

Conceitos Básicos

Linguagens de Anotação de Documentos

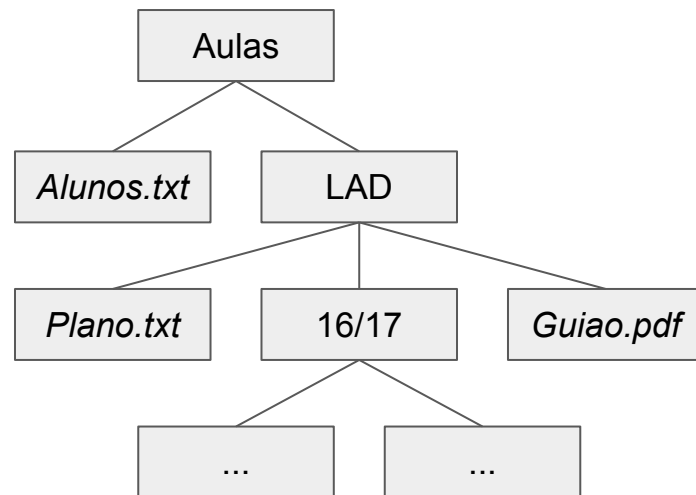
Edição de Documentos

Sistema de Ficheiros

- Gere todos os *dados* armazenados num computador
- Armazenada em dispositivos persistentes (disco duro, *flash drives*, ...)
- Introduz a noção de *ficheiro*, pedaços de informação bem identificados
- Ficheiros agrupados em *diretorias* (ou pastas)
- Têm geralmente *meta-dados* associados (“marcas” temporais, permissões)

Sistema de Ficheiros

- Estrutura de directorias introduz uma hierarquia em *árvore*
- Geralmente navegada por interfaces gráficas do sistema operativo



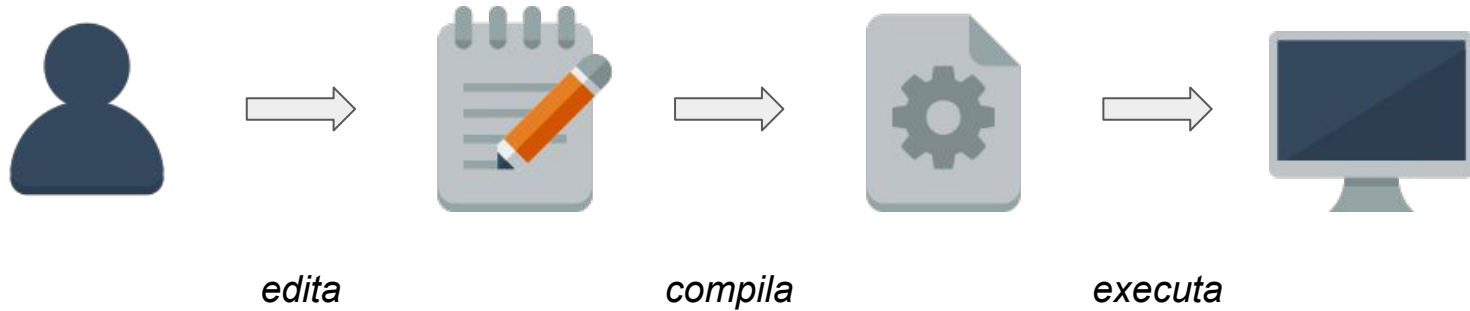
Sistema de Ficheiros

- Vários tipos de ficheiros, geralmente identificados pela extensão
- Distinção relevante:
 - Ficheiros de texto puro (e.g., *txt*)
 - Ficheiros “de aplicação” (e.g., *doc*, *pdf*)
 - Ficheiros executáveis (e.g., *exe*)
- Ficheiros de texto ou “de aplicação” são portáveis entre sistemas operativos
- Os executáveis são específicos para cada sistema operativo
- Apenas os ficheiros de texto puro são legíveis pelos humanos

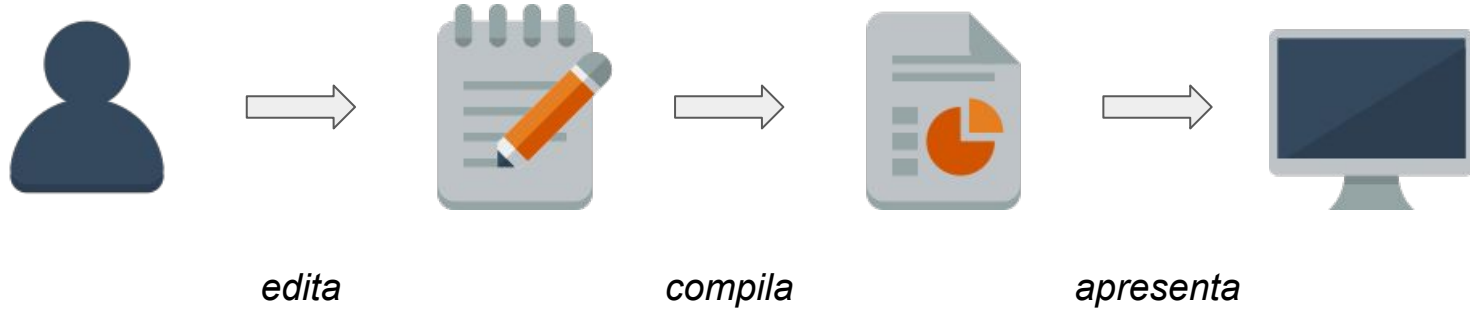
Sistema de Ficheiros

- Fundamental nas ciências de computação é o conceito de *compilação*
- Processo que transforma ficheiros em texto puro (legível por humanos) em ficheiros binários (computáveis pela máquina)
- Geralmente aplica-se à transformação de código em programas executáveis
- No nosso contexto, refere-se à geração de documentos (e.g., *pdf*) de textos anotados (e.g., LaTeX)

Sistema de Ficheiros



Sistema de Ficheiros



Permissões

- Funcionalidade dos sistemas de ficheiros que limita os direitos de acesso dos utilizadores (meta-dados)
- Permite definir o que cada utilizador consegue fazer com cada ficheiro
 - Geralmente leitura, escrita ou execução
- Particularmente importante quando os dados são partilhados entre utilizadores

Escrita de Documentos

- Existem diferentes abordagens à escrita de documentos
- Diferentes conceitos são abordados de maneira diferente:
 - Conteúdo
 - Estrutura
 - Aspecto
 - Disseminação
- Escrever e estruturar o conteúdo de um documento é *conceptualmente* diferente de definir o seu aspecto e formatação

WYSIWYG

- Paradigma de edição “*What You See Is What You Get*” (WYSIWYG)
- O utilizador edita ficheiro no seu aspecto final
- Paradigma implementado em *processadores de texto*
 - *Microsoft Word*
 - *Openoffice*
 - *LibreOffice*
 - *Google Docs*

Processadores de Texto

- Mistura a escrita do documento com a sua formatação, a componente gráfica
- O autor perde tempo com tarefas que não são relevantes até à distribuição final do documento
 - Fontes, alinhamento, páginas...
 - Não afeta propriamente o conteúdo do documento
- Por muito tempo que se invista, é provável que resistam sempre incongruências
- Em particular porque a estrutura é *implícita*

Processadores de Texto

- Edição de ficheiros de aplicação
- O seu formato pode ser *proprietário* (e.g., *docx*):
 - Limita a disseminação e partilha (o *Microsoft Office* não é grátis)
 - Alternativas podem formatar ficheiros de maneira diferente

Processadores de Texto

- Em muitas situações processadores de texto são as soluções mais práticas
- E.g., documentos curtos, temporários e dinâmicos
- Passa a ideia de como o ficheiro ficará impresso
- No entanto, para documentos longos (como garantir que a formatação está congruente ao longo de centenas de páginas?) há alternativas

Linguagens Anotadas

- Ficheiros de texto puro, com anotações sobre meta-informação
- A estrutura torna-se *explícita*
 - Em vez de formatarmos uma “secção” de maneira diferente, dizemos explicitamente que é uma “secção”
- A formatação não é gerida explicitamente
- Quando o ficheiro final for gerado, todas as “secções” são garantidamente formatadas da mesma maneira
- E.g., *LaTeX*, *HTML*, *XML*, *Wiki*

Linguagens Anotadas

- Curva de aprendizagem mais acentuada: não se edita o documento visualmente
- Conhecer as *anotações* disponíveis para cada linguagem (daí editores de texto adequados serem essenciais)
- Pode não ser evidente as consequências no documento final
- Torna-se mais rígido, se quisermos fazer alterações de formatação muito específicas
- Geralmente passa por um processo de compilação (ou interpretação) para gerar documentos finais (e.g., *pdf*)

Editores de Texto

- Visto agora lidarmos com texto puro, são usados *editores de texto*:
 - *Notepad*
 - *TextEdit*
 - *Gedit*
 - *Sublime Text*
- O texto puro é totalmente *portável*
- Editores podem oferecer funcionalidades avançadas como “realce” de sintaxe (*highlighting*) ou “auto-complemento” (*auto-complete*)

Codificação de Caracteres

- Quando lidamos com ficheiros de texto puro levanta-se a questão de como codificar os diferentes caracteres dum sistema de escrita
 - Este tipo de ficheiros é portátil e computável, por isso os esquemas têm que estar bem definidos
- A codificação de caracteres (*encoding*) define o conjunto de símbolos disponíveis para escrever em texto puro
- O espaço ocupado por cada carácter depende de quantos símbolos estiverem disponíveis

Codificação de Caracteres

- Historicamente o esquema mais usado é o chamado ASCII
- Devido a limitações de espaço e de memória permite apenas 128 caracteres diferentes
 - Caracteres alfa-números e estruturas de controlo
- Não inclui, e.g., caracteres acentuados
- Daí ser boa regra em geral não usar acentos ou cedilhas em nomes de ficheiros

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	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	Neg. 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 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	Substitution	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	□

Codificação de Caracteres

- A solução (temporária) foi estender ligeiramente o conjunto (256) mas atribuir diferentes símbolos dependendo do *encoding* selecionado
 - Para ler um texto em Português, o utilizador tinha que saber que o *encoding* era da Europa Ocidental
- Após muita discussão, a comunidade conseguiu acordar num *encoding* uniforme, o extremamente complexo *Unicode*

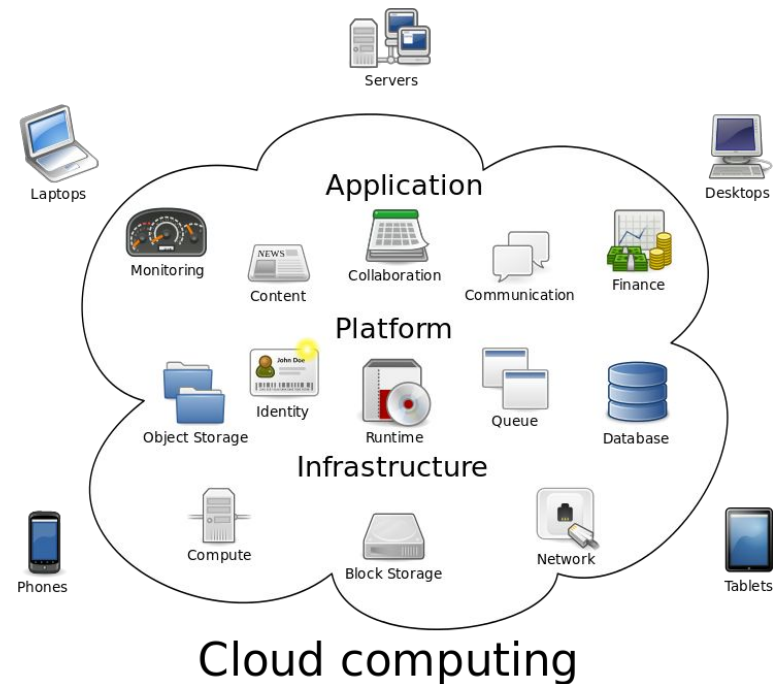
Codificação de Caracteres

- O Unicode permite codificar todos os símbolos de todas as linguagens conhecidas (1,112,064 códigos)
 - Assumindo que a aplicação os sabe apresentar!
 - O que não parece ser o caso para a escrita cuneiforme suméria □□□
 - Mas sim do alfabeto fonético internacional (IPA, [aɪ pi: eɪ])
- Solução técnica: cada carácter pode ter dimensão variada, dependendo do contexto
- A codificação UTF-8 é hoje em dia universalmente usada

Trabalho Colaborativo na Nuvem

Computação em Nuvem

- Partilha de recursos (armazenamento e processamento) na *internet*
- Exemplo mais comum: serviços de *email* (a não ser que seja um servidor local)
- Dados e aplicações *acessíveis* de qualquer lado através de qualquer dispositivo (desde que haja internet!)
- Problema de *privacidade*: os dados estão efetivamente do lado da nuvem



Serviços de Armazenamento

- Permitem armazenar dados na nuvem
- Abstraem as funcionalidades de um sistema de ficheiros
- *Backups* permanentes
 - Uma *pen drive* é em geral menos robusta que os “*data centers*” da Google
- Propício a trabalho *colaborativo*, visto os dados estarem também na nuvem
- Exemplos: *DropBox*, *Google Drive*, *One Drive*

Software com um Serviço








- *Software as a Service* (SaaS) na nuvem, evita a instalação local de ferramentas
- Acessíveis de qualquer dispositivo, sempre com a versão mais atualizada
- Abstrai toda a infraestrutura técnica
- Exemplos: *Google Docs*, *Microsoft Office 365*

Software como um Serviço

- Limitações: como se acessam através de *browsers* têm menos funcionalidades e pior desempenho
- Necessita de ligação permanente: os dados não são persistidos localmente
- Sempre a questão da privacidade: o modelo de negócio da Google envolve explorar as nossas preferências

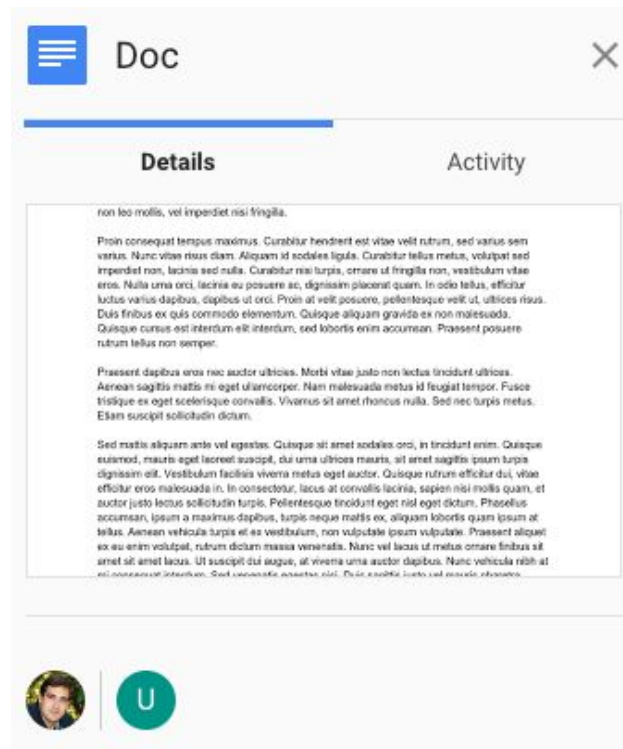
Trabalho Colaborativo

- Paradigma tradicional:
 - Editar ficheiro localmente
 - Transferir para o resto da equipa (*pen drive, email*)
 - Como gerir mudanças em conflito?
 - Como gerir diferentes versões?

Name	
	doc-final copy.txt
	doc-final-backup.txt
	doc-final-final.txt
	doc-final.txt
	doc-final2.txt
	doc-nmm.txt
	doc.txt

Trabalho Colaborativo

- Paradigma moderno:
 - Ficheiro está na nuvem e é editado remotamente
 - Partilhado com colaboradores, gestão de permissões
 - Edição em paralelo
 - Histórico de alterações, comentários



Colaboração na Nuvem

- Sempre versão mais atualizada dos documentos
- Sempre *backups* automáticos
- Comentários e troca de mensagem em tempo real
- Gestão de permissões (leitura, edição)
- Alertas, atribuição de tarefas
- Centrado no documento (os comentários estão no documento, e não num *email* ao lado)

Controlo de Versões

- Quando um projeto é editado por uma equipa torna-se necessário controlar as mudanças
- Isto é particularmente importante na nuvem, visto não existirem versões locais dos documentos para recuperar
- Funcionalidades mais básicas incluem mostrar o histórico de edições e a autoria
- Sistemas mais avançados permitem reverter mudanças, gerir edições paralelas em conflito, etc