
```
1  Pegue o dicionário
2  Abra o meio do dicionário
3  Olhe a página
4  Se Jogo estiver na página
5      Leia o significado de Jogo
6  Senão se Jogo estiver antes no livro
7      Abra o meio da metade esquerda do livro
8      Retorne à linha 3
9  Senão se Jogo estiver depois no livro
10     Abra o meio da metade direita do livro
11     Retorne à linha 3
12 Senão
13     Pare
```



PSEUDOCÓDIGO

Algumas dessas linhas começam com verbos ou ações. Passaremos a chamá-los de funções:

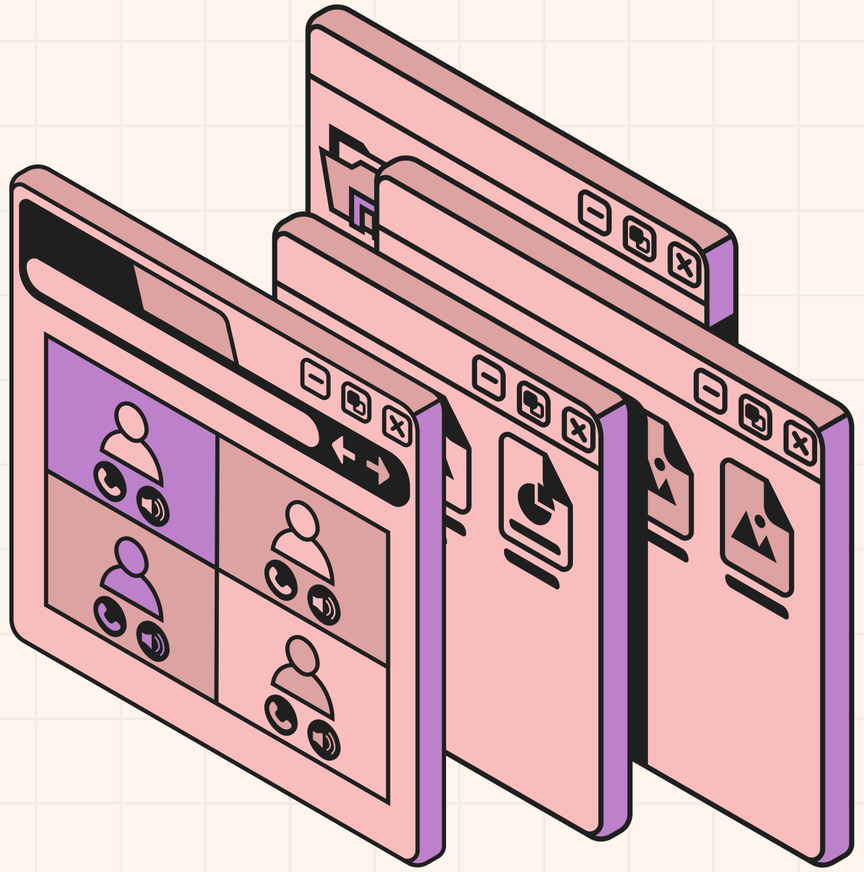
```
1  Pegar o dicionário
2  Abrir no meio do dicionário
3  Olhar a página
4  Se Jogo estiver na página
5      Ler o significado de Jogo
6  Senão se Jogo estiver antes no livro
7      Abrir no meio da metade esquerda do livro
8      Retornar à linha 3
9  Senão se Jogo estiver depois no livro
10     Abrir no meio da metade direita do livro
11     Retornar à linha 3
12 Senão
13     Parar
```



PSEUDOCÓDIGO

Também temos ramificações que levam a caminhos diferentes, como bifurcações na estrada, que chamaremos de condições:

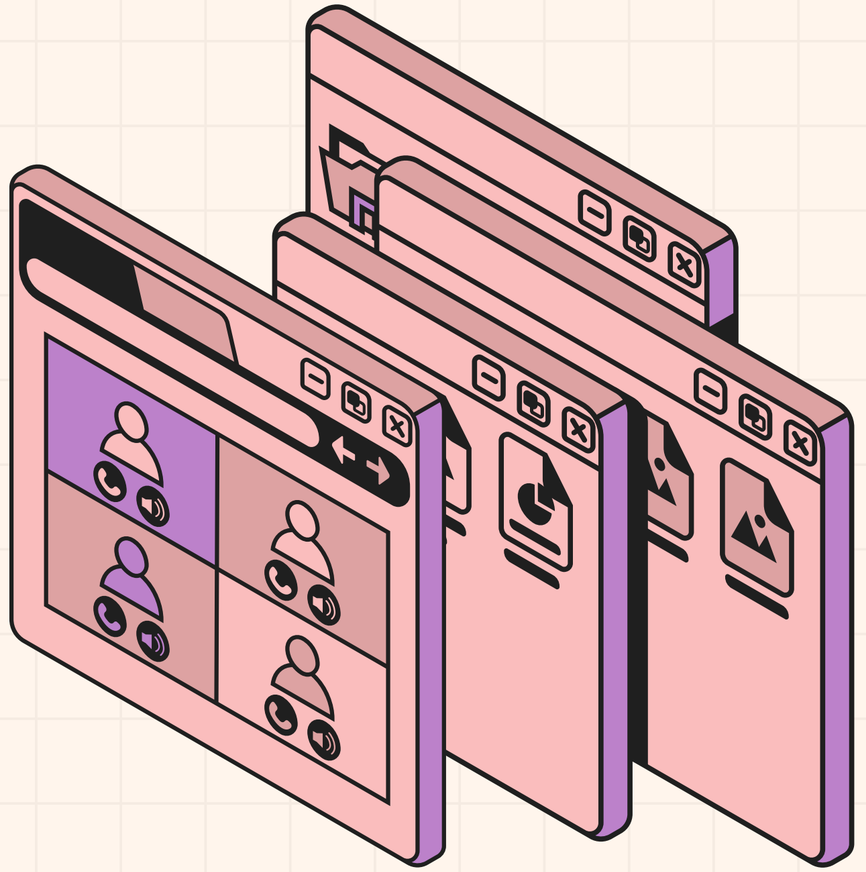
```
1  Pegar o dicionário
2  Abrir no meio do dicionário
3  Olhar a página
4  Se Jogo estiver na página
5      Ler o significado de Jogo
6  Senão se Jogo estiver antes no livro
7      Abrir no meio da metade esquerda do livro
8      Retornar à linha 3
9  Senão se Jogo estiver depois no livro
10     Abrir no meio da metade direita do livro
11     Retornar à linha 3
12 Senão
13     Parar
```



PSEUDOCÓDIGO

E as perguntas que decidem para onde vamos são chamadas de expressões booleanas, que eventualmente resultam em um valor verdadeiro ou falso:

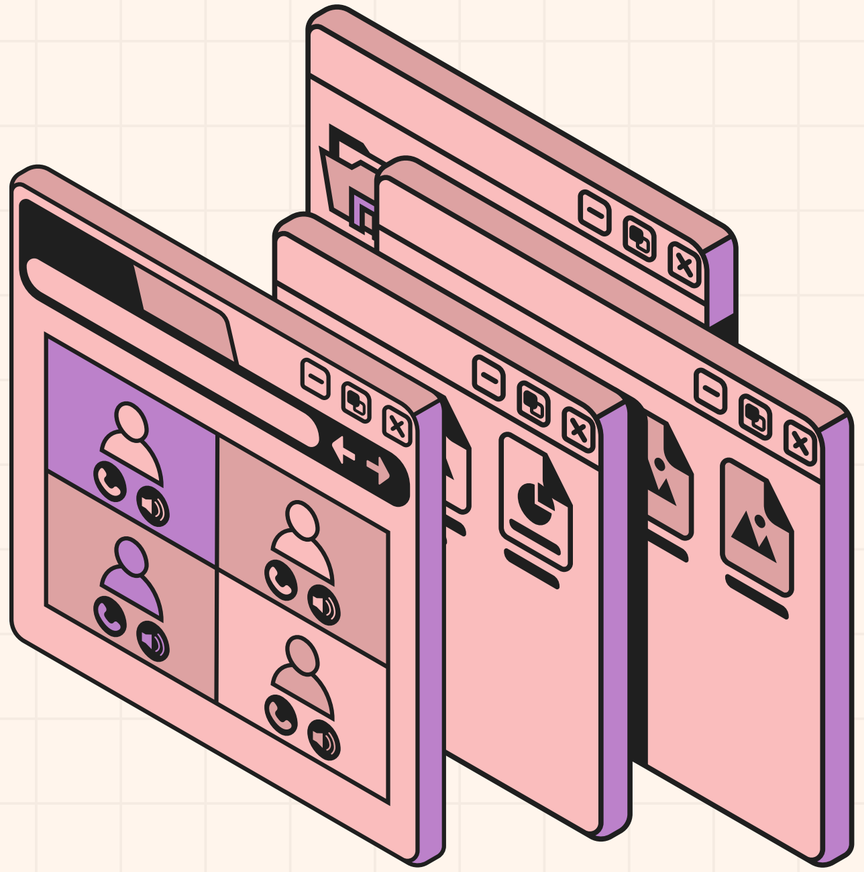
```
1  Pegar o dicionário
2  Abrir no meio do dicionário
3  Olhar a página
4  Se Jogo estiver na página
5      Ler o significado de Jogo
6  Senão se Jogo estiver antes no livro
7      Abrir no meio da metade esquerda do livro
8      Retornar à linha 3
9  Senão se Jogo estiver depois no livro
10     Abrir no meio da metade direita do livro
11     Retornar à linha 3
12 Senão
13     Parar
```



PSEUDOCÓDIGO

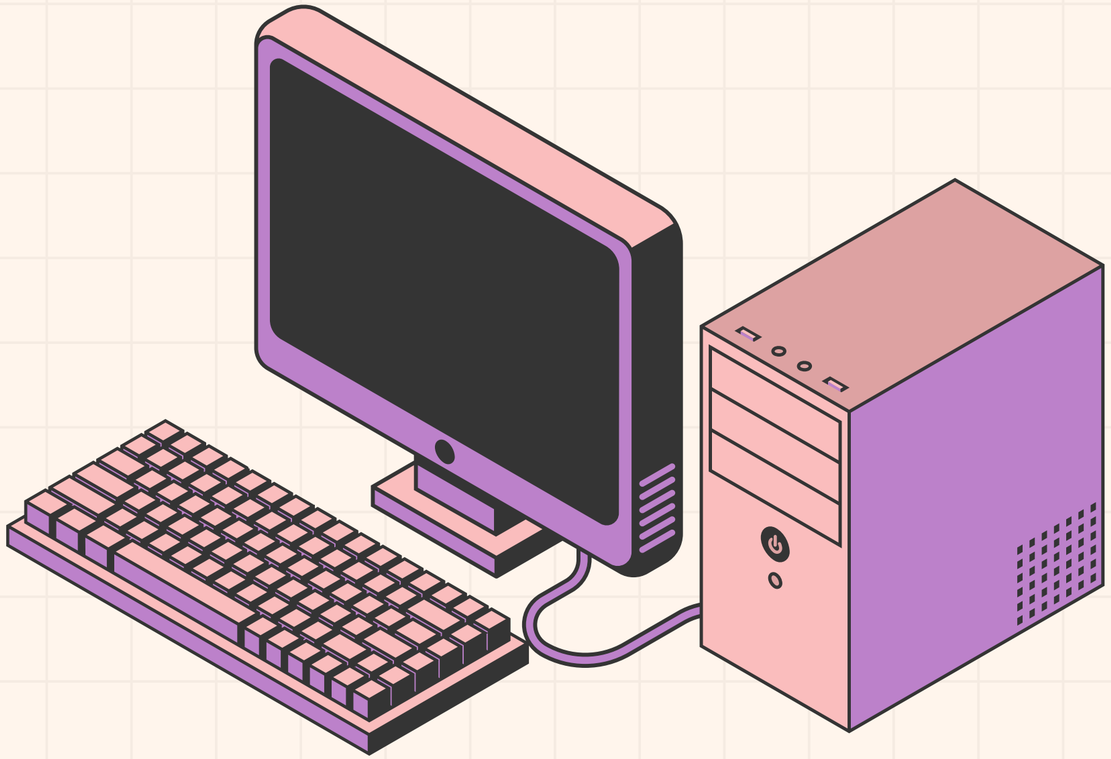
Finalmente, temos palavras que levam a ciclos, onde podemos repetir partes do nosso programa, chamadas de loops:

```
1  Pegar o dicionário
2  Abrir no meio do dicionário
3  Olhar a página
4  Se Jogo estiver na página
5      Ler o significado de Jogo
6  Senão se Jogo estiver antes no livro
7      Abrir no meio da metade esquerda do livro
8      Retornar à linha 3
9  Senão se Jogo estiver depois no livro
10     Abrir no meio da metade direita do livro
11     Retornar à linha 3
12 Senão
13     Parar
```



SCRATCH

- Podemos escrever programas com os blocos de construção que acabamos de descobrir:
  - **funções**
  - **condições**
  - **expressões booleanas**
  - **laços**
- Usaremos uma linguagem de programação gráfica chamada Scratch, na qual arrastaremos e soltaremos blocos que contêm instruções.



Firjan SENAI

SCRATCH

- Atividades e prática no scratch