

TEXTO, HIPERTEXTO E HIPERMÉDIA



TELEMÉDIA

2006/2007

Sumário

- Texto
 - Representação do conteúdo textual
 - Representação da aparência
 - Operações de processamento de texto
- Hipertexto e Hipermedia
 - Âncoras e ligações
 - Um modelo
 - Particularidades do hipermedia em rede

Texto de apoio:

Multimédia e Tecnologias Interactivas, *Nuno Ribeiro*, FCA, 2004

(Secção 4.1 do Capítulo 4)

Introdução

- Texto é o media mais usado na interacção homem-máquina:
 - Os primeiros monitores dos computadores não exibiam gráficos nem imagens;
 - Os teclados baseiam-se em texto;
- Hoje em dia os elementos gráficos são cada vez mais usados no software, porque:
 - permitem uma interpretação mais rápida
 - ocupam menos espaço no monitor
- Três tipos de texto:
 - Texto não formatado (plain text) – texto simples
 - Texto formatado (rich text) – inclui a formatação (fontes, etc)
 - Hipertexto (ex: html) – texto com ligações a outros textos

Texto

- Natureza dupla do texto:
 - **Conteúdo léxico** – caracteres que constituem as palavras, sinais de pontuação e outros símbolos;
 - **Aparência** – atributos visuais dos caracteres (fonte, tamanho, disposição no ecran (*layout*), etc.)
- Identidade fundamental dos caracteres:
 - **Caracter abstracto**;
 - **Representação gráfica**;
Por exemplo, o caracter abstracto “A” pode ter uma infinidade de representações gráficos, incluindo “*A*”, “**A**”, “*A*”, “*a*”, “*a*”, “**a**”

Texto – caracteres abstractos

- **Caracteres abstractos** são os caracteres representados apenas quanto à sua natureza léxica:
 - Caracteres abstractos são agrupados em **alfabetos**;
 - Cada **idioma** ou **grupo de idiomas** usa um alfabeto;
- **Repositórios de caracteres** (*character sets*) – tabelas mantidas pelo sistema operativo que consistem numa correspondência entre os códigos e os caracteres
 - Incluem maiúsculas, minúsculas, sinais de pontuação, números e símbolos matemáticos;

Texto – caracteres abstractos

- Vantagens da utilização de repositórios de caracteres:
 - É vital guardar os caracteres na forma de códigos:
 - para poder *editar* (alterar) e *pesquisa* de texto;
 - Para facilitar a *comparação* de caracteres (basta comparar códigos!)
 - Permitem associar os caracteres dos teclados à representação desses caracteres:
 - Por exemplo, quando se pressiona um A no teclado, esse caracter é procurado na tabela de caracteres para depois ser apresentado no monitor
 - **Normalização** é o mais importante, pois os códigos universais podem facilmente ser trocados entre máquinas distintas e que usam sistemas operativos distintos.

Texto – caracteres abstractos

- O primeiro conjunto de caracteres normalizado (1968), foi o **ASCII** (*American Standard Code for Information Interchange*):
 - Utiliza apenas **7 bit**... ou seja... **2⁷** combinações, portanto um total de 128 caracteres distintos...
 - Chega para o *Inglês*, mas não serve para mais nenhuma língua!...
 - Tentou-se resolver o problema com variantes nacionais – a norma ISO 646 – que não foi bem sucedida...
 - As múltiplas variantes contrariam a noção de norma e criam dificuldades na comunicação entre sistemas que usam diferentes variantes da mesma tabela!
- Solução mais óbvia: usar 8 bit em vez de 7 – **ASCII Estendido**

Tabela ASCII (7bit)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	:	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Tabela ASCII - Estendido

128	Ç	144	É	161	í	177	⣿	193	⊥	209	⌘	225	β	241	±
129	ù	145	æ	162	ó	178	⣻	194	⌞	210	⌠	226	Γ	242	≥
130	é	146	Æ	163	û	179		195	⌟	211	⌡	227	π	243	≤
131	â	147	ô	164	ñ	180	⌈	196	—	212	⌢	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	⌋	197	+	213	⌣	229	σ	245	∫
133	à	149	ò	166	ª	182	⌌	198	⌜	214	⌤	230	μ	246	÷
134	å	150	û	167	º	183	⌍	199	⌝	215	⌥	231	τ	247	≈
135	ç	151	ù	168	¿	184	⌎	200	⌞	216	⌦	232	Φ	248	°
136	ê	152	—	169	—	185	⌏	201	⌟	217	⌧	233	⊙	249	.
137	ë	153	Ö	170	¬	186	⌐	202	⌠	218	⌨	234	Ω	250	.
138	è	154	Û	171	½	187	⌑	203	⌡	219	■	235	δ	251	√
139	ï	156	£	172	¼	188	⌒	204	⌢	220	■	236	ω	252	_
140	î	157	¥	173	¡	189	⌓	205	=	221	■	237	φ	253	z
141	ì	158	—	174	«	190	⌔	206	⌣	222	■	238	ε	254	■
142	Ä	159	f	175	»	191	⌕	207	⌤	223	■	239	∩	255	
143	Å	160	á	176	⣿	192	⌖	208	⌥	224	α	240	≡		

Source: www.LookupTables.com

Texto – caracteres abstractos

- O ASCII Estendido não resolveu o problema, porque os 256 caracteres eram insuficientes para todos os idiomas...
- Norma **ISO 8859** - Durante a década de 80 procedeu-se à normalização dos conjuntos de 8 bits:
 - 7 bits (ASCII) + 1 Bit para a variante (ASCII Estendido).
 - Várias variantes regionais (10 variantes);
 - A primeira variante (**ISO 8859-1**), designada por **ISO Latin1**
- **Outras:**
 - Latin 2 ISO-8859-2 (Europa Oriental: Checo, Eslovaco e Croata)
 - ISO-8859-5 (Cirílico); ISO-8859-7 (Grego Moderno) e ISO-8859-8 (Hebreu)

Texto – caracteres abstractos

- A opção pelas variantes ISO-8859 acaba por não conseguir resolver bem o problema:
 - 7+1 bit são claramente insuficientes para representar todas as línguas (Chinês e Japonês)
 - E os textos multilingue? Como se trabalha com várias línguas em simultâneo?
- Solução: mais bits!
- Norma **ISO-10646** (32 bit) de 1991:
 - Permite representar 4.294.967.296 caracteres distintos (2^{32})
 - Desvantagem: qual a diferença entre representar um texto de 50 caracteres em ASCII-Estendido e ISO-10646 em termos de memória ocupada?

Texto – caracteres abstractos

- Em simultâneo com a norma ISO-10646, um consórcio (Apple, Adobe, Microsoft, HP, IBM, Oracle, SAP, SUN e Unysys) introduziu um conjunto de caracteres de 16 bits:
 - UNICODE
 - 16 bits permitem a representação de caracteres de vários idiomas, incluindo os orientais, no mesmo texto
 - Apesar de permitir representar 65.536 caracteres (2^{16}), o UNICODE apenas usa 39.000, deixando os restantes reservados para utilização futura;
 - É compatível com ISO-10646 porque é um subconjunto dessa norma.

Texto – caracteres abstractos

- Conclusão:

Os caracteres abstractos são representados usando repositórios de caracteres, recorrendo a várias normas que têm vindo a evoluir no sentido de ser permitida a escrita de textos multi-língua.



- Uma página WEB ou uma mensagem de correio electrónico podem ser criadas e posteriormente visualizadas em sistemas muito distintos! As representações dos caracteres ganham aí enorme importância!
- Visite a página da disciplina e experimente:
 - Observar no menu “Ver” os códigos escolhidos pelo seu browser;
 - Altere a escolha para Cirilico, Grego e Europa Oriental. Que se passou? Como?
 - Verificar no código fonte HTML (“Ver-> Ver Código Fonte”), logo nas primeiras duas linhas qual o alfabeto (character set) em que foi criada;
- Repita a experiência para outras páginas (Ex: www.google.com)

Texto – representação gráfica

- Apresentação do texto: a **forma dos caracteres**
 - A representação visual de um caracteres designa-se por **Glifo**
 - Um caracter pode ser representado por uma infinidade de glifos;
 - Os **glifos** agrupam-se em colecções que se designam por **Fontes**
 - Os glifos de uma fonte partilham um conjunto comum de características visuais (existe uma semelhança entre os glifos da mesma fonte);
 - São desenvolvidos para que combinem bem entre si, tanto no monitor como no papel;
 - As fontes ***digitais*** são versões das fontes tradicionais (algumas do séc XV)
 - As **fontes** podem ser vistas como **tabelas de correspondência** entre os *caracteres abstractos* e a sua *representação gráfica*

Texto – representação gráfica

- Poderá surgir a seguinte questão:
 - *Se*
 1. *Os caracteres abstractos são codificados usando códigos*
 2. *Depois convertidos em representações gráficas para serem apresentados*
 - Porque motivo não são logo codificados na sua representação gráfica digital?

Texto – representação gráfica

- Caracteres de fontes distintas seriam interpretados como caracteres distintos (não existiria a noção de carácter abstracto)
- Uma vez convertidos para forma gráfica, a operação inversa não é tão simples – implica operações de reconhecimento!
 - Não poderia ser facilmente editado, pesquisado, convertido, etc.
- A representação gráfica é ineficiente (consome muito mais memória que os códigos em abstracto);
 - Não permite tirar partido das repetições do mesmo carácter (compressão)

Texto – representação gráfica

- Onde existem as **Fontes** que fornecem os **Glifos** para representar os textos?
- Há duas hipóteses apenas:
 - Ou são guardadas em ficheiros próprios e **instaladas no sistema** operativo;
 - São partilhadas por todos os ficheiros e todas as aplicações!
 - Quando são requeridas e não existem tem de ser trocadas por fontes alternativas;
 - Ou são **embebidas** nos próprios ficheiros de texto;
 - Vantagem importante para o *designer* de uma aplicação multimédia pois é livre de usar qualquer fonte no seu trabalho;
 - Os ficheiros de texto ocupam muito mais espaço;
 - Não se partilham as fontes entre documentos que usam as mesmas fontes;

Texto – representação gráfica

- Exercício com as **fontes** da Universidade do Minho:



- Descarregue da página da disciplina os ficheiros **ExercicioFontes.pdf** e **ExercícioFontes.ppt**. Visualize-os e compare o resultado.
- O ficheiro utiliza a fonte usada na nova imagem da Universidade do Minho: o NewsGotT
 - Procure na pasta “Painel de Controlo -> Tipos de Letra” o ficheiro da fonte respectiva
 - Descarregue-a da página da disciplina e copie-a para essa pasta;
 - Volte a visualizar os dois ficheiros anteriores observando o resultado;
- Verifica nas opções (Ferramentas->Opções) de um dos programas do Office como embeber as fontes:
 - Crie um texto que use as fontes acima e grave-o com fontes embebidas e sem fontes embebidas; verifique as diferenças de tamanho dos ficheiros;

Texto – classificação das fontes

- **Mono-espaçadas** versus **proporcionais**

- Nas fontes **mono-espaçadas** cada caracter ocupa o mesmo espaço na horizontal, independentemente da sua forma (como nas máquinas de escrever tradicionais)

- Exemplo: `Courier`, `Courier NEW`

- Nas fontes **proporcionais** o espaço ocupado depende da largura da forma; As fontes proporcionais produzem texto mais legível que as mono-espaçadas;

- Exemplos: `Garamond`, `Helvética` e **Arial**

Texto – classificação das fontes

- **Fontes com serif** versus **Fontes sem serif**

- Os *Serif* são traços minúsculos que se acrescentam nas extremidades dos glifos; tiveram a sua origem nas marcas produzidas pelos cinzéis dos Romanos nas inscrições em pedra; Daí que também se designam por **fontes Romanas**

- Fontes sem serif:

- Exemplos: **Helvética**, **Arial** e **Lucida**

- As fontes com serif

- Exemplos: Garamond, Times New Roman, **Century**

Texto – classificação das fontes

- Fontes com **forma vertical** versus Fontes com **forma itálica**
 - Nas fontes com forma vertical as linhas dos glifos são perfeitamente verticais
 - Nas fontes de forma itálica
 - As linhas verticais são ligeiramente curvadas para a direita;
 - Algumas são apenas variantes das outras fontes usadas para evidenciar texto;
 - *Times New Roman em Itálico*
 - Outras tentam imitar a caligrafia
 - Exemplo: *Lucida Calligraphy, Lucida Handwriting*

Texto – classificação das fontes

- Fontes **pesadas** versus Fontes **leves**
 - Fontes **pesadas**
 - Traço dos glifos é espesso;
 - Podem ser variantes das outras fontes;
 - Não devem ser usadas para frases longas, apenas para destacar palavras;
 - Exemplo: **Arial Black**
 - Fontes leves
 - Traço dos glifos é fino;

Texto – classificação das fontes

- Fontes **para texto contínuo** versus Fontes **para texto isolado**
 - Designação mais subjectiva;
 - **Texto contínuo**: corpo de um livro ou de um artigo
 - Fáceis de ler;
 - Discretas, não intrusivas;
 - Exemplo: Arial, Times New Roman
 - Texto isolado:
 - Pretende chamar a atenção, fontes intrusivas;
 - Usado em slogans publicitários;
 - Uso de fontes decorativas ou ornamentais;
 - Exemplo: **Impact**, Comics Sans

Texto – classificação das fontes

- Ao conjunto de variações de uma fonte (normal, itálico, bold) dá-se o nome de **família de fontes**;
- Medição das dimensões da fonte:
 - Pontos (pt) – **1 pt** aproximadamente = 0,3528mm
 - Exemplo 20pt: Dimensão do corpo do caracter, altura entre o topo do caracter mais alto e o fundo do caracter mais baixo;
 - Picas (pc) – **1 pc** aproximadamente = 4,2333mm
 - Utiliza-se para medir espaçamento entre linhas;

Texto – classificação das fontes

- Quanto á forma como são armazenadas em ficheiro:
 - Fontes **Outline** versus Fontes **Bitmap**
 - As fontes outline são guardadas sob a forma de gráficos vectoriais;
 - As fontes bitmap são armazenadas como mapas de bits;
 - As fontes outline são mais utilizadas que as fontes bitmap, porque possuem a mesma forma e aparência qualquer que seja a plataforma;
 - O formato mais comum de fonte outline é o **Adobe Type 1**, sendo os glifos gerados por pequenos programas escritos em PostScript que descrevem curvas Bézier cúbicas; As **TrueType** são descritas por curvas quadráticas;

Texto

Devem evitar-se passagens longas de texto nas aplicações multimédia porque como são visualizadas no monitor tendem a tornar-se cansativas

Texto

- Disposição do conteúdo textual (**layout**)
 - forma como os caracteres se combinam em palavras, frases, linhas, parágrafos e outras unidades de divisão do texto tais como, Secções e Capítulos, isto é, a forma como o texto se dispõe no ecrã ou no papel
 - os documentos formatados possuem uma estrutura interna constituída por elementos estruturais:
 - Cabeçalhos, Títulos, Capítulos, Parágrafos, Listas, Notas de rodapé, Referências, Índices
- Duas formas para definir o **layout**:
 - Formatos de descrição de estrutura:
 - Formatos de descrição de páginas;

Texto

- Formatos de descrição de estrutura:
 - Contêm **marcas** que correspondem à informação de controlo que é adicionada ao corpo do texto (*body text*)
 - Linguagens de **Markup** (baseadas em etiquetas)
 - O SGML (*Standard Generalized Markup*) é um formato de descrição de estrutura usada descrever vários tipos de documentos, por exemplo o HTML.
 - RTF e DOC usados pelo Microsoft Word
 - Latex
 - Simples editores de texto podem criar documentos (Ex: Latex e HTML):
 - **WYWIWYG**: What You Want Is What You Get
 - Necessário browser ou equivalente para processar e visualizar!
 - Resultado depende do dispositivo de saída: mas é essa a ideia

Texto

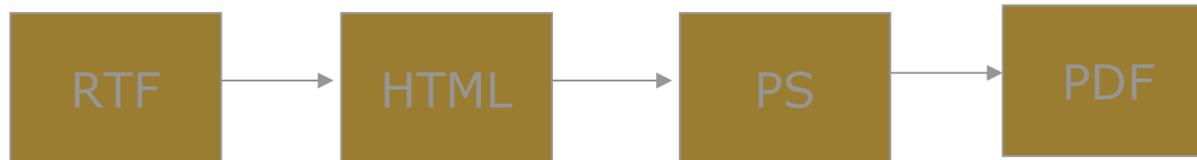
- Processadores de Texto (MS-Word, WordPerfect) ou aplicações desktop publishing (Adobe In Design, Quark Express):
 - **WYSIWYG**: What You See Is What You Get
 - Todas as operações de formatação são geradas por comandos e visualizadas de imediato!
 - Na verdade, internamente, são geradas e guardadas marcas de formatação!
 - Normalmente só se consegue ver o resultado da formatação com a própria aplicação que a permite gerar;
 - Podem-se exportar para formatos com marcas (ex: HTML)

Texto

- Formatos de descrição de páginas:
 - São formatos baseados em comandos que descrevem o conteúdo de páginas que podem ser interpretados pelas impressoras ou por uma aplicação que os apresenta no ecrã
 - Exemplos: PS (PostScript) e PDF (Portable Document Format), ambos da Adobe
 - Não é necessário ter software específico para poder imprimir estes formatos (grande parte das impressoras lê os comandos contidos no ficheiro), embora para poder visualizá-los possa ser necessário um *reader* que não é a mesma aplicação que os produziu

Texto

- Exercício Conversão de formatos:



- De **Rich Text Format** para **HTML**, depois **PostScript** e finalmente **PDF**



- **Nota:** Exercício completo no site
- Produzir um texto RTF (texto.rtf) com o Word; Gravar também em HTML (texto.html) e ver no browser o resultado;
- Instalar uma impressora qualquer PostScript a cores, associada a um ficheiro (ex: HPColorLaser 8500PS que não temos de facto!)
- Imprimir para ficheiro (texto.ps)
- Instalar software "GSView" para ver e converter PostScript
- Ver o PS e converter para PDF
- Identifique os formatos de descrição de página e de estrutura

Texto

- Operações sobre o texto:
 - Operações sobre caracteres
 - Operações sobre *strings* (sequências de caracteres)
 - Edição de texto;
 - Formatação de texto;
 - Compressão de texto;
 - Cifragem de texto;
 - Verificação ortográfica e gramatical do texto;

Texto

- Conclusão

- O aspecto visual é dado por fontes embebidas no próprio texto ou pré instaladas no sistemas operativo;
- Existem vários tipos de fontes e podem ser classificadas de várias formas;
- O texto pode ser disposto de várias formas e existem formatos de descrição de estrutura e formatos de descrição de página;
- Sob o texto podem-se fazer várias operações

Hipertexto e Hipermedia

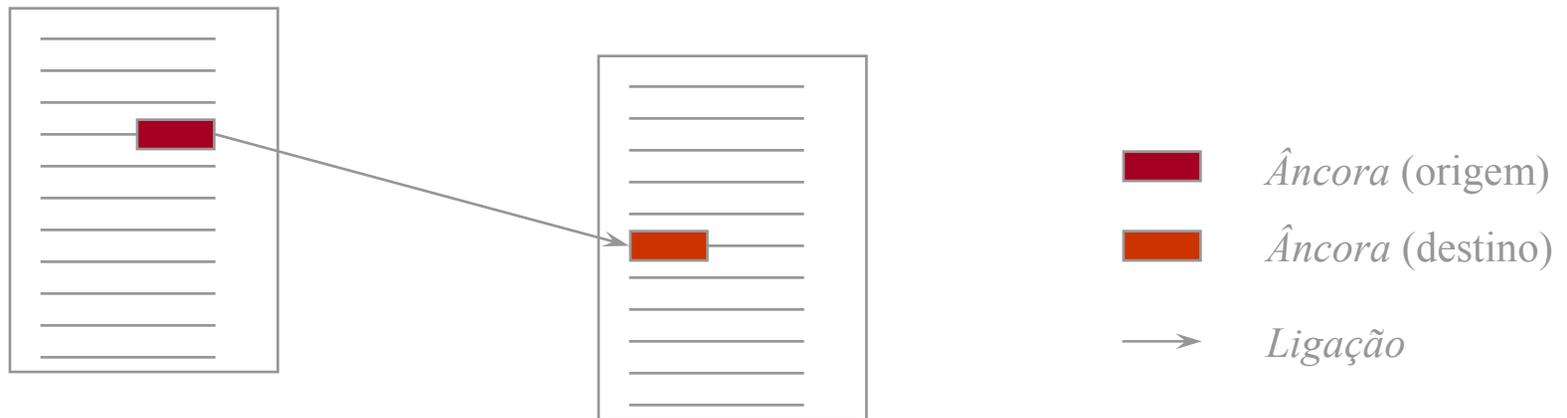
- Um conceito que engloba:
 - *a apresentação* (forma como o documento é exibido)
 - *o acesso* (forma de aceder e obter os docs.)
 - *a estrutura* (forma com está estruturado)
 - *o armazenamento* (diferentes formas de armazenamento)

Hipertexto é texto com ligações a outros textos. Os documentos hipertexto não são estritamente sequenciais, pois podem conter *referências a outras partes do documento* ou mesmo a *outros documentos*.

Hipermedia é média com ligação a outros média. Trata-se da aplicação do conceito de hipertexto, a documentos multimédia.

Hipertexto e Hipermedia

- Modelo teórico: *nós* com *ligações* entre si



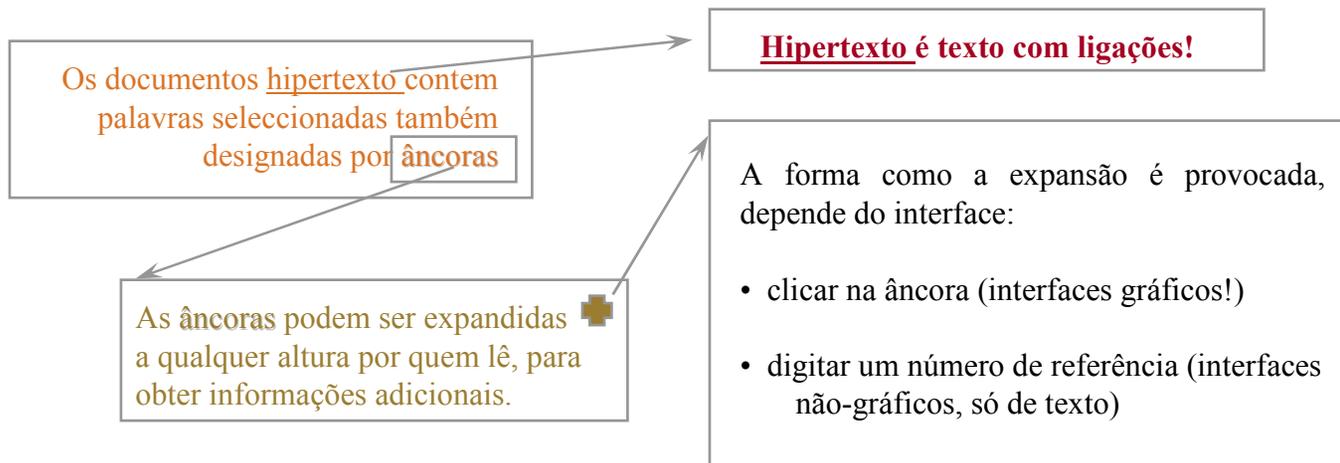
Âncoras são fragmentos de informação (palavras, frases, etc.) dentro de um documento, aos quais se podem associar *ligações*.

Ligações são referências ou apontadores, de uma âncora para outra. Devem identificar o documento destino (*Qual o nome?*), sua localização (*Onde está?*) e forma de acesso (*Como obter?*)

Hipertexto e Hipermedia

- Trata-se de uma ideia muito antiga:
 - *Notas de rodapé e referências internas* (ver cap. Y)
 - *Índices e tabelas de conteúdos*
 - *bibliografias*

O que há de novo é apenas a facilidade com que se seguem os *links*, quer estes sejam *internos* quer *externos*: Basta um *click*!



Hipertexto e Hipermedia

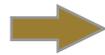
- Podemos ter *cadeias de ligações* com dois objectivos diferentes e complementares:

Focagem

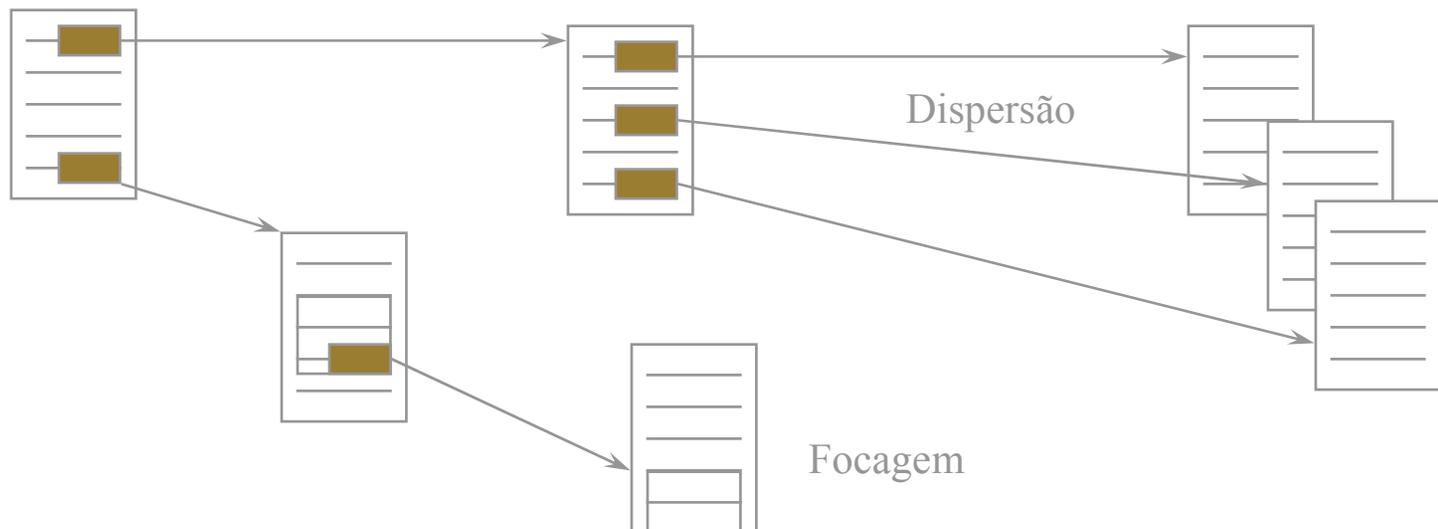


O utilizador progride de documento em documento, restringindo a cada salto o domínio de informação e convergindo para um tópico específico.

Dispersão



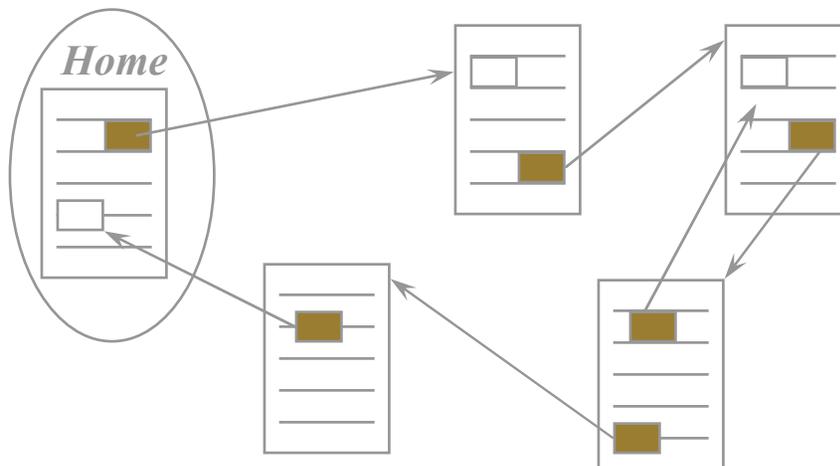
O utilizador não tem uma ideia precisa daquilo em que está interessado, e o sistema abre-lhe novas possibilidades com colecções de referências.



Hipertexto e Hipermedia

- As cadeias de ligações não tem de ser necessariamente hierárquicas:
 - A *estrutura topológica* é normalmente uma *teia*, podendo haver *ciclos* (caminhos que conduzem ao ponto de partida)

Como evitar que os utilizadores se percam nestes labirintos, enquanto *navegam*?



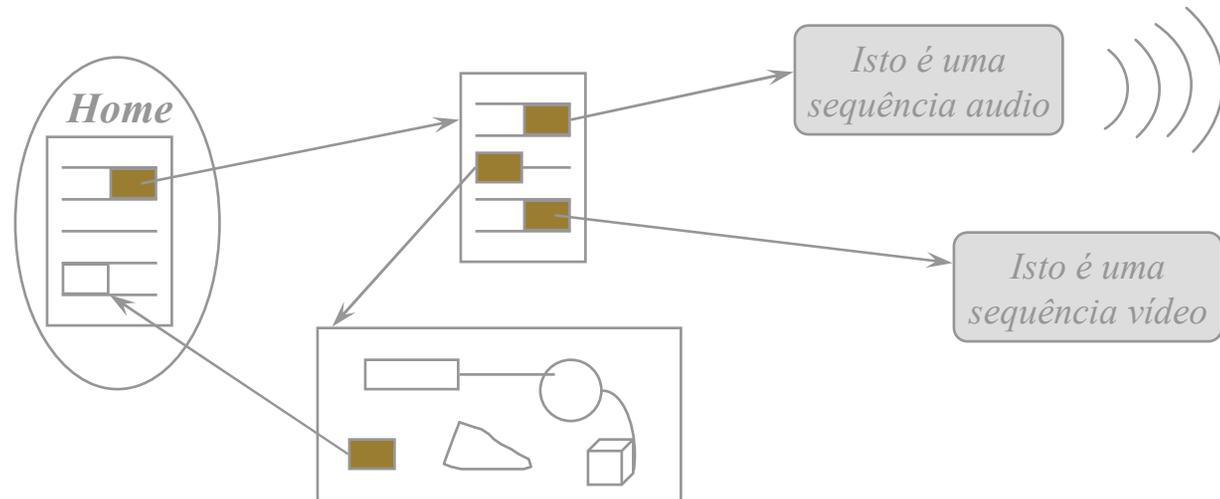
1. Documento inicial (*Home*)
2. Caminho percorrido é caminho de retorno

Hipertexto e Hipermedia

- Outras soluções a que os autores podem recorrer:
 - Mapas com a estrutura, ou diagramas gerais do *site*
 - Visitas Guiadas previamente preparadas
 - Barras de navegação com sugestões de direcção em cada instante
 - Utilização de Frames, fixando numa parte da janela informação de localização (menus, pág. inicial, etc.)
 - Possibilidade de pesquisa pelo conteúdo em certos pontos da navegação
- Conceitos válidos, independentemente da localização:
 - CD-ROM local, no disco rígido, em *servidores* distintos, dispersos geograficamente!

Hipertexto e Hipermedia

- Particularidades dos sistemas hipermedia em *rede*:



- relacionamento temporal entre os objectos!...
- é necessário minimizar *tempos* de transferência:
 - evitar imagens grandes, sons e vídeo em contínuo...
 - objectos de grandes dimensões só a pedido do utilizador:
 - notificar os utilizadores do tamanho e do tempo de transferência aproximado!