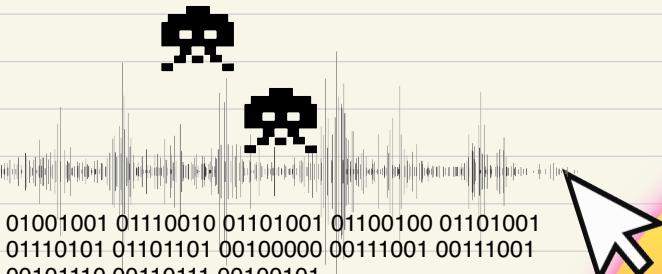
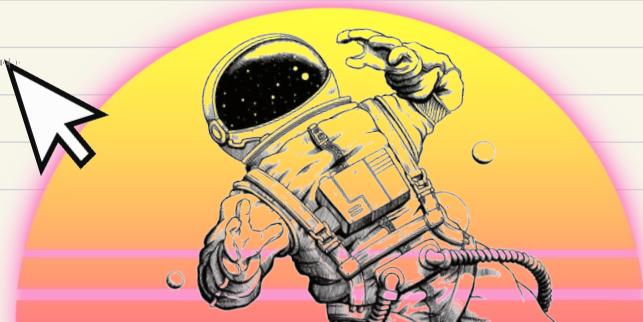
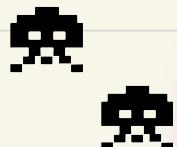


Números Binários

O sistema binário é um sistema de numeração que usa apenas dois dígitos 0 e 1. Isso porque os computadores trabalham com eletricidade, que pode estar em dois estágios: ligado (1) ou desligado (0). Dessa forma, tudo o que o computador faz é representado por sequências desses dois valores: 0 e 1.



01001001 01110010 01101001 01100100 01101001
01110101 01101101 00100000 00111001 00111001
00101110 00110111 00100101



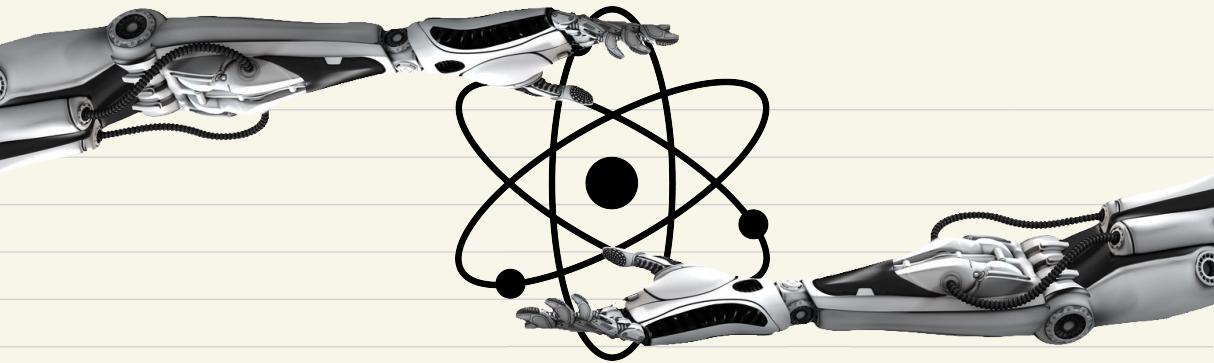
Entering the hardware

→ O fato dos computadores sempre representarem a eletricidade com 0 e 1 está no conhecimento da **eletrônica digital**, pois, dentro do computador tudo funciona com circuitos eletrônicos, e estes mesmos circuitos só operam com tensões da eletricidade, então eles usam apenas 0 e 1, porque trabalhar com outros valores de voltagem como 0V, 0.5V, 1.2V, 2.8V seria um pesadelo, pois eles precisariam ser extremamente precisos em calcular essas voltagens.

e a própria energia é uma matéria instável, então pequenas variações nas voltagens poderiam ocasionar erros, além da própria dificuldade em construir circuitos confiáveis e o próprio processamento seria mais lento, porque o computador precisaria ter outros números, aumentando a complexidade e diminuindo sua escalabilidade. Por isso operam em:

0 → baixa tensão (0V)

1 → alta tensão (5V / 3.3V)



1 byte = 1 número binário

Ex: O sistema binário é baseado em potências de 2, diferente dos números decimais que apresentam potências de 10. Nessa forma podemos estabelecer que cada bit estabelecido pelo computador possui um valor dentro da exponenciação, ou seja, os valores vão se multiplicando até chegar a 7-bit.

Bit (posição)	Potência de 2	Valor Decimal
7º bit	2^7	128
6º bit	2^6	64
5º bit	2^5	32
4º bit	2^4	16
3º bit	2^3	8
2º bit	2^2	4
1º bit	2^1	2
0º bit	2^0	1

6 → Na tabela acima podemos entender as diversas combinações do 7º bit, que permitiu que nós pudéssemos representar os 128 caracteres estadunidenses no computar, como números, letras e acentuações.

Binário	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

 Nessa imagem temos a noção de construção de uso dos bits. Como existem apenas três números, temos três bits, e podemos ter até no máximo 7 representações, já que $2^3 = 2 \times 2 \times 2 = 8$, mas devemos

começar o raciocínio a partir
do zero, então: $2^3 = 7$ bits

1 2 3 4 5 6 7 ~~8~~
→ 0 1 2 3 4 5 6 7

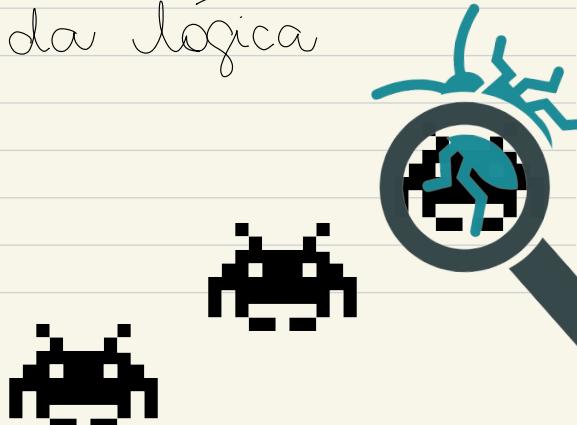
Binário	Decimal
00000000	0
00000001	1
00000010	2
00000011	3
...	...
11111110	254
11111111	255

 No quadro acima vamos ter
8 números binários, ou seja, 8 formas
de representá-los matematicamente.

Todavia, como o numero binário é sempre representado por um expoente de 2, devemos pensar que:

$$8\text{-bit} = 2^8 = 256 - 1 (-1=0) = 255$$

Nessa equação, podemos entender as seguintes abstrações: para 8-bit, vamos ter 256 possibilidades de representação em binário, mas sempre teremos que subtrair 1, pois assim, consideraremos a contagem a partir de zero sendo um número ativo e representá-lo dentro da lógica computacional.



Bits (n)	Total de combinações (2^n)	Maior número decimal ($2^n - 1$)
1	2	1
2	4	3
3	8	7
4	16	15
8	256	255
16	65536	65535
32	4294967296	4294967295
64	18446744073709551616	18446744073709551615

↗ Nesse quadro podemos entender como o bit trabalha, já que vamos ter um número decimal específico para combinações binárias, ou seja, dentro de 3-bit vamos ter 7, pois será $2^3 = 2 \times 2 \times 2 = 8 - 1 = 7$, e esse mesmo padrão se repete por $2^1, 2^2, 2^3, 2^4, 2^8, 2^{16}, 2^{32}, 2^{64}$ e todos eles terão as representações em n^2 , e apenas os números decimais usados que precisaram ser -1, pois considera o zero

ASCII → Código Padrão

Americano para Intercâmbio
de Informações

→ No fim das contas, para que precisamos de tantas formas de combinação assim? Será que realmente usamos 64 bits (2^{64})? São inúmeras perguntas que só podem ser respondidas com outra pergunta:

“o mundo inteiro seria capaz de falar e se expressar em apenas 1 língua?”

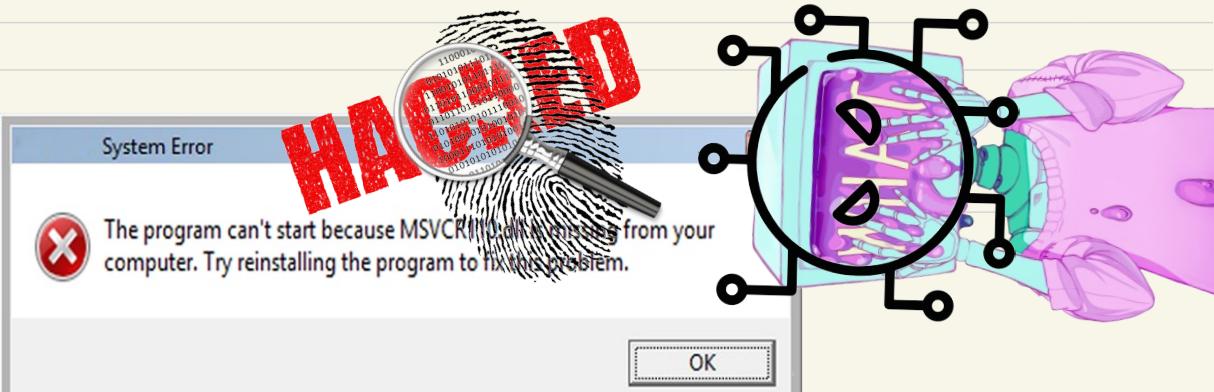
Ao longo da história da humanidade algumas línguas

foram faladas mais que outras e desempenharam um papel central em algumas culturas, mas jamais conseguiríamos falar apenas uma língua, mas o computador consegue!

0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	'	112	P
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	
13	CR	29	GS	45	-	61	=	77	M	93]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	-
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

→ Esse quadro é a representação visual do ASCII e foi retirado da aula de David Malan, professor no

curso C550 da universidade de Harvard. Nesse quadro vamos entender que estamos trabalhando com binário 7-bit, pois temos o limite de 127 caracteres, sendo assim:
 $2^7 = 128 - 1 (-1=0) = 127$. Com essa tabela podemos ordenar que o computador expresse qualquer função que esteja no quadro ASCII, mas vale lembrar que só possui os caracteres estadunidenses



4 - 8 - 16 - 32

→ Os números acima representam a quantidade de bits disponíveis para serem usados atualmente. Mas a pergunta que fica é: para que é necessário termos tantos bits?

→ Vale lembrar que o ASCII trabalha apenas em 7-bit e só pode representar os caracteres próprios do Estados Unidos. Mas e o restante das linguagens humanas? Como o sânscrito, chinês, ou até mesmo emoji e imagens em 3D

Tudo o que operaiona no nosso planeta pode ser lido ou interpretado através dos números binários. Cada letra, símbolo ou música, é apenas representado como uma sequência de números binários pelo computador, e nada foge dessa regra.

How to discovery binary?

Podemos descobrir os números binários facilmente, analisando o número representado em cada tabela!

Por exemplo: o símbolo "%" é apenas representado pelo decimal 37.

Vamos ter a possibilidade de calcular quais são os decimais através de uma regra: como os números binários serão sempre dois números, vamos dividir o decimal pelo binário

$$37 \div 2 = 18,5$$

Nesse caso vamos entender que a divisão por 2, dava justamente 0,5 já que $18 + 0,5 = 18,5$. Assim, quando nos falamos em divisões inteiros, não é possível considerar números frazionados, então, o 0,5 será representado como 1, e assim continuaremos a equação:

$$37 \div 2 = 18 \text{ sobrou } 1$$

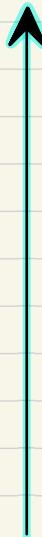
$$18 \div 2 = 9 \text{ sobrou } 0$$

$$9 \div 2 = 4 \text{ sobrou } 1$$

$$4 \div 2 = 2 \text{ sobrou } 0$$

$$2 \div 2 = 1 \text{ sobrou } 0$$

$$1 \div 2 = 0 \text{ sobrou } 1$$



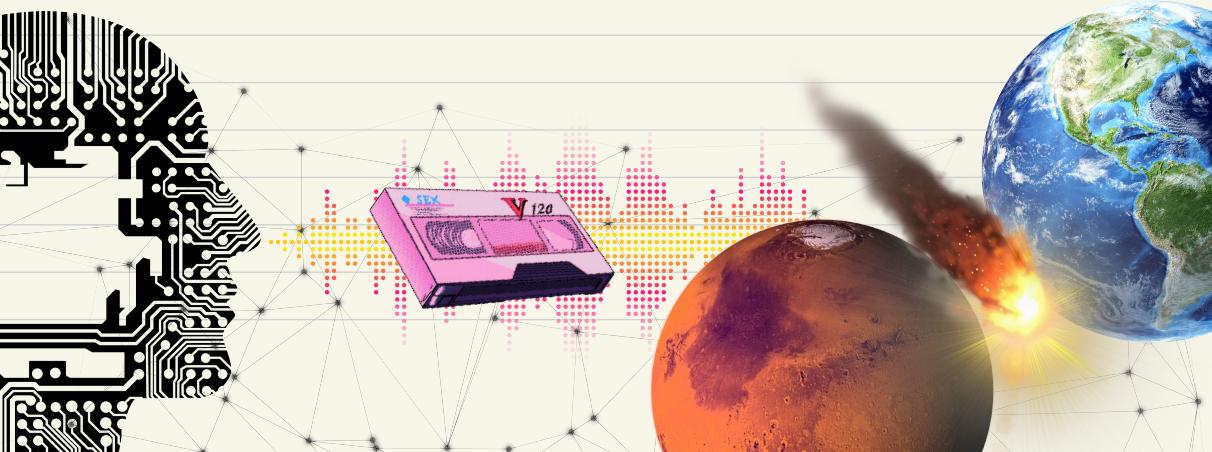
→ Nesse caso, vamos pensar que o número decimal que representa o caractere % terá o valor de 37 na tabela ASCII, mas para migra-lo para binário, devemos dividir o mesmo numeral decimal (37) por dois até que não seja mais possível dividir.



Evidentemente chegamos a diversos resultados dividindo o de cima! Agora, para representá-lo, mas vamos pensar de baixo para cima!

$$\% = 67 = 100\ 101$$

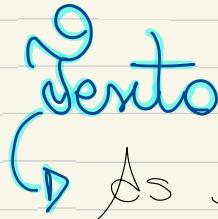
Nessa representação podemos entender que: o símbolo de porcentagem é descrito como 67 em decimal, mas também pode ser descrito como o numeral "100 101" em binário.



The World in Binary

Entendemos que os números na computação são a base de toda a informação digital. Qualquer dado — seja texto, imagem, som, vídeos, símbolos ou programas — é representado por sequências numéricas organizadas de forma lógica e estruturada. Essa estrutura numérica permite que os computadores interpretem, processem, armazenem e transmitam informações por meio de algoritmos.

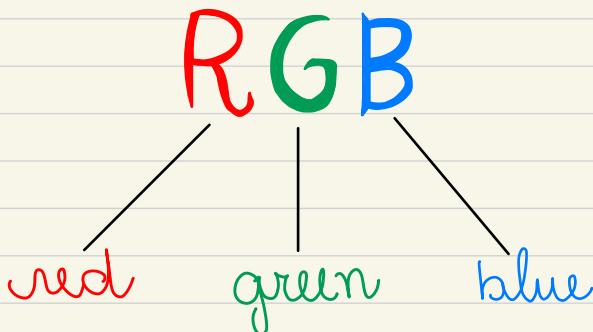
Os algoritmos são conjuntos de instruções matemáticas e lógicas que interpretam e manipulam números, transformando-os em informações compreensivas para humanos e máquinas. Cada tipo de dado digital possui um método específico para ser representado e processado

 **Título**
As letras do alfabeto e os sinais de pontuação, são associados aos números através do ASCII (7 bits) ou Unicode (8, 16, 32 bits)

Imagens

As imagens digitais são compostas por pixels, e cada pixel é representado por um número associado a uma cor.

★ Quando estudamos computação, principalmente áreas de design e/ou imagens, vamos precisar ter uma noção básica de colorimetria, ou seja, saber como as cores se formam.



As telas de celulares, monitores, TVs e projetores funcionam com o modelo de RGB, pois cada pixel da tela emite luzes nessas três cores e uniam sua identidade para poder formar todas as outras.

* O RGB é uma mistura aditiva, e ocorre quando feixes de luz de diferentes cores são combinados

$$\text{Red circle} + \text{Green circle} = \text{Yellow circle}$$

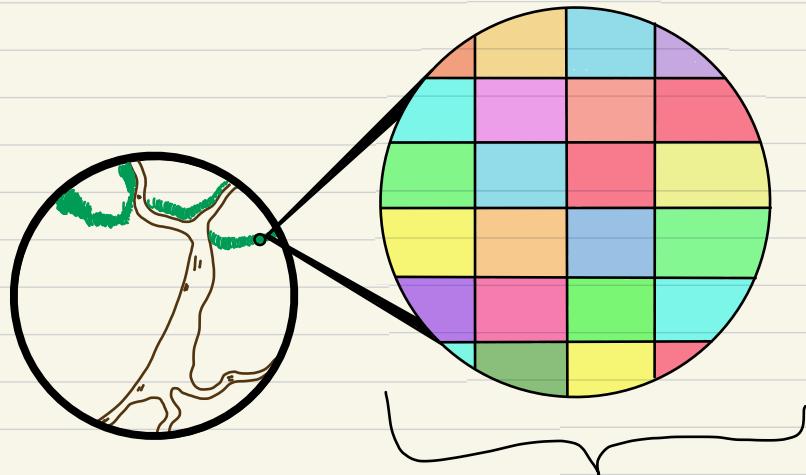
$$\text{Red circle} + \text{Blue circle} = \text{Purple circle}$$

$$\text{Green circle} + \text{Blue circle} = \text{Cyan circle}$$

$$\text{Red circle} + \text{Green circle} + \text{Blue circle} = \text{Grey circle}$$

$$\text{no light} = \text{Black circle}$$

Naturalmente em uma imagem, vamos ter todas as combinações de cores para formar algo visível na tela. Se aproximarmos uma imagem, podemos ver que ela é formada por pequenos quadrados chamados pixels. Cada pixel terá 24 bits preenchendo seu espaço



1 pixel = 24 bytes

Áudio (Música)

O som analógico é transformado em números por um processo que se chama amostragem, onde a onda sonora contínua é capturada em pequenos intervalos de tempo. Cada uma dessas amostras é armazenada como um número binário, como 16 ou 24 bits, determinando a precisão do som digital. Quanto mais amostra por segundo (Kilohertz-amostragem) e quanto mais bits por amostra, maior será a fidelidade do áudio.

8kHz - ruim; 44.1kHz - bom; 96kHz ótimo

Vídeos

→ O vídeo reproduzido na televisão celular ou cinema, nada mais é do que uma fusão entre imagens e som, isto é: o vídeo é formado a partir de uma sequência de imagens (**frames**) exibidas rapidamente juntas com um áudio sincronizado.

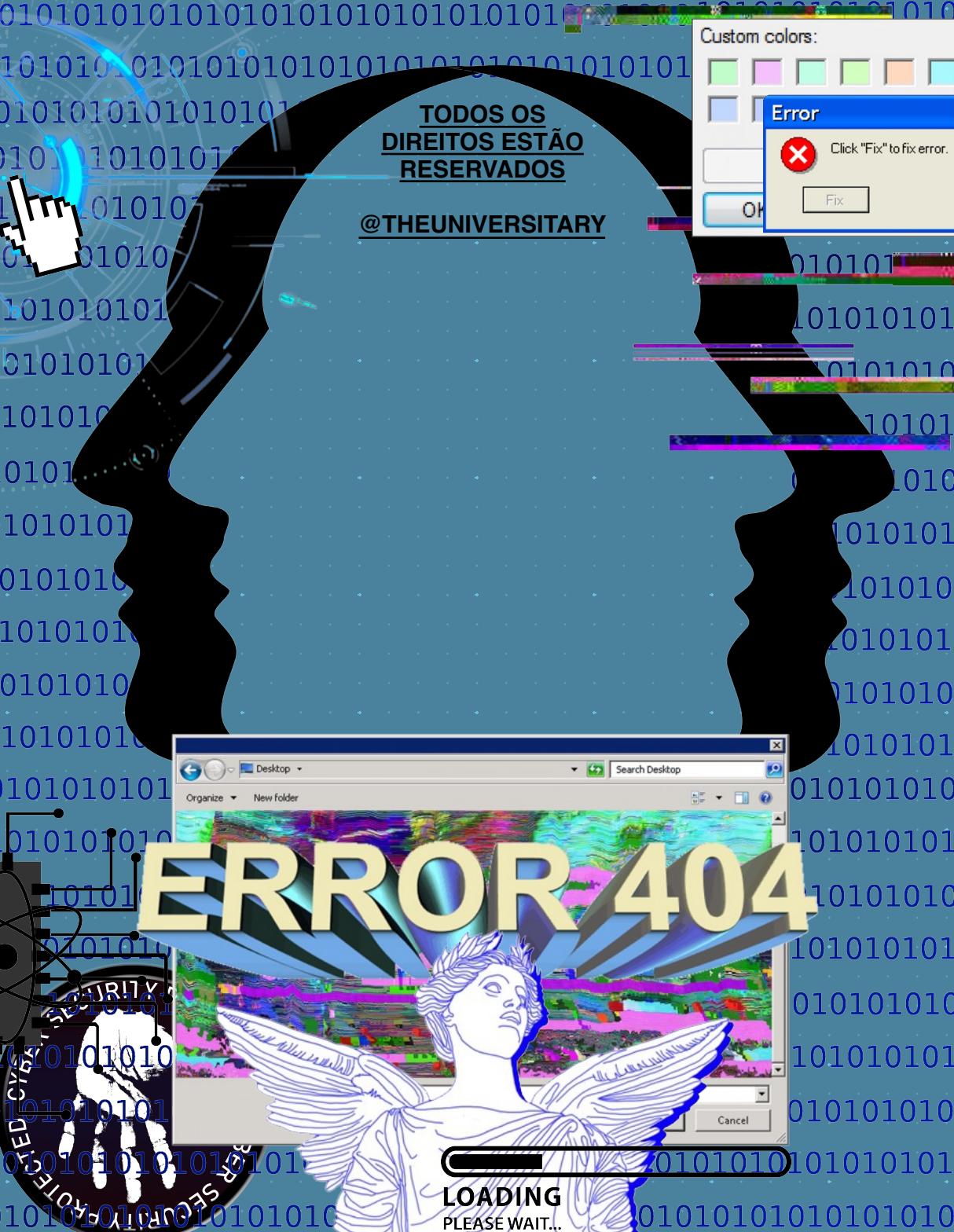
Cada frame é uma imagem digital composta por pixels e cores (RGB). O som é gravado separadamente e sincronizado com os frames para criar uma reprodução fluida.

Unicode

O Unicode é um padrão de codificação universal de caracteres criado para representar textos de qualquer idioma, símbolos especiais e até emojis no computador. Antes dele, diferentes sistemas usavam codificações próprias, como o ASCII que era limitado a 128 caracteres (7 bits) e só suportava o alfabeto inglês. Isso criava problemas de compatibilidade, pois um mesmo caractere podia ser exibido de

representado da mesma forma. Com o formato do Unicode, permitiu que **milhares de caracteres** fossem representados de forma padronizada, garantindo que textos, símbolos e emojis sejam exibidos corretamente em qualquer dispositivo ou software.

Com a popularização da internet e da comunicação global, o Unicode se tornou essencial para garantir padronização, compatibilidade e acessibilidade em texto, símbolos, fotos, vídeos, emojis, etc, tornando a própria computação verdadeiramente universal.



**TODOS OS
DIREITOS ESTÃO
RESERVADOS**

@THEUNIVERSITY

Error



Click "Fix" to fix error.

OK

Fix

ERROR 404

LOADING

PLEASE WAIT...