

07

Fundamentos de TI

Aula 07

Representação da Informação

Prof. Dr. Dilermando Piva Jr.

Site Disciplina:  <http://fundti.blogspot.com.br/>

Organização e representação de dados e Informações em computador.

Introdução

“As informações e o conhecimento compõem recursos estratégicos essenciais para o sucesso de uma organização.”

O Analista de Sistemas de Informações ou Analista de Negócios deve entender que a INFORMAÇÃO é um dos bens mais valiosos de uma organização.

☞ Entendendo os conceitos de DADOS e INFORMAÇÃO

✓ DADOS

São os fatos em sua forma primária e isolada, que uma vez organizados, rearranjados e processados de uma forma lógica e significativa tornam-se uma INFORMAÇÃO.

➔ Dados são classificados em TIPOS:

Numérico: de 0 a 9

Alfabético: de A até Z

Alfanumérico: de 0 a 9, de A até Z, e símbolos: &, @, *, %, \$, etc.

Numéricos podem ser:

Inteiro: 100 200 1000

Real (com decimais): 100,30 205,45 1985,65

Positivo: +100 +100,25

Negativo: -100 -100,25.

Dados possuem uma DESCRIÇÃO (Quantidade, Preço, Salário atual, etc.).

➔ **No Computador:**

✓ Os dados estão contidos em unidades de memória denominado BYTE.

O **BYTE** por sua vez é **dividido em 8 bits**, que recebem ou não energia elétrica provocando uma mudança de estado.

Dizemos que quando **BIT = 1** significa que está passando corrente elétrica pelo **circuito condutor** desse bit, e o **BIT = 0** quando não está passando corrente elétrica.

A combinação desses estados do bit, dentro do Byte irá representar o caractere armazenado nele.

✓ Após o processamento dos dados, os mesmos são então organizados em registros e gravados em meios de armazenamento, magnéticos e/ou óticos.

Representação de DADOS e INFORMAÇÃO

Desde tempos remotos o homem vem utilizando símbolos para registrar e transmitir informação, conforme mostra a cronologia abaixo.

Pré-história ➔ usava e símbolos para registrar a informação.

Meio: Paredes de rochas.

Antiguidade → usava e símbolos, sinais e mais tarde o alfabeto.

Meio: Papiro.

Idade Média → Usava o alfabeto, símbolos e signos.

Meio: Papel → Livro.

Modernidade → Usamos o alfabeto, símbolos, signos, figuras, etc., para representar dados/informações

Meio: Computador (meios magnéticos, óticos....).

Na sociedade atual, em sua vida cotidiana o homem lida, do ponto de vista numérico, com o sistema decimal e do ponto de vista alfabético, com determinada linguagem.

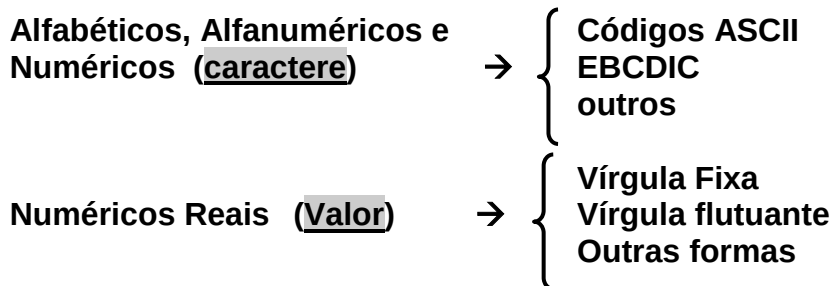
Da mesma forma, ***o computador, dado suas características físicas, ele lida sob ambos os aspectos com o sistema binário***, utilizando sistema de códigos para representação da informação, tanto numérica com alfabética e alfanumérica.

Tanto o sistema decimal quanto o binário têm por base os mesmos princípios, ou seja, a representação de um número se faz através de cadeias de símbolos.

Por motivos técnicos, os computadores são projetados, na maioria dos casos, para reconhecer sinais elétricos do tipo digital, sendo necessário que os métodos de codificação interna se baseiem no sistema binário, permitindo a representação de todo tipo de informação e instruções que um computador executa.

➔ Representação Interna

Os dados e informações que um sistema informático manipula internamente podem ser representados, conforme sua característica, das seguintes formas:



➔ Códigos Alfanuméricos e Numéricos

Nos códigos alfanuméricos cada caractere é representado através de um BYTE (8 bits), de modo que todo tipo de informação pode ser utilizado internamente, formando cadeias de bytes sucessivos que irão representar cadeia de caracteres (**dados e informações**).

Nem todos os padrões de códigos utilizam **todos os 8 bits** de um byte para representar um caractere.

Entretanto, devido à **necessidade de processamento de dados, sons e imagens**, atualmente os computadores usam a totalidade dos bits de um byte para representar dados e informações.

- **Primeiros códigos utilizados usavam 6 bits para representar um caractere.**

Permitiam a representação de $2^6 = 64$ caracteres.

Os bits: **543210** podiam representar:

26 letras maiúsculas – **10** algarismos – **28** caracteres especiais.

5	4	3	2	1	0
1	0	0	1	1	0

- **Códigos ASCII – American Standard Code for Information Interchange**

As necessidades de representação de minúsculas e de caracteres de controle de periféricos, exigidas pelo desenvolvimento das linguagens de programação de alto nível **motivaram o desenvolvimento dos códigos de 7 bits.**

O Código ASCII utiliza 7 bits para representação de caracteres.

Os bits: **6543210** permitem a representação de $2^7 = 128$ caracteres.

6	5	4	3	2	1	0
1	0	1	0	0	0	1

representa a letra **Q**

Bits 3210	654 000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	\	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

EOT	<i>End of transmition (fim de transmissão).</i>
ENQ	<i>Enquiry (solicitação de transmissão).</i>
ACK	<i>Acknowledge (reconhecimento de transmissão).</i>
BEL	<i>Bell (sinal sonoro, campainha ou alarme).</i>
BS	<i>Backspace (retrocesso).</i>
HT	<i>Horizontal tabulation (tabulação horizontal).</i>
LF	<i>Line feed (avanço de linha).</i>

Tabela de códigos ASCII – 7 bits

► **Códigos EBCDIC – Extended Binary-Coded Decimal Interchange Code**

O **Código EBCDIC utiliza 8 bits** para representação de caracteres, sendo muito utilizado em computadores IBM de grande porte.

Os bits: **76543210** permitem a representação de $2^8 = 256$ caracteres.

7	6	5	4	3	2	1	0
1	1	0	1	1	0	0	0

Exemplo de Tabela de códigos EBCDIC

Bits 3210	7654 0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	DLE	DS		SP	&	-						[	}	\	φ
0001	SOH	DC1	SOS						a	j	-		A	J		1
0010	STX	DC2	FS	SYN					b	k	s		B	K	S	2
0011	ETX	DC3							c	l	t		C	L	T	3
0100	PF	RES	BYP	PN					d	m	u		D	M	U	4
0101	HT	NL	LF	RS					e	n	v		E	N	V	5
0110	LC	BS	EOB	UC					f	o	w		F	O	W	6
0111	DEL	IL	ESC	EOT					g	p	x		G	P	X	7
1000		CAN							h	q	y		H	Q	Y	8
1001	RLF	EM						\	i	r	z		I	R	Z	9
1010	SMM	CC	SM		α	!	!	:								
1011	VT				'	\$	'	#								
1100	FF	IFS		DC4	<	*	%	@								
1101	CR	IGS	ENQ	NAK	(	)	-	'								
1110	SO	IRS	ACK		+	:	>	=								
1111	SI	IUS	BEL	SUB		~	?	"								

Os significados dos caracteres de controle são:

NUL *Null*

SOH *Start of heading*

STX *Start of text*

ETX *End of text*

PF *Punch off*

HT *Horizontal tabulation*

Exercício:

Como o computador, internamente, representaria a palavra **INFORMATICA** , usando o sistema de código ASCII.

Tabela de códigos ASCII – 7 bits

<i>Bits</i> 3210	654 <i>000</i>	<i>001</i>	<i>010</i>	<i>011</i>	<i>100</i>	<i>101</i>	<i>110</i>	<i>111</i>
0000	NUL	DLE	SP	0	@	P	\	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	*	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Caractere	6	5	4	3	2	1	0
I							
N							
F							
O							
R							
M							
A							
T							
I							
C							
A							

Exercício:

Como o computador, internamente, representaria a palavra **INFORMATICA**, usando o sistema de código EBCDIC.

Exemplo de Tabela de códigos EBCDIC

Bits	7654															
3210	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	DLE	DS		SP	&	-						{	}	\	φ
0001	SOH	DC1	SOS						a	j	-		A	J		1
0010	STX	DC2	FS	SYN					b	k	s		B	K	S	2
0011	ETX	DC3							c	l	t		C	L	T	3
0100	PF	RES	BYP	PN					d	m	u		D	M	U	4
0101	HT	NL	LF	RS					e	n	v		E	N	V	5
0110	LC	BS	EOB	UC					f	o	w		F	O	W	6
0111	DEL	IL	ESC	EOT					g	p	x		G	P	X	7
1000		CAN							h	q	y		H	Q	Y	8
1001	RLP	EM						\	i	r	z		I	R	Z	9
1010	SMM	CC	SM		α	!	!	:								
1011	VT				'	\$	'	#								
1100	FF	IFS		DC4	<	*	%	@								
1101	CR	IGS	ENQ	NAK	(	)	-	'								
1110	SO	IRS	ACK		+	:	>	=								
1111	SI	IUS	BEL	SUB		~	?	"								

Caractere	7	6	5	4	3	2	1	0
I								
N								
F								
O								
R								
M								
A								
T								
I								
C								
A								

➔ Representação de Números (caracteres)

Exercício:

Como o computador, internamente, representaria o número **28032006**, usando o sistema de código ASCII.

Tabela de códigos ASCII – 7 bits

<i>Bits</i> 3210	<i>654</i> 000	<i>001</i>	<i>010</i>	<i>011</i>	<i>100</i>	<i>101</i>	<i>110</i>	<i>111</i>
0000	NUL	DLE	SP	0	@	P	\	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Caractere	6	5	4	3	2	1	0
2							
8							
0							
3							
2							
0							
0							
6							

➔ **Representação de Números (valores numéricos)**

O número de bits que representa um valor numérico (inteiro ou real) é chamado palavra.

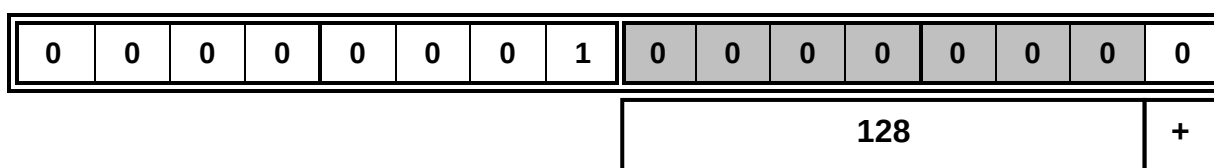
O tamanho da palavra depende do computador utilizado e corresponde à quantidade de bits que este é capaz de transferir numa determinada operação.

▶ **Valor numérico inteiro**

A representação de um valor numérico inteiro depende da linguagem de programação utilizada para definir a variável que irá conter esse valor.

Exemplo: valor inteiro +128 (variável definida em linguagem Pascal).

Tipo de dado *Integer*: Utiliza 2 bytes



O bit mais a direita contém o sinal, sendo seu valor = 0 (bit desligado) quando o número é positivo, ou 1 se negativo.

▶ **Valor numérico Real (com decimais)**

O computador utiliza o sistema binário como método de representação interna de dados, o número de bits que representa um número real (com decimais) é limitado, de modo que números reais sofrem truncamento na sua parte fracionária.

Dessa forma temos os recursos do ponto decimal (vírgula) fixo e flutuante.

✓ **Vírgula Fixa**

O ponto decimal está situado em uma posição fixa dentro da *palavra*. A vírgula fixa é utilizada exclusivamente para a representação de números inteiros, supondo-se a vírgula decimal implicitamente situada à direita dos dígitos.

Existem três formas de representar números com vírgulas fixas:

- ▶ **Binário puro;**
- ▶ **Decimal não compactado;**
- ▶ **Decimal compactado.**

✓ **Vírgula Flutuante**

A Vírgula Flutuante surge da necessidade de representar números reais com uma faixa maior àquela propiciada pela vírgula fixa e com o objetivo de permitir ao computador o tratamento de números extremamente grandes e extremamente pequenos.

Existem muitas formas de representação em Vírgula Flutuante, sendo as mais comuns:

- ▶ **Com precisão simples – (Short-precision floating point)**
- ▶ **Com precisão dupla – (Extended-precision floating point).**



Exemplos de Tabela (EBCDIC) de conversão para Binário, utilizada por Analistas e Programadores de MAINFRAMES na década de 70 e 80.

CODE TRANSLATION TABLE

9

Dec.	Hex	Instruction (IRI)	Graphics and Controls BCDIC EBCDIC(1) ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
0	00		NUL	NUL	12-0-1-8-9	0000 0000
1	01		SOH	SOH	12-1-9	0000 0001
2	02		STX	STX	12-2-9	0000 0010
3	03		ETX	ETX	12-3-9	0000 0011
4	04	SPM	PF	EOT	12-4-9	0000 0100
5	05	BALR	HT	ENQ	12-5-9	0000 0101
6	06	BCTR	LC	ACK	12-6-9	0000 0110
7	07	BCR	DEL	BEL	12-7-9	0000 0111
8	08	SSK		BS	12-8-9	0000 1000
9	09	ISK		HT	12-1-8-9	0000 1001
10	0A	SVC	SMM	LF	12-2-8-9	0000 1010
11	0B		VT	VT	12-3-8-9	0000 1011
12	0C		FF	FF	12-4-8-9	0000 1100
13	0D		CR	CR	12-5-8-9	0000 1101
14	0E	MVCL	SO	SO	12-6-8-9	0000 1110
15	0F	CLCL	SI	SI	12-7-8-9	0000 1111
16	10	LPR	DLE	DLE	12-11-1-8-9	0001 0000
17	11	LNR	DC1	DC1	11-1-9	0001 0001
18	12	LTR	DC2	DC2	11-2-9	0001 0010
19	13	LCR	TM	DC3	11-3-9	0001 0011
20	14	NR	RES	DC4	11-4-9	0001 0100
21	15	CLR	NL	NAK	11-5-9	0001 0101
22	16	OR	BS	SYN	11-6-9	0001 0110
23	17	XR	IL	ETB	11-7-9	0001 0111
24	18	LR	CAN	CAN	11-8-9	0001 1000
25	19	CR	EM	EM	11-1-8-9	0001 1001
26	1A	AR	CC	SUB	11-2-8-9	0001 1010
27	1B	SR	CU1	ESC	11-3-8-9	0001 1011
28	1C	MR	IFS	FS	11-4-8-9	0001 1100
29	1D	DR	IGS	GS	11-5-8-9	0001 1101
30	1E	ALR	IRS	RS	11-6-8-9	0001 1110
31	1F	SLR	IUS	US	11-7-8-9	0001 1111
32	20	LPDR	DS	SP	11-0-1-8-9	0010 0000
33	21	LNDR	SOS	!	0-1-9	0010 0001
34	22	LTDR	FS	"	0-2-9	0010 0010
35	23	LCDR		#	0-3-9	0010 0011
36	24	HDR	BYP	\$	0-4-9	0010 0100
37	25	LRDR	LF	%	0-5-9	0010 0101
38	26	MXR	ETB	&	0-6-9	0010 0110
39	27	MXDR	ESC	'	0-7-9	0010 0111
40	28	LDR		(	0-8-9	0010 1000
41	29	CDR		)	0-1-8-9	0010 1001
42	2A	ADR	SM	*	0-2-8-9	0010 1010
43	2B	SDR	CU2	+	0-3-8-9	0010 1011
44	2C	MDR		,	0-4-8-9	0010 1100
45	2D	DDR	ENQ	-	0-5-8-9	0010 1101
46	2E	AWR	ACK	.	0-6-8-9	0010 1110
47	2F	SWR	BEL	/	0-7-8-9	0010 1111
48	30	LPER		0	12-11-0-1-8-9	0011 0000
49	31	LNDR		1	1-9	0011 0001
50	32	LTER	SYN	2	2-9	0011 0010
51	33	LCER		3	3-9	0011 0011
52	34	HER	PN	4	4-9	0011 0100
53	35	LRER	RS	5	5-9	0011 0101
54	36	AXR	UC	6	6-9	0011 0110
55	37	SXR	EOT	7	7-9	0011 0111
56	38	LER		8	8-9	0011 1000
57	39	CER		9	1-8-9	0011 1001
58	3A	AER		:	2-8-9	0011 1010
59	3B	SER	CU3	;	3-8-9	0011 1011
60	3C	MER	DC4	<	4-8-9	0011 1100
61	3D	DER	NAK	=	5-8-9	0011 1101
62	3E	AUR		>	6-8-9	0011 1110
63	3F	SUR	SUB	?	7-8-9	0011 1111

CODE TRANSLATION TABLE (Contd)

(10)

Dec.	Hex	Instruction (RX)	Graphics and Controls BCDIC EBCDIC(1) ASCII			7-Track Tape BCDIC(2)	Card Code	Binary
64	40	STH	Sp	Sp	@	(3)	no punches	0100 0000
65	41	LA			A		12-0-1-9	0100 0001
66	42	STC			B		12-0-2-9	0100 0010
67	43	IC			C		12-0-3-9	0100 0011
68	44	EX			D		12-0-4-9	0100 0100
69	45	BAL			E		12-0-5-9	0100 0101
70	46	BCT			F		12-0-6-9	0100 0110
71	47	BC			G		12-0-7-9	0100 0111
72	48	LH			H		12-0-8-9	0100 1000
73	49	CH			I		12-1-8	0100 1001
74	4A	AH	¢	¢	J		12-2-8	0100 1010
75	4B	SH	.	.	K	B A 8 2 1	12-3-8	0100 1011
76	4C	MH	□	<	L	B A 8 4	12-4-8	0100 1100
77	4D		[	(	M	B A 8 4 1	12-5-8	0100 1101
78	4E	CVD	<	+	N	B A 8 4 2	12-6-8	0100 1110
79	4F	CVB	*	!	O	B A 8 4 2 1	12-7-8	0100 1111
80	50	ST	& +	& &	P	B A	12	0101 0000
81	51				Q		12-11-1-9	0101 0001
82	52				R		12-11-2-9	0101 0010
83	53				S		12-11-3-9	0101 0011
84	54	N			T		12-11-4-9	0101 0100
85	55	CL			U		12-11-5-9	0101 0101
86	56	O			V		12-11-6-9	0101 0110
87	57	X			W		12-11-7-9	0101 0111
88	58	L			X		12-11-8-9	0101 1000
89	59	C			Y		11-1-8	0101 1001
90	5A	A	!	!	Z		11-2-8	0101 1010
91	5B	S	\$	\$	[	B 8 2 1	11-3-8	0101 1011
92	5C	M	*	*	\	B 8 4	11-4-8	0101 1100
93	5D	D	]	)	]	B 8 4 1	11-5-8	0101 1101
94	5E	AL	:	:	^	B 8 4 2	11-6-8	0101 1110
95	5F	SL	Δ	—	—	B 8 4 2 1	11-7-8	0101 1111
96	60	STD	-	-	^	B	11	0110 0000
97	61		/	/	a	A 1	0-1	0110 0001
98	62				b		11-0-2-9	0110 0010
99	63				c		11-0-3-9	0110 0011
100	64				d		11-0-4-9	0110 0100
101	65				e		11-0-5-9	0110 0101
102	66				f		11-0-6-9	0110 0110
103	67	MXD			g		11-0-7-9	0110 0111
104	68	LD			h		11-0-8-9	0110 1000
105	69	CD			i		0-1-8	0110 1001
106	6A	AD			j		12-11	0110 1010
107	6B	SD			k	A 8 2 1	0-3-8	0110 1011
108	6C	MD	%	%	l	A 8 4	0-4-8	0110 1100
109	6D	DD	Y	—	m	A 8 4 1	0-5-8	0110 1101
110	6E	AW	\	>	n	A 8 4 2	0-6-8	0110 1110
111	6F	SW	≠	?	o	A 8 4 2 1	0-7-8	0110 1111
112	70	STE			p		12-11-0	0111 0000
113	71				q		12-11-0-1-9	0111 0001
114	72				r		12-11-0-2-9	0111 0010
115	73				s		12-11-0-3-9	0111 0011
116	74				t		12-11-0-4-9	0111 0100
117	75				u		12-11-0-5-9	0111 0101
118	76				v		12-11-0-6-9	0111 0110
119	77				w		12-11-0-7-9	0111 0111
120	78	LE			x		12-11-0-8-9	0111 1000
121	79	CE			y		1-8	0111 1001
122	7A	AE	¢	:	z	A	2-8	0111 1010
123	7B	SE	#	#	{	8 2 1	3-8	0111 1011
124	7C	ME	@	@		8 4	4-8	0111 1100
125	7D	DE	:	:	}	8 4 1	5-8	0111 1101
126	7E	AU	>	*	~	8 4 2	6-8	0111 1110
127	7F	SU	✓	"	DEL	8 4 2 1	7-8	0111 1111

CODE TRANSLATION TABLE (Contd)

11

Dec.	Hex	Instruction and Format	Graphics and Controls BCDIC EBCDIC(1) ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
128	80	SSM -S			12-0-1-8	1000 0000
129	81		a a		12-0-1	1000 0001
130	82	LPSW -S	b b		12-0-2	1000 0010
131	83	Diagnose	c c		12-0-3	1000 0011
132	84	WRD } SI	d d		12-0-4	1000 0100
133	85	RDD	e e		12-0-5	1000 0101
134	86	BXH	f f		12-0-6	1000 0110
135	87	BXLE	g g		12-0-7	1000 0111
136	88	SRL	h h		12-0-8	1000 1000
137	89	SLL	i i		12-0-9	1000 1001
138	8A	SRA			12-0-2-8	1000 1010
139	8B	SLA -RS	i		12-0-3-8	1000 1011
140	8C	SRDL	≤		12-0-4-8	1000 1100
141	8D	SLOL	(		12-0-5-8	1000 1101
142	8E	SRDA	+		12-0-6-8	1000 1110
143	8F	SLDA	+		12-0-7-8	1000 1111
144	90	STM			12-11-1-8	1001 0000
145	91	TM } SI	j j		12-11-1	1001 0001
146	92	MVI	k k		12-11-2	1001 0010
147	93	TS -S	l l		12-11-3	1001 0011
148	94	NI	m m		12-11-4	1001 0100
149	95	CLI } SI	n n		12-11-5	1001 0101
150	96	OI	o o		12-11-6	1001 0110
151	97	XI	p p		12-11-7	1001 0111
152	98	LM -RS	q q		12-11-8	1001 1000
153	99		r r		12-11-9	1001 1001
154	9A				12-11-2-8	1001 1010
155	9B				12-11-3-8	1001 1011
156	9C	SIO, SIOF	□		12-11-4-8	1001 1100
157	9D	TIO, CLRIO	]		12-11-5-8	1001 1101
158	9E	HIO, HDV	±		12-11-6-8	1001 1110
159	9F	TCH	■		12-11-7-8	1001 1111
160	A0		-		11-0-1-8	1010 0000
161	A1		°		11-0-1	1010 0001
162	A2		s s		11-0-2	1010 0010
163	A3		t t		11-0-3	1010 0011
164	A4		u u		11-0-4	1010 0100
165	A5		v v		11-0-5	1010 0101
166	A6		w w		11-0-6	1010 0110
167	A7		x x		11-0-7	1010 0111
168	A8		y y		11-0-8	1010 1000
169	A9		z z		11-0-9	1010 1001
170	AA				11-0-2-8	1010 1010
171	AB		⌞		11-0-3-8	1010 1011
172	AC	STNSM } SI	⌞		11-0-4-8	1010 1100
173	AD	STOSM	[		11-0-5-8	1010 1101
174	AE	SIGP -RS	≥		11-0-6-8	1010 1110
175	AF	MC -SI	●		11-0-7-8	1010 1111
176	B0		0		12-11-0-1-8	1011 0000
177	B1	LRA -RX	1		12-11-0-1	1011 0001
178	B2	See below	2		12-11-0-2	1011 0010
179	B3		3		12-11-0-3	1011 0011
180	B4		4		12-11-0-4	1011 0100
181	B5		5		12-11-0-5	1011 0101
182	B6	STCTL } RS	6		12-11-0-6	1011 0110
183	B7	LCTL	7		12-11-0-7	1011 0111
184	B8		8		12-11-0-8	1011 1000
185	B9		9		12-11-0-9	1011 1001
186	BA	CS } RS			12-11-0-2-8	1011 1010
187	BB	CDS	⌞		12-11-0-3-8	1011 1011
188	BC		⌞		12-11-0-4-8	1011 1100
189	BD	CLM } RS	]		12-11-0-5-8	1011 1101
190	BE	STCM	+		12-11-0-6-8	1011 1110
191	BF	ICM	-		12-11-0-7-8	1011 1111

CODE TRANSLATION TABLE (Contd)

12

Dec.	Hex	Instruction (SS)	Graphics and Controls BCDIC EBCDIC(1) ASCII			7-Track Tape BCDIC(2)	Card Code	Binary
192	C0		?	{		B A 8 2	12-0	1100 0000
193	C1		A	A	A	B A 1	12-1	1100 0001
194	C2		B	B	B	B A 2	12-2	1100 0010
195	C3		C	C	C	B A 2 1	12-3	1100 0011
196	C4		D	D	D	B A 4	12-4	1100 0100
197	C5		E	E	E	B A 4 1	12-5	1100 0101
198	C6		F	F	F	B A 4 2	12-6	1100 0110
199	C7		G	G	G	B A 4 2 1	12-7	1100 0111
200	C8		H	H	H	B A 8	12-8	1100 1000
201	C9		I	I	I	B A 8 1	12-9	1100 1001
202	CA						12-0-2-8-9	1100 1010
203	CB						12-0-3-8-9	1100 1011
204	CC			J			12-0-4-8-9	1100 1100
205	CD						12-0-5-8-9	1100 1101
206	CE			Y			12-0-6-8-9	1100 1110
207	CF						12-0-7-8-9	1100 1111
208	D0		!	}		B 8 2	11-0	1101 0000
209	D1	MVN	J	J	J	B 1	11-1	1101 0001
210	D2	MVC	K	K	K	B 2	11-2	1101 0010
211	D3	MVZ	L	L	L	B 2 1	11-3	1101 0011
212	D4	NC	M	M	M	B 4	11-4	1101 0100
213	D5	CLC	N	N	N	B 4 1	11-5	1101 0101
214	D6	OC	O	O	O	B 4 2	11-6	1101 0110
215	D7	XC	P	P	P	B 4 2 1	11-7	1101 0111
216	D8		Q	Q	Q	B 8	11-8	1101 1000
217	D9		R	R	R	B 8 1	11-9	1101 1001
218	DA						12-11-2-8-9	1101 1010
219	DB						12-11-3-8-9	1101 1011
220	DC	TR					12-11-4-8-9	1101 1100
221	DD	TRT					12-11-5-8-9	1101 1101
222	DE	ED					12-11-6-8-9	1101 1110
223	DF	EDMK					12-11-7-8-9	1101 1111
224	E0		#	\		A 8 2	0-2-8	1110 0000
225	E1						11-0-1-9	1110 0001
226	E2		S	S	S	A 2	0-2	1110 0010
227	E3		T	T	T	A 2 1	0-3	1110 0011
228	E4		U	U	U	A 4	0-4	1110 0100
229	E5		V	V	V	A 4 1	0-5	1110 0101
230	E6		W	W	W	A 4 2	0-6	1110 0110
231	E7		X	X	X	A 4 2 1	0-7	1110 0111
232	E8		Y	Y	Y	A 8	0-8	1110 1000
233	E9		Z	Z	Z	A 8 1	0-9	1110 1001
234	EA						11-0-2-8-9	1110 1010
235	EB						11-0-3-8-9	1110 1011
236	EC			rd			11-0-4-8-9	1110 1100
237	ED						11-0-5-8-9	1110 1101
238	EE						11-0-6-8-9	1110 1110
239	EF						11-0-7-8-9	1110 1111
240	F0	SRP	0	0	0	8 2	0	1111 0000
241	F1	MVO	1	1	1	1	1	1111 0001
242	F2	PACK	2	2	2	2	2	1111 0010
243	F3	UNPK	3	3	3	2 1	3	1111 0011
244	F4		4	4	4	4	4	1111 0100
245	F5		5	5	5	4 1	5	1111 0101
246	F6		6	6	6	4 2	6	1111 0110
247	F7		7	7	7	4 2 1	7	1111 0111
248	F8	ZAP	8	8	8	8	8	1111 1000
249	F9	CP	9	9	9	8 1	9	1111 1001
250	FA	AP					12-11-0-2-8-9	1111 1010
251	FB	SP					12-11-0-3-8-9	1111 1011
252	FC	MP					12-11-0-4-8-9	1111 1100
253	FD	DP					12-11-0-5-8-9	1111 1101
254	FE						12-11-0-6-8-9	1111 1110
255	FF						12-11-0-7-8-9	1111 1111

Exemplo de representação dos dados em computador.

Conforme visto no tópico anterior, a representação de informações em um sistema computadorizado ocorre segundo tabelas de códigos.

Dependendo da plataforma computacional (Micros, Minis e Mainframes), temos diferentes tabelas (ASCII, EBCDIC, BCDIC, etc.).

A representação e armazenamento de dados, enquanto caracteres (Numéricos, Alfabéticos e/ou Alfanuméricos) ocorrem segundo essas tabelas de códigos, enquanto o armazenamento e representação de valores numéricos (Inteiros, Reais, etc.) dependem da linguagem de programação utilizada, ou do gerenciador das bases dados contidas no sistema.

Para exemplificar construímos um programa em linguagem Pascal, para gravar diferentes tipos de dados em um arquivo magnético (HD). Após a execução do programa, usando um utilitário do SO. DOS (o Debug), tiramos um “foto” do registro gravado, para analisar como os dados foram gravados.

Representação de dados em um registro de arquivo magnético

- Programa fonte em linguagem Pascal, usado para gravar dados em um arquivo contido em HD.

```
Program Representa_Dados;
Uses Crt;
Type
  TRegistro = Record
    Caracter_AlfaNum : String[12];
    Caracter_Num      : String[4];
    Val_Num_inteiro   : Integer;
    Caracter_Num1     : String[3];
    Val_Num_Real      : Real;
    Caracter_Alfabet  : String[3];
  End;
Var
  Registro : TRegistro;
  ArqLogico : File of TRegistro;
Const
  ArqFisico = 'C:\INTRCOMP\Arquivo.DAT';
```

```
Begin;
  ClrScr;

  Assign(ArqLogico,ArqFisico);

  Rewrite(ArqLogico);

  Registro.Caracter_AlfaNum := 'AKMZ 012 &@$';

  Registro.Caracter_Num      := '5678';
  Registro.Val_Num_Inteiro   := 4096;
  Registro.Caracter_Num1     := '4096';
  Registro.Val_Num_Real      := 256.64;
  Registro.Caracter_Alfabet  := 'Flm';

  Write(ArqLogico,Registro);

  Close(ArqLogico);
End.
```

O programa acima está gravando os seguintes dados:

Caracteres Alfanuméricos = **AKMZ 012 &@\$** e **Flm**

Caracteres Numéricos = **5678** e **4096**

Valor numérico inteiro = **4096**

Valor numérico real = **256.64**

- ✓ Resultado da execução do utilitário: DEBUG (do DOS), que permite tirar uma foto do registro gravado pelo programa em PASCAL.

The screenshot shows a DOS DEBUG window with the following content:

```

C:\WINNT\system32\cmd.exe - debug arquivo.dat
C:\>cd intrcomp
C:\INTRCOMP>dir
O volume na unidade C é HD_SERVER
O número de série do volume é 3A4F-E20F

Pasta de C:\INTRCOMP
30/03/2006 15:17 <DIR> .
30/03/2006 15:17 <DIR> ..
30/03/2006 16:01          34 ARQUIVO.DAT
                1 arquivo(s)          2 bytes
                2 pasta(s) 68.719.640,57% bytes disponíveis

C:\INTRCOMP>debug arquivo.dat
-d
0D21:0100  0C 41 4B 4D 5A 20 30 31-32 20 26 40 24 04 05 36  .AKMZ 012 &Q$.56
0D21:0110  37 38 00 10 03 34 30 39-89 1F 85 EB 51 00 03 46  78...409....Q..F
0D21:0120  49 6D 69 6A 61 20 6E 6F-20 63 61 6D 69 6E 6F  Imida no caminho
0D21:0130  20 64 65 20 6C 6E 63 61-6C 69 7A 61 87 C6 6F 0D  de localiza..o.
0D21:0140  0A 16 44 67 73 70 6F 73-69 74 69 76 6F 20 69 6E  ..Dispositivo in
0D21:0150  76 A0 6C 67 64 6F 0D 0A-1B 46 4F 52 20 6E C6 6F  v.lido...FOR n.o
0D21:0160  20 70 6F 67 65 20 73 6F-72 20 61 6E 69 6E 68  pode ser aninha
0D21:0170  64 6F 0D 0A 6E 45 72 72 72 20 6E 6F 20 61 72 72  do...Erro no arq
  
```

Callouts explaining the hex dump structure:

- 12 Bytes contendo os dados alfanuméricos (points to the first 12 bytes of the hex dump)
- Início dos 4 Bytes contendo dados numéricos (points to the first 4 bytes of the hex dump)
- Final dos 4 Bytes contendo dados numéricos (points to the last 4 bytes of the hex dump)
- Byte informando o tamanho da string de (points to the first byte of the hex dump)
- 2 Bytes contendo um valor numérico (points to the next 2 bytes of the hex dump)
- Byte informando o tamanho da string de (points to the next byte of the hex dump)
- Byte - EOT informando o final da string de (points to the next byte of the hex dump)

Importante: Os dados gravados no registro em disco estão no formato binário, porém quando pedimos uma representação dos mesmos, estes são mostrados no formato Hexadecimal.

Se desejarmos saber **como os bits estão arranjados cada Byte**, temos que efetuar a **conversão de Base Numérica Hexadecimal para Base Numérica Binária**.

A conversão entre bases numéricas será assunto do próximo tópico.

- Definição de um registro de dados, em Programa fonte em linguagem COBOL (código ASCII).

01 Registro.

02 Character_AlfaNum	Pic X(12)	Value Spaces.
02 Character_Num	Pic X(4)	Value Spaces.
02 Val_Num_Inteiro	Pic S9(4)	Value Zeros.
02 Character_Num1	Pic X(4)	Value Spaces.
02 Val_Num_Real	Pic 9(7)V9(2)	Value Zeros.
02 Character_Alfabet	Pic A(3)	Value Spaces.

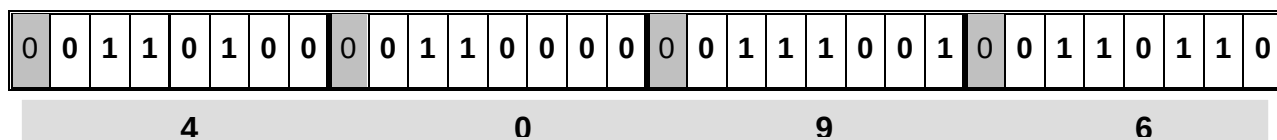
Usando a linguagem Cobol, tanto os caracteres numéricos quanto os valores numéricos são armazenados em tantos bytes quantos forem definidos pelo programador.

Por exemplo, após as instruções em Cobol:

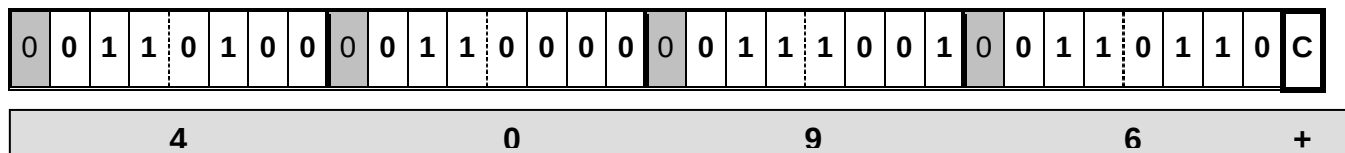
Move '4096' To Character_Num1.

Move 4096 To Val_Num_Inteiro.

Teremos 4 Bytes armazenando os caracteres numéricos 4096, com a seguinte configuração:



Teremos 4 bytes para armazenar o valor numérico 4096, com a mesma configuração acima, porém como se trata de valor numérico, é considerado também o sinal (+ ou -), que ocupará mais um byte à direita.



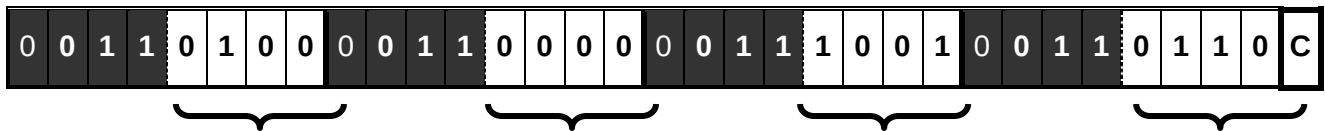
Fundamentos de TI

Ainda, em linguagem Cobol, se a definição da variável for efetuada da seguinte forma:

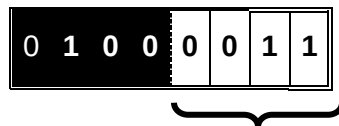
02 Val_Num_Inteiro Pic S9(4) Value Zeros COMP-3.

Isso significa que estamos querendo armazenar um valor numérico em **formato compactado**.

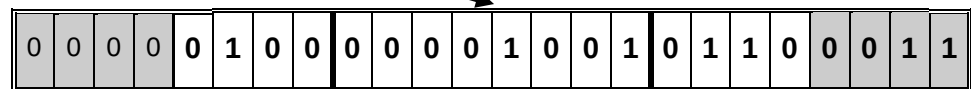
Note-se que todos os Bytes representando números possuem a **configuração dos 4 bits mais à esquerda do Byte exatamente iguais**.



Sinal: + (C) →



A compactação ocorre eliminando essas configurações iguais e juntando os 4 bits mais à direita de cada Byte, mantendo-se o sinal.



A descompactação segue o processo inverso ao da compactação.