

Manual do Curso de Licenciatura em

Engenharia Informática

2º Ano

Disciplina: Tecnologia Web

Código:

Total Horas/1º Semestre:

Créditos (SNATCA):

2022



Direitos de autor (copyright)

Este manual é propriedade doInstituto Superior de Ciências e Educação a Distância (ISCED), e contém reservado todos os direitos. É proibida a duplicação ou reprodução parcial ou total deste manual, sob quaisquer formas ou por quaisquer meios (electrónicos, mecânico, gravação, fotocópia ou outros), sem permissão expressa da entidade editora (Instituto Superior de Ciências e Educação a Distância (ISCED)).

A não observância do acima estipuladooo infractor é passível a aplicação de processos judiciais em vigor no País.

Universiade Aberta ISCED
Rua Dr. Almeida Lacerda, Nº 212 Ponta - Gêa

Beira - Moçambique

Telefone: +258 23 323501

Cel: +258 82 3055839

E-mail: unisced@unisced.edu.mz

Website:www.unisced.edu.mz

Agradecimentos

O Instituto Superior de Ciências e Educação a Distância (ISCED) agradece a colaboração dos seguintes indivíduos e instituições na elaboração deste manual:

Autor Dr. Flávio Miguel Pinto Monteiro

Coordenação	Direcção Académica
Design	Universidade Aberta ISCED
Financiamento e Logística	Instituto Africano de Promoção da Educação a Distância (IAPED)
Revisão Científica e Linguística	Dr. Flávio Miguel Pinto Monteiro
Ano de Publicação	2019
Local de Publicação	ISCED – BEIRA

Índice

Visão geral	1
Benvindo à Disciplina/Módulo de Tecnologias Web.....	1
Objectivos do Módulo	1
Quem deveria estudar este módulo	1
Como está estruturado este módulo.....	2
Ícones de actividade	Erro! Marcador não definido.
Habilidades de estudo.....	3
Precisa de apoio?.....	5
Tarefas (avaliação e auto-avaliação)	6
Avaliação	6
DESCRIPÇÃO GERAL DO MÓDULO	9
Tecnologias Web e Programação Web.....	9
Pré-requisitos	10
Materiais	10
Objectivos	11
Temas.....	12
1. Introdução a web 2.0	12
2. Utilização de redes sociais para partilha de informação (Facebook, Whatsapp...)	12
3. Ferramentas de vídeo para fóruns visuais.....	13
4. Ferramentas para partilha de documentos (Dropbox, Wetransfer, Google Drive,...)	13
5. Blog	13
6. Gestionários de conteúdos com Joomla	14
7. Hipertexto, hipermédia, protocolos para internet e para Web	14
8. Linguagem de marcação (HTML, CSS).....	14
9. Programação do lado do cliente (Javascript)	15
Unidade Temática 1.1: Web	17
Unidade Temática 1.2: Aplicação Web.....	22
Fig. 1.3. Aplicações Web Móveis	25
Unidade Temática 1.3: Modelo cliente-servidor.....	25
Unidade Temática 1.4: Desenvolvimento Web	28
Exercícios de AUTO-AVALIAÇÃO	31
Unidade Temática 2.1: Redes sociais <i>online</i>	40
Unidade Temática 2.2: Marketing nas redes sociais	42
Unidade Temática 2.3: Formas de redes sociais	43
Unidade Temática 2.4: Análise das redes sociais	44
Exercícios de AUTO-AVALIAÇÃO	46
5. Em quais redes sociais devo estar presente?	47
6. O que é melhor postar?.....	47

7. Como devo interagir?	47
8. Por que devo contractar uma agência se eu mesmo faço?	47
9. Como avaliar o resultado?	47
10. Com que frequência devo postar?	47
11. Posso utilizar uma ferramenta para programar as postagens?	47
Exercícios para AVALIAÇÃO	47
Unidade Temática 3.1: Os 5 melhores editores de vídeo on-line para edição de vídeo on-line e os 5 melhores editores de imagens.....	54
Unidade Temática 3.2: Os 5 melhores editores de vídeo online.....	56
Unidade Temática 3.3: Os 5 melhores editores de imagens	61
Exercícios de AUTO-AVALIAÇÃO	67
Unidade Temática 4.1: Dropbox	76
Unidade Temática 4.2: Wetransfer.....	79
Unidade Temática 4.3: Google Drive.....	84
Exercícios de AUTO-AVALIAÇÃO	87
Unidade Temática 5.1: O que é Blog?.....	95
Unidade Temática 5.2: Origens <i>Blog</i>	97
Unidade Temática 5.3: Tipos de <i>Blog</i>	98
Unidade Temática 5.4: Componentes <i>Blog</i>	99
Exercícios de AUTO-AVALIAÇÃO	102
Unidade Temática 6.1: Sistema de Gestão de Conteúdo	110
Unidade Temática 6.2: Os Principais Sistemas CMS Gratuitos	113
Exercícios de AUTO-AVALIAÇÃO	118
Unidade Temática 7.1: Hipertexto e Hipermédia	127
Unidade Temática 7.2: <i>HyperText Transfer Protocol (HTTP)</i>	129
Unidade Temática 7.3: Modelo OSI.....	130
Camada Protocolo.....	132
Unidade Temática 7.4: Os Principais Protocolos da Internet	136
.....	144
Exercícios de AUTO-AVALIAÇÃO	168
TEMA 8. LINGUAGEM DE MARCAÇÃO (HTML, CSS)	174
Unidade Temática 8.1: O que é HTML?	175
Unidade Temática 8.2: Codificando em HTML	179
O que são <i>Tags</i> , Elementos e Atributos?	179

Unidade Temática 8.3: Estrutura Básica HTML.....	183
Unidade Temática 8.4:CSS	187
Unidade Temática 8.5: Escrevendo CSS	191
Exercícios de AUTO-AVALIAÇÃO	208

TEMA 9. PROGRAMAÇÃO DO LADO DO CLIENTE (JAVASCRIPT) 215

Unidade Temática 9.1: O que é Javascript?.....	216
Unidade Temática 9.2: Codificando em Javascript.....	223
Unidade Temática 9.3: Operadores	233
Unidade Temática 9.4: Funções	237
Unidade Temática 9.5: Declarações (Statements)	240
Unidade Temática 9.6: Arrays	244
Unidade Temática 9.7: O que são Objectos?	247
Unidade Temática 9.8: O que é um Framework?	252
Unidade Temática 9.9: O que é jQuery?	253
Exercícios de AUTO-AVALIAÇÃO	255
Referências Bibliográficas	261

Visão geral

Benvindo à Disciplina/Módulo de Tecnologias Web

Objectivos do Módulo

Este módulo vai capacitar os estudantes a compreender o funcionamento, a utilizar na íntegra os seus recursos, a projectar, a desenhar e a implementar aplicações Web modernas.

Ao terminar o estudo deste módulo de Tecnologias Web deverá ser capaz de:

Demonstrar continuamente com uso de suplemento contínuo de pequenas pesquisas de leitura e experimento, demonstrar conhecimentos sobre as principais e quaisquer outras tecnologias Web; Instalar qualquer aplicação Web moderna e utilizar os recursos na sua íntegra; Projectar, desenhar e implementar qualquer tipo de aplicações Web modernas;

Quem deveria estudar este módulo

Este Módulo foi concebido para estudantes do 1º ano do curso de Licenciatura em Engenharia Informática do ISCED. Poderá ocorrer, contudo, que haja leitores

Como está estruturado este módulo

Este módulo de Tecnologias Web, para estudantes do 1º ano do curso de Licenciatura em Engenharia Informática, à semelhança dos restantes do ISCED, está estruturado como se segue:

Páginas introdutórias

- Um índice completo.
- Uma visão geral detalhada dos conteúdos do módulo, resumindo os aspectos-chave que você precisa conhecer para melhor estudar. Recomendamos vivamente que leia esta secção com atenção antes de começar o seu estudo, como componente de habilidades de estudos.

Conteúdo desta Disciplina/Módulo

Este módulo está estruturado em Temas. Cada tema, por sua vez comporta certo número de unidades temáticas ou simplesmente unidades. Cada unidade temática se caracteriza por conter uma introdução, objectivos, conteúdos.

Os exercícios de avaliação têm as seguintes características: Puros exercícios teóricos/práticos, problemas não resolvidos e actividades práticas, incluído estudo de caso.

Outros recursos

A equipa dos académicos e pedagogos do ISCED, pensando em si, num cantinho, recôndito deste nosso vasto Moçambique e cheio de dúvidas e limitações no seu processo de aprendizagem, apresenta uma lista de recursos didácticos adicionais ao seu módulo para você explorar. Para tal o ISCED disponibiliza na biblioteca do seu centro de recursos mais material de estudos relacionado com o seu curso como: Livros e/ou módulos, CD, CD-ROOM, DVD. Para além deste material físico ou electrónico disponível na biblioteca, pode ter acesso a Plataforma digital moodle para alargar mais ainda as possibilidades dos seus estudos.

Auto-avaliação e Tarefas de avaliação

Tarefas de **auto-avaliação** para este módulo encontram-se no final de cada unidade temática e de cada tema. As tarefas dos exercícios de auto-avaliação apresentam duas características: primeiro apresentam exercícios resolvidos com detalhes. Segundo, exercícios que mostram apenas respostas.

Tarefas de **avaliação** devem ser semelhantes às de auto-avaliação mas sem mostrar os passos e devem obedecer o grau crescente de dificuldades do processo de aprendizagem,umas a seguir a outras. Parte das tarefas de avaliação será objecto dos trabalhos de campo a serem entregues aos tutores/docentes para efeitos de correção e subsequentemente nota. Também constará do exame do fim do módulo. Pelo que, caro estudante, fazer todos os exercícios de avaliação é uma grande vantagem.

Comentários e sugestões

Use este espaço para dar sugestões valiosas, sobre determinados aspectos, quer de natureza científica, quer de natureza didáctico-Pedagógica, etc, sobre como deveriam ser ou estar apresentadas. Pode ser que graças às suas observações que, em gozo de confiança classificámo-las de úteis, o próximo módulo venha a ser melhorado.

Habilidades de estudo

Durante a formação e desenvolvimento de competências, para facilitar a aprendizagem e alcançar melhores resultados, implicará empenho, dedicação e disciplina no estudo. Isto é, os bons resultados apenas se conseguem com estratégias eficientes e eficazes. Por isso é importante saber **como, onde e quando** estudar. Apresentamos algumas sugestões com as quais esperamos que caro estudante possa rentabilizar o tempo dedicado aos estudos, procedendo como se segue:

1º Praticar a leitura. Aprender a Distância exige alto domínio de leitura.

2º Fazer leitura diagonal aos conteúdos (leitura corrida).

3º Voltar a fazer leitura, desta vez para a compreensão e assimilação crítica dos conteúdos (ESTUDAR).

4º Fazer seminário (debate em grupos), para comprovar se a sua aprendizagem confere ou não com a dos colegas e com o padrão.

5º Fazer TC (Trabalho de Campo), algumas actividades práticas ou as de estudo de caso se existirem.

IMPORTANTE: Em observância ao triângulo **modo-espacotempo**, respectivamente **como, onde e quando**...estudar, como foi referido no início deste item, antes de organizar os seus momentos de estudo reflecta sobre o ambiente de estudo que seria ideal para si: Estudo melhor em casa/biblioteca/café/outrolugar? Estudo melhor à noite/de manhã/de tarde/fins-de-semana/ao longo da semana? Estudo melhor com música/num sítio sossegado/num sítio barulhento!?

É impossível estudar numanoite tudo o que devia ter sido estudado durante um determinado período de tempo; Deve estudar cada ponto da matéria em profundidade e passar só ao seguinte quando achar que já domina bem o anterior.

Privilegia-se saber bem (com profundidade) o pouco que puder ler e estudar, que saber tudo superficialmente! Mas a melhor opção é juntar o útil ao agradável: Saber com profundidade todos conteúdos de cada tema, no módulo.

Dica importante: não recomendamos estudar seguidamente por tempo superior a uma hora. Estudar por tempo de uma hora intercalado por 10 (dez) a 15 (quinze) minutos de descanso (chama-se descanso à mudança de actividades). Ou seja que durante o intervalo não se continuar a tratar dos mesmos assuntos das actividades obrigatórias.

Uma longa exposição aos estudos ou ao trabalho intelectual obrigatório pode conduzir ao efeito contrário: baixar o rendimento da aprendizagem. Por que o estudante acumula um elevado volume de trabalho, em termos de estudos, em pouco tempo, criando interferência entre os conhecimentos, perde sequência lógica, por fim ao perceber que estuda tanto mas não aprende, cai em insegurança, depressão e desespero, por se achar injustamente incapaz!

Não estude na última da hora; quando se trate de fazer alguma avaliação. Aprenda a ser estudante de facto (aquele que estuda sistematicamente), não estudar apenas para responder a questões de alguma avaliação, mas sim estude para a vida, sobre tudo, estude pensando na sua utilidade como futuro profissional, na área em que está a se formar.

Organize na sua agenda um horário onde define a que horas e que matérias deve estudar durante a semana; Face ao tempo livre que resta, deve decidir como o utilizar produtivamente, decidindo quanto tempo será dedicado ao estudo e a outras actividades.

É importante identificar as ideias principais de um texto, pois será uma necessidade para o estudo das diversas matérias que compõem o curso: A colocação de notas nas margens pode ajudar a estruturar a matéria de modo que seja mais fácil identificar as partes que está a estudar e Pode escrever conclusões, exemplos, vantagens, definições, datas, nomes, pode também utilizar a margem para colocar comentários seus relacionados com o que está a ler; a melhor altura para sublinhar é imediatamente a seguir à compreensão do texto e não depois de uma primeira leitura; Utilizar o dicionário sempre que surja um conceito cujo significado não conhece ou não lhe é familiar;

Precisa de apoio?

Caro estudante, temos a certeza que por uma ou por outra razão, o material de estudos impresso, lhe pode suscitar algumas dúvidas como falta de clareza, alguns erros de concordância, prováveis erros ortográficos, falta de clareza, fraca visibilidade, página trocada ou invertidas,etc). Nestes casos, contacte os serviços de atendimento e apoio ao estudante do seu Centro de Recursos (CR), via telefone, SMS, E-mail, se tiver tempo, escreva mesmo uma carta participando a preocupação.

Uma das atribuições dos Gestores dos CR e seus assistentes (Pedagógico e Administrativo), é a de monitorar e garantir a sua aprendizagem com qualidade e sucesso. Daí a relevância da comunicação no Ensino a Distância (EAD), onde o recurso às TIC se torna incontornável: entre estudantes, estudante – Tutor, estudante – CR, etc.

As sessões presenciais são um momento em que você caro estudante, tem a oportunidade de interagir fisicamente com staff do seu CR, com tutores ou com parte da equipa central do ISCEDindigitada para acompanhar as suas sessões presenciais. Neste

período pode apresentar dúvidas, tratar assuntos de natureza pedagógica e/ou administrativa.

O estudo em grupo, que está estimado para ocupar cerca de 30% do tempo de estudos a distância, é muita importância, na medida em que lhe permite situar, em termos do grau de aprendizagem com relação aos outros colegas. Desta maneira ficará a saber se precisa de apoio ou precisa de apoiar aos colegas. Desenvolver hábito de debater assuntos relacionados com os conteúdos programáticos, constantes nos diferentes temas e unidade temática, no módulo.

Tarefas (avaliação e auto-avaliação)

O estudante deve realizar todas as tarefas (exercícios, actividades e auto-avaliação), contudo nem todas deverão ser entregues, mas é importante que sejam realizadas. As tarefas devem ser entregues duas semanas antes das sessões presenciais seguintes.

Para cada tarefa serão estabelecidos prazos de entrega, e o não cumprimento dos prazos de entrega, implica a não classificação do estudante. Tenha sempre presente que a nota dos trabalhos de campo conta e é decisiva para ser admitido ao exame final da disciplina/módulo.

Os trabalhos devem ser entregues ao Centro de Recursos (CR) e os mesmos devem ser dirigidos ao tutor/docente.

Podem ser utilizadas diferentes fontes e materiais de pesquisa, contudo os mesmos devem ser devidamente referenciados, respeitando os direitos do autor.

O plágio¹é uma violação do direito intelectual do(s) autor(es).Uma transcrição à letra de mais de 8 (oito) palavras do texto de um autor, sem o citar, é considerado plágio. A honestidade, humildade científica e o respeito pelos direitos autorais devem caracterizar a realização dos trabalhos e seu autor (estudante do ISCED).

Avaliação

Muitos perguntam: Com é possível avaliar estudantes à distância, estando eles fisicamente separados e muito distantes do docente/tutor! Nós dissemos: Sim é muito possível, talvez seja uma avaliação mais fiável e consistente.

¹Plágio - copiar ou assinar parcial ou totalmente uma obra literária, propriedade intelectual de outras pessoas, sem prévia autorização.

Você será avaliado durante os estudos à distância que contam com um mínimo de 90% do total de tempo que precisa de estudar os conteúdos do seu módulo. Quando o tempo de contacto presencial conta com um máximo de 10% do total de tempo do módulo. A avaliação do estudante consta detalhada do regulamentado de avaliação.

Os exames são realizados no final da cadeira disciplina ou modulo e decorrem durante as sessões presenciais. Os exames pesam no mínimo 75%, o que adicionado aos 25% da média de frequência, determinam a nota final com a qual o estudante conclui a cadeira.

A nota de 10 (dez) valores é a nota mínima de conclusão da cadeira.

Nesta cadeira o estudante deverá realizar pelo menos 2 (dois) trabalhos e 1 (um) (exame).

Algumas actividades práticas, relatórios e reflexões serão utilizados como ferramentas de avaliação formativa.

Durante a realização das avaliações, os estudantes devem ter em consideração a apresentação, a coerência textual, o grau de científicidade, a forma de conclusão dos assuntos, as recomendações, a identificação das referências bibliográficas utilizadas, o respeito pelos direitos do autor, entre outros.

Os objectivos e critérios de avaliação constam do Regulamento de Avaliação.

Descrição Geral do Módulo**Tecnologias Web e Programação Web**

Este módulo centraliza-se nas Tecnologias Web, nas tecnologias usadas para desenhar e implementar aplicações web, como páginas web por exemplo, usadas para desenhar e implementar programas e sistemas web, como os demais programas que utilizamos sempre que estamos conectados a internet. Portanto, todos esses programas que utilizamos na internet (ou na web) funcionam com base nos conhecimentos e habilidades desenvolvidas nas diferentes áreas das Tecnologias Web.

Este módulo vai ajudar os estudantes a compreenderem como funcionam as tecnologias Web, assim como ajudá-los a projetar, desenhar e implementar aplicações web modernas.

Este é um módulo importante porque a Web é hoje o centro da economia digital e ter experiência em desenvolvimento Web é uma vantagem frente a concorrência num mercado centrado no uso das Tecnologias.

Pré-requisitos

São necessários conhecimentos básicos de utilização do computador; conhecimentos básicos de informática, como directórios, ficheiros; conhecimentos básicos de sistemas operativos (Windows, Linux, OS X); fundamentos de linguagens de programação (Java) e os princípios básicos de programação (variáveis, atribuição, loops, estruturas de controle, etc).

Materiais

O estudante deve ter disponível um computador com sistema operativo Windows, Linux ou OS X, com conexão a internet.

Objectivos

Este módulo vai capacitar os estudantes a compreender o funcionamento, a utilizar na íntegra os seus recursos, a projectar, a desenhar e a implementar aplicações Web modernas.

No fim deste módulo, o estudante deve ser capaz de:



Objectivos específicos

- Demonstrar continuamente com uso de suplemento contínuo de pequenas pesquisas de leitura e experimento, demonstrar conhecimentos sobre as principais e quaisquer outras tecnologias Web;
- Instalar qualquer aplicação Web moderna e utilizar os recursos na sua íntegra;
- Projectar, desenhar e implementar qualquer tipo de aplicações Web modernas;

Temas**1. Introdução a web 2.0**

Esta unidade vai ajudá-lo a conhecer os conceitos, as boas práticas e as tecnologias por detrás do bom funcionamento da internet e de todas as suas aplicações Web.

Nesta unidade vai aprender também os conceitos e práticas sobre a programação Web do lado do cliente e do lado do servidor. A arquitectura cliente-servidor.

**2. Utilização de redes sociais para partilha de informação
(Facebook, Whatsapp...)**

Esta unidade vai ajudá-lo a conhecer as melhores teorias e as melhores práticas de utilização das redes sociais para uma partilha de informação ou exposição estável e segura. Os prós e contras da utilização das redes sociais para a partilha de informação.

Quais os objectivos com maiores vantagens e desvantagens no uso das redes sociais? Qual o impacto político que o devido e/ou indevido uso das redes sociais pode ter? Quais as desvantagens de uma utilização não resguardada das redes sociais?

3. Ferramentas de vídeo para fóruns visuais

Esta unidade vai ajudar o estudante a conhecer as principais ferramentas para elaboração de vídeos *online* a serem partilhados em fóruns de estudo, de discussão ou de partilha de conteúdos online.

Esses vídeos podem ser de interacção com o utilizador, pelo que as ferramentas que serão aqui apresentadas vão guiar o estudante na produção dos seus próprios vídeos com ou sem interacção como forma de enriquecerem e distinguirem as vossas apresentações.

4. Ferramentas para partilha de documentos (Dropbox, WeTransfer, Google Drive,...)

Esta unidade vai dar a conhecer as ferramentas para partilha de documentos, disponibilizar os mecanismos para a instalação e a utilização na íntegra dos recursos das principais aplicações Web utilizadas para a partilha de documentos na internet.

5. Blog

Esta unidade vai dar a conhecer as ferramentas para a criação e a publicação de *Blogs*. Disponibilizar os mecanismos para a melhor indexação dos seus *Blogs* de modo a que possam causar os melhores números de visualizações esperados.

6. Gestionários de conteúdos com Joomla

Esta unidade vai dar a conhecer as ferramentas para a criação e a gestão dos seus próprios conteúdos *online*, para a criação e administração dos seus próprios websites.

Vai aprender nesta unidade o processo para hospedagem dum *website*.

7. Hipertexto, hipermédia, protocolos para internet e para Web

Esta unidade vai ensinar os principais conceitos e protocolos que estão na vanguarda do bom funcionamento da internet e das suas aplicações Web.

Nesta unidade vai aprender as regras de funcionamento dos principais protocolos que regem o funcionamento da internet (HTTP, FTP, SSH, etc).

8. Linguagem de marcação (HTML, CSS)

Nesta unidade vai aprender as regras de funcionamento das linguagens de escrita para a criação de páginas Web (HTML, CSS) para apresentação, interação e coleta de dados e informação através de páginas Web.

9. Programação do lado do cliente (Javascript)

Nesta unidade vai aprender a linguagem de programação Web do lado do cliente, Javascript.

Como escrever, criar e publicar páginas Web com conteúdos interactivos Javascript.

TEMA 1. INTRODUÇÃO ÀS TECNOLOGIAS WEB

Tecnologias Web

São todas as tecnologias por detrás do funcionamento da Web.

Esta unidade vai ajudá-lo a conhecer os conceitos, as boas práticas e as tecnologias por detrás do bom funcionamento da internet e de todas as suas aplicações Web.



**Objectivos
Específicos**

Nesta unidade vai aprender também os conceitos e práticas sobre a programação Web do lado do cliente e do lado do servidor. A arquitectura cliente-servidor.

Unidade Temática 1.1: Web

O World Wide Web (WWW) é um termo técnico (e anglicismo de tecnologia da informação) que foi traduzido para a língua portuguesa como rede mundial de computadores, também conhecido como Web, que designa um sistema de documentos em hipermídia (hipermédia) que são interligados e executados na Internet.

Os documentos podem estar na forma de vídeos, sons, hipertextos e imagens. Para consultar a informação, pode-se usar um programa de computador chamado navegador web (Internet Explorer, Google Chrome, Mozilla Firefox, Microsoft Edge, Opera etc), para descarregar informações (chamadas "documentos" ou "páginas") de servidores web (ou "sítios") e mostrá-los na tela do usuário (ecrã do utilizador). O usuário (utilizador) pode então seguir as hiperligações na página para outros documentos ou mesmo enviar informações de volta para o servidor para interagir com ele. O ato de seguir hiperligações é comumente chamado "navegar" ou "surfar" na Web.



Fig. 1.1. Aplicações Web Desktop

Visualizar uma página web ou outro recurso disponibilizado normalmente inicia ou ao digitar uma *URL* no navegador ou seguindo (acessando) uma hiperligação. Primeiramente, a parte da URL referente ao servidor de rede é separada e transformada em um endereço IP, por um banco de dados da Internet chamado *domain name system* (DNS). O navegador estabelece, então, uma conexão TCP-IP com o servidor web localizado no endereço IP retornado.

O próximo passo é o navegador enviar uma requisição http ao servidor para obter o recurso indicado pela parte restante da *URL* (retirando-se a parte do servidor). No caso de uma página web típica, o texto HTML é recebido e interpretado pelo navegador, que realiza então requisições adicionais para figuras, arquivos de formatação, arquivos de *script* e outros recursos que fazem parte da página.

O navegador, então, renderiza (reconstitui) a página na tela do usuário (utilizador), assim como descrita pelos arquivos que a compõe.

Web 1.0, Web 2.0 e Web 3.0

A web 1.0 é considerada como estática, sendo que seus conteúdos não podem ser alterados pelos usuários (utilizadores) finais. Todo o conteúdo da página é somente leitura, por isso o termo estático. Na web 1.0, não existia a interatividade do usuário com a página, onde somente o *webmaster* ou o programador pode realizar alterações ou actualizações da página.

A web 2.0 é o que usamos actualmente, destaca-se por ser dinâmica, ao contrário da web 1.0 que é estática. Referindo-se à web 2.0, dinâmico indica a interatividade e participação do usuário final com a estrutura e conteúdo da página. Nela, o usuário final pode postar comentários, enviar imagens, compartilhar arquivos e fazer milhares de outras coisas que a web 1.0 não permitia. Outra grande mudança entre a web 1.0 e a web 2.0 foi que o usuário diminuiu a taxa de carregamento (*download*) e aumentou a de envio (*upload*), o que indica que o usuário está interagindo mais com a web e trocando mais informações por compartilhamento.

A web 2 é chamada de participativa ou colaborativa.

A web 3.0 é uma evolução da 2.0, pois tem o intuito de mudar as formas de pesquisas para facilitar a vida do usuário da web 2.0, a fim de que possa suprir as necessidades de hoje que são consideradas extravasamento de dados, ou seja, o usuário está postando muitos dados aleatoriamente e isso dificulta a localização. A web 3.0 também vem incrementar a interactividade entre homem e máquina, melhorando as linguagens de programação para que o homem e a máquina falem a mesma língua. Como exemplo, podemos utilizar o Google que inovou seu *site* com uma nova forma de pesquisa interativa: o usuário pode encontrar informações sobre o arquivo que ele adicionou na barra de pesquisa do *site* do Google. Essa é uma das formas que podemos apresentar a web 3.0 que ainda é só um conceito que está chegando a sua fase final e entrando em aplicação.

A Web 3 é chamada de semântica ou *marketing*.



Fig. 1.2. Aplicações Web Móveis

Navegador ou *browser*

O navegador é um programa de computador usado para visualizar recursos da WWW, como páginas web, imagens e vídeos. Com ele também é possível por comunicar-se com o servidor web a fim de receber ou enviar informações. O primeiro navegador desenvolvido no CERN foi o *WorldWideWeb*, pelo próprio Tim Berners-Lee, para plataforma NeXTSTEP em 1990. Mas, mais adiante, surgiram outros navegadores, como o Viola, da Pei Wei (1992). Marc Andreessen, da NCSA lançou um navegador chamado "Mosaic para X" em 1993 que causou um tremendo aumento na popularidade da web entre usuários novos. Andreessen fundou a Mosaic Communication Corporation (hoje Netscape Communications). Características adicionais como conteúdo dinâmico, música e animação podem ser encontrados em navegadores modernos. Frequentemente, as capacidades técnicas de navegadores e servidores avançam muito mais rápido que os padrões conseguem se ajustar, por isso não é incomum que essas características não funcionem propriamente em todos os computadores.

A necessidade de encontrar exactamente a informação desejada surgiu com a WWW: desta constatação, vieram os primeiros motores de busca.

Motores de Busca

Um motor de busca é feito para auxiliar a procura de informações armazenadas na rede mundial (WWW), dentro de uma rede corporativa ou de um computador pessoal. Ele permite que uma pessoa solicite conteúdo de acordo com um critério específico (tipicamente contendo uma dada palavra ou frase) e responde com uma lista de referências que combinam com tal critério, ou seja é uma espécie de catálogo mágico. Ao se realizar uma consulta, a lista de ocorrências de assuntos é criada previamente por meio de um conjunto de softwares de computadores, conhecidos como Web crawler, que vasculham toda a Web em busca de ocorrências de um determinado assunto em uma página. Ao encontrar uma página com muitos links,

os spiders embrenham-se por eles, conseguindo, inclusive, vasculhar os diretórios internos - aqueles que tenham permissão de leitura para usuários - dos *sites* nos quais estão trabalhando.

Os motores de busca usam regularmente índices actualizados para funcionar de forma rápida e eficiente. Sem maior especificação, ele normalmente refere-se ao serviço de busca Web, que procura informações na rede pública da Internet. Outros tipos incluem motores de busca para empresas (Intranets), motores de busca pessoais e motores de busca móveis. De qualquer forma, enquanto diferente seleção e relevância podem aplicar-se em diferentes ambientes, o utilizador provavelmente perceberá uma pequena diferença entre as operações neles. Alguns motores também extraem dados disponíveis em grupos de notícias, grandes bancos de dados ou diretórios abertos como a DMOZ.org (Open Directory Project). Ao contrário dos directórios Web, que são mantidos por editores humanos, os serviços de busca funcionam algorítmicamente. A maioria dos *sites* que chamam os motores de busca são, na verdade, uma "interface" (front end) para os sistemas de busca de outras empresas.

Java

Um avanço significativo da web foi a plataforma Java, desenvolvida pela Sun Microsystems. Ela permite que páginas web incrustem pequenos programas (chamados *applets*) directamente dentro da informação enviada que será rodada no computador do usuário. Esses *applets* são executados na própria máquina cliente, fornecendo uma experiência mais rica para o usuário. Essa tecnologia nunca ganhou a popularidade que a Sun esperava, por uma variedade de razões, incluindo falta de integração com outros conteúdos e o facto de que a JVM (máquina virtual necessária para a execução do conteúdo) ter que ser instalada antes do uso. Actualmente, o Adobe Flash realiza várias das funções originalmente visadas aos *applets* Java, como apresentação de vídeo, animação e interfaces gráficas ricas.

Javascript

O JavaScript é uma linguagem de computador interpretada desenvolvida originalmente para uso em páginas web, cuja versão

padronizada é ECMAScript. Ainda que seu nome seja similar ao da linguagem Java, o JavaScript foi desenvolvido pela Netscape e não possui semelhanças com o Java. Em conjunto com a tecnologia de Document Object Model, o JavaScript tornou-se um método bastante poderoso de manipulação de páginas web.

Unidade Temática 1.2: Aplicação Web

Em computação, aplicação web designa, de forma geral, sistemas de informática projectados para utilização através de um navegador, através da internet ou aplicativos desenvolvidos utilizando tecnologias web HTML, JavaScript e CSS. Pode ser executado a partir de um servidor HTTP (Web Host) ou localmente, no dispositivo do usuário.

Uma aplicação web também é definida em tudo que se é processado em algum servidor, exemplo: quando você entra em um e-commerce a página que você acessa antes de vir até seu navegador é processada em um computador ligado a internet que retorna o processamento das regras de negócio nele contido. Por isso se chama aplicação e não simplesmente *site* web.

A função do servidor web é receber uma solicitação (requisição) e devolver (resposta) algo para o cliente. O *browser* permite ao usuário solicitar um recurso e quando o servidor responde a uma solicitação são encontrados recursos como: páginas HTML, figuras e documento PDF que são exibidas depois para o usuário. Geralmente os servidores enviam instruções para o *browser* escritas em HTML. O HTML diz ao *browser* como apresentar conteúdo ao usuário web.

O servidor em si tem alguns recursos, mas por algumas deficiências não consegue processar tudo sozinho como: criações de páginas dinâmicas e o armazenamento de dados em um banco de dados.

Páginas dinâmicas – Quando a aplicação roda no servidor, este disponibiliza somente páginas estáticas. Porém, para efectuar essa comunicação é necessário o auxílio de uma outra aplicação de ajuda que é passada através de *Servlet*.

Armazenar dados no servidor – Para efectuar essa ação o servidor

precisa de uma aplicação de apoio (*Servlet*), fazendo com que o servidor envie esses parâmetros para o *Servlet*.

As falhas de segurança podem surgir em diferentes etapas, tais como: análise de requisito; especificação; Implementação. Os riscos de aplicação na vulnerabilidade de uma empresa pode causar impactos.

O HTTP usa um modelo de solicitações e respostas. Uma solicitação ocorre quando o usuário faz uma solicitação HTTP e o servidor web devolve uma resposta HTTP, sendo que o browser verifica como tratar esse conteúdo. Se a resposta que vem do servidor for uma página HTML, então é inserido na resposta HTTP.

As diferenças entre as solicitações GET e POST são que enquanto o GET anexa dados do formulário no final da *URL* o POST inclui dados do formulário no corpo da solicitação.



Fig. 1.3. Aplicações Web Móveis**Unidade Temática 1.3: Modelo cliente-servidor**

O modelo cliente-servidor (em inglês *client/server model*), em computação, é uma estrutura de aplicação distribuída que distribui as tarefas e cargas de trabalho entre os fornecedores de um recurso ou serviço, designados como servidores, e os requerentes dos serviços, designados como clientes.

Geralmente os clientes e servidores comunicam através de uma rede de computadores em computadores distintos, mas tanto o cliente quanto o servidor podem residir no mesmo computador.

Um servidor é um host que está executando um ou mais serviços ou programas que compartilham recursos com os clientes. Um cliente não compartilha qualquer de seus recursos, mas solicita um conteúdo ou função do servidor. Os clientes iniciam sessões de comunicação com os servidores que aguardam requisições de entrada.

O modelo cliente-servidor foi desenvolvido na Xerox PARC durante os anos 70. Este modelo é actualmente o predominante nas redes informáticas. Email, a World Wide Web e redes de impressão são exemplos comuns deste modelo.

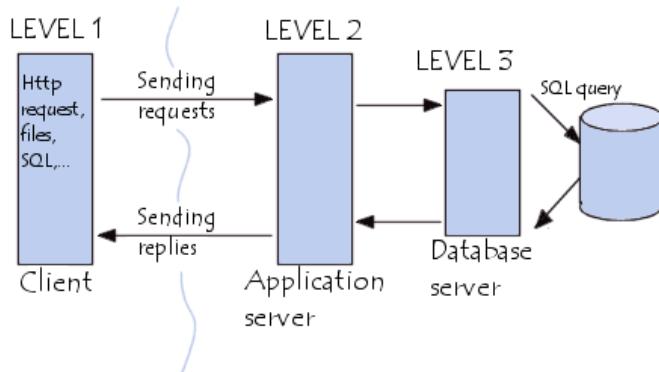


Fig. 1.4. Modelo Cliente-Servidor

A característica do modelo cliente-servidor, descreve a relação de programas numa aplicação. O componente de servidor fornece uma função ou serviço a um ou mais clientes, que iniciam os pedidos de serviço.

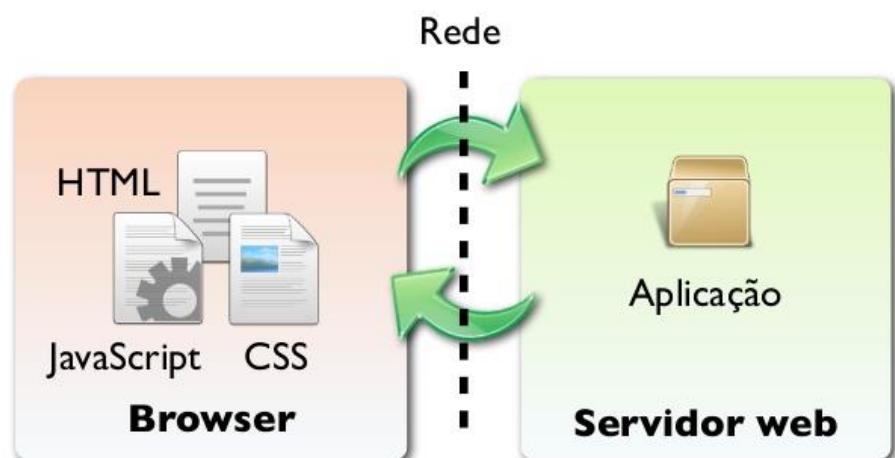
Funcionalidades como a troca de e-mail, acesso à internet ou acesso a um banco de dados, são construídos com base no modelo cliente-servidor. Por exemplo, um navegador web é um programa cliente, em execução no computador do usuário, que acede às informações armazenadas num servidor web na internet. Usuários de serviços bancários, acedendo do seu computador, usam um cliente web para enviar uma solicitação para um servidor web num banco. Esse programa pode, por sua vez, encaminhar o pedido para o seu próprio programa de banco de dados do cliente que envia uma solicitação para um servidor de banco de dados noutro computador do banco para recuperar as informações da conta. O saldo é devolvido ao cliente de banco de dados do banco, que por sua vez, serve de volta ao cliente navegador exibindo os resultados para o usuário.

O modelo cliente-servidor, tornou-se uma das ideias centrais de computação de rede. Muitos aplicativos de negócios, escritos hoje, utilizam o modelo cliente-servidor. O termo também tem sido utilizado para distinguir a computação distribuída por computadores dispersos da "computação" monolítica centralizada em mainframe.

Cada instância de software do cliente pode enviar requisições a vários servidores. Por sua vez, os servidores podem aceitar esses pedidos, processá-los e retornar as informações

solicitadas para o cliente. Embora este conceito possa ser aplicado por uma variedade de razões e para diversos tipos de aplicações, a arquitectura permanece fundamentalmente a mesma.

Arquitectura Cliente/Servidor



37

Fig. 1.5. Arquitectura Modelo-Servidor

Características do cliente:

- Inicia pedidos para servidores;
- Espera por respostas;
- Receber respostas;
- Conecta-se a um pequeno número de servidores de uma só vez;

Normalmente interage directamente com os servidores através de seu *software* aplicação específico, que lhe possibilita a comunicação com o servidor;

Utiliza recursos da rede.

Características do servidor:

Sempre espera por um pedido de um cliente;

Atende os pedidos e, em seguida, responde aos clientes com os dados solicitados;

Podem se conectar com outros servidores para atender uma solicitação específica do cliente; jamais podem se comunicar.

Fornece recursos de rede.

Normalmente interage directamente com os usuários finais através de qualquer interface com o usuário;

Estrutura o sistema.

Unidade Temática 1.4: Desenvolvimento Web

Desenvolvimento web é o termo utilizado para descrever o desenvolvimento de *sites*, na Internet ou numa intranet. Este é o profissional que trabalha desenvolvendo *websites*, podendo ser um *Web Designer* (Desenvolvedor do Layout), ou *Web Developer* (Desenvolvedor de sistemas). O desenvolvimento refere-se a um processo de construção e testes do *software* específico para a web, com a finalidade de se obter um conjunto de programas, que satisfazem as funções pretendidas, quer em termos de usabilidade dos usuários ou compatibilidade com outros programas existentes. O desenvolvimento

web pode variar desde simples páginas estáticas a aplicações ricas, comércios electrónicos ou redes sociais.

Plataformas para o desenvolvimento Web

Codificação no cliente (Front-end)

CSS

HTML

XHTML

Javascript

AJAX

Flash

Microsoft Silverlight

Swipty

spdropkit

Codificação no servidor (Back-end)

PHP

ASP

.NET

Node.js (JavaScript)

Perl (via CGI, FastCGI e, recentemente, PSGI)

Java, J2EE, WebObjects

SSJS, Aptana Jaxer, Mozilla Rhino

Python, Django

Ruby, Ruby on Rails

Smalltalk Seaside

ColdFusion

Lotus Domino

Websphere



Bancos de dados

MySQL

PostgreSQL

SQLite

Microsoft SQL Server

Firebird

Apache Derby

Oracle

DB2

Fig. 1.6. Aplicações Web na Internet

Exercícios de AUTO-AVALIAÇÃO

1. Como funciona a Internet e a *World Wide Web*?

A Internet é uma rede que interconecta computadores e outros dispositivos como o seu celular em escala global para a transferência de dados entre eles. Já a *World Wide Web* é uma aplicação onde páginas são interligadas através de *links* e que se utiliza da Internet para funcionar.

2. Como foi criado o www?

A *World Wide Web* completa 20 anos, conheça como ela surgiu. Você provavelmente já ouviu falar naquela história de que a Internet nasceu nos Estados Unidos durante a década de 1960, em meio à Guerra Fria. ... Quando o meio mais democrático da actualidade surgiu sequer se chamava Internet. Seu primeiro nome foi ARPANET.

3. Quem é o criador da *World Wide Web*?

Tim Berners-Lee
Robert Cailliau

4. O que é e para que serve o www?

Significando literalmente a teia (de aranha) mundial, normalmente chamada de Web e mundialmente conhecido por WWW, o *World Wide Web* é um sistema de documentos públicos colocados e disponibilizados na Internet, no formato de hipertexto acessível a partir de um programa instalado no computador chamado de browser.

5. O que é e como funciona a Internet?

A Internet é uma rede de computadores mundial de acesso de público ilimitado que utiliza uma infra-estrutura de telecomunicações homogénea. A comunicação entre computadores da rede é feita através do protocolo TCP/IP que gerencia e encaminha mensagens de dados entre os computadores.

6. O que quer dizer *World Wide Web*?

World Wide Web significa em português rede de alcance mundial, também conhecida como Web ou WWW. *World Wide Web* é um sistema de documentos em hipermeídia que são interligados e executados na Internet.

7. Para que serve a *World Wide Web*?

A *World Wide Web* (que em português significa “Rede de Alcance Mundial”, também conhecida como Web e WWW) é um sistema de documentos em hipermeídia que são interligados e executados na Internet. Os documentos podem estar na forma de vídeos, sons, hipertextos e figuras.

8. Como funciona um servidor web?

Servidor web é um *software* responsável por aceitar pedidos em HTTP de clientes, geralmente os navegadores, e servi-los com respostas em HTTP, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos em HTML com objectos embutidos (imagens, etc.) ou um computador que executa um programa que provê a funcionalidade descrita anteriormente. O mais popular, e mais utilizado no mundo, é o servidor Apache (*software* livre). A Microsoft possui a sua própria solução denominada *IIS* (*Internet Information Services*).

9. Quando e como surgiu o HTML?

O HTML foi criado em 1991, por Tim Berners-Lee, no CERN (*European Council for Nuclear Research*) na suíça. Inicialmente o HTML foi projectado para interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade.

10. Quem foi o verdadeiro criador da internet?

Foi no início da década de 80, em 1983, que os pesquisadores começaram a montar redes que se tornaram a internet que conhecemos hoje. O mundo *online* assumiu a forma que tem actualmente em 1990, quando o cientista da informática Tim Berners-Lee inventou a *World Wide Web* (www).

11. Quais as hashtags mais usadas?Quais são as hashtags mais usadas no Instagram?

- #love: 1,221B.
- #instagood: 704M.
- #photooftheday: 478,6M.
- #fashion: 456,5M.
- #beautiful: 445,0M.
- #happy: 413,8M.
- #cute: 404,3M.
- #tbt: 401,4M.

12. O que é um provedor e como funciona?

A internet é, portanto, uma rede de redes compartilhadas. O acesso à internet de modo externo se dá quando a sua rede local se conecta a outra rede maior, no caso, o seu provedor de internet, por meio da tecnologia TCP/IP, um modo de comunicação baseado no endereço de *IP (Internet Protocol)*.

13. Qual é o significado de HTTP?

O que é HTTP: HTTP é sigla de *HyperText Transfer Protocol* que em português significa "Protocolo de Transferência de Hipertexto". É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na *World Wide Web (Internet)*.

14. Quem foi o inventor da tecnologia?

1860 - O inventor belga Jean-Joseph-Etienne Lenoir desenvolve o primeiro motor a explosão. 1876 - O americano Alexander Graham Bell inventa o telefone, possibilitando a comunicação entre pessoas situadas a longas distâncias. 1879 - O americano Thomas Alva Edison inventa a lâmpada elétrica.

15. Qual a função do servidor web Apache?

Trata-se de um servidor web muito popular, utilizado principalmente no Linux. A Apache Software Foundation é a responsável pelo projecto, além de desenvolver e trabalhar com outras tecnologias de transmissão via web, processamento de dados e execução de aplicativos distribuídos.

16. Quem foi o criador da Internet Explorer?

Internet Explorer (anteriormente Microsoft Internet Explorer e Windows Internet Explorer, e comumente abreviado como IE ou MSIE) é uma série de navegadores web gráficos desenvolvidos pela Microsoft e inclusos como parte da linha de sistemas operacionais Microsoft Windows, iniciado em 1995.

17. O que é o motor de busca?

Motor de pesquisa ou ferramenta de busca ou buscador (em inglês: *search engine*) é um programa desenhado para procurar palavras-chave fornecidas pelo utilizador em documentos e bases de dados.

18. Quais são os principais sites de busca?

Confira abaixo a lista completa, publicada pelo Business Insider, com os maiores *sites* do mundo:

Facebook.
Google.com.
YouTube.com.
Yahoo.com.
Wikipedia.org.
Live.com.
QQ.com.
Microsoft.com.

19. Como se usa a *hashtag* no Twitter?

Inclua uma *hashtag* existente no seu tweet. Você pode fazer isso simplesmente digitando a frase no formato "# tópico" no seu tweet. Depois de clicar em "Tweetar", o seu tweet aparecerá na sua lista de tweets com a hashtag em azul. Role com o mouse até a *hashtag* e clique nela para ir à página com aquela *hashtag*.

20. Como é a internet via rádio?

Para converter o sinal de rádio em internet, a antena é conectada a um *modem*. Este equipamento, por sua vez, transforma o sinal e o

transmite para a placa de rede do computador. Para fornecer internet via rádio, o provedor deve possuir várias torres, que são distribuídas pela localidade em que ele vai operar o serviço.

21. Onde começa a internet?

Origem da Internet, nesse momento, surge o protótipo da primeira rede de internet, a "Arpanet" (*Advanced Research Projects Agency Network*). Assim, no dia 29 de Outubro de 1969 foi estabelecida a primeira conexão entre a Universidade da Califórnia e o Instituto de Pesquisa de Stanford.

22. O que é o https e para que serve?

Acrônimo de *Hyper Text Transfer Protocol Secure*, o HTTPS, que também pode ser chamado de protocolo de transferência de hipertexto seguro, é uma implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL/TLS.

23. Qual a diferença de http para https?

Hypertext Transfer Protocol Secure, ou simplesmente HTTPS, é uma versão idêntica do protocolo HTTP sobre uma camada SSL. Essa camada adicional permite que os dados sejam transmitidos através de uma conexão criptografada e que se verifique a autenticidade do servidor e do cliente através de certificados digitais.

24. Quando se iniciou a tecnologia?

Desta forma, foi desenvolvendo e inventando instrumentos tecnológicos com o objectivo de superar dificuldades. Podemos dizer que a necessidade é a mãe das grandes invenções tecnológicas. Acompanhe abaixo um histórico da evolução tecnológica:

1291 - Na Itália surgem os primeiros espelhos.

1440 - Foi criada a tecnologia da impressão por GUTENBERG.

1454 - O alemão Johann Gutenberg inventa a máquina chamada de

Imprensa. Com está máquina o homem passou a produzir de forma mais rápida e eficiente, os livros. Esse invento causou uma revolução na cultura da época.

1590 - O holandês Zacharias Janssen (1580-1638?) fabrica o microscópio, utilizando técnicas usadas na fabricação de lentes para óculos.

1592 - O astrônomo e inventor italiano cria o primeiro termômetro utilizando o sistema de água para a medição de temperatura.

1643 - O cientista italiano Evangelista Torricelli inventa o barômetro para medir a pressão atmosférica.

1707 - O físico inglês John Floyer inventa o relógio de pulso.

1712 - O engenheiro inglês Thomas Newcomen inventa a máquina a vapor.

1800 - O físico italiano Alessandro Volta cria a bateria elétrica.

1822 - Foi produzida a primeira imagem reconhecida como fotografia pelo francês JOSEPH NICÉPHORE NIÉPCE.

1839 - O artista e pesquisador francês Louis-Jacques-Mandé Daguerre tira a primeira fotografia, com sua máquina chamada daguerreótipo

1860 - O inventor belga Jean-Joseph-Etienne Lenoir desenvolve o primeiro motor a explosão.

1874 - Foi inventado o rádio.

1876 - O americano Alexander Graham Bell inventa o telefone, possibilitando a comunicação entre pessoas situadas a longas distâncias.

1879 - O americano Thomas Alva Edison inventa a lâmpada elétrica.

1895 - Foi criado o cinematógrafo pelos irmãos LUMIÉRE.

1901 - É criado o rádio pelo italiano Guglielmo Marconi.

1903 - Os irmãos Wright pilotam o primeiro avião.

1904 - Criadas pelo engenheiro inglês John Ambrose Fleming surgem as válvulas eletrônicas.

1906 - O brasileiro Alberto Santos Dumont voa em paris no 14 bis e passa também a ser considerado um dos pais da aviação junto com os irmãos Wright.

1941 - O engenheiro inglês Frank Whittle desenvolve o avião a jacto.

1943 - A empresa japonesa Motorola lança no mercado o walkie-talkie.

1945 - Os EUA detonam no deserto do Novo México a primeira bomba atômica.

1946 - O engenheiro americano Vannevar Bush desenvolve um computador usando válvulas de rádio.

1947 - A televisão começa a chegar nos lares de pessoas de todo o mundo.

1948 - Começam a ser utilizados os chips de silício e as válvulas eletrônicas.

1956 -O pager é lançado nos Estados Unidos.

1961 - Lançada a Vostok, a primeira nave espacial tripulado por ser humano a sair da atmosfera terrestre.

1965 - Lançados os primeiros satélites de comunicação. Inaugura uma nova era na transmissão de dados eletrônicos.

1972 - Os discos laser são lançados revolucionando a indústria fonográfica.

1977 - Lançado nos Estados Unidos o primeiro telefone celular.
1981 - Primeira viagem de um ônibus espacial.

1995 - Dave Wineland e Chris Monroe desenvolvem o primeiro transistor do tamanho de um átomo.

1998 - Lançado no Brasil os primeiros DVDs.

1999 - A Internet cresce no mundo todo em velocidade impressionante. Os arquivos de MP3 começam a ser usados e transmitidos pelas ondas da rádio.

25. Quando surgiu a tecnologia Digital?

Tecnologia digital é um conjunto de tecnologias que permite, principalmente, a transformação de qualquer linguagem ou dado em números, isto é, em zeros e uns (0 e 1). As tecnologias digitais surgiram no século XX e revolucionaram a indústria, a economia, a sociedade.

26. Como foi chamado a 1ª versão da internet?

Internet Explorer 1.0 é um navegador web gráfico que foi lançado em 16 de agosto de 1995 pela Microsoft, sendo a primeira versão do navegador Internet Explorer. É uma versão remodulada do Spyglass Mosaic que a Microsoft licenciou da Spyglass, Inc.

27. Em que ano foi criada a primeira página de internet?

Criada por Tim Berners-Lee usando um computador NeXT, a World Wide Web começou a existir em 1991, e há mais de 20 anos de sua criação, o primeiro *site* do mundo está armazenado pela W3C, que o conserva para que seja parte da história da internet.

28. Qual foi a primeira tecnologia do mundo?

O foguete de combustível sólido foi inventado na China por volta de 1150, cerca de 200 anos após a invenção da pólvora (que era o seu principal combustível) e 500 anos após a invenção dos fósforos.

29. O que é o método HTTP GET?

O HTTP define um conjunto de métodos de solicitação para indicar a acção desejada a ser executada para um determinado recurso. Embora eles também possam ser substantivos, esses métodos de solicitação são às vezes chamados de verbos HTTP. O método GET solicita uma representação do recurso especificado. Solicitações usando GET só devem recuperar dados.

30. O que é o método HTTP POST?

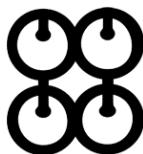
Na computação, o POST é um método de solicitação suportado pelo HTTP usado pela World Wide Web. Por design, o método de solicitação POST solicita que um servidor da Web aceite os dados contidos no corpo da mensagem de solicitação, muito provavelmente para armazená-los. É frequentemente usado ao fazer upload de um arquivo ou ao enviar um formulário da web completo.

TEMA 2. UTILIZAÇÃO DE REDES SOCIAIS PARA PARTILHA DE INFORMAÇÃO

Rede social é uma estrutura social composta por pessoas ou organizações, conectadas por um ou vários tipos de relações, que compartilham valores e objectivos comuns. Uma das fundamentais características na definição das redes é a sua abertura, possibilitando relacionamentos horizontais e não

hierárquicos entre os participantes. "Redes não são, portanto, apenas uma outra forma de estrutura, mas quase uma não estrutura, no sentido de que parte de sua força está na habilidade de se fazer e desfazer rapidamente."

Muito embora um dos princípios da rede seja sua abertura e porosidade, por ser uma ligação social, a conexão fundamental entre as pessoas se dá através da identidade.



"Os limites das redes não são limites de separação, mas limites de identidade. (...) Não é um limite físico, mas um limite de expectativas, de confiança e lealdade, o qual é permanentemente mantido e renegociado pela rede de comunicações."

Esta unidade vai ajudá-lo a conhecer as melhores teorias e as melhores práticas de utilização das redes sociais para uma partilha de informação ou exposição estável e segura. Os prós e contras da utilização das redes sociais para a partilha de informação.

Quais os objectivos com maiores vantagens e desvantagens no uso das redes sociais? Qual o impacto político que o devido e/ou indevido uso das redes sociais pode ter? Quais as desvantagens de uma utilização não resguardada das redes sociais?

Objectivos Específicos

Unidade Temática 2.1: Redes sociais *online*

As redes sociais *online* podem operar em diferentes níveis, como, por exemplo, redes de relacionamentos (Facebook, Twitter, Instagram, Google+, Youtube, MySpace, Badoo), redes profissionais (Linkedin), redes comunitárias (redes sociais em bairros ou cidades), redes políticas, redes militares, dentre outras, e permitem analisar a forma como as organizações desenvolvem a sua actividade, como os indivíduos alcançam os seus objectivos ou medir o capital social – o valor que os indivíduos obtêm da rede social.

As redes sociais têm adquirido importância crescente na sociedade moderna. São caracterizadas primariamente pela autogeração de seu desenho, pela sua horizontalidade e sua descentralização.

Um ponto em comum dentre os diversos tipos de rede social é o compartilhamento de informações, conhecimentos, interesses e esforços em busca de objectivos comuns. A intensificação da formação das redes sociais, nesse sentido, reflecte um processo de fortalecimento da Sociedade Civil, em um contexto de maior participação democrática e mobilização social.

Com as diversas redes sociais, os consumidores estão em contacto constante com as marcas. Este fato resulta da revolução tecnológica que, com tablets e smartphones, coloca o mundo nas mãos do consumidor, e tem, como consequência, uma revolução ao nível do *marketing* e da forma como as empresas se comunicam com os consumidores. Devido a este fenómeno, as marcas, nas mais diversas áreas de negócio, estão a perceber-se de que as técnicas de marketing tradicionais estão a tornar-se cada vez menos eficazes e mais dispendiosas, isto porque a segmentação no marketing "tradicional" é muito menos eficaz e a medição do impacto não é imediata, o que acontece com o marketing digital e, principalmente, com o marketing das redes sociais.

As marcas já não conseguem controlar nem a sua comunicação nem o que se diz sobre elas nas redes sociais. O consumidor assumiu o controle e é participativo em todo o processo de comunicação, chegando mesmo ao ponto de as marcas terem que apresentar conteúdos relevantes, pertinentes e adequados aos desejos e necessidades dos consumidores de forma a terem "permissão" para falar com eles. Este fenómeno está directamente ligado a um conceito

emergente no *marketing* e comunicação: *earned media* ("mídia conquistada").

Unidade Temática 2.2: Marketing nas redes sociais

As redes sociais criaram uma nova forma de exploração das dinâmicas de relacionamento entre as empresas e os seus clientes, o marketing nas redes sociais é mais pessoal e interactivo do que qualquer outro meio, no entanto é também necessário ter em atenção de que estas novas ferramentas que surgem todos os dias são cada vez mais complexas e de difícil entendimento para o utilizador mais comum. Uma gestão profissional de marketing nas redes sociais exige o estudo e conhecimento de outras áreas como por exemplo as comunidades virtuais e as conexões. O marketing e a comunicação atravessam, portanto, hoje, um caminho que se pauta pela existência de diversificados canais de comunicação, com duas vias, através dos quais as marcas dialogam com os seus consumidores e vice-versa.

As redes sociais *online* apresentam-se actualmente como uma plataforma de interacção com clientes e potenciais clientes, promovendo a fidelização. Permitem ainda que esta mesma interacção seja em tempo real uma vez que estando estas Redes ligadas 24 horas/dia, 365 dias/ano as empresas e os seus utilizadores estão conectados quase que automaticamente, sabendo sempre a opinião de cada um. Por exemplo, a empresa lança um produto novo mas quer saber se o mesmo irá ter ou não sucesso no mercado, poderá efectuar um questionário aos seus seguidores e após tratamento das respostas sabe se será aceite ou não consoante está ou terá que ser alterado. Isto tudo em tempo real ou numa questão de dias, enquanto que através de outros sistemas antigos isto era tudo muito mais complicado e demorado. As redes sociais permitem mesmo à empresa uma espécie de estudo de mercado com um custo baixíssimo.

Unidade Temática 2.3: Formas de redes sociais

As redes sociais costumam reunir uma motivação comum, porém podem se manifestar de diferentes formas. As principais são:

Redes comunitárias: estabelecidas em bairros ou cidades, em geral tendo a finalidade de reunir os interesses comuns dos habitantes, melhorar a situação do local ou prover outros benefícios. Redes profissionais: prática conhecida como networking, tal como o LinkedIn, que procura fortalecer a rede de contactos de um indivíduo, visando futuros ganhos pessoais ou profissionais. Redes sociais *online* tais como Facebook, WhatsApp, VK, Google+, MySpace, Twitter, Badoo WorldPlatform (normalmente estamos acostumados a redes sociais públicas, mas existem privadas. Normalmente, existem estágios de tempo em cada rede social até que se torne pública) que é um serviço online, plataforma ou site que foca em construir e reflectir redes sociais ou relações sociais entre pessoas, que, por exemplo, compartilham interesses e/ou actividades, bate-papo, jogar com os amigos, entre outras funções.

Como já dito acima, existem redes sociais públicas, em que o registo está desbloqueado para todos. As privadas podem pedir o endereço electrónico e só depois de uma resposta é que o registo fica disponível, nesse tipo de rede nem sempre são aceitos todos os tipos de pessoas. Existem ainda as redes sociais pessoais, para família ou amigos, pouco conhecidas na Internet.

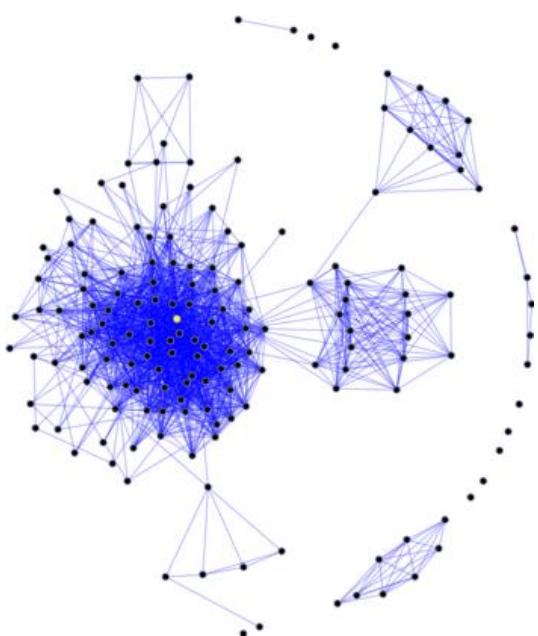


Fig. 2.1. Redes Sociais Online

Unidade Temática 2.4: Análise das redes sociais

A análise de redes sociais surgiu como uma técnica chave na sociologia moderna. O conceito surgiu na sociologia e na antropologia social. No final do século XX, o termo passou a ser olhado como um novo paradigma das ciências sociais, vindo ser aplicada e desenvolvida no âmbito de disciplinas tão diversas como a antropologia, a biologia, os estudos de comunicação, a economia, a geografia, as ciências da informação, a psicologia social, a sociolinguística e, sobretudo, no serviço social.

A ideia de rede social começou a ser usada há cerca de um século atrás,

para designar um conjunto complexo de relações entre membros de um sistema social a diferentes dimensões, desde a interpessoal à internacional.

Em 1954, J. A. Barnes começou a usar o termo sistematicamente para mostrar os filhos da pátria, incorporando os conceitos tradicionalmente usados quer pela sociedade quer pelos cientistas sociais: grupos bem definidos (exemplo: tribos, famílias) e categorias sociais (exemplo: género, grupo étnico).

Académicos como S.D. Berkowitz, Stephen Borgatti, Ronald Burt, Kathleen Carley, Martin Everett, Katherine Faust, Linton Freeman, Mark Granovetter, David Knoke, David Krackhardt, Peter Marsden, Nicholas Mullins, Anatol Rapoport, Stanley Wasserman, Barry Wellman, Douglas R. White ou Harrison White expandiram e difundiram o uso sistemático da análise de redes sociais.

Em teoria, na estrutura das redes sociais, os atores sociais se caracterizam mais pelas suas relações do que pelos seus atributos (género, idade, classe social). Estas relações têm uma densidade variável, a distância que separa dois atores é maior ou menor e alguns atores podem ocupar posições mais centrais que outros. Este fenómeno é explicado por alguns teóricos apontando a existência de laços fortes e fracos e a dos buracos estruturais onde se encontram os actores que não podem comunicar entre si a não ser por intermédio dum terceiro.

No estudo da estrutura das redes sociais, é necessário incluir as relações de parentesco de seus membros, redes sociométricas, capital social, redes de apoio, de mobilização, interconexões entre empresas e redes de escola.

É composta por três elementos básicos:

Nós ou atores

Vínculos

Fluxos de informação (unidireccional ou bidimensional)

Exercícios de AUTO-AVALIAÇÃO

- 1. Como as redes sociais afectam minha privacidade?**

- 2. Como as redes sociais afectam meu tempo?**

- 3. Como as redes sociais afectam minha reputação?**

- 4. Como as redes sociais afectam minha escolha de amizades?**

5. Em quais redes sociais devo estar presente?

6. O que é melhor postar?

7. Como devo interagir?

8. Por que devo contactar uma agência se eu mesmo faço?

9. Como avaliar o resultado?

10. Com que frequência devo postar?

11. Posso utilizar uma ferramenta para programar as postagens?

Exercícios para AVALIAÇÃO

12. Em quais plataformas de mídia social a minha marca deve estar presente?

Primeiro, não há dúvidas de que sua empresa deve estar nas mídias sociais, independentemente do sector. O objectivo por trás de ter uma presença na mídia social, no entanto, pode diferir de uma empresa para outra. Como marca, você precisa saber onde está o seu público e começar de lá. Transmitir mensagens para a turma errada terminará em muita insatisfação para todas as partes envolvidas.

Tenha em mente que alguns canais estão focados no B2B, enquanto outros adoptam uma abordagem B2C. A decisão depende do produto ou serviço que você está fornecendo, bem como do seu público.

Com isso fora do caminho, o mínimo inclui uma conta no Facebook, Twitter e Linkedin. Os dois primeiros destinam-se a actualizações de B2C mais descontraídas, enquanto o segundo é usado principalmente para iniciativas de RH e relacionamentos B2B. A partir daqui, você pode expandir para o Instagram, Snapchat, Pinterest, Google+ ou outros, dependendo de quem você quer abordar e com que finalidade.

13. O que devo postar?

A resposta a essa pergunta depende novamente do que sua empresa faz e de sua demografia. Cada plataforma de mídia social é atraente à sua maneira, mas usa o tipo errado de conteúdo no público errado e você será rotulado como não profissional

Todas as plataformas suportam conteúdo multimídia incorporável e descrições de texto, mas em algumas delas também é aceitável postar actualizações somente de texto. O Facebook e o Twitter também oferecem suporte a vídeos ao vivo, algo que você deve ter em mente caso sua marca tenha uma forte identidade visual. O Snapchat e o Instagram também colocam muito foco no conteúdo visual, mas vale a pena considerar essas duas plataformas apenas se você pegar a estrada B2C. Dado o principal objectivo do LinkedIn, actualizações não relacionadas à vida profissional serão desaprovadas.

Para ter uma ideia melhor do que funciona em cada plataforma e por quê, dê uma olhada em um de nossos eBooks, *The Psychology of Social Media Marketing*.

14. Está tudo bem publicar o mesmo conteúdo em cada plataforma todos os dias?

Ou devemos publicar actualizações diferentes em *sites* diferentes? Em suma: não. As pessoas seguem sua marca em diferentes plataformas por diferentes motivos. Isso significa que você não pode esperar ter exactamente o mesmo público em todas as plataformas. Dito isto, unificar o público é algo que você pode conseguir através da narrativa.

Já discutimos isso em nossa

comparação do Snapchat vs. Instagram. As marcas que se concentram em conteúdo visual não precisam necessariamente depender apenas de uma das duas plataformas. Em vez disso, eles poderiam começar uma história publicando o “making of” no Snapchat e as fotos mais refinadas no Instagram.

15. Como obtenho mais seguidores?

Aqui é onde a diferença entre qualidade e quantidade se torna óbvia. Postar actualizações frequentes onde você está pedindo para as pessoas seguirem você ou trazer mais seguidores obviamente não funcionará. A compra de seguidores pode parecer uma solução atraente, mas não é viável. Os seguidores pagos não são particularmente conhecidos por se envolver com conteúdo e nem são leais. Você pode impulsionar o envolvimento fornecendo conteúdo de alta qualidade para as pessoas certas no momento certo. Conteúdo criativo e exclusivo será mais do que isso. Ele também pode determinar seus seguidores para compartilhá-lo com seus familiares, amigos ou contactos profissionais. Se isso se transformar em uma bola de neve que você não pode parar de rolar (não que você queira), seu conteúdo se tornou viral. Leia mais sobre como usar 5 ingredientes para engajar campanhas de marketing de conteúdo visual.

16. Como lidar com comentários negativos nas mídias sociais?

Algumas marcas têm esse equívoco comum de que a mídia social é um lugar cheio de negatividade. Mesmo que fosse assim, é melhor saber que existe um lugar onde os clientes expressam todo tipo de opinião, em vez de ficarem inconscientes. Quando se trata de lidar com esse tipo de comentário, você só tem duas opções.

- Em primeiro lugar, você pode optar por abordar as preocupações de seus clientes em uma tentativa de recuperar sua confiança. Tenha em mente que, em alguns casos, não haverá retenção se você não estiver fazendo isso. Mesmo os clientes satisfeitos começarão a questionar seus padrões de atendimento ao cliente se perceberem essas abordagens.
- Em segundo lugar, você pode recorrer às soluções de moderação de conteúdo fornecidas pelas plataformas de mídia social. Facebook e Instagram são apenas duas das redes sociais que oferecem opções de filtragem de comentários.

17. Qual é a frequência certa se as pessoas continuarem perdendo as coisas que postamos nas mídias sociais?

A melhor frequência é realmente algo que você precisa descobrir sozinho. Algo que você precisa levar em conta é o quanto activo o seu público é. Depois disso, você precisa encontrar um equilíbrio com o qual o público esteja familiarizado. Você definitivamente não quer que eles começem a deixar de segui-lo depois de inundar seus newsfeeds. Nem você quer que eles perguntam a si mesmos se você caiu da face da Terra. Existem algumas diretrizes, mas você só pode encontrar a melhor frequência experimentando.

18. A publicidade social paga é uma boa idéia?

Se você está tendo dificuldades para aumentar o reconhecimento de seus produtos ou serviços, a publicidade em mídias sociais pode ser a melhor opção que você tem. Você não deve confiar nessa abordagem toda vez que publicar uma actualização, mas aumentar as actualizações de vez em quando pode ter um grande impacto. Pesquisa realizada pela eMarketer no final de 2015 concluiu que 76% dos entrevistados B2C confiaram em posts promovidos e apenas 61% desses usuários acreditavam que os anúncios pagos são eficazes. Dos 74% dos entrevistados afirmaram usar o LinkedIn, Facebook e Twitter por seus esforços. Além disso, 59% dos entrevistados classificaram essas três redes com 4 ou 5 pontos em 5 pontos para eficácia. As coisas não parecem tão boas no departamento B2B, onde menos de 50% dos profissionais de marketing social acreditam que os anúncios pagos são eficazes. Para garantir que você está jogando dinheiro fora da janela, verifique se a plataforma escolhida é aquela em que seu público-alvo ideal passa a maior parte do tempo.

19. Como medir o ROI nas mídias sociais?

Depois de selecionar as plataformas corretas e planejar as campanhas, você deve decidir quais métricas verificar para determinar o ROI. Em uma de nossas postagens sobre esse tópico, discutimos o alcance, o tráfego, o engajamento e as conversões nas mídias sociais e por que cada um deles é importante para medir o retorno do investimento.

Social Media ROI = $(\text{Retorno da mídia social} - \text{custo do marketing de mídia social}) / \text{custo do marketing de mídia social}$

Usando a fórmula acima, você pode determinar como os resultados se comparam aos objectivos iniciais. É melhor analisar o maior número possível de métricas para garantir que as alterações sejam pequenas.

20. Posso levar uma estratégia social de uma plataforma para outra?

O conteúdo geralmente determina o tipo de estratégia que você precisa adoptar. À medida que diferentes plataformas de mídia social se concentram em diferentes tipos de conteúdo, as chances de levar uma estratégia de um para o outro são pequenas. Há conteúdo específico da plataforma e, depois, há restrições específicas da plataforma. Por muito tempo, o Twitter teve o limite de 140 caracteres, e a única maneira de transmitir mais foi incorporar uma imagem. Facebook, LinkedIn e Google+ são bem mais liberais nessa perspectiva. Mesmo que você decida publicar conteúdo nessas três plataformas, é necessário lembrar que as audiências podem ser diferentes.

21. Quais erros comuns de marketing de mídia social o meu negócio deve evitar?

Spamming, ter vários perfis em cada plataforma e não interagir com o público são algumas das práticas que você deve evitar. Algumas dessas acções causam confusão, enquanto outras determinam seu público para deixar de seguir e / ou mesmo denunciar você. Inconsistência é uma das principais armadilhas em que você pode cair. Isso pode

significar começar em uma plataforma e abandoná-la apenas para tentar outra. Isso também pode significar publicar actualizações de maneira frenética. Colocar alguma lógica em como, quando e onde você postar as postagens de mídia social pode estreitar o relacionamento com seu público. Se não fosse por qualquer outra coisa, seu público poderia saber quando esperar suas actualizações. Uma plataforma de gerenciamento de mídia social pode ajudar você a se tornar mais consistente. Ao mesmo tempo, pode ajudar você a acompanhar o engajamento e outras métricas relevantes. Experimente o Swat.io e veja como ele atende às suas necessidades.

TEMA 3. FERRAMENTAS DE VÍDEO PARA FÓRUNS VISUAIS

Esta unidade vai ajudar o estudante a conhecer as principais ferramentas para elaboração de vídeos *online* a serem partilhados em fóruns de estudo, de discussão ou de partilha de conteúdos *online*.

Esses vídeos podem ser de interacção com o utilizador, pelo que as ferramentas que serão aqui apresentadas vão guiar o estudante na produção dos seus próprios vídeos com ou sem interacção como forma de enriquecerem e distinguirem as vossas apresentações.

Objectivos Específicos



Unidade Temática 3.1: Os 5 melhores editores de vídeo on-line para edição de vídeo on-line e os 5 melhores editores de imagens

Com a criação de redes sociais de compartilhamento de vídeos como YouTube, Vimeo, Instagram e Vine, editar e fazer vídeos deixou de ser de domínio exclusivo de profissionais. Tudo que você precisa agora é um bom sistema de computador e bingo você é um editor de vídeo. Na verdade, esta não é a única opção disponível. Você também pode editar seus vídeos para o conteúdo do seu coração sem baixar e instalar qualquer *software* - tudo que você precisa é de uma conexão decente com a internet. Existem vários *sites* que permitem que você faça o upload de suas imagens, imagens de vídeo brutais e sons, e ajuste-os ao seu conteúdo, depois publique-o onde quiser (ou não, se assim decidir). Vamos falar sobre as 10 soluções de edição de vídeo mais recomendadas que você pode acessar *online* se tiver uma conexão de internet decente.

Se você precisa de opções de edição mais avançadas e belos efeitos de vídeo, a animação *online* não é a melhor escolha. Aqui nós recomendamos o Wondershare Filmora 9 para a ferramenta que você não pode perder. Basta baixá-lo para tentar explorar os recursos para iniciantes e profissionais.

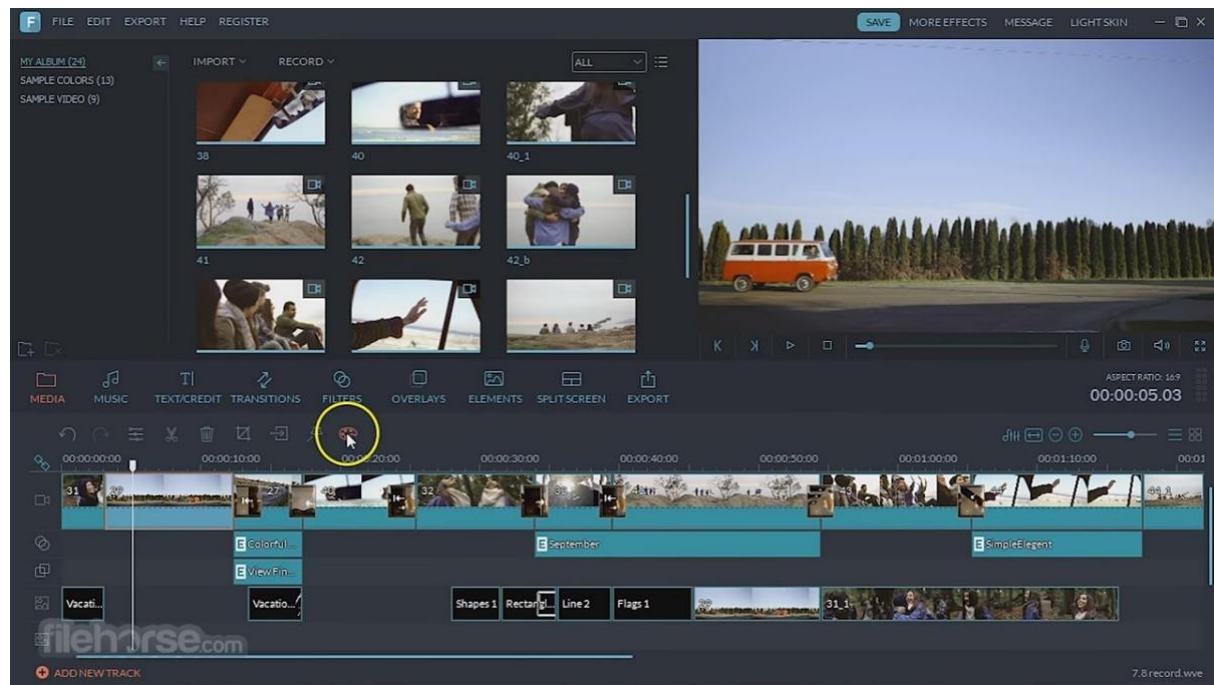


Fig. 3.1. Ambiente de Trabalho do Wondershare Filmora 9

O software de edição de vídeo doméstico mais fácil de usar, com recursos especiais de edição, permite unir cliques de vídeo, músicas e textos, aplicar efeitos e obter um filme caseiro de aparência profissional em minutos. Todas as funções clássicas de edição de vídeo, como cortar, recortar, reduzir, cortar, dividir, bem como efeitos de vídeo impressionantes (PIP, Tela Verde, Face-off, etc) são fornecidos. Até 100 faixas de áudio e vídeo para usar.

Unidade Temática 3.2: Os 5 melhores editores de vídeo online**Youtube Video Editor (Editor de Vídeo Youtube)**

É uma das ferramentas de edição de vídeo mais utilizadas na Internet. Com o Editor de vídeos do YouTube, você pode compartilhar facilmente seus projectos editados com milhões de espectadores conectados ao canal do YouTube. Esta ferramenta de edição de vídeo fácil de usar não apenas corta cliques de vídeo em qualquer tamanho necessário, mas ao mesmo tempo pode combinar vários cliques de vídeo, fotos ou cliques de áudio em um único arquivo para criar um projeto personalizado. Ele tem o recurso Panorâmico e Zoom, além de poder ajustar as sobreposições de texto em cliques de vídeo, além de ajudar os usuários a atualizar o contraste e o nível de brilho dos vídeos.

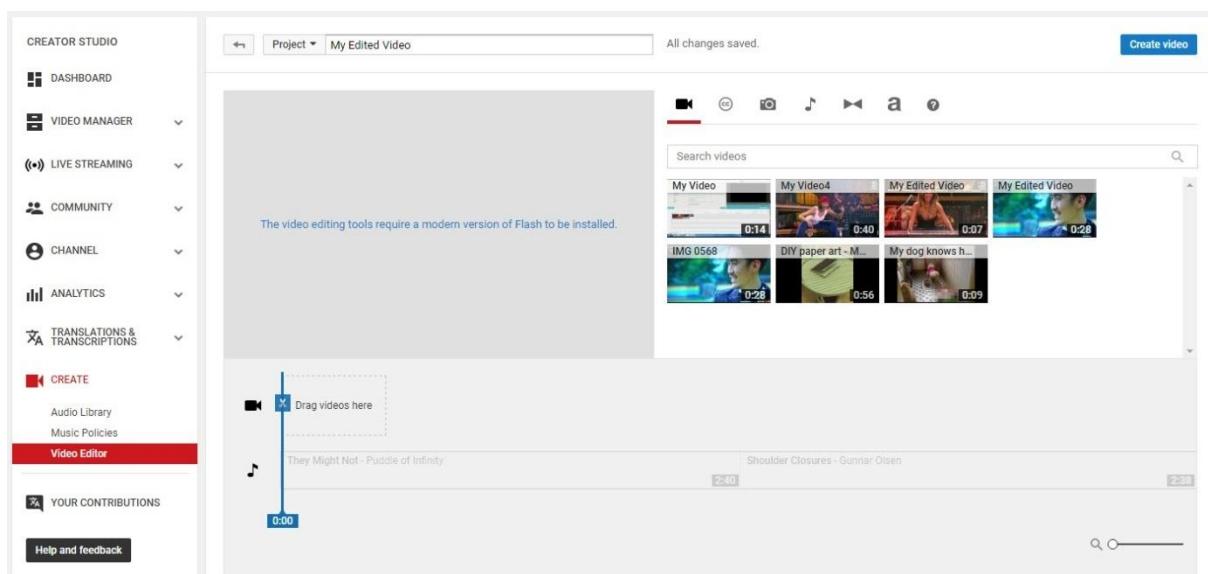


Fig. 3.2. Ambiente de Trabalho do Youtube Video Editor

WeVideo

WeVideo é uma ferramenta de criação de vídeos on-line compartilhada. Isso significa que você pode ter várias pessoas trabalhando em um projecto a qualquer momento. Há muita diversão no ambiente colaborativo promovido no WeVideo. Outra grande coisa sobre o WeVideo é o aplicativo do Google Drive. Você não precisa se preocupar em ocupar espaço no seu computador para salvar projectos incompletos; O WeVideo Google Drive permite que você salve todos os seus projectos, completos ou incompletos, directamente na sua conta do Google Drive. O aplicativo para Android também permite que você capture imagens brutais de seus projectos directamente de seu telefone.



Fig. 3.3. Ambiente de Trabalho do WeVideo

PowToon

O PowToon é uma óptima maneira de criar vídeos que terão uma finalidade explicativa. Se você precisa de um vídeo que sirva como um guia ou professor, então o PowToon é definitivamente o site que você quer usar. Ele permite que você use o que é essencialmente uma versão digital de recortes de papel sobrepostos em fundos coloridos. A edição de estilo arrastar e soltar deixa você com um vídeo que lembra o estilo Common Craft. Você pode optar por seleccionar um dos muitos desenhos de pessoas e objectos do PowToon, que você pode organizar em uma tela em branco ou em um fundo colorido de sua escolha. A única dificuldade em usar o PowToon é que você só pode criar ou editar 45 segundos de vídeo a qualquer momento, a menos que você opte por comprar a versão completa.



Fig. 3.4. Ambiente de Trabalho do PowToon

Wideo

O Wideo é outro óptimo *site* para criar vídeos regulares, vídeos animados e outros vídeos comuns do estilo Craft. Usando os elementos arrastar e soltar do serviço de edição do *site*, você cria uma sequência de animações. O Wideo permite que você use os diferentes elementos quantas vezes quiser para criar o seu vídeo e, ao contrário de alguns *sites*, você não está limitado a usar elementos de estoque (texto, desenhos animados, desenhos e imagens) e componentes. Você é livre para enviar suas próprias imagens, vídeos e arquivos de áudio.



Fig. 3.5. Ambiente de Trabalho do Wideo

ClipChamp

É muito mais fácil obter os melhores resultados para as necessidades de gravação, conversão e compactação de vídeo usando a ferramenta de *software* Clip Champ. A melhor parte é que sua plataforma de trabalho intuitiva permite que os iniciantes concluam sua edição de vídeo em muito menos tempo e a saída pode ser compartilhada directamente em *sites* de mídia social populares como Vídeo, Facebook, YouTube e Google Drive, etc. para cortar, cortar, girar e virar vídeos conforme sua necessidade. Além disso, também é possível ajustar o brilho, a saturação e o contraste com facilidade.

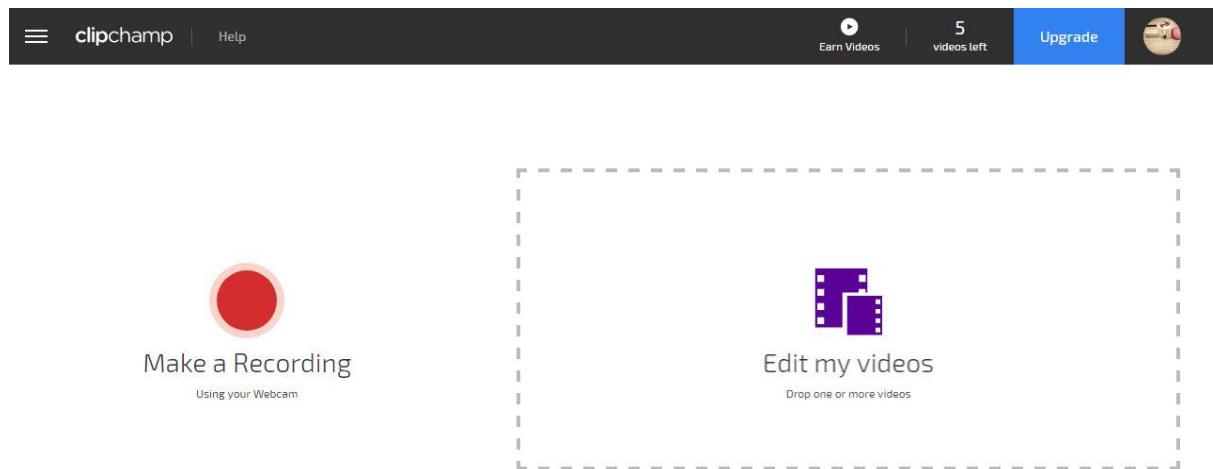


Fig. 3.6. Ambiente de Trabalho do ClipChamp

Unidade Temática 3.3: Os 5 melhores editores de imagens

Como você escolhe o melhor *software* de edição de fotos? Escolhemos os programas de edição de fotos mais bem pagos que funcionam tanto no Mac quanto no PC, e procuramos facilidade de uso, qualidade de resultados, versatilidade ou capacidade de processamento de *pixels*. Também existem editores de fotos gratuitos, móveis ou *online*, mas os manteremos para outro guia.

A resposta para a melhor pergunta do editor de fotos costumava ser fácil - obter o Photoshop. No entanto, ficou claro que os fotógrafos e entusiastas querem mais do que o Photoshop pode oferecer. Ele não cataloga sua biblioteca de fotos em crescimento e não fornece idéias e inspiração para alimentar sua visão criativa. Basicamente, é apenas uma grande caixa de chaves.

Então, vamos começar com o pequeno ecossistema complexo que é o Plano de Fotografia baseado em assinatura da Adobe, e depois analisar alternativas que ofereçam melhor qualidade, melhor organização, melhor inspiração ou simplesmente um melhor valor. Eles não estão em nenhuma ordem específica, pois cada programa tem seus pontos fortes, portanto, continue até o final da lista, pois há algo para todos.

Adobe Photoshop CC

É o aplicativo de edição de fotos mais poderoso do mundo e pode haver ocasiões em que nada mais funcionará.

O Photoshop ainda é a ferramenta de edição de imagens para artistas, ilustradores e designers, mas os fotógrafos têm um grupo diferente de necessidades que podem ser melhor atendidas nos dias de hoje por uma ferramenta de catalogação / aprimoramento como o Lightroom ou uma ferramenta de efeitos como o Alien Skin Exposure X3. . As ferramentas de criação de camadas, mascaramento e retoque do Photoshop ainda são o padrão pelo qual todas as outras são avaliadas, mas são projectadas para trabalhos minuciosos em imagens únicas ou composições com várias camadas, em vez de edições rápidas no dia-a-dia. A única maneira de obter o Photoshop agora é por meio de uma assinatura da Adobe. O plano de fotografia regular é o melhor valor e também permite que você use o Lightroom CC e o Lightroom Classic.

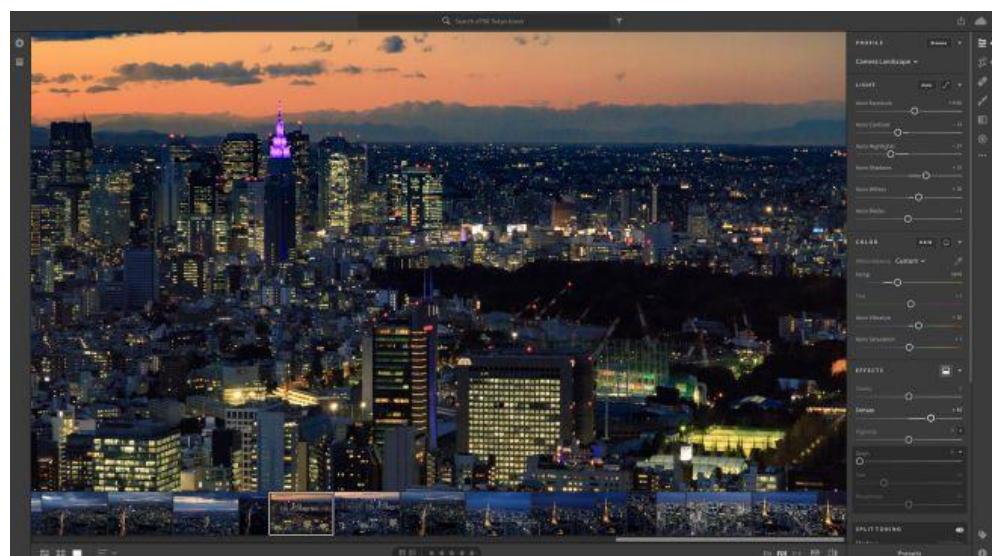


Fig. 3.7. Ambiente de Trabalho do Adobe Photoshop CC

Adobe Photoshop Lightroom CC

As coisas ficaram confusas. O "novo" Lightroom é uma ferramenta despojada baseada em armazenamento em nuvem.

Onde o Photoshop é uma manipulação detalhada, o Lightroom se concentra na organização de imagens e aprimoramentos regulares de fotos. Agora, porém, existem duas versões. O "antigo" foi renomeado como Lightroom Classic (veja abaixo), enquanto o "novo" Lightroom CC oferece uma interface simplificada e armazenamento em nuvem integrado. Você pode obter o Lightroom CC e o armazenamento de 1 GB pelo mesmo preço do Plano de fotografia normal, mas não obtém o Photoshop, o que é uma desvantagem significativa. O Lightroom CC é super elegante de usar, mas está faltando algumas ferramentas no Lightroom Classic e não oferece suporte a *plug-ins* e editores de imagem externos, exceto o Photoshop. Se você quiser o armazenamento Lightroom e Photoshop AND 1GB, o plano custa o dobro...



Fig. 3.8. Ambiente de Trabalho do Adobe Photoshop Lightroom CC

Adobe Photoshop Lightroom Classic CC

É o "antigo" Lightroom com um novo nome e se adere ao armazenamento de imagens baseado em desktop.

O Lightroom e o Photoshop são o duplo ato perfeito. Um cuida de organizar e melhorar suas fotos enquanto o outro lida com qualquer manipulação de imagem baseada em camadas mais complexa. O Lightroom Classic é a antiga versão "completa" do Lightroom. Parece um pouco mais pesado e complicado do que o CC Lightroom baseado em nuvem, mas é mais poderoso e suporta plug-ins. Você obtém o Photoshop e o Lightroom CC / Classic como parte do Plano de Fotografia baseado em assinatura da Adobe e, para ser honesto, essa combinação é de bom valor e leva um certo tempo. Para muitos, no entanto, a ideia de pagar uma assinatura para usar software é demais para engolir, e é por isso que vamos passar rapidamente para o restante da nossa lista.



Fig. 3.9. Ambiente de Trabalho do Adobe Photoshop Lightroom Classic CC

Phase One Capture One Pro 12

Caro, mas bonito, o Capture One é um concorrente directo do Lightroom e lançado com firmeza em profissionais.

O Capture One abrange quase exactamente o mesmo território do Adobe Lightroom Classic, oferecendo ferramentas de catalogação, processamento bruto contínuo, ferramentas de aprimoramento manual de imagens juntamente com efeitos predefinidos e um fluxo de trabalho não destrutivo que permite revisitar seus ajustes a qualquer momento. Suas conversões brutas são mais nítidas e menos barulhentas do que as da Adobe, mas não são compatíveis com uma ampla variedade de formatos de câmera raw ou com um grande número de perfis de correcção de lentes. Ele também não possui aplicativos para dispositivos móveis e opções de sincronização on-line da Adobe, mas oferece ferramentas de tethering de nível profissional para fotógrafos de estúdio que capturam imagens por meio de um computador. Também possui um sistema melhor para aplicar ajustes locais, usando camadas de ajuste e máscaras. É caro, mas muito, muito bom.

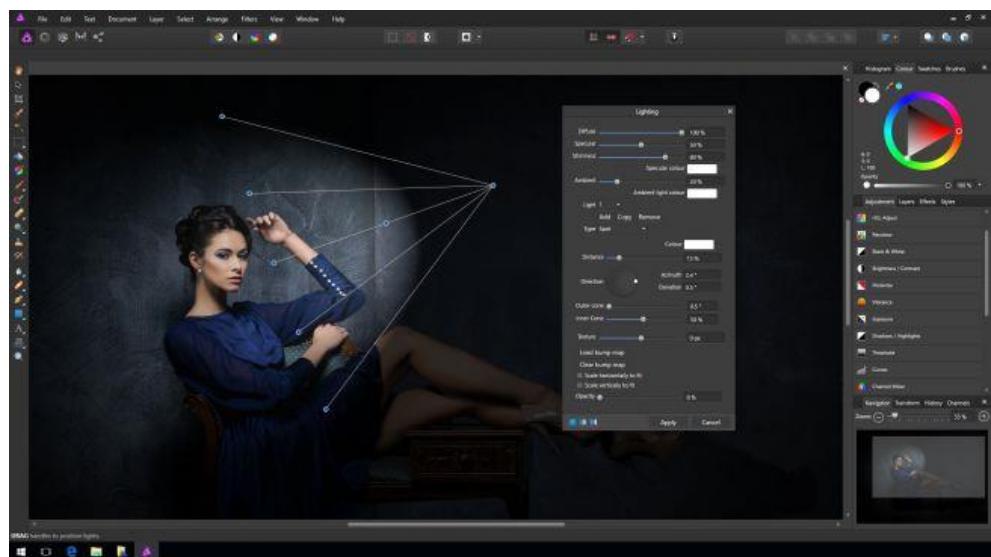
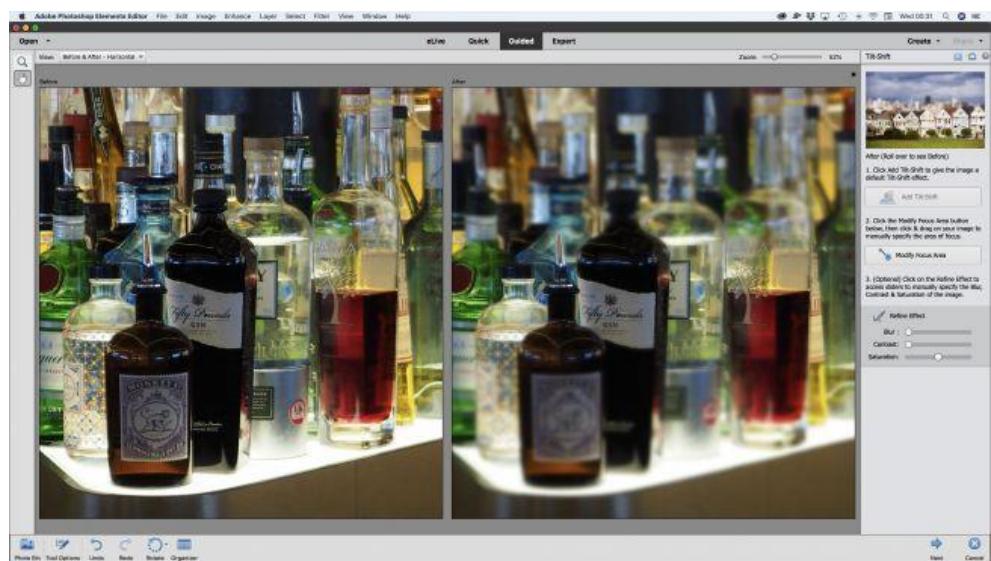


Fig. 3.10. Ambiente de Trabalho do Phase One Capture One Pro 12**Serif Affinity Photo 1.6**

Se você deseja o Photoshop, mas não deseja o plano de assinatura da Adobe, esta é a resposta!

A Serif construiu sua reputação com base nas versões de baixo custo das ferramentas gráficas profissionais do Windows, mas com sua nova linha Affinity, ela se livrou do orçamento passado para sempre. O Affinity Photo pode ter um preço acessível, mas é um rival completo e completo do Photoshop para profissionais, que pode até ensinar um equivalente ou dois ao equivalente da Adobe. Suas ferramentas de camadas, mascaramento e retoque são tão poderosas quanto os do Photoshop, seus efeitos de filtro podem ser aplicados "ao vivo" e seu mapeamento de tons HDR e as ferramentas de espaço de trabalho são excelentes. Como o Photoshop, porém, ele é focado apenas na manipulação técnica detalhada de imagens. Não possui suas próprias ferramentas de navegação e catalogação e não apresenta efeitos predefinidos instantâneos. A Affinity Photo trará as ferramentas, mas você terá que trazer a visão.

**Fig. 3.11. Ambiente de Trabalho do Serif Affinity Photo 1.6**

Exercícios de AUTO-AVALIAÇÃO

1. Como remover a marca d'água, uma limitação da versão teste do Wondershare Filmora?

Por favor tenha em atenção que a marca d'água é uma limitação da versão teste. Você precisa comprar e registar o programa, carregar o arquivo de projecto e exportá-lo novamente. Para conferir todos os detalhes, por favor siga os passos em baixo:

- Após comprar o programa, você irá receber um *email* com o código de registo e o email da licença. De seguida clique aqui e siga as instruções para registar o programa. Caso você não receba o código de registo, pode recuperá-lo no *link* em baixo.

<http://support.wondershare.com/pt-br/contact-support/retrieve-registration-code.html>

- Após registar o programa, você pode fazer duplo-clique directamente no arquivo de projecto criado pela versão teste. Ou você pode clicar no ícone do programa no canto superior esquerdo e seleccionar **Abrir ou Abrir Recente**, para carregar o **arquivo de projeto (.wve)**., agora exporte o arquivo e a marca d'água desaparecerá.

2. Como registar o Wondershare Filmora?

Para registar o programa, você pode acessar o **Modo Fácil** ou o **Modo Completo de Recursos**, e inserir seu **código de registo e email da licença**.

Após você seleccionar o Modo Fácil para acessar o programa, por favor seleccione a **Ajuda** por baixo do **Menu**, e clique na opção **Registar...**. De seguida **copie e cole** o código de registo e o *email* da licença para registar o programa. Por favor certifique-se de inserir correctamente seu código de registo e o *Email* da licença.

Em alternativa, você pode seleccionar o Modo Completo de Recursos para acessar o programa, e clicar na opção **Registar...**no Menu no canto superior direito. De seguida **copie e cole** o código de registo e o *email* da licença para activar o programa.

3. Como faço para activar o meu Filmora 9.0?

Você pode usar a mesma conta do Wondershare ID para activar esta nova versão 9.0. Ou pode usar o mesmo código para activar a versão através da activação *offline*.

4. Como eu compartilho e baixo os pacotes gratuitos?

Há dois meios para conseguir os pacotes de efeitos gratuitos. Você pode obtê-los pelo website oficial da Loja de Efeitos , ou você pode fazer isso dentro do Filmora. Para evitar problemas de compatibilidade com tipos diferentes de navegadores, é altamente sugerível baixar os pacotes gratuitos através do Filmora. Estes são os passos para a sua referência:

Inicie o Filmora, seleccione o modo de recursos completos.

Seleccione **MAIS EFEITOS** no canto superior direito da janela de prévia.

Seleccione **Entrar** e faça o login na Loja de Efeitos do Filmora. Se você for um usuário novo, por favor seleccione **Cadastrar Uma Nova Conta** e registe uma conta.

Desça até o fim da página, aperte a letra "f" . Ou você pode clicar na miniatura Irá abrir uma nova janela, por favor escolha **Compartilhar Download Grátis**.

Faça o login na sua conta do Facebook, poste suas palavras, seleccione Publicar no Facebook para finalizar o compartilhamento.

O download irá começar automaticamente.

Depois da instalação, todo o conteúdo do pacote será incluído no Filmora.

5. Como eu baixo todos os efeitos embutidos

Existem três meios de baixar os efeitos embutidos.

Encontre os efeitos/modelos com a seta azul para baixo no canto superior direito, clique duas vezes neles para começar o *download*.

Ou você pode clicar no ícone "+" em cada modelo para começar a baixar. Deste modo, abrirá uma janela para lhe mostrar o progresso.

Para lhe poupar tempo, você pode clicar aqui para baixar o instalador completo de todos os efeitos embutidos directamente.

6. Como fazer Uso dos efeitos do Filmora

Há dois tipos de efeitos fornecidos pelo Filmora:

1. Quanto aos efeitos embutidos encontrados dentro do Filmora, incluindo faixas musicais, transições, textos e títulos, efeitos sonoros, etc, todas as faixas musicais são para uso pessoal enquanto os outros podem ser usados para uso comercial apenas se você comprar uma licença comercial.
2. Quanto aos Efeitos complementares vendidos na loja de Efeitos (<https://filmora.wondershare.com/pt-br/effects-store/>), já que todos eles são feitos pelos *designers* da Wondershare e têm todos os direitos reservados à Wondershare, você pode usá-los para uso comercial directamente após a sua compra de uma licença comercial.

7. Quais as características de funcionamento e desempenho melhorados no Serif Affinity? Um dos melhores softwares de edição de imagens.

Panorâmica e *zoom* ainda mais suaves;

Compatibilidade para o *plug-in* do Photoshop melhorada;

Renderização mais rápida de filtros dinâmicos;

Melhorias de exportação em PDF, incluindo a exportação vectorial de gradientes com várias marcas;

8. Enumere um conjunto de ferramentas rápidas e dinâmicas no Serif Affinity.

Aplique panorâmica e *zoom* a 60 fps e superior se suportado, mesmo com imagens enormes;

Ferramentas dinâmicas, visualizações dinâmicas e edição em tempo real;

Processamentos RAW e regular juntos em um aplicativo;

9. Enumere ferramentas profissionais no Serif Affinity.

Espaços de cor e perfis profissionais;

Formatos de arquivo profissionais como PSD suportados;

Formatos RAW de câmera modernos suportados;

Conjunto completo de filtros de 16 bits;

Precisão do subpixel e ajuste perfeito do *pixel*;

Barras de ferramentas personalizáveis, atalhos de teclado e painéis;

10. Como limpar espaços de trabalho no Serif Affinity?

Personas (modos) dedicadas a tarefas e conjuntos de ferramentas específicos;

IUs bem organizadas para a edição RAW, edição de fotografia, panoramas, dissoluções e mapeamento de tons;

Exportar a persona para um controle rigoroso das áreas e configurações de exportação;

11. Idiomas do *software* de edição de imagens Serif Affinity?

Inglês (EUA e RU);

Alemão;

Francês;

Espanhol;

Português (Brasileiro);

Japonês;

Italiano;

Chinês (Simplificado);

Russo.

12. Como você sobrepõe textos no WeVideo?

Você pode facilmente adicionar vários gráficos de texto à sua edição no WeVideo. Seleccione a guia de texto e você pode clicar em cada gráfico de texto para visualizá-los. Depois de encontrar um que você goste, adicioná-lo à sua edição é tão simples quanto arrastá-lo e soltá-lo em uma trilha de vídeo.

13. O que é o WeVideo?

A WeVideo é uma empresa de *Software* como um Serviço, com sede nos EUA que fornece uma plataforma de edição de vídeo colaborativa baseada na Web que funciona em qualquer navegador.

14. O 4k Video Downloader é seguro?

A resposta é: 4K Video Downloader é seguro de usar, desde que seja baixado de seu *site* oficial 4kdownload.com. No entanto, se seu 4kvideodownloader.exe vem de sites de "download de EXE" de terceiros, não podemos dar qualquer garantia de que não há vírus, *trojans*, *malware*, *spyware* ou outros.

15. Como você exclui um clipe no aplicativo WeVideo?

Clique no clipe que você deseja remover excluir da linha do tempo. Você verá quatro ícones de acção. Clique no ícone da lixeira. Você também pode clicar e arrastar um clipe em direcção à janela de visualização para remover um clipe rapidamente.

16. O que é o tempo de publicação no WeVideo?

Tempo de Publicação é o subsídio de vídeos que você pode terminar por mês, da maneira que desejar. Por exemplo, se sua conta tiver 2 horas por mês, isso permitirá que você termine 20 vídeos em 6 minutos cada, ou 12 vídeos em 10 minutos cada, etc. O tempo de publicação é consumido toda vez que um usuário inicia o processo de finalização.

17. Onde posso baixar filmes completos de graça?

Dê um passeio por esta lista dos principais *sites* gratuitos de download de filmes, onde você pode baixar filmes completos sem ter que se registar para uma conta.

- Megabox HD (Android App) ...
- YouTube (apenas nos EUA) ...
- Hotstar (Índia) ...
- Sony Crackle. ...
- PopCornFlix.
- O arquivo da Internet.
- Vimeo.
- Retrovisão.

18. Como você acelera um vídeo no WeVideo?

Seleccione o clipe que você pretende acelerar ou desacelerar e seleccione editar. Navegue até a guia Velocidade.

1. Ajuste o final do intervalo para seleccionar o horário de início e o horário de término do efeito.
2. Mova o valor nob para a velocidade desejada

19. Como faço para criar uma marca d'água?

Para fazer isso, abra a janela Exportar e seleccione "Editar marcas d'água" no menu suspenso na área Opções de marca d'água. Acima, no canto superior direito, seleccione a opção de estilo 'Graphic' watermark. Localize o logotipo que você deseja definir como marca d'água do Lightroom. Sob Watermark Effects você tem opções de opacidade, tamanho e posição.

20. Como você salva um vídeo em seu computador?

Download de um vídeo do YouTube para o seu computador

1. Primeiro Passo: Seleccione o vídeo do YouTube. Encontre o vídeo desejado no YouTube e copie seu URL.
2. Passo Dois: Use o KeepVid.com. Abra o seu navegador e vá para KeepVid.com. Cole o URL do YouTube no campo URL e seleccione o botão "DOWNLOAD" (acima dos ícones de *download* e instalação grandes).

21. Você pode baixar vídeos do WeVideo?

Baixando o projecto de vídeo para um computador. Assim que o seu vídeo é carregado para o WeVideo, pode encontrá-lo no separador Exportações no topo de cada página do WeVideo. Sob esta guia, cada projecto tem uma seta que você pode clicar para ver um menu suspenso com opções para assistir, editar, renomear, fazer *download* ou excluir o projecto.

22. Como você adiciona músicas ao WeVideo?

Para adicionar uma música ou faixa de áudio ao seu vídeo, abra o aplicativo de upload. Você pode fazer isso clicando no botão de upload dentro da galeria de mídia no editor ou clicando no botão verde no canto superior direito do painel. Em seguida, seleccione de onde você quer extrair sua música.

23. Como você importa vídeos para o WeVideo?

Para importar mídia no editor, clique no ícone de importação e seleccione os clipes em seu computador, clicando em "Procurar para seleccionar" na janela de importação. Depois de seleccionar seus arquivos, eles aparecerão na biblioteca e poderão ser usados imediatamente.

24. O WeVideo salva automaticamente?

Salvando seu projeto. WeVideo salva automaticamente o seu projeto enquanto você está trabalhando nele. Se você editar o projeto e tentar ir para uma área diferente do WeVideo, um pop-up perguntará se você deseja salvar as alterações feitas no seu projeto.

25. Quanto custa o WeVideo?

Ao inscrever-se, você concorda com os Termos de Uso e a Política de Privacidade do WeVideo. Planos de destaque premium começam em apenas US \$ 4,99 por mês.

TEMA 4. FERRAMENTAS PARA PARTILHA DE DOCUMENTOS (DROPBOX, WETRANSFER, GOOGLE DRIVE)

Esta unidade vai dar a conhecer as ferramentas para partilha de documentos, disponibilizar os mecanismos para a instalação e a utilização na íntegra dos recursos das principais aplicações Web utilizadas para a partilha de documentos na internet.

**Objectivos Específicos**

Unidade Temática 4.1: Dropbox

Dropbox é um serviço para armazenamento e partilha de arquivos. É baseado no conceito de "computação em nuvem" ("cloud computing"). Ele pertence ao Dropbox Inc., sediada em San Francisco, Califórnia, EUA.

A empresa desenvolvedora do programa disponibiliza centrais de computadores que armazenam os arquivos de seus clientes. Uma vez que os arquivos sejam devidamente copiados para os servidores da empresa, passarão a ficar acessíveis a partir de qualquer lugar que tenha acesso à Internet. O princípio é o de manter arquivos sincronizados entre dois ou mais computadores que tenham o aplicativo do Dropbox instalado.

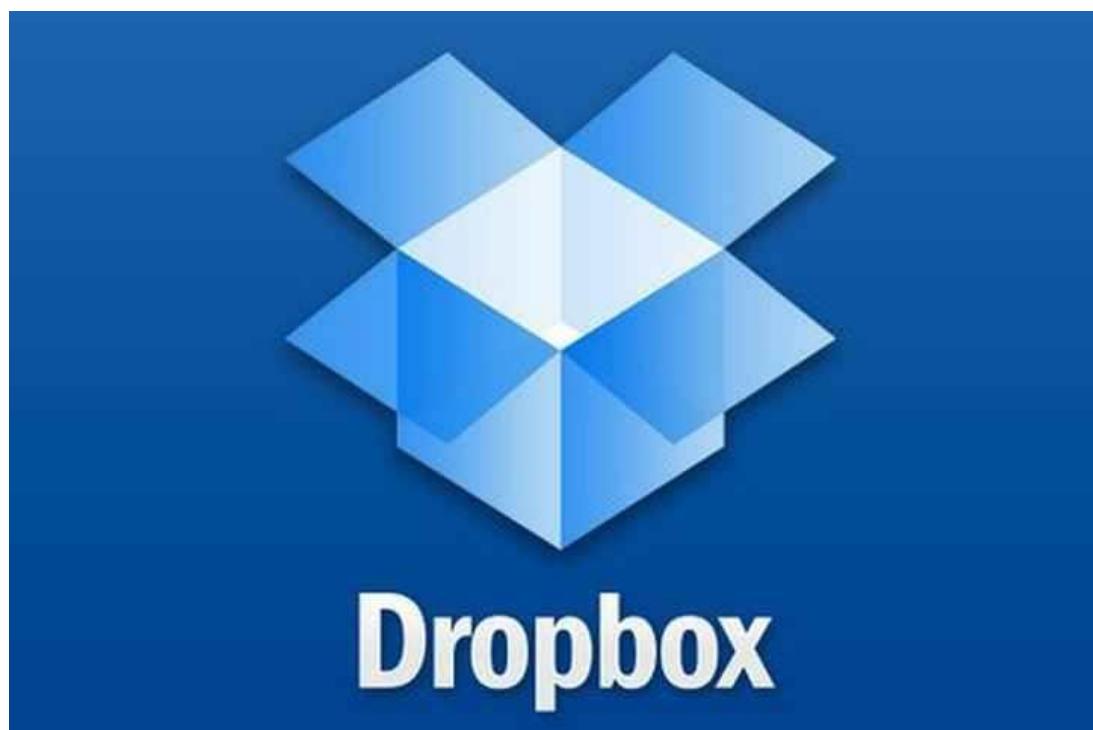


Fig. 4.1. Logotipo ou *Icon* da Empresa e do Software Dropbox

O Dropbox é utilizado por mais de 400 milhões de pessoas no mundo, sendo que mais de 100 mil são clientes empresariais. O Dropbox oferece 2 GB de espaço gratuito na rede para sincronizar e compartilhar arquivos com outros usuários. Em 2014, a Dropbox Inc foi avaliada em US\$ 10 bilhões.

A ideia do Dropbox surgiu quando por acaso um dos seus futuros criadores (Drew Houston) esqueceu seu pendrive quando viajava em um ônibus e precisava dele para uma consultoria que daria em uma empresa. Frustrado com a perda, Drew começou a escrever um código sem nenhuma pretensão, e que futuramente viria se chamar Dropbox. Drew também cita em entrevistas que já estava saturado de ter que ficar abrindo seu *email* a todo instante para anexar arquivos e poder acessá-los remotamente mais tarde. Seu trabalho inicial era apenas para resolver seu problema pessoal, só depois, Houston viu que sua ideia tinha um bom potencial de mercado. Formado no MIT em ciência da computação, Drew deu início a seus trabalhos no projecto, e convidou seu colega Arash Ferdowsi a participar do mesmo. Arash na época ainda não se tinha formado no MIT e interrompeu seu curso para ajudar no projecto. Eles conseguiram um contrato com a empresa de aceleração de crescimento Y Combinator em 2007, se mudaram para o Vale do Silício, para sediar a empresa, e colocaram o produto no ar em 2008.

O sucesso foi imediato. A empresa teve um crescimento muito rápido, e em 2011 já tinha contrato firmado com a multinacional japonesa Sony Ericsson, para que seu *software* fosse pré-instalado em seus dispositivos. Em 2010 o produto foi bloqueado pelo governo Chinês e voltou a ter esses problemas em 2011. Nesse mesmo ano o Dropbox já atingia a marca de mais de 50 milhões de usuários. Mas em 2012 sentiu um impacto da concorrência das gigantes Microsoft, com o SkyDrive, e da Google, com o Google Drive.

O Dropbox já recebeu cerca de US\$ 257 milhões em investimentos de empresas de capital de risco. Um dos investidores ilustres da empresa é o vocalista da banda irlandesa U2, Bono Vox.

O Dropbox é um serviço freemium (o cliente tem a opção de usá-lo gratuitamente, mas pode pagar para obter algumas funções extras) de armazenamento remoto de 2 GB de arquivos e pode ter até 1 TB se optar por um plano pago. Os arquivos

podem ser carregados nos servidores do Dropbox a partir de qualquer dispositivo que possua o seu *software* e conexão com a internet, a partir daí, esses arquivos podem ser acessados de qualquer dispositivo com internet. Além da sincronização dos arquivos, o Dropbox mantém as versões anteriores, permitindo que o cliente retorne a uma versão mais antiga dos seus dados, isso é muito útil no uso de planilhas.

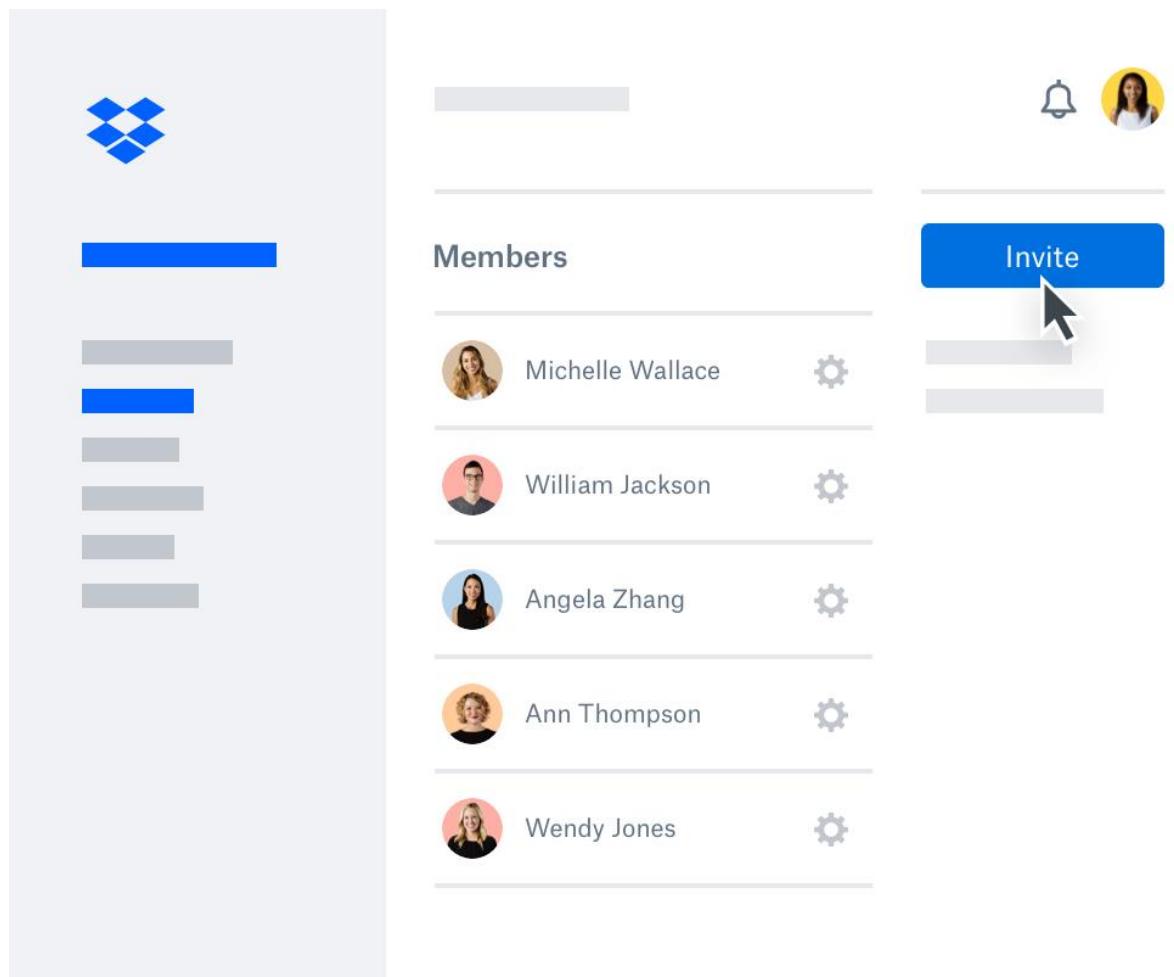


Fig. 4.2. Ambiente de Trabalho do Dropbox

A empresa acredita no modelo de negócio freemium inclusive como uma estratégia de marketing, em entrevista o CEO da empresa afirmou que o crescimento rápido tem muito a ver com os usuários que não pagam, esses chamam outros que possivelmente virão a pagar no futuro, em pesquisa realizada constatou-se que 96% dos clientes possuem contas gratuitas. A diferença das contas gratuitas para as pagas está apenas no espaço disponível para os dados. A empresa lançou também um modelo de marketing onde quem traz amigos para usar o produto ganha mais espaço para uso

Tudo sobre como usar o dropbox, aceda:

<https://www.dropbox.com/help/>

Unidade Temática 4.2: Wetransfer

O WeTransfer é um serviço de transferência de arquivos baseado na nuvem. A empresa foi fundada em Amsterdã (Holanda) por Bas Beerens e Nalden. O serviço principal é gratuito, com mais recursos disponíveis para contas premium. Usuários gratuitos podem enviar arquivos de até 2 GB; o WeTransfer "Plus" pago permite que os usuários enviem até 20 GB por vez e oferece recursos como proteção por senha, personalização de canal e armazenamento de 100 GB. Em Outubro de 2018, a WeTransfer relançou seu aplicativo com o nome "Collect by WeTransfer".

A WeTransfer foi fundada em 2009 e evoluiu a partir da frustração de Beerens por não poder compartilhar facilmente arquivos grandes. WeTransfer tinha que ser "simples o suficiente para seus pais usarem". Nalden, que dirigiu o blog Nalden.net, foi abordado por Beerens e adicionou a ideia de imagens em tela cheia. O WeTransfer foi mais tarde transformado em um negócio autônomo.



Fig. 4.3. Logotipo ou *Icon* da Empresa e do Software WeTransfer

Em 2012, a WeTransfer implementou um novo *design* e introduziu sua oferta premium "Plus", que incluiu transferências de arquivos maiores, além de armazenamento, catálogo de endereços e a capacidade de proteger arquivos por senha

Autofinanciada desde o seu lançamento em 2009, a WeTransfer alcançou a lucratividade em 2013. Em 2014, a WeTransfer lançou o creativeclass.tv, uma série de vídeos em andamento. Nesse mesmo ano, a WeTransfer também iniciou suas primeiras bolsas através de uma parceria colaborativa com a Central Saint Martins. Essas bolsas apoiaram dois estudantes de todo o mundo em seus estudos em período integral.

Em 2015, a WeTransfer arrecadou uma rodada de financiamento da Série A de US \$ 25 milhões da Highland Capital Partners Europe.

Nesse mesmo ano, a empresa acrescentou ao seu conselho Troy Carter, capitalista de risco e gerente de talentos para artistas como Lady Gaga e Meghan Trainor.

Em 2016, a WeTransfer anunciou a aquisição do estúdio de *design* e inovação digital Present Plus. Fundada em 2010 por Damian Bradfield e pelo co-fundador da WeTransfer, Nalden, a Present Plus focou em tornar a tecnologia simples e útil, e criar conteúdo na esperança de que ela fosse compartilhada. Seu trabalho atraiu clientes como a Adidas, a MR PORTER, a British Airways, a Sonos e a Christie's.

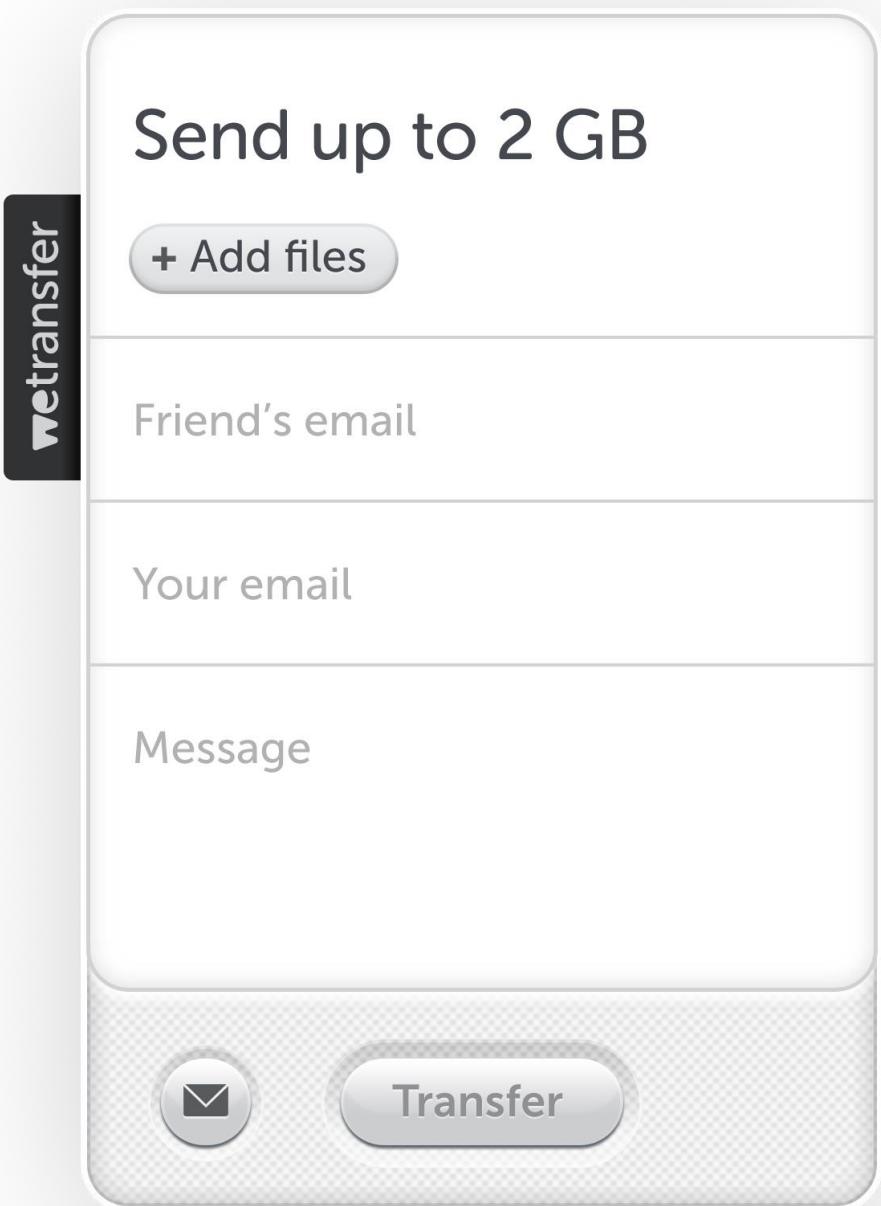


Fig. 4.4. Ambiente de Trabalho do WeTransfer

Em Setembro de 2016, o WeTransfer abriu seu primeiro escritório nos Estados Unidos, em Venice Beach, Los Angeles.

No início de 2017, Gordon Willoughby tornou-se o CEO da empresa, substituindo o fundador e presidente-executivo Bas Beerens, que se afastou das operações do dia-a-dia para se tornar presidente executivo.

Em agosto de 2018, a WeTransfer adquiriu o desenvolvedor de aplicativos FiftyThree, cujo portfólio incluiu o aplicativo de desenho Paper e o aplicativo de apresentação colaborativa Paste.

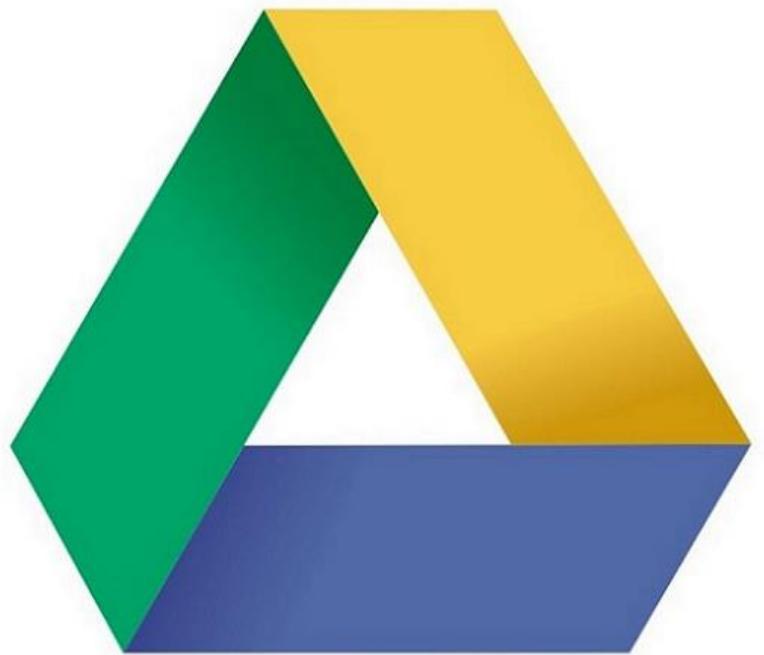
Tudo sobre como usar o WeTransfer, acesa:

<https://wetransfer.com/help>

Unidade Temática 4.3: Google Drive

Google Drive é um serviço de armazenamento e sincronização de arquivos que foi apresentado pela Google em 24 de Abril de 2012. Google Drive abriga agora o Google Docs, um leque de aplicações de produtividade, que oferece a edição de documentos, folhas de cálculo, apresentações, e muito mais. O Google Drive é considerado uma "evolução natural" do Google Docs (uma vez activado substitui a URL docs.google.com por drive.google.com). Rumores sobre o Google Drive começaram a circular no início de Março de 2006. Com o lançamento do Google Drive, o Google aumentou o espaço de armazenamento do Gmail para 15GB.

O Google Drive baseia-se no conceito de computação em nuvem, pois o internauta poderá armazenar arquivos através deste serviço e acedê-los a partir de qualquer computador ou outros dispositivos compatíveis, desde que ligados à internet. Para além disso o Google Drive disponibiliza vários aplicativos via *online*, sem que esses programas estejam instalados no computador da pessoa que os utiliza.



Google Drive

Fig. 4.5. Logotipo ou *Icon* do Software Google Drive

O Google Drive dá ao usuário 15GB grátis de armazenamento no início. Um utilizador consegue espaço extra, que é compartilhado entre Picasa e Google Drive, que vai de 25 GB até 16TB pagando uma quantia mensal (2,49 dólares/mês ou menos de 2,00 euros/mês). Pode ser adquirido através da contratação de um plano de pagamento mensal.

Em agosto de 2018, Google lançou o Google XX para substituir planos não gratuitos do Google Drive.

Há aplicativos de terceiros que podem ser instalados pela Chrome Web Store. Esses aplicativos, rodando no Google Chrome, acessam os arquivos *online*, e podem ser utilizados para editar imagens e vídeos, enviar fax assinar documentos, gerenciar projectos, criar fluxogramas etc. Além da página na internet, o Google Drive necessita de um cliente local para sincronização de arquivos. No lançamento do serviço havia clientes para os seguintes dispositivos: Mac Mountain Lion (10.8), Lion (10.7) e Snow Leopard (10.6); Windows XP, Windows Vista e Windows

7; Android (2.1+); iPhone e iPad (iOS 5.0+). O Google prometeu que haverá suporte ao Linux em breve. De acordo com o engenheiro do Google, Sundar Pichai, o Google Drive será totalmente integrado às próximas versões do Chrome OS.



Fig. 4.6. Modelo ou Arquitectura do Software Google Drive

Tudo sobre como usar o Google Drive, acesa:

<https://support.google.com/drive/?hl=en#topic=14940>

Exercícios de AUTO-AVALIAÇÃO

1. Como criar um mostruário no Dropbox?

Para criar um mostruário:

Faça login no dropbox.com.

Clique em Showcase na barra lateral esquerda.

Seleccione um mostruário existente ou clique em Criar mostruário para criar um novo.

Você também pode fazer uma cópia duplicada de um mostruário existente. Para fazer isso:

Abra o mostruário que você deseja copiar.

Clique em "..." (reticências).

Clique em Fazer uma cópia.

Clique em Fazer uma cópia novamente.

Nota: A cópia duplicada do mostruário terá (Cópia 1) anexada ao título.

2. Como adicionar arquivos e imagens a um mostruário no Dropbox?

Você pode adicionar vários arquivos, do Dropbox ou de outro lugar, a um único mostruário. Por exemplo, um único mostruário pode incluir arquivos de imagem, documentos do Dropbox Paper e planilhas e documentos que não são do Dropbox. Os arquivos serão exibidos como uma visualização com *layout* e aparência personalizáveis. As pessoas com quem você compartilha seu mostruário têm a opção de visualizar ou baixar as versões ".jpg" e ".pdf" dos arquivos que você inclui no seu mostruário, desde que possam ser visualizadas no Dropbox.

Existem três maneiras de adicionar arquivos e imagens a um mostruário:

Clique no botão "+" (mais) no canto superior esquerdo do mostruário e, em seguida, clique em Escolher a partir do Dropbox ou Enviar arquivo;

Role até a barra de ferramentas na parte inferior do mostruário e clique no ícone Escolher do Dropbox ou Enviar arquivo;

Passe o mouse sobre uma secção, clique no sinal "+" (mais) que aparece à esquerda e, em seguida, clique em Escolher do Dropbox ou Enviar arquivo.

3. Como adicionar e editar uma imagem em destaque em uma vitrine no Dropbox?

Você pode adicionar uma imagem em destaque ao seu mostruário no Dropbox que aparecerá no topo. A fonte da imagem em destaque deve ter um mínimo de 350 x 350px e um máximo de 1000 x 350px.

Para adicionar uma imagem em destaque, clique em Adicionar uma imagem em destaque na parte superior do mostruário.

Para alterar uma imagem em destaque, clique uma vez na imagem para destacá-la e, em seguida, clique no ícone Substituir imagem na barra de ferramentas.

Para alterar a posição de uma imagem em destaque, clique uma vez na imagem para destacá-la e escolha um dos ícones na barra de ferramentas.

Para alterar a aparência da imagem (Reduzir, Ampliar ou Preencher), clique uma vez na imagem para destacá-la, clique no ícone Ajustar imagem na barra de ferramentas e escolha um dos ícones.

Observação: se você passar o *mouse* sobre os ícones nas barras de ferramentas, verá uma descrição do que cada um faz.

4. Como adicionar, substituir e reutilizar um logotipo em uma vitrine no Dropbox?

Depois de carregar um logotipo em um mostruário, você poderá reutilizá-lo em exibições futuras. Você também pode fazer o upload de vários logotipos e escolher entre eles. Para adicionar e reutilizar logotipos: Clique em Adicionar um logotipo na parte superior do mostruário. Clique em Novo logotipo para enviar um logotipo ou clique em um logotipo que você enviou anteriormente para o Showcase. Clique em Escolher. Para alterar um logotipo, clique nele e clique no ícone Substituir imagem na barra de ferramentas.

5. Como personalizar a aparência de um arquivo de imagem em uma vitrine?

Você pode alterar a aparência (Ampliar, Reduzir, Ajustar ou Preencher) de um arquivo de imagem que você adicionou ao seu mostruário. Para alterar a aparência de um arquivo de imagem, clique na imagem para destacá-la e, em seguida, clique no ícone Editar aparência (lápis) e escolha entre as opções na barra de ferramentas.

6. Como adicionar um vídeo do YouTube ou outro link da Web que não seja do Dropbox a um mostruário?

Você pode adicionar *links* da web a um mostruário que será exibido como imagens ou vídeos clicáveis.

Para adicionar um *link* da Web ao seu Showcase:

Clique no botão “+” (mais) no canto superior esquerdo do mostruário.

Clique no *link* Inserir.

Cole um *link*.

Clique em Inserir.

Nota: Você também pode clicar nos ícones de *link* Inserir à esquerda de uma seção ou na parte inferior de um mostruário.

A imagem ou o vídeo exibido para um *link* da Web é determinado pelo website ao qual você está vinculando e não pode ser alterado no Showcase. Por exemplo, um *link* para um vídeo do YouTube é exibido como um vídeo reproduzível. Outros *links* da Web serão exibidos como uma imagem em destaque no site de origem ou na página da Web do *link*.

Nota: Alguns *links* da Web não renderizam uma imagem ou um vídeo. Esses links são exibidos com um símbolo de *link* simples.

Se o *link* da Web for exibido como uma imagem, você poderá editar a aparência da imagem (Ampliar, Reduzir, Ajustar e Preencher). Para fazer isso, clique uma vez na imagem, clique no ícone Editar aparência (lápis) e escolha entre as opções na barra de ferramentas.

Observação: se você passar o *mouse* sobre os ícones nas barras de ferramentas, verá uma descrição do que cada um faz.

7. A personalização do título de um arquivo em um mostruário também altera o nome desse arquivo no meu Dropbox?

Não. Se você personalizar o título de um arquivo em uma vitrine, o nome do arquivo original (ou "fonte") na sua conta do Dropbox permanecerá o mesmo. A personalização de um título de arquivo em um mostruário não altera os metadados para o arquivo de origem de nenhuma maneira.

8. Um mostruário usa o espaço de armazenamento do Dropbox?

Não, um mostruário não usa espaço de armazenamento adicional do Dropbox. Apenas os arquivos dentro do mostruário usam seu espaço de armazenamento.

Os arquivos que não foram armazenados anteriormente na sua conta do Dropbox vão directamente para uma pasta da sua conta chamada Uploads de demonstração. Os arquivos que já existem na sua conta do Dropbox, mas que você enviou de outra fonte (sua pasta de downloads ou em outro lugar), também vão directamente para os uploads do Showcase. Os únicos arquivos no seu mostruário que não são armazenados nos Uploads de demonstração são arquivos que já estavam na sua conta do Dropbox e foram enviados para o seu mostruário directamente da sua conta do Dropbox.

9. Tenho que manter os arquivos no meu mostruário sincronizados com a minha conta do Dropbox?

Os arquivos no seu Showcase são armazenados em uma pasta na sua conta do Dropbox chamada Uploads de demonstração. Se você excluir um item da sua pasta de uploads do Showcase, o arquivo desaparecerá do seu mostruário e será substituído por um ícone.

10. As pessoas com quem eu compartilho uma vitrine precisam ter uma conta do Dropbox?

Não. Os destinatários podem ver uma vitrine que você compartilhou com eles, mesmo sem uma conta do Dropbox.

No entanto, as pessoas com quem você compartilhou uma vitrine precisarão de uma conta do Dropbox - e estar conectada a essa conta - para deixar comentários no seu mostruário.

11. Quem pode editar um mostruário?

Apenas o proprietário (ou criador) de um mostruário pode editá-lo.

No entanto, se você compartilhou arquivos no seu mostruário, qualquer pessoa com quem você compartilhou esses arquivos pode editá-los. Essas alterações apareceriam no mostruário.

12. Um showcase funcionará em um telefone ou tablet?

Sim. Se você compartilhar um mostruário com alguém, ele poderá abri-lo e visualizá-lo em qualquer dispositivo: computador, telefone ou tablet. Todas as imagens e texto serão exibidos, independentemente do dispositivo. Nota: actualmente não é possível criar ou editar um mostruário em um smartphone ou tablet.

13. E se eu editar a fonte dos arquivos que estão no meu mostruário?

Se você editar um dos arquivos compartilhados em um mostruário, essas alterações serão reflectidas no próprio mostruário. Qualquer pessoa que visualize, abra ou faça *download* dos arquivos nesse mostruário também verá essas alterações. Desta forma, um mostruário funciona como uma pasta compartilhada do Dropbox: aqueles com acesso sempre verão as cópias mais recentes dos arquivos.

14. Como posso enviar arquivos grandes de graça?

Basta pressioná-lo, escolher o arquivo e enviá-lo como um anexo comum. Alternativamente, o Dropbox permite que você envie arquivos grandes e, em seguida, envie um *link* da web por email ou texto para o destinatário. Com o nível gratuito do Dropbox, você receberá 2 GB de espaço de armazenamento.

15. Por quanto tempo os arquivos permanecem no WeTransfer?

Quando você fez o upload usando nosso serviço gratuito, os transfers estão disponíveis por 7 dias. Após essa semana, eles são excluídos para liberar espaço para novas transferências em nossos servidores. Os arquivos enviados através do Perfil Plus de uma pessoa (não o seu) estão disponíveis por quatro semanas, em vez de sete dias.

16. Quanto custa o WeTransfer?

A assinatura anual custa US \$ 120, o que dá aos usuários um desconto de 17%. A opção gratuita é fornecida com um limite de envio de até 2 GB, enquanto os assinantes do WeTransfer Plus podem enviar até 20 GB por vez e estender a data de expiração de suas transferências.

17. O WeTransfer compacta fotos?

A verdade é que o WeTransfer usa compactação sem perdas, e o tamanho dos arquivos transferidos nunca diminui ou diminui. Quando você faz o upload para o WeTransfer, mostramos o tamanho real do arquivo e, ocasionalmente, ele pode parecer menor que o "tamanho no disco".

18. Como uso o WeTransfer no meu Iphone?

Transfira arquivos do seu dispositivo móvel. Para começar a usar o WeTransfer for Android oriOS, primeiro você precisa fazer o *download* do aplicativo da loja de aplicativos que você usa e depois iniciar o aplicativo. Quando você abre o aplicativo pela primeira vez, recebe uma mensagem explicando o que fazer. Deslize para cima a partir da parteinferior da tela para se livrar dessa mensagem.

19. O Dropbox é mais seguro que o *email*?

Isso é mais seguro do que enviar o arquivo como um anexo de *email*, já que o serviço de *email* do destinatário pode não ter segurança forte. Este método não é perfeito (já que, novamente, o Dropbox pode ver seus arquivos), mas é quase certamente melhor do que usar o anexo do *email*.

20. O WeTransfer é gratuito?

O WeTransfer é um serviço de transferência de arquivos baseado em nuvem. Freeusers podem enviar arquivos de até 2 GB; O WeTransfer 'Plus pago permite que os usuários enviem até 20 GB por vez e oferece recursos com protecção por senha, personalização de canal e armazenamento de 100 GB.

21. O WeTransfer é mais rápido que o Dropbox?

Por valor, o Dropbox fornece mais, já que você está recebendo 1TB de armazenamento em nuvem e sincronização de arquivos, enquanto o WeTransfer só é bom para enviar arquivos. Se tudo o que importa é enviar arquivos, no entanto, o WeTransfer é o melhor negócio, já que uma assinatura de US \$ 12 por mês permite que você registe senhas e datas de expiração.

22. Por que a WeTransfer demora tanto?

Isso significa que, se você está tendo um upload lento, é bem provável que ele tenha algo a fazer com seu provedor de serviços de Internet (ISP), conexão de rede ou algo que bloqueie nossa conexão de funcionar correctamente. Se você não estiver usando um *software* antivírus, a velocidade mais lenta poderá ser causada por problemas de conexão com a Internet.

23. Quão seguro é o WeTransfer?

Seus arquivos são criptografados quando estão sendo transferidos (TLS) e quando são armazenados (AES-256). Depois que seus arquivos são armazenados com segurança, eles só podem ser acessados usando os *links* exclusivos enviados ao remetente e ao destinatário.

24. Como faço para enviar via WeTransfer?

Siga estas etapas para compartilhar um arquivo usando o WeTransfer:

Passo 1: Abra um navegador no computador onde o arquivo está localizado.

Passo 2: Vá para a barra de endereços e digite myrealdata.wetransfer.com e pressione Enter.

Passo 3: Clique em Add Files.

Passo 4: Clique em "Seu Email" e adicione seu endereço de *email*.

Passo 5: Clique em “Transferir”.

25. O WeTransfer é legítimo?

Ignore *emails* de *scam* do 'WeTransfer'. Actualmente, há *emails* de embuste circulando em poses que vêm do WeTransfer. O assunto dos *emails falsos* é '[O remetente do e-mail] enviou um arquivo via WeTransfer'. A linha de assunto fornece o endereço de *email* do remetente, o que torna a mensagem confiável.



TEMA 5. BLOG

Esta unidade vai dar a conhecer as ferramentas para a criação e a publicação de *Blogs*. Disponibilizar os mecanismos para a melhor indexação dos seus *Blogs* de modo a que possam causar os melhores números de visualizações esperados.

Objectivos Específicos

Unidade Temática 5.1: O que é Blog?

Um blogue (em inglês: *blog*) (contracção dos termos em inglês web e log, "diário da rede") é um sítio electrónico cuja estrutura permite a actualização rápida a partir de acréscimos dos chamados artigos, ou postagens ou publicações. Estes são, em geral, organizados de forma cronológica inversa, tendo como foco a temática proposta do *blog*, podendo ser escritos por um número variável de pessoas, de acordo com a política do *blog*.

Muitos *blogs* fornecem comentários ou notícias sobre um assunto em particular; outros funcionam mais como diários *online*. Um *blog* típico combina texto, imagens e hiperligações para outros *blogs*, páginas da Web e mídias relacionadas a seu tema. A capacidade de leitores deixarem comentários de forma a interagir com o autor e outros leitores é uma parte importante de muitos *blogs*.

Alguns sistemas de criação e edição de *blogs* são muito atractivos pelas facilidades que oferecem, disponibilizando ferramentas próprias que dispensam o conhecimento de HTML. A maioria dos *blogs* são primariamente textuais, embora uma parte seja focada em temas exclusivos como arte, fotografia, vídeos, música ou áudio, formando uma ampla rede de mídias sociais. Outro formato é o microblogging, que consiste em *blogs* com textos curtos.

Em Dezembro de 2007, o motor de busca de *blogs* Technorati rastreou a existência de mais de 112 milhões de blogs. Com o advento do videoblog, a palavra *blog* assumiu um significado ainda mais amplo, implicando qualquer tipo de mídia onde um indivíduo expresse sua opinião ou simplesmente discorra sobre um assunto qualquer.



SITE

Os sites são feitos sob demanda, já que normalmente são espaços usados por empresas para promover seus produtos e serviços e é necessário um cuidado maior com a identidade visual.

BLOG

Já os blogs são muito mais básicos e normalmente são elaborados em plataformas que já estão prontas; além de possuírem limitações em relação as ações do autor dentro da plataforma e ao layout da mesma.

Fig. 5.1. Diferença entre *Site* e *Blog***Unidade Temática 5.2: Origens *Blog*****Etimologia**

O termo *weblog* foi criado por Jorn Barger em 17 de Dezembro de 1997. A abreviação *blog*, por sua vez, foi criada por Peter Merholz, que, de brincadeira, desmembrou a palavra *weblog* para formar a frase “*we blog*” (“nós blogamos”) na barra lateral de seu *blog* Peterme.com, em abril ou maio de 1999. Pouco depois, Evan Williams do Pyra Labs usou *blog* tanto como substantivo quanto verbo (“*to blog*” ou “blogar”), significando “editar ou postar em um *weblog*”), aplicando a palavra *blogger* em conjunção com o serviço *Blogger*, da Pyra Labs, o que levou à popularização dos termos.

Origens

Antes do formato *blog* se tornar amplamente conhecido, havia vários formatos de comunidades digitais como o Usenet, serviços comerciais online como o GEnie, BiX e Compuserve, além das listas de discussão e do Bulletim Board System (BBS). Em 1990, softwares de fóruns de discussão como o WebEx criaram os diálogos via threads.

O *blog* actual é uma evolução dos diários online, onde pessoas mantinham informações constantes sobre suas vidas pessoais. Estes primeiros *blogs* eram simplesmente componentes de sites, actualizados manualmente no próprio código da página. A evolução das ferramentas que facilitavam a produção e manutenção de artigos postados em ordem cronológica facilitaram o processo de publicação, ajudando em muito na popularização do formato. Isso levou ao aperfeiçoamento de ferramentas e hospedagem próprios para *blogs*.

Popularização

A mensagem passou a modelar o meio, quando no início de 2000, o *Blogger* introduziu uma inovação – o *permalink*, conhecido em português como ligação permanente ou apontador permanente – que transformaria o perfil dos *blogs*. Os *permalinks* garantiam a cada publicação num *blog* uma localização permanente - uma *URL* – que poderia ser referenciada. Anteriormente, a recuperação em arquivos de *blogs* só era garantida através da navegação livre (ou cronológica). O *permalink* permitia então que os blogueiros pudessem referenciar publicações específicas em qualquer *blog*.

Em seguida, *hackers* criaram programas de comentários aplicáveis aos sistemas de publicação de *blogs* que ainda não ofereciam tal capacidade. O processo de se comentar em *blogs* significou uma democratização da publicação, consequentemente reduzindo as barreiras para que leitores se tornassem escritores.

A blogosfera, termo que representa o mundo dos *blogs*, ou os *blogs* como uma comunidade ou rede social, cresceu em ritmo espantoso. Em 1999 o número de *blogs* era estimado em menos de 50; no final de 2000, a estimativa era de poucos milhares. Menos de três anos depois, os números saltaram para algo em torno de 2,5 a 4 milhões. Actualmente existem cerca de 112 milhões de *blogs* e cerca de 120 mil são criados diariamente, de acordo com o estudo “State of Blogosphere”.

Unidade Temática 5.3: Tipos de *Blog*

Existem diversos tipos de *blogs* actualmente. Entretanto é possível dividi-los em três grandes ramos:

Pessoais

Os *blogs* pessoais são os mais

populares, normalmente são usados como um género de diário ou com temas definidos, com postagens voltadas para os acontecimentos da vida e as opiniões do usuário. Também são largamente utilizados por celebridades que buscam manter um canal de comunicação com seus fãs.

Corporativos e organizacionais

Muitas empresas vêm utilizando *blogs* como ferramentas de divulgação e contacto com clientes. Tanto é assim que já existe a profissão de blogueiro, ou seja, profissionais são contratados pelas empresas com o cargo de blogueiro para a realização de *blogs* internos ou externos para registar as diversas actividades corporativas respectivamente para públicos internos (colaboradores) de forma mais privativa e externos como clientes e fornecedores. A empresa líder em *blogs* pelo mundo é a Microsoft com um total de 4500 *blogs*.

Temáticos

Por fim há *blogs* com um género específico, que tratam de um assunto dominado pelo usuário, ou grupo de usuários. Estes são os *blogs* com o maior número de acessos. Sendo que eles podem apresentar conteúdos variados, como humorísticos, notícias, informativos ou de variedades, com contos, opiniões políticas e poesias. Algumas categorias de *blogs* recebem denominações específicas, como: *blogs* educativos, *blogs* literários, *Metablogs*, etc.

Unidade Temática 5.4: Componentes *Blog*

Componentes:

Blogger

Blogueiro (português brasileiro) ou blóquer ou bloguista (português

europeu) ou ainda blogger são palavras utilizadas para designar aquele que escreve em blogues. O universo dos blogueiros (a soma de tudo o que está relacionado a este grupo e este grupo em si) é conhecido como blogosfera.

No dia 31 de agosto, comemora-se o Dia do *Blog* (devido a semelhança da data 31.08 com a palavra *blog*), que se propõe a promover a descoberta de novos blogues e de novos blogueiros.

Artigos

Conhecidos também como post, a forma substantiva anglófona do verbo "postar", refere-se a uma entrada de texto efetuada num *weblog/blog*. As postagens são organizadas tradicionalmente de forma cronologicamente inversa na página, de forma que as informações mais actualizadas aparecem primeiro, ou colocada ao contrário, a postagem mais antiga aparece em primeiro, sendo opção do blogueiro.

Um artigo deve seguir a temática proposta pelo *blog* e, embora permita uma enorme liberdade opinativa, seu conteúdo está sujeito às mesmas regras legais de outras fontes, de modo que seu autor pode vir a ser responsabilizado juridicamente por aquilo que escreve.

Actualmente, a maioria dos *blogs* é compatível com o recurso de inserção de imagens, vídeos, áudio nos artigos.

Comentários

Um recurso característico dos *blogs* é a possibilidade de interacção do visitante, respondendo ou opinando em relação aos artigos publicados.



Fig. 5.2. Site do Blogger.com

Para criar os seus *blogs*, aceda:

<https://wordpress.com/start/user?ref=create-blog-ip>

https://www.blogger.com/about/?r=1-null_user

<https://www.wix.com>

Exercícios de AUTO-AVALIAÇÃO

1. O *blog* ainda é relevante em 2018?

Há muitas razões pelas quais os *blogs* ainda são muito relevantes, até mesmo essenciais para o marketing digital e de conteúdo em 2018 e no futuro previsível. Uma delas é que continua sendo o formato de conteúdo de maior sucesso na web. Mas o *blog* também rege muitas outras partes do marketing de conteúdo.

2. Você pode realmente ganhar dinheiro blogando?

Sim, é possível ganhar dinheiro blogando. Há um número cada vez maior de pessoas ganhando dinheiro com *blogs* a tempo parcial e a tempo integral - embora elas ainda sejam minoria.

3. O *blog* ainda é relevante em 2019?

Sim, vale a pena fazer *blogs* em 2019. Os *blogs* surgiram desde a sua primeira aparição em 1990. Ele é usado como uma plataforma única para todos compartilharem seus pensamentos e expressões, mas hoje o *blog* cresceu para um uso de plataforma de milhões, mesmo de pessoa para empresa.

4. Como faço para começar um *blog* e ganhar dinheiro?

Como iniciar um *blog* (no lado) em 10 etapas fáceis:

1. Escolha o nome e o domínio do seu *blog*.
2. Encontre um provedor de hospedagem.
3. Escolha uma plataforma de *blogs* (CMS)

4. Encontre o seu nicho e decida sobre o que sobre o *blog*.
5. Escolha um tema e crie seu *blog*.
6. Crie suas páginas essenciais (sobre, contacto, contratação)
7. Desenvolva sua estratégia de conteúdo.

5. Por que a maioria dos *blogs* falha?

Há outra razão pela qual os *blogs* falham. A segunda razão mais comum para o fracasso dos *blogs* é quase tão mortal. Nesse caso, o blogueiro escreve e escreve, mas não coloca tempo (ou planeamento) para direcionar o tráfego. Grandes blogueiros criam conteúdo com promoção em mente.

6. Quão bem sucedidos são os *blogs*?

Como iniciar um *blog* de sucesso em 2019:

- Escolha o seu nome de domínio e *host*.
- Configure seu *blog* com o WordPress.
- Crie seu *blog* com um tema.
- Seleccione os melhores plugins para o seu *blog*.
- Comece a blogar com conteúdo atraente.

7. Como posso tornar meu *blog* lucrativo?

5 Estratégias comprovadas para construir um *blog* rentável:

1. Seja aberto sobre quem você é.
2. Conheça seu público e dê a eles o que precisam.
3. Crie sua lista de *email* desde o início.
4. Concentre-se em um grupo principal de produtos afiliados.
5. Crie seu próprio produto.

8. Quem é o blogueiro mais bem pago?

Sem mais delongas, vamos dar uma olhada nos 10 maiores blogueiros profissionais pagos de 2018:

- Rand Fishkin: Moz;
- Michael Arrington: TechCrunch;
- Pete Cashmore: Mashable;
- Martin Lewis: MoneySavingExpert.com;
- Brian Clark: CopyBlogger;
- Perez Hilton: PerezHilton.com;
- Univision Communications: Gizmodo;
- Smashing Magazine.

9. Ganhar dinheiro blogando é realista?

Outra pesquisa, essa de 1.500 leitores ProBlogger que disseram que estão tentando ganhar dinheiro com *blogs*, descobriu que 9% fazem entre US \$ 1.000 e US \$ 10.000 por mês e 4% fazem mais de US \$ 10.000 por mês. Mas a grande maioria ganha menos de US \$ 3,50 por dia. (A maioria deles eram blogs com menos de dois anos de idade).

10. Os *websites* ainda são relevantes em 2019?

É 2019 e a Internet nunca encontrou uma mente que não pudesse consumir. A mídia social é conhecida por ser uma das maneiras mais rápidas de estabelecer uma presença digital - com a capacidade de se conectar rapidamente com os clientes, a utilidade de um *website* é questionada.

11. Qual plataforma de *blog* é melhor para ganhar dinheiro?

Com tudo isso em mente, vamos comparar as melhores plataformas de *blogs* para iniciantes:

1. WordPress.org é o *software* de *blog* mais popular do mundo;
2. O Wix é uma plataforma hospedada para construir *sites*;
3. WordPress.com;
4. Blogger;
5. Tumblr;
6. Médio;
7. Squarespace;
8. Joomla.

12. O que devo escrever sobre o *blog*?

- Considere suas paixões;
- Considere os *blogs* que você lê;
- Ensine tudo o que você sabe;
- Pense em perder conteúdo de seus *blogs* favoritos;
- Concentre-se em apresentar soluções para problemas;
- Escreva como um aprendiz líder;
- Pense no corpo de trabalho que você deseja ter em cinco anos.
- Escreva para uma pessoa específica.

13. Quantos *blogs* existem em 2018?

Estimamos que o número total de *blogs online* em 2018 é de cerca de 505 milhões de *blogs*.

14. Como você inicia uma revisão de *blog*?

Primeiro, dê uma olhada nas suas paixões mais profundas.

Passo 1: Comece observando suas paixões;

Passo 2: Encontre o seu nicho;

Etapa 3: Pesquise seu mercado;

Etapa 4: Escolha um domínio;

Etapa 5: Decida de onde o \$\$\$ virá;

Etapa 6: crie o *site* ou *blog*;

Passo 7: Escreva três artigos para começar;

Passo 8: Inicie o seu *blog* de revisão.

15. Quem é o *Vlogger* mais bem pago?

De irmãos controversos a um garoto de 7 anos que adora brincar com brinquedos, aqui está a lista dos *YouTubers* mais bem pagos, segundo a Forbes:

- Logan Paul - US \$ 14,5 milhões;
- PewDiePie - US \$ 15,5 milhões;
- Jacksepticeye - US \$ 16 milhões;

- VanossGaming - US \$ 17 milhões;
- Markiplier - US \$ 17,5 milhões;
- Jeffree Star - US \$ 18 milhões;
- DanTDM - US \$ 18,5 milhões.

16. Os blogueiros do Youtube são pagos?

A maioria dos blogueiros começa a ganhar dinheiro com o Google Adsense em seus *blogs*. Os paysucks, com uma média entre US \$ 0,006 e US \$ 0,015 por visualização de página, são rápidos e fáceis. O YouTube compartilha 55% de sua receita de anúncios com criadores de vídeos e livros, em média US\$ 7,60 por 1.000 visualizações de anúncios.

17. Porquê o *website* é necessário?

Hoje, cada vez mais usuários usam a internet para procurar os produtos ou serviços de que precisam. Sua empresa vai ganhar credibilidade por ter um *website*. Aumentar a visibilidade é um factor importante que torna importante ter um *site*. ... Ter uma estratégia de presença *online* e de presença permite que você comercialize seu negócio *online*.

18. Quem é o *YouTuber* 2018 mais rico?

Aqui está uma lista dos *YouTubers* mais ricos em 2018:

- Ryan Toysreview - US \$ 11 milhões;
- Jake Paul - US \$ 11,5 milhões;
- Felix Kjelberg (Pewdiepie) - US \$ 12 milhões;
- Mark Fischbach (Markiplier) - US \$ 12,5 milhões;
- Logan Paul - US \$ 12,5 milhões;
- Dude Perfect - USD \$ 14m;
- Evan Fong (vanossgaming) - US \$ 15,5 milhões;

- Daniel Middleton (dantdm) - US \$ 16,5 milhões.

19. Como os *Youtubers* conseguem dinheiro?

Prepare seu canal para ganhar:

Etapa 1: active seu canal para monetização;

Etapa 2: conecte seu canal do YouTube a uma conta do Google AdSense para ganhar dinheiro e receber o pagamento de seus vídeos monetizados;

Etapa 3: tire um minuto para conhecer os tipos de vídeos que você pode gerar receita e os diferentes formatos de anúncio.

20. Quanto dinheiro você ganha por mil visualizações no YouTube?

Você poderia calcular que para fazer um vídeo no YouTuber e postá-lo no canal deles, você estaria pagando cerca de US \$ 10.000,00 por 100.000 visualizações, o que significa US \$ 100,00 por mil visualizações.

21. Os *Youtubers* são pagos por subscritores?

24 de fevereiro de 2016 - Enquanto escrevo isto, o AdSense (publicidade do YouTube) está pagando em média entre 1 e 2 dólares por 1.000 visualizações. Então, se você quiser ganhar US \$ 1000 dólares por mês, você precisa entre 500k a 1 milhão de visualizações mensais. A publicidade é apenas uma das várias maneiras de ganhar dinheiro no YouTube!

22. Os YouTubers são pagos pelos *Likes* dos subscritores?

Os anunciantes pagam apenas quando alguém clica em um anúncio ou assiste por 30 segundos. É por isso que

você não pode vincular suas visualizações do canal a dólares. Se o seu vídeo obtiver 10 milhões de visualizações, mas ninguém assiste ou clica nos anúncios, você não ganha dinheiro. É assim que eu ganho \$ 1 por 25 visualizações.



TEMA 6. GESTIONÁRIOS DE CONTEÚDOS COM JOOMLA

Esta unidade vai dar a conhecer as ferramentas para a criação e a gestão dos seus próprios conteúdos *online*, para a criação e administração dos seus próprios *websites*.

Vai aprender nesta unidade o processo para hospedagem dum *website*.

Objectivos Específicos

Unidade Temática 6.1: Sistema de Gestão de Conteúdo

Sistema de gerenciamento de conteúdo ou sistema de gestão de conteúdo (em inglês: *Content management system*, CMS) é um aplicativo usado para criar, editar, gerenciar e publicar conteúdo em plataformas digitais, permitindo que o mesmo seja modificado, removido e adicionado sem o conhecimento de linguagem de programação. CMSs são frequentemente usados para armazenar, controlar, prover documentação, empresarial tais como notícias, artigos, manuais de operação, manuais técnicos, guias de vendas e brochuras de marketing. O conteúdo pode incluir imagens, áudios, vídeos, documentos e conteúdo Web.

Podemos dizer que um CMS é semelhante a um framework (um quadro de referência) de website pré-estruturado, com recursos básicos de: usabilidade, visualização e administração já prontamente disponíveis. É um sistema que permite a criação, armazenamento e administração de conteúdos de forma dinâmica, através de uma interface de acesso via Internet.

Um CMS permite que a empresa tenha total autonomia sobre o conteúdo e evolução da sua presença na internet e dispense a assistência de terceiros ou empresas especializadas para manutenções de rotina.



Fig. 6.1. Logotipo ou *Icon* do Software Joomla

O principal intuito de uma empresa optar pelo uso de um CMS é tornar mais ágil o processo de actualização de ambientes web, permitindo que colaboradores da empresa actualizem directamente suas respectivas áreas de responsabilidade em *sites* internet e intranet. Ampliando esta experiência na busca pela agilidade nas actualizações, sistemas CMS de qualidade não devem requerer habilidades além dos conhecimentos necessários para um usuário de um simples editor de texto.

Sistemas CMS de qualidade devem permitir ainda a alteração da aparência de um *website* por meio da utilização de temas. Estes temas são pacotes que contemplam imagens e estilos de fonte que podem ser facilmente mudados, duplicados e alterados conforme a necessidade de cada *website*. Altera-se a aparência de um *site* meramente com a alteração do tema utilizado pelo site.

O uso de um sistema CMS dispensa o uso de profissionais de internet em uma empresa?

A resposta é não. Nenhum CMS de mercado, gratuito ou não, dispensa o papel de uma pessoa responsável pelo ambiente de hospedagem do CMS ou pela alimentação do conteúdo.

O termo "profissionais de internet" é muito amplo para responder de forma objectiva e satisfatória essa pergunta. Exemplificando, podemos separar esses profissionais da seguinte forma:

Responsáveis pela tecnologia, infra-estrutura e desenvolvimento ou manutenção da ferramenta;

Responsáveis pela criação de conteúdos para a internet;

Responsáveis pela aprovação de conteúdos, antes de sua divulgação ao público.

Observe que a adopção de um CMS pela empresa simplesmente visa facilitar o desenvolvimento de ferramentas para gestão de *websites*, como recursos de organização e cadastro de notícias, por exemplo.

Assim, sempre teremos um ambiente a ser mantido, e conteúdos a serem produzidos.

Um sistema de gerenciamento de conteúdos requer uma infra-estrutura de hospedagem (servidores). Bem como, um CMS requer manutenções periódicas para garantir sua evolução, sua actualização e seu crescimento futuro. E ainda, para a implantação de novos *websites*, faz-se necessário o conhecimento específico do CMS sobre sua estrutura de navegação, sua estrutura de informações, sua organização de conteúdos e o reaproveitamento de recursos ali disponíveis.

Este trabalho envolve análise de sistemas, e um certo conhecimento técnico que permitirá a evolução, a qualidade e o futuro dos projectos de internet de uma empresa.

Um CMS trabalha com gestão de CONTEÚDO. O que não é possível sem um ambiente estável, estruturado e organizado.

Unidade Temática 6.2: Os Principais Sistemas CMS Gratuitos

Principais sistemas CMS gratuitos disponíveis no mercado

Os sistemas de gerenciamento de conteúdo mais populares são o Wordpress e o Joomla! que são sistemas em PHP, Open Source. Ambos são utilizados para criação de *sites* e portais. Outros exemplos de CMS são o Plone, o Liferay e o Drupal onde é facilmente realizada a edição de conteúdos a partir do próprio *site*, podendo também ser usado para o gerenciamento de intranet. O Joomla! também é muito utilizado para intranets.

Joomla! (pronúncia: [djumla]) é um sistema livre *open source* de gestão de conteúdo web (do inglês: *Content Management System - CMS*) desenvolvido em PHP e com base de dados MySQL, executado em um servidor interpretador. Criado pelo grupo de desenvolvedores OSM em 2005, foi projectado com separação entre *design*, programação e conteúdo, proporcionando rapidez na produção de sites com flexibilidade, design personalizados e baixo investimento. Um é esqueleto de *website* pré-configurado com recursos básicos, com fácil manutenção e administração via web.

É um projecto *open source* (licença GNU/GPL) e a sua última versão estável é a 3.8.3 (dezembro de 2017); a sua última versão legada é a 1.5.26.

O Joomla! auxilia na publicação e administração de um conteúdo na web para leigos em desenvolvimento de *sites*. É utilizado para o desenvolvimento de lojas virtuais, *blogs*, revistas *online*, portais de conteúdo, jornais, catálogos de produtos/serviços e diversos outros formatos de conteúdo. No CMS vem pré-configurado as características básicas de um *site*, como *login* de usuários e administradores, edição e publicação de conteúdo, publicação de "banners" de propaganda.

O CMS conta com extensões que não estão em sua instalação básica, mas que podem ser adquiridos através de *download* dos módulos e componentes (10.071 mil) que aumentam sua funcionalidade com rapidez. Além disso, existem vários

templates prontos para agilizar o processo de criação de *sites* com investimento mínimo. Possuindo versões em 64 idiomas.

Os CMS ajudam a resolver um problema, antes complexo durante muito tempo, a necessidade de investir um custo alto em pessoas capacitadas para construir e gerenciar um *site* ou portal.

É resultado da separação entre a equipe de desenvolvedores do Mambo CMS e a empresa australiana Miro, devido a transferência do controle do CMS para a fundação Mambo Foundation, prejudicando assim os desenvolvedores que teriam apenas uma participação passiva e pouco representativa. Esses desenvolvedores, preocupados com a integridade do projecto e com o futuro dos utilizadores, não aceitaram a transferência e, em 2005, criaram o "Joomla 1.0", também "*open source*", a partir do código-fonte do Mambo 4.5.2 (fork).

Como afirma Sady Jacques, o embaixador da Associação Software Livre: "O *software* livre é fundamental neste contexto de mudança de visão, ele traz o compartilhamento e colaboração como filosofia". Para o usuário iniciante existem diversas comunidades locais de Joomla onde pode ser conseguida ajuda no seu processo de aprendizado e mesmo na construção de projectos profissionais.

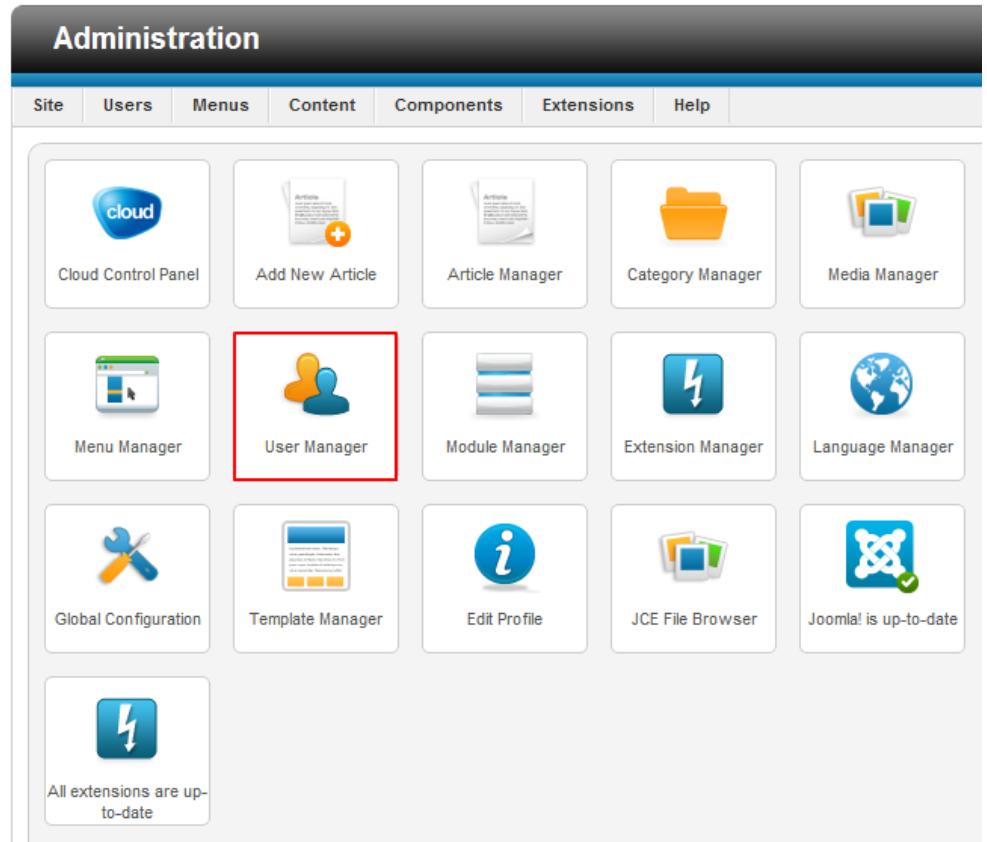


Fig. 6.2. Ambiente de Trabalho (Administração) do Joomla

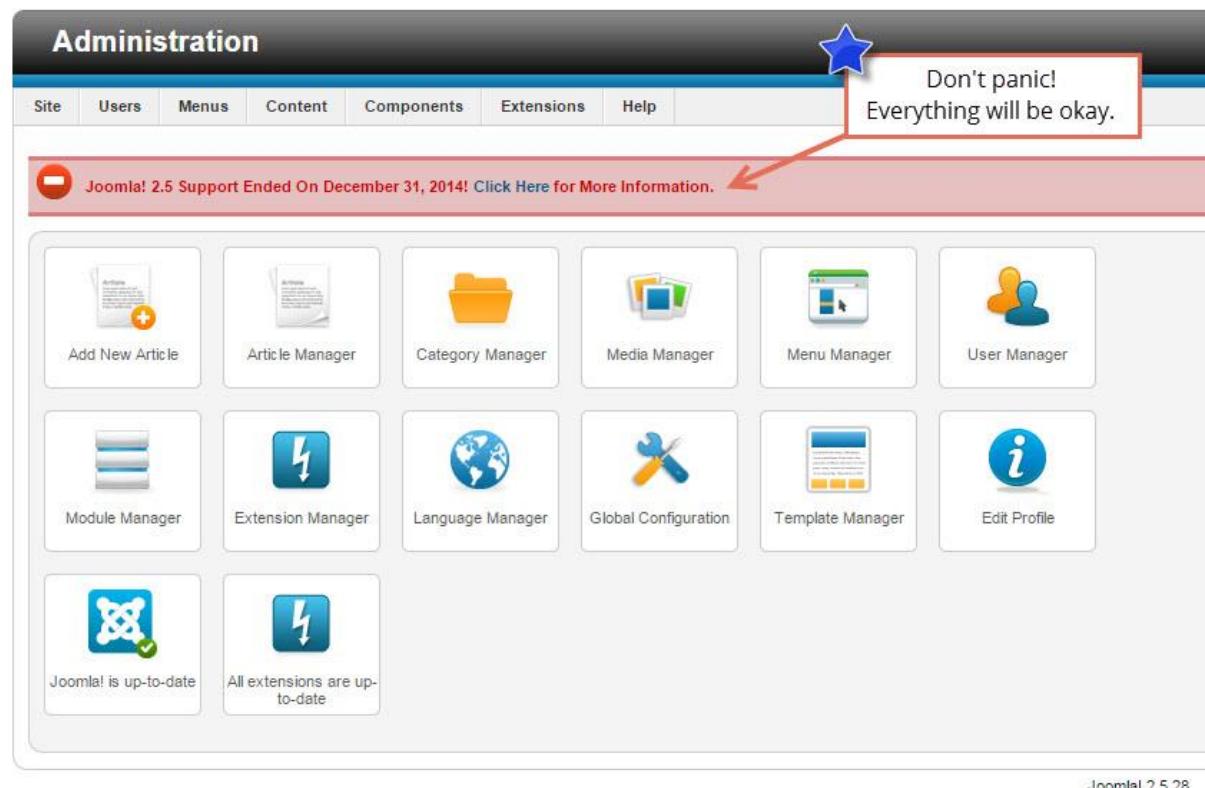


Fig. 6.3. Ambiente de Trabalho (Administração) do Joomla

Premiações

Em 2005 recebeu o prémio Linux Awards, como o “melhor projecto de software livre”.

Em 2011 recebeu o prémio de o "melhor CMS Open Source".

Em 2014 recebeu prémio de o "melhor CMS PHP".

Em 2016 recebeu prémio de o "Best Free CMS".

Suas principais características são:

Código aberto (Licença GPL);

Sistema simples de fluxo de aprovação;

Arquivamento para conteúdo não utilizado (lixo);
Gerenciamento de *banners*;

Sistema de publicação para o conteúdo;

Sumário de conteúdo no formato RSS;

Busca optimizada (qualquer palavra registada);

Frontend traduzido em várias línguas;

Fácil instalação de novos templates, módulos e componentes;

Hierarquia para grupos de usuários;

Estatísticas básicas de visitantes (módulos mais complexos podem ser adicionados);

Editor de conteúdo WYSIWYG;

Sistema de enquete simples (com acompanhamento de resultado em tempo real);

Sistemas de índices de avaliação;

Extensões livres em diversos *sites* (Módulos, Componentes, Templates, Traduções).

Compatibilidade com versão anterior.

Tudo sobre como usar o Joomla, aceda:

https://docs.joomla.org/J3.x:Installing_Joomla/pt

Exercícios de AUTO-AVALIAÇÃO

1. O que faz um CMS?

Um sistema de gerenciamento de conteúdo (CMS) é um aplicativo de *software* ou conjunto de programas relacionados que são usados para criar e gerenciar conteúdo digital. Os CMSs são normalmente usados para gerenciamento de conteúdo corporativo (ECM) e gerenciamento de conteúdo da Web (WCM). Um WCM facilita a criação colaborativa de *sites*.

2. O que significa CMS?

Sistema de gerenciamento de conteúdo CMS tornou-se uma palavra da moda na indústria de desenvolvimento web por causa de seus benefícios. CMS significa *Content Management System*. É um termo e não foi endossado com uma definição sólida. Um CMS pode ter vários significados, dependendo dos cenários e dos objectivos da pessoa ou do projecto.

3. O que é o desenvolvimento de *sites* com CMS?

Um CMS ou um '*Content Management System*' literalmente permite que você controle e gerencie o conteúdo do seu *website* - sem treinamento técnico. Usando este sistema descomplicado você pode facilmente adicionar, excluir imagens e editar texto em seu *website*, com a maior facilidade.

4. Quantos tipos de CMS existem?

Existem três tipos amplos de *software* CMS: código aberto, proprietário e CMS de *software* como serviço, incluindo soluções baseadas em nuvem.

5. Qual é o propósito de usar o CMS?

Um sistema de gerenciamento de conteúdo (CMS) gerencia a criação e modificação de conteúdo digital. Geralmente, ele suporta vários usuários em um ambiente colaborativo. Os recursos do MSMS variam amplamente. A maioria dos CMSs inclui publicação baseada na Web, gerenciamento de formato, edição de histórico e controle de versão, indexação, pesquisa e recuperação.

6. Qual é o melhor CMS?

Os sistemas CMS mais populares em detalhes:

- WordPress. Com cerca de 18 milhões de instalações, o WordPress é o CMS de código-fonte aberto mais utilizado em todo o mundo;
- Joomla! Com 2,5 milhões de instalações em todo o mundo, o Joomla! é o segundo maior agente no mercado de CMS;
- Drupal;
- TYPO3;
- Contao;
- Craft;
- Grav.

7. O que é o *software* do site CMS?

Um sistema de gerenciamento de conteúdo da Web (WCM ou WCMS) é um sistema de gerenciamento de conteúdo de *software* (CMS), especificamente para conteúdo da Web. Ele fornece ferramentas de administração de *website*, colaboração e administração que ajudam usuários com pouco conhecimento de linguagens de programação da

Web ou linguagens de marcação a criar e gerenciar conteúdo de websites.

8. Qual é o melhor CMS para *Website Designing*?

5 principais sistemas de gerenciamento de conteúdo:

- Drupal O Drupal é facilmente o CMS *open source* mais funcional disponível atualmente;
- WordPress. O WordPress começou como um simples sistema de *blogs*, mas se tornou um dos CMSs mais poderosos da Internet;
- CMS radiante. Um simples CMS desenvolvido por Ruby on Rails;
- Magento;
- Silverstripe.

9. O que é um exemplo de CMS?

O WordPress, que mostramos acima, é o melhor exemplo de um sistema popular de gerenciamento de conteúdo. Além do *software* WordPress auto-hospedado, outros sistemas de gerenciamento de conteúdo populares incluem: Joomla e Drupal.

10. Quais são os benefícios do CMS?

Flexibilidade para desenvolvedores - Como o CMS permite que usuários não técnicos publiquem facilmente conteúdo, isso libera os desenvolvedores técnicos para se concentrarem na funcionalidade e nos recursos aprimorados. O *design* é separado do conteúdo - você pode manipular o conteúdo sem medo de alterar accidentalmente o design.

11. O que é o CMS em HTML?

Novos *sites* podem ser criados usando HTML ou CMS. HTML é a linguagem de codificação que alimenta a maioria das páginas da web na Internet. Um CMS (sistema de gerenciamento de conteúdo) é uma plataforma de *software* que é alimentada por um banco de dados e fornece aos usuários uma interface fácil de usar para criar e manter um *website*.

12. Quais são os tipos de CMS?

Os 10 principais sistemas de gerenciamento de conteúdo mais utilizáveis:

- WordPress;
- Drupal;
- Joomla!;
- ExpressionEngine;
- TextPattern;
- CMS radiante;
- CMS Cushy;
- SilverStripe.

13. Como eu faço o *download* do Joomla?

Nós nos concentramos na instalação do Joomla 3.9, pois é o mais recente ramo estável oficialmente recomendado para todos os novos projectos:

Faça o *download* do pacote de instalação do Joomla 3.9;

Envie os arquivos do Joomla para o seu servidor;

Crie um banco de dados MySQL para o Joomla usar;

Percorra o processo de instalação.

14. Como eu instalo um template Joomla?

Etapas para instalar um modelo Joomla:

Passo 1: Primeiro de tudo você tem que entrar no seu painel de controle do Joomla usando seu nome de usuário e senha;

Etapa 2: clique em extensões;

Passo 3: Na página de instalação, seleccione a opção “*Upload Package File*”;

Passo 4: Faça o *upload* do arquivo de instalação do seu modelo clicando na opção “Escolher arquivo”.

15. Como faço para instalar o Joomla no Windows 7?

Veja como você faz isso:

Etapa 1: instale o WAMP. Baixe o WAMP para o seu computador;

Passo 2: Baixe o Joomla. Vá para Joomla.org e clique no botão preto dizendo "Download Joomla";

Etapa 3: Mova o Joomla para o WAMP;

Etapa 4: Configure nosso banco de dados;

Etapa 5: Instale o Joomla;

Etapa 6: Excluir/renomear o directório de instalação.

16. O que é o *plugin* Joomla?

Um plugin é uma espécie de Joomla! extensão. Plugins fornecem funções associadas a eventos de *trigger*. O Joomla fornece um conjunto de eventos de *plugins* principais, mas qualquer extensão pode disparar eventos (personalizados). Esta é uma maneira poderosa de estender a funcionalidade do Joomla!

17. Qual é o melhor Joomla ou Wordpress?

Enquanto Joomla é o melhor para o *website* de redes sociais e eCommerce. WordPress vem com fácil instalação de cinco minutos, enquanto o Joomla não é tão fácil de instalar como o WordPress. Ambos permitem que os usuários alterem a aparência do *site* usando temas e modelos. Wordpress e Joomla são CMSs (*Content Management Systems*).

18. O Joomla é livre para usar?

Joomla! é um sistema de gerenciamento de conteúdo (CMS) gratuito e de código aberto para publicação de conteúdo da web. Ao longo dos anos Joomla! ganhou vários prémios. Ele é construído em uma estrutura de aplicativo da web controlador de visualização de modelo que pode ser usada independentemente do CMS que permite a criação de aplicativos *online* avançados.

19. Como faço para instalar um template?

Para instalar, vamos seguir os mesmos passos da instalação do template:

Navegue até o seu Gerenciador de extensão no painel de administração do Joomla;

Clique na guia Carregar arquivo do pacote;

Clique em Escolher arquivo e seleccione um dos plugins;

E finalmente clique em Upload & Install para instalá-lo.

20. Como faço para editar um *site* do Joomla?

Faça o *login* no seu painel administrativo do joomla 3;

No menu superior, clique em Extensões e, em seguida, clique em Gerenciador de Modelos;

No menu esquerdo, clique em Modelos;

Você verá uma lista de modelos que você instalou;

Clique no arquivo que você deseja editar;

Edite o arquivo e clique no botão Salvar.

21. O Joomla ainda é relevante?

Muitos ainda acreditam que o Joomla é bom para criar *sites* de pequenas e médias empresas. Joomla! ainda é considerada uma boa opção para criar sites governamentais. Joomla!, se posicionar-se de forma inteligente, pode lidar com isso, pois a sua gestão e desenvolvimento é muito mais fácil do que o Drupal.

22. Como faço para recuperar minha senha de administrador do Joomla?

Como redefinir a senha do administrador do Joomla:

Entre no seu Painel de Controle;

Clique na guia Sistema;

Em Minhas Assinaturas, seleccione seu nome de domínio;

Clique em Websites & Domains;

Seleccione Bancos de Dados;

Clique no nome do seu banco de dados;

No menu Ferramentas, escolha Webadmin para abrir o phpMyAdmin;

Clique na tabela * _users.

23. Como eu mudo o cabeçalho no Joomla?

Como faço para actualizar o código do cabeçalho e rodapé:

Faça o login na área de administrador do seu *site* Joomla;

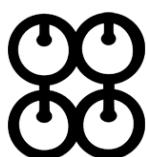
Seleccione o menu Extensões e o menu Gerenciador de modelos;

Abra o modelo que você instalou no *site*;

Você verá agora o painel de controle;

Abra a guia do código personalizado de cabeçalho e rodapé;

Você verá dois campos para o cabeçalho e rodapé.

**TEMA 7. HIPERTEXTO, HIPERMÉDIA, PROTOCOLOS PARA INTERNET E PARA WEB**

Esta unidade vai ensinar os principais conceitos e protocolos que estão na vanguarda do bom funcionamento da internet e das suas aplicações Web.

Nesta unidade vai aprender as regras de funcionamento dos principais protocolos que regem o funcionamento da internet (HTTP, FTP, SSH, etc.).

Objectivos Específicos

Unidade Temática 7.1: Hipertexto e Hipermédia

Hipertexto é o termo que remete a um texto ao qual se agregam outros conjuntos de informação na forma de blocos de textos, palavras, imagens ou sons, cujo acesso se dá através de referências específicas, no meio digital denominadas hiperligações. Estas hiperligações ocorrem na forma de termos destacados no corpo de texto principal, ícones gráficos ou imagens e têm a função de interconectar os diversos conjuntos de informação, oferecendo acesso sob demanda às informações que estendem ou complementam o texto principal. O conceito de "linkar" ou de "ligar" textos foi criado por Ted Nelson nos anos 1960 e teve como influência o pensador e sociólogo francês Roland Barthes, que concebeu em seu livro *S/Z* o conceito de "Lexia", que seria a ligação de textos com outros textos.

Em termos mais simples, o hipertexto é uma ligação que facilita a navegação dos internautas. Um texto pode ter diversas palavras, imagens ou até mesmo sons que, ao serem clicados, são remetidos para outra página onde se esclarece com mais precisão o assunto do *link* abordado.

O sistema de hipertexto mais conhecido actualmente é a *World Wide Web*, no entanto a Internet não é o único suporte onde este modelo de organização da informação e produção textual se manifesta.

Uma hiperligação, um liame/ligame, ou simplesmente uma ligação (em inglês, *hyperlink* e *link*), é uma referência dentro de um documento em hipertexto a outras partes desse documento ou a outro documento. Um programa informático utilizado para

visualizar e criar esse documento chama-se um sistema de hipertexto, normalmente um usuário pode criar uma hiperligação ou simplesmente uma ligação. Um usuário que siga as ligações está a navegar o hipertexto ou a navegar a web.

Hiperligações são partes dos fundamentos das linguagens usadas para construção de páginas na *World Wide Web* e outros meios digitais e são designadas elementos clicáveis, em forma de texto ou imagem, que levam a outras partes de um sítio ou para recursos variados. Um dos símbolos (ícones) mais comuns para representar um *link* numa página da Internet é quando aparece uma mão fechada com o dedo indicador levantado no momento que o mouse está posicionado em cima desse *link*.

A palavra inglesa *link* entrou na língua portuguesa por via de redes de computadores (em especial a Internet), servindo de forma curta para designar as hiperligações do hipertexto. O seu significado é "atalho", "caminho" ou "ligação". Através dos *links* é possível produzir documentos não lineares interconectados com outros documentos ou arquivos a partir de palavras, imagens ou outros objectos.

Navegar ou "surfar" na Internet é seguir uma sequência de *links*, que agregam interactividade no documento. Ao leitor torna-se possível localizar rapidamente conteúdo sobre assuntos específicos, no contexto.

O *Hypertext Transfer Protocol*, sigla HTTP (em português Protocolo de Transferência de Hipertexto) é um protocolo de comunicação (na camada de aplicação segundo o Modelo OSI) utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos. Ele é a base para a comunicação de dados da *World Wide Web*.

Hipertexto é o texto estruturado que utiliza ligações lógicas (hiperlinks) entre nós contendo texto. O HTTP é o protocolo para a troca ou transferência de hipertexto.

Coordenado pela *World Wide Web Consortium* e a *Internet Engineering Task Force*, culminou na publicação de uma série de *Requests for Comments*; mais notavelmente o RFC 2616, de Junho de 1999, que definiu o HTTP/1.1. Em Junho de 2014 foram publicados 6 RFC's para maior clareza do protocolo HTTP/1.1. Em Março de 2015, foi divulgado o lançamento do HTTP/2. A actualização deixará o navegador com um

tempo de resposta melhor e mais seguro. Ele também melhorará a navegação em *smartphones*.

Para acedermos a outro documento a partir de uma palavra presente no documento actual podemos utilizar hiperligações (ou âncoras). Estes documentos se encontram no sítio com um endereço de página da Internet – e para acessá-los deve-se digitar o respectivo endereço, denominado *URI (Universal Resource Identifier ou Identificador Universal de Recurso)*, que não deve ser confundido com *URL (Universal Resource Locator ou Localizador Universal de Recurso)*, um tipo de *URI* que pode ser directamente localizado.

Hipermédia é a reunião de várias mídias num ambiente computacional, suportada por sistemas electrónicos de comunicação. Hipermédia, diferentemente de multimédia, não é a mera reunião dos meios existentes, e sim a fusão desses meios a partir de elementos não-lineares.

Uma forma bastante comum de Hipermédia é o Hipertexto, no qual a informação é apresentada sob a forma de texto interactivo. As informações são acessadas pelo monitor de um computador, pela tela de um *smartphone*, entre outros dispositivos. O usuário é capaz de ler de forma não linear, ou seja, ele escolhe entre o início, o meio ou o fim do conteúdo. Segundo Bugay, a Hipermédia pode ser considerada uma extensão do Hipertexto, entretanto, inclui, além de textos comuns, sons, animações e vídeos.

Unidade Temática 7.2: *HyperText Transfer Protocol (HTTP)*

O *HyperText Transfer Protocol* é um protocolo de aplicação responsável pelo tratamento de pedidos e respostas entre cliente e servidor na *World Wide Web*. Ele surgiu da necessidade de distribuir informações pela Internet e para que essa distribuição fosse possível foi necessário criar uma forma padronizada de comunicação entre os clientes e os servidores da Web e entendida por todos os computadores ligados à Internet. Com isso, o protocolo HTTP passou a ser utilizado para a comunicação entre computadores na Internet e a especificar como

seriam realizadas as transacções entre clientes e servidores, através do uso de regras básicas.

Este protocolo tem sido usado pela WWW desde 1990. A primeira versão de HTTP, chamada HTTP/0.9, era um protocolo simples para a transferência de dados no formato de texto ASCII pela Internet, através de um único método de requisição, chamado GET. A versão HTTP/1.0 foi desenvolvida entre 1992 e 1996 para suprir a necessidade de transferir não apenas texto. Com essa versão, o protocolo passou a transferir mensagens do tipo MIME44 (*Multipurpose Internet Mail Extension*) e foram implementados novos métodos de requisição, chamados POST e HEAD.

No HTTP/1.1, versão actual do protocolo descrito na RFC 2616, foi desenvolvido um conjunto de implementações adicionais ao HTTP/1.0, como por exemplo: o uso de conexões persistentes; o uso de servidores *proxy* que permitem uma melhor organização da cache; novos métodos de requisições; entre outros. Afirma-se que o HTTP também é usado como um protocolo genérico para comunicação entre os agentes de utilizadores e *proxies/gateways* com outros protocolos, como o SMTP, NNTP, FTP, Gopher, e WAIS, permitindo o acesso a recursos disponíveis em aplicações diversas.

Uma sessão HTTP é uma sequência de transacções de rede de requisição-resposta. Um cliente HTTP inicia uma requisição estabelecendo uma conexão *Transmission Control Protocol* (TCP) para uma porta particular de um servidor (normalmente a porta 80. Veja Lista de portas dos protocolos TCP e UDP). Um servidor HTTP ouvindo naquela porta espera por uma mensagem de requisição de cliente. Recebendo a requisição, o servidor retorna uma linha de estado, como "HTTP/1.1 200 OK", e uma mensagem particular própria. O corpo desta mensagem normalmente é o recurso solicitado, apesar de uma mensagem de erro ou outra informação também poder ser retornada.

Unidade Temática 7.3: Modelo OSI

O Modelo OSI (acrónimo do inglês *Open System Interconnection*) é um modelo de rede de computador referência da ISO dividido em camadas de funções, criado em 1971 e formalizado em 1983, com objectivo de

ser um padrão, para protocolos de comunicação entre os mais diversos sistemas em uma rede local (*Ethernet*), garantindo a comunicação entre dois sistemas computacionais (*end-to-end*).

Este modelo divide as redes de computadores em 7 camadas, de forma a se obter camadas de abstracção. Cada protocolo implementa uma funcionalidade assinalada a uma determinada camada.



Fig. 7.1. Modelo de Referência OSI

Segundo Tanenbaum o Modelo OSI não é uma arquitectura de redes, pois não especifica os serviços e protocolos exactos que devem ser usados em cada camada. Ele apenas informa o que cada camada deve fazer.

O Modelo OSI permite comunicação entre máquinas heterogéneas e define directivas genéricas para a construção de redes de computadores (seja de curta, média ou longa distância) independente da tecnologia utilizada.

Camada	Protocolo
7. Aplicação:	HTTP, RTP, SMTP, FTP, SSH, Telnet, SIP, RDP, IRC, SNMP, NNTP, POP3, IMAP, BitTorrent, DNS ...
6. Apresentação:	XDR, TLS ...
5. Sessão:	NetBIOS ...
4. Transporte	NetBEUI, TCP, UDP, SCTP, DCCP, RIP ...
3. Rede:	IP (IPv4, IPv6), IPsec, ICMP, ARP, RARP, NAT, ...
2. Enlace:	
Subcamada LLC:	
Subcamada MAC:	Ethernet, IEEE 802.1Q, HDLC, Token ring, FDDI, PPP, Switch, Frame relay, ATM, ...
1. Física:	Modem, 802.11 Wi-Fi, RDIS, RS-232, EIA-422, RS-449, Bluetooth, USB, 10BASE-T, 100BASE-TX, ISDN, SONET, DSL, ...

Modelo de Referência OSI

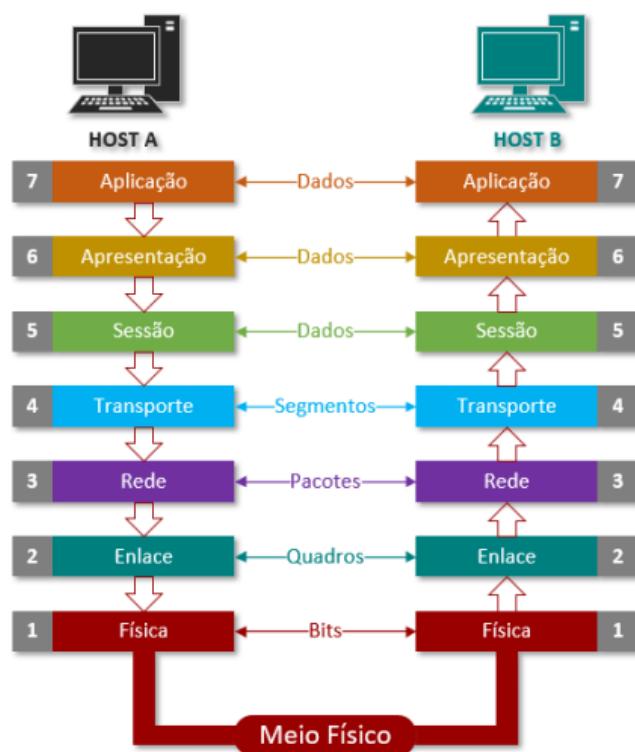


Fig. 7.2. Camadas Modelo de Referência OSI

- | | |
|-------------------|---------------------------------------|
| 7-Aplicação | Interfaces com aplicativos |
| 6-Apresentação | Formatos / Criptografia |
| 5-Sessão | Controle de Sessões entre Aplicativos |
| 4-Transporte | Conexão entre hosts / Portas |
| 3-Rede | Endereço lógico / Roteadores |
| 2-Enlace de Dados | Endereço físico / Pontes e Switches |
| 1-Física | Hardware / Sinal elétrico / bits |

Fig. 7.3. Funções Camadas Modelo de Referência OSI

CAMADA	FUNÇÃO
APLICAÇÃO	Funções especializadas (transferência de arquivos, terminal virtual, e-mail)
APRESENTAÇÃO	Formatação de dados e conversão de caracteres e códigos
SESSÃO	Negociação e estabelecimento de conexão com outro nó
TRANSPORTE	Meios e métodos para a entrega de dados ponta-a-ponta
REDE	Roteamento de pacotes através de uma ou várias redes
ENLACE	Detecção e correção de erros introduzidos pelo meio de transmissão
FÍSICA	Transmissão dos bits através do meio de transmissão

Fig. 7.4. Funções Camadas Modelo de Referência OSI

Unidade Temática 7.4: Os Principais Protocolos da Internet

RTP

Em ciência da computação, do inglês *Real-time Transport Protocol* conhecido pelo acrônimo RTP, é um protocolo de redes utilizado em aplicações de tempo real como, por exemplo, entrega de dados áudio ponto-a-ponto, como Voz sobre IP. Ele funciona como uma subcamada na camada de transporte, camada 4 do Modelo OSI, e define como deve ser feita a fragmentação do fluxo de dados de áudio, adicionando a cada fragmento informação de sequência e de tempo de entrega, sendo o controle é realizado pelo RTCP - *Real Time Control Protocol*. Ambos utilizam o UDP como real protocolo de transporte, o qual não oferece qualquer garantia que os pacotes serão entregues num determinado intervalo.

RTP -Real Time Transport Protocol

- 36 O Protocolo de transporte mais utilizado para aplicações é o TCP., porém, para aplicações em tempo real, o TCP não possui as características desejadas.
- 37 Algumas funções do TCP o desqualificam para o seu uso como protocolo de transporte das aplicações de tempo real, como:

Fig. 7.5. Diferença entre RTP e TCP

RTP -Real Time Transport Protocol

- 38 Outro protocolo de transporte largamente usado é o UDP, ele não possui as duas primeiras características listadas, mas da mesma forma que o TCP, não provê informações de temporização. Sozinho, o UDP também não provê as ferramentas para as aplicações de tempo real.

Fig. 7.6. Diferença entre RTP e UDP

É um pouco difícil dizer em que camada o RTP está. Como ele funciona no espaço do usuário e está vinculado ao programa aplicativo, certamente parece ser um protocolo de aplicação. Por outro lado ele é um protocolo genérico e independente das aplicações que apenas fornecem recursos de transporte, e assim também é semelhante a um protocolo de transporte.

O RTP permite que seja atribuída a cada fonte (i.e, câmaras ou microfones) sua própria corrente independente de pacotes RTP. Por exemplo, para uma videoconferência entre dois participantes, quatro correntes RTP podem ser abertas — duas correntes para transmitir o áudio (uma em cada direção) e duas para transmitir o vídeo (uma em cada direção).

Os protocolos RTP/RTCP são definidos pela RFC 3550 do IETF (Internet Engineering Task Force).

Real Time Streaming Protocol (RTSP)

- O Real Time Streaming Protocol (RTSP) é utilizado para fornecer conteúdo como uma transmissão em sequência unicast (ligação um-para-um entre o servidor e um cliente), em que cada cliente recebe uma transmissão em sequência distinta e apenas esses clientes que solicitam a transmissão em sequência podem recebê-la.
- É um protocolo de controlo que funciona em conjunto com a entrega de dados RTP (Real Time Protocol) para fornecer conteúdo aos clientes, tendo sido criado especificamente para controlar a entrega de dados em tempo real, como, por exemplo, conteúdo de áudio e vídeo. Serve para estabelecer e controlar um ou vários streams sincronizados de meios de comunicação de massa (imprensa, rádio, televisão, etc.).

Fig. 7.7. Como funciona o RTSP

SMTP

Simple Mail Transfer Protocol (abreviado SMTP, traduzido do inglês, significa "Protocolo de transferência de correio simples") é o protocolo padrão para envio de *emails* através da Internet, definido na RFC 821.

É um protocolo relativamente simples, em texto plano, onde um ou vários destinatários de uma mensagem são especificados (e, na maioria dos casos, validados) sendo, depois, a mensagem transferida. Esse protocolo usa por padrão a porta TCP 25 (ou 465 para conexão criptografada via SSL). No Brasil, porém, desde 2013, provedores e operadoras de internet passaram a utilizar a porta 587, como medida de segurança para diminuir o número de spams.

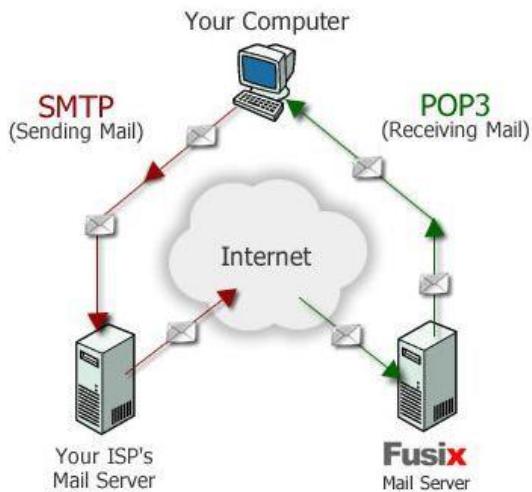


Fig. 7.8. Modelo ou Arquitectura do SMTP e POP3

O SMTP é um protocolo de envio apenas, o que significa que ele não permite que um usuário descarregue as mensagens de um servidor. Para isso, é necessário um cliente de *email* com suporte ao protocolo POP3 ou IMAP, o que é o caso da maioria dos clientes atuais.

A resolução DNS de um servidor SMTP de um dado domínio é possibilitada por sua entrada MX (*Mail eXchange*).

FTP

FTP ou *File Transfer Protocol* (em português, Protocolo de Transferência de Arquivos) é uma forma de transferir arquivos (Portugal: conhecidos como ficheiros).

Pode referir-se tanto ao protocolo quanto ao programa que implementa este protocolo (Servidor FTP, neste caso, tradicionalmente aparece em letras minúsculas, por influência do programa de transferência de arquivos do Unix).



FTP (*File Transfer Protocol*) Protocolo de Transferencia de Arquivos



Fig. 7.9. Modelo ou Arquitectura do FTP

A transferência de dados em redes de computadores envolve normalmente transferência de arquivos e acesso a sistemas de arquivos remotos (com a mesma interface usada nos arquivos locais). O FTP (RFC 959) é baseado no TCP, mas é anterior à pilha de protocolos TCP/IP, sendo posteriormente adaptado para o TCP/IP. É o padrão da pilha TCP/IP para transferir arquivos, é um protocolo genérico independente de *hardware*.

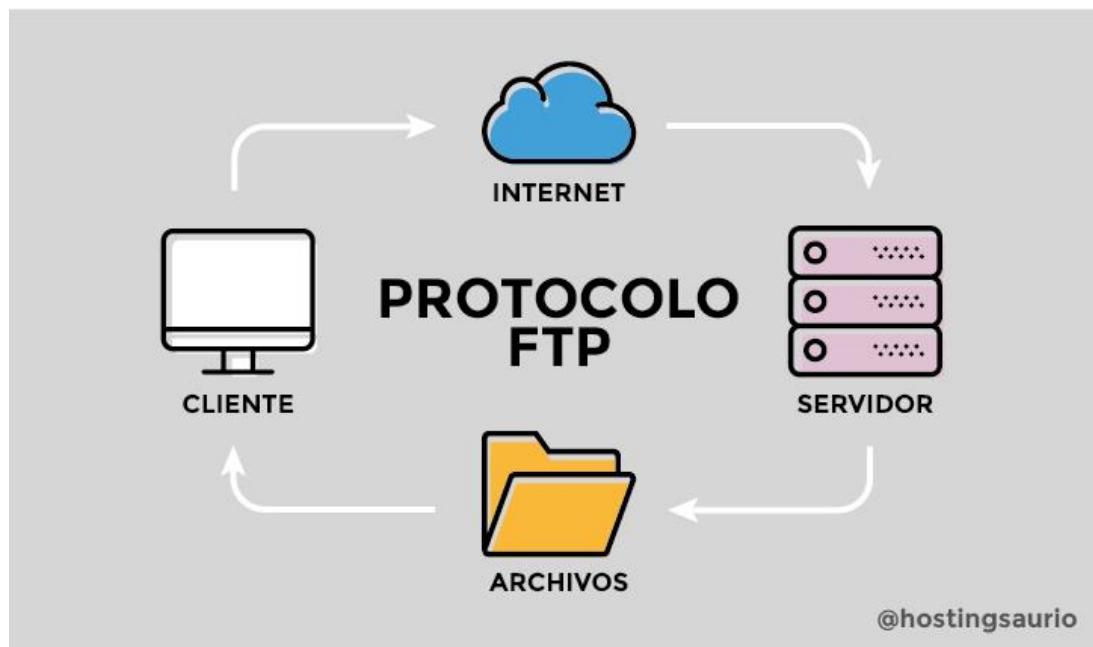


Fig. 7.10. Modelo ou Arquitectura do FTP

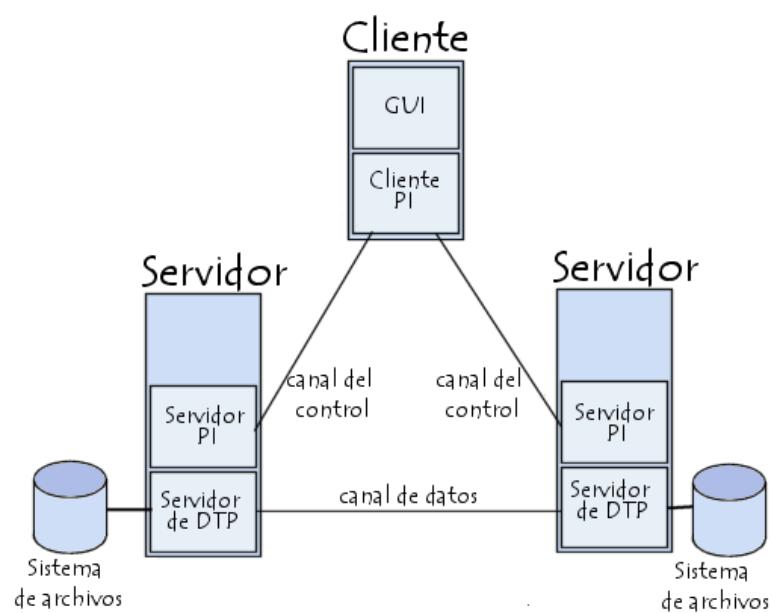


Fig. 7.11. Modelo ou Arquitectura do FTP

SSH

Secure Shell (SSH) é um protocolo de rede criptográfico para operação de serviços de rede de forma segura sobre uma rede insegura. O melhor exemplo de aplicação conhecido é para login remoto de usuários a sistemas de computadores.

O SSH fornece um canal seguro sobre uma rede insegura em uma arquitectura cliente-servidor, conectando uma aplicação cliente SSH com um servidor SSH. Aplicações comuns incluem login em linha de comando remoto e execução remota de comandos, mas qualquer serviço de rede pode ser protegido com SSH. A especificação do protocolo distingue entre duas versões maiores, referidas como SSH-1 e SSH-2.

A aplicação mais visível do protocolo é para acesso a contas shell em sistemas operacionais do tipo Unix, mas também verifica-se algum uso limitado no Windows. Em 2015, a Microsoft anunciou que incluiriam suporte nativo para SSH em uma liberação futura.

O SSH foi projectado como um substituto para o Telnet e para protocolos de shell remotos inseguros como os protocolos Berkeley rlogin, rsh e rexec. Estes protocolos enviam informações, notavelmente senhas, em texto puro, tornando-os susceptíveis à interceptação e divulgação usando análise de pacotes. A criptografia usada pelo SSH objectiva fornecer confidencialidade e integridade de dados sobre uma rede insegura, como a Internet, apesar dos arquivos vazados por Edward Snowden indicarem que a Agência de Segurança Nacional Americana pode algumas vezes descriptografar o SSH, permitindo-os ler o conteúdo de sessões SSH.

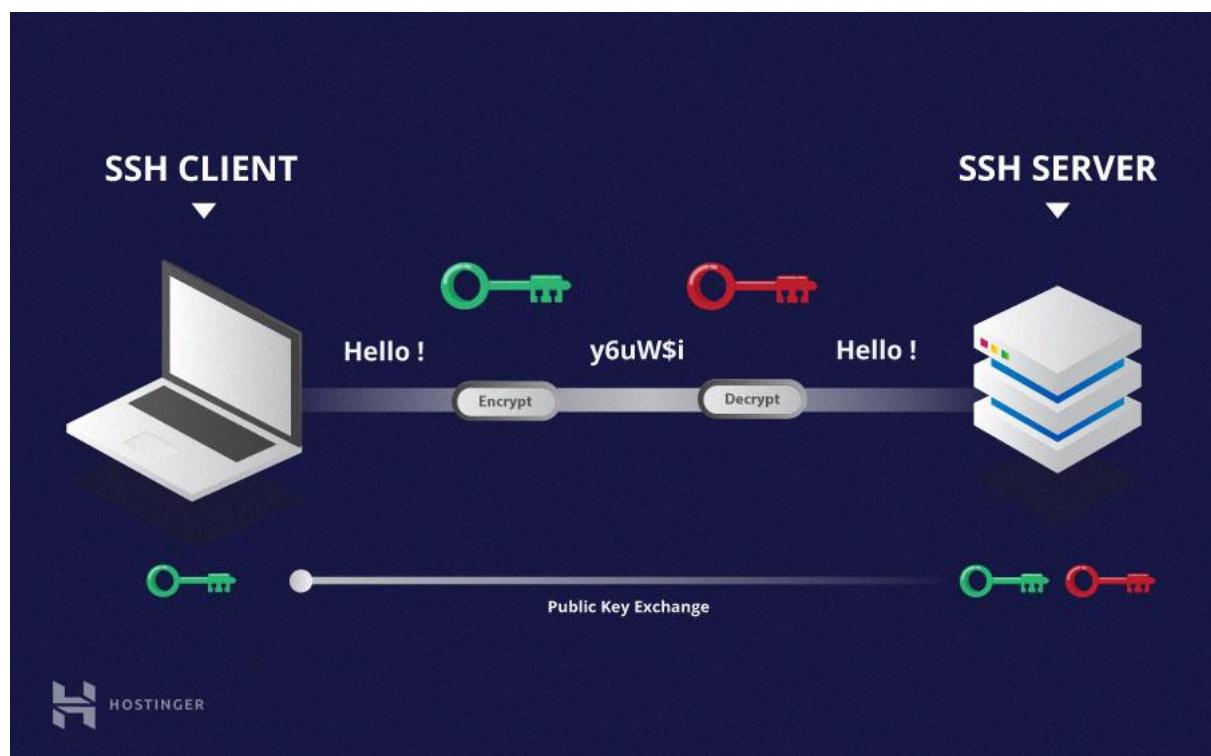


Fig. 7.12. Modelo ou Arquitectura do SSH

O SSH usa criptografia de chave pública para autenticar o computador remoto e permiti-lo autenticar o usuário, se necessário. Há várias maneiras de usar o SSH, uma é usar pares de chave pública-privada geradas automaticamente para simplesmente encriptar uma conexão de rede e então usar autenticação por senha para logar.

Outra maneira é usar um par de chaves pública-privada geradas manualmente para realizar a autenticação, permitindo que usuários ou programas loguem sem ter que especificar uma senha. Neste cenário, qualquer um pode produzir um par correspondente de chaves diferentes (pública e privada). A chave pública é colocada em todos os computadores que devem permitir o acesso ao proprietário da chave privada correspondente (o proprietário mantém o segredo da chave privada). Uma vez que a autenticação é baseada na chave privada, a chave em si nunca é transferida por meio da rede durante a autenticação. O SSH apenas verifica se a mesma pessoa que oferece a chave pública também possui a chave privada correspondente. Em todas as versões do SSH é importante verificar chaves públicas desconhecidas, isto é, associar as chaves públicas com as identidades, antes de aceitá-las como válidas.

Aceitar a chave pública de um atacante sem validação autorizará o atacante como um usuário válido.

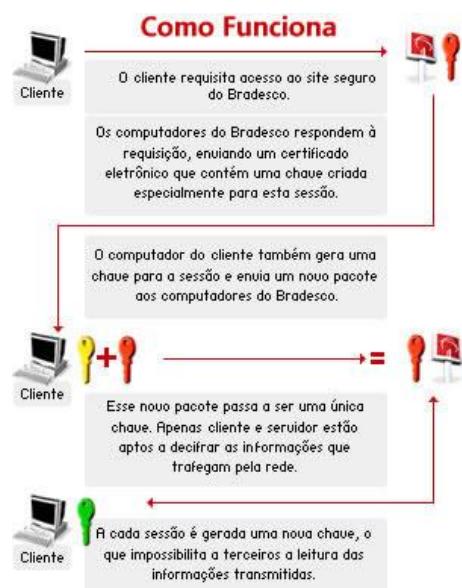


Fig. 7.13. Como funciona o SSH

O SSH é normalmente usado para login em uma máquina remota e execução de comandos, mas também suporta tunelamento, redirecionamento de portas TCP e conexões X11. Ele pode transferir arquivos usando os protocolos *SSH file transfer* (SFTP) ou *secure copy* (SCP). O SSH utiliza o modelo cliente-servidor.

A porta TCP padrão 22 tem sido usada para contactar servidores SSH.

Telnet

Telnet é um protocolo de rede utilizado na Internet ou redes locais para proporcionar uma facilidade de comunicação baseada em texto interactivobidireccional usando uma conexão de terminal virtual. Os dados do usuário são intercalados em banda com informações de controle Telnet em um byte de conexão 8-bit de dados orientado sobre o *Transmission Control Protocol* (TCP).

O Telnet foi desenvolvido em 1969 com a chegada do RFC 15, prorrogado no RFC 854, e padronizado como *Internet Engineering Task Force* (IETF) Internet STD Padrão 8, um dos primeiros padrões da Internet.

O protocolo Telnet é um protocolo *standard* de Internet que permite a interface de terminais e de aplicações através da Internet. Este protocolo fornece as regras básicas para permitir ligar um cliente (sistema composto de uma afixação e um teclado) a um intérprete de comando (do lado do servidor).

O Telnet existe há mais de 40 anos, muito antes de aparecer a Internet. Este sistema de transmissão de dados foi inventado pelas Forças Armadas Americanas para transmissão de dados entre bases militares. Foi disponibilizado ao público em 1977, tendo sido os radioamadores os primeiros a aproveitá-lo.

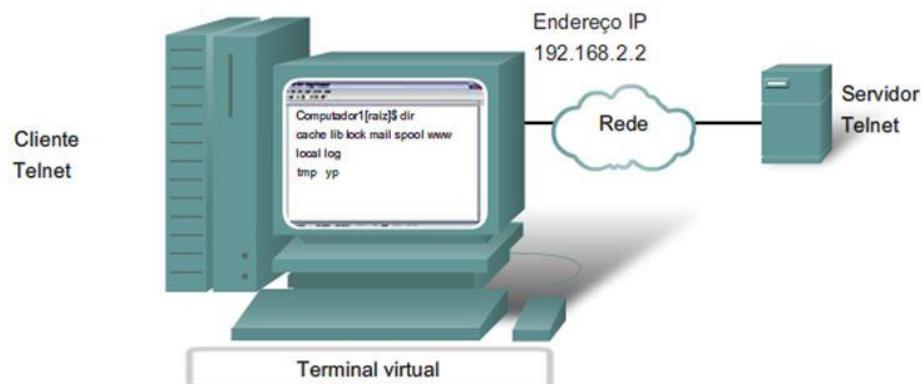
Portanto, pode-se dizer que a Internet trabalha por cima do Telnet, servindo-se do seu sistema para funcionar. A transmissão de dados pelo Telnet utiliza *software* específico que os codifica, permitindo utilizar centenas de portas por nós definidas e reencaminhá-las, para o PC que pretendemos. Se tivermos uma rede interna com vários pc's instalados, utilizando um *router*, abrimos uma porta para cada PC e, com o mesmo IP, os dados fluem direcionados e reencaminhados simultaneamente, sem qualquer problema. O utilizador que recebe dados combina com o seu correspondente a porta de passagem para o sistema funcionar. A máquina que envia os dados fá-lo em pacotes. Informa o correspondente que tem dados. Este, por sua vez, dá o OK para a transmissão. O pacote é enviado com a informação do número de bits que este tem. Só depois do correspondente ter informado que recebeu os bits todos, é que pede o segundo pacote. Se por qualquer motivo informa que os bits não chegaram todos, o envio do pacote é repetido. Muita coisa fica ainda por dizer sobre este sistema. O protocolo baseia-

se numa conexão TCP para enviar dados em formato ASCII codificado em 8 bits entre os quais se intercalam sequências de controle para o Telnet. Fornece assim um sistema orientado para a comunicação, bidireccional (*half-duplex*), codificado em 8 bits fácil de aplicar. Com essa conexão é possível o acesso remoto para qualquer máquina ou equipamento que esteja sendo executado em modo servidor.

O protocolo Telnet assenta em três conceitos fundamentais que serão explicados a seguir:

O paradigma do terminal rede virtual (NVT, *Network Virtual Terminal*);
O princípio de opções negociadas; As regras de negociação.

Protocolo telnet



O Telnet fornece um modo de usar um computador, conectado através da rede, para acessar um dispositivo de rede como se o teclado e o monitor estivessem diretamente conectados ao dispositivo.

kraemer

Fig. 7.14. Modelo ou Arquitectura do Telnet

Terminal virtual

O Telnet consiste em criar abstracções no terminal, fazendo com que qualquer cliente ou servidor se comunique com outro host sem conhecer as suas características.

A comunicação de NVT, comumente chamada de Terminal Rede Virtual, fornece uma base padrão de:

Caractéres de 7 bits ASCII

Três caracteres de controle

Cinco caracteres de controle opcionais

Um jogo de sinais de controle básico

Opções negociadas

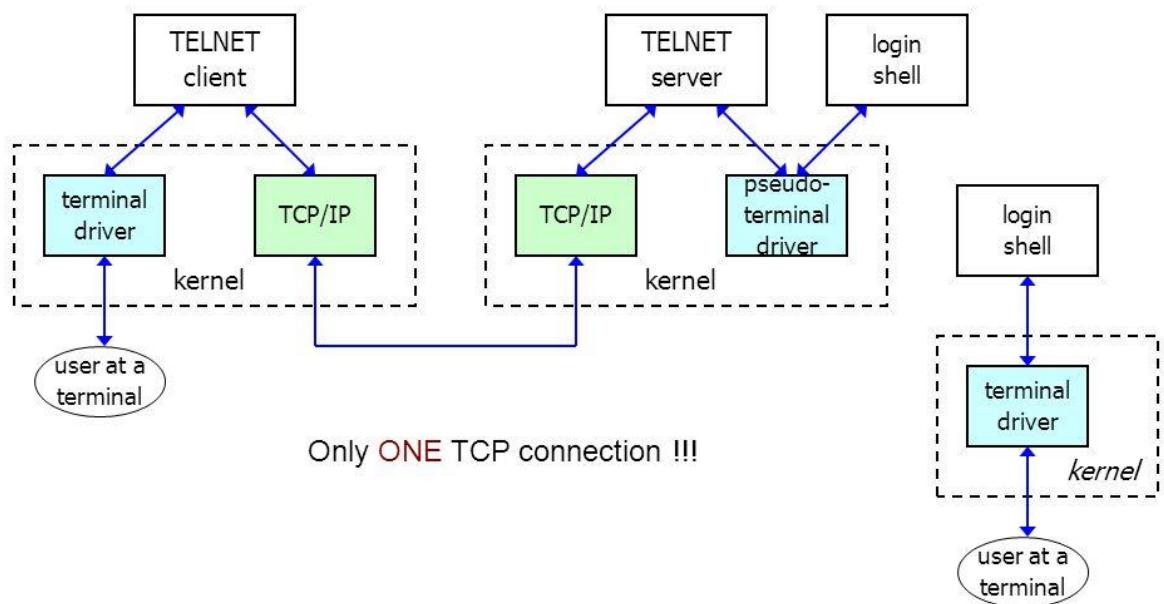
As opções negociadas permitem que alguns terminais proponham serviços adicionais que não são definidos nas especificações básicas. Esses serviços adicionais permitem a utilização de funções avançadas em forma de opções, fazendo iniciar os pedidos para solicitar a autorização ao sistema distante a activação desse serviço ou não. Qualquer um dos lados da rede pode emitir o sinal e logo em seguida a outra deve responder se aceita ou não a liberação da opção requerida. Caso o pedido seja para desactivar a opção, quem recebe a mensagem não deve recusar a mensagem, para ser compatível com o modelo NVT.

Regras de negociação

As regras de negociação de opções podem evitar um caso de bloqueio do pedido, pois os pedidos só devem ser emitidos quando acontece a mudança de um modo. Caso exista um envio para mudança esse deve ser inserido no fluxo de dados apenas no lugar onde tem efeito, o receptor da mensagem só deve adoptá-lo quando não se encontrar no mesmo modo pedido.

Este protocolo é um protocolo básico, no qual se apoiam outros protocolos da sequência TCP/IP (FTP, SMTP, POP3, ...). As especificações de Telnet não mencionam autenticação porque o Telnet está totalmente separado das aplicações que o utilizam (o protocolo FTP define uma sequência de autenticação acima do Telnet). Além disso, o protocolo Telnet é um protocolo de transferência de dados não seguro, o que quer dizer que os dados que veicula circulam às claras na rede (de maneira não codificada). Quando o protocolo Telnet é utilizado para ligar um hóspede distante à máquina na qual é aplicado como servidor, este protocolo é atribuído à porta 23.

TELNET Client and Server



2007

CSCE515 – Computer Network Programming

Fig. 7.15. Modelo ou Arquitectura do FTP

SNMP

Simple Network Management Protocol (SNMP), em português Protocolo Simples de Gerência de Rede, é um "protocolo padrão da Internet para gerenciamento de dispositivos em redes IP". Dispositivos que normalmente suportam SNMP incluem roteadores, computadores, servidores, estações de trabalho, impressoras, racks modernos e etc. SNMP é usado na maioria das vezes em sistemas de gerenciamento de rede para monitorar dispositivos ligados a rede para condições que garantem atenção administrativa. SNMP é um componente do conjunto de protocolos da Internet como definido pela *Internet Engineering Task Force* (IETF). Ele consiste de um conjunto de padrões de gerenciamento de rede, incluindo um protocolo da camada de aplicação, um esquema de banco de dados, e um conjunto de objectos de dados.

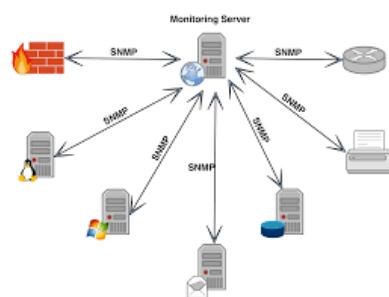


Fig. 7.16. Modelo ou Arquitectura do SNMP

Exemplo

Monitoramento utilizando o protocolo SNMP.

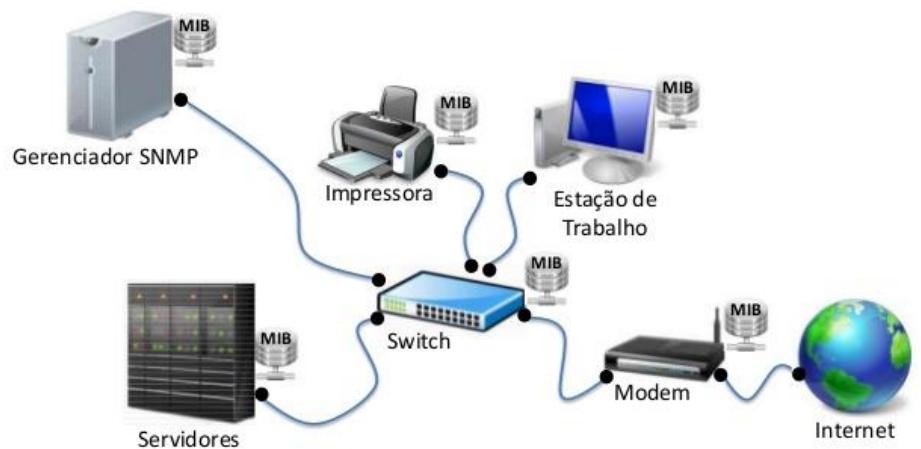


Fig. 7.17. Monitoramento de Dispositivos com SNMP

O software de gerência de redes não segue o modelo cliente-servidor convencional, pois para as operações GET e SET, a estação de gerenciamento se comporta como cliente e o dispositivo de rede a ser analisado ou monitorado se comporta como servidor, enquanto que na operação TRAP ocorre o oposto, pois no envio de alarmes é o dispositivo gerenciado que toma iniciativa da comunicação. Por conta disso, os sistemas de gerência de redes evitam os termos 'cliente' e 'servidor' e optam por usar "gerente" para a aplicação que roda na estação de gerenciamento e "agente" para a aplicação que roda no dispositivo de rede.

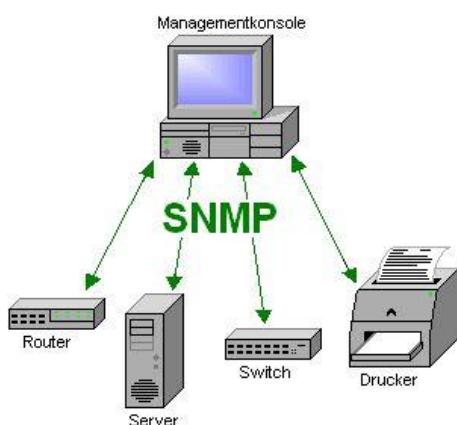


Fig. 7.18. Modelo ou Arquitectura do SNMP

POP3

O *Post Office Protocol* (termo em inglês que, traduzido, significa "Protocolo dos correios"), ou POP3, é um protocolo utilizado no acesso remoto a uma caixa de correio electrónico. Ele está definido no RFC 1939 e permite que todas as mensagens contidas numa caixa de correio electrónico possam ser transferidas sequencialmente para um computador local. Dessa maneira, o utilizador pode ler as mensagens recebidas, apagá-las, responder-lhes, armazená-las etc.

Este protocolo utiliza as portas TCP 110 (porta padrão) ou TCP 995 (conexão criptografada via SSL). A porta TCP 109 foi utilizada na versão anterior do protocolo (POP2).

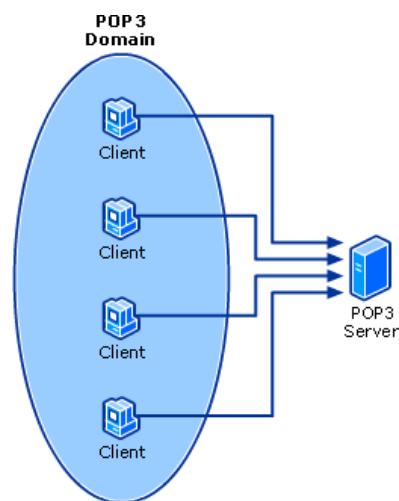


Fig. 7.19. Modelo ou Arquitectura do POP3

O funcionamento do protocolo POP3 diz-se *offline*, uma vez que o processo suportado se baseia nas seguintes etapas:

É estabelecida uma ligação TCP entre a aplicação cliente de *email* (*User Agent - UA*) e o servidor onde está a caixa de correio (*Message Transfer Agent - MTA*);

O utilizador autentica-se;

Todas as mensagens existentes na caixa de correio são transferidas sequencialmente para o computador local;

As mensagens são apagadas da caixa de correio (opcionalmente, o protocolo pode ser configurado para que as mensagens não sejam apagadas da caixa de correio; se esta opção não for utilizada, deve-se utilizar sempre o mesmo computador para ler o correio electrónico, para poder manter um arquivo das mensagens);

A ligação com o servidor é terminada;

O utilizador pode agora ler e processar as suas mensagens (*offline*).

A característica *offline* do protocolo POP3 é particularmente útil para utilizadores que se ligam à Internet através de redes públicas comutadas, em que o custo da ligação é proporcional ao tempo de ligação (exemplo: a rede telefónica convencional ou a rede RDIS). Com o POP3, a ligação apenas precisa de estar activa durante a transferência das mensagens, e a leitura e processamento das mensagens pode depois ser efetuada com a ligação inactiva.

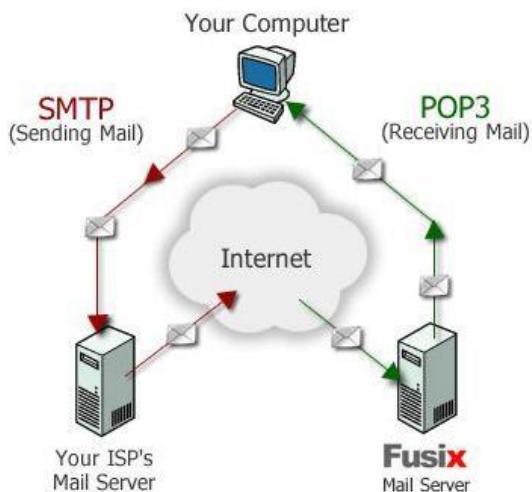


Fig. 7.20. Modelo ou Arquitectura do SMTP e POP3

IMAP

IMAP (*Internet Message Access Protocol*, traduzido do inglês, significa "Protocolo de acesso a mensagem da internet") é um protocolo de gerenciamento de correio electrónico.

Utiliza, por padrão, as portas TCP 143 ou 993 (conecção criptografada via SSL). O mais interessante é que as mensagens ficam armazenadas no servidor e o utilizador pode ter acesso a suas pastas e mensagens em qualquer computador, tanto por *webmail* como por cliente de correio electrónico (como o Mozilla Thunderbird, Outlook Express ou o Evolution). Outra vantagem deste protocolo é o compartilhamento de caixas postais entre usuários membros de um grupo de trabalho. Além disso, é possível efectuar pesquisas por mensagens directamente no servidor, utilizando palavras-chave.

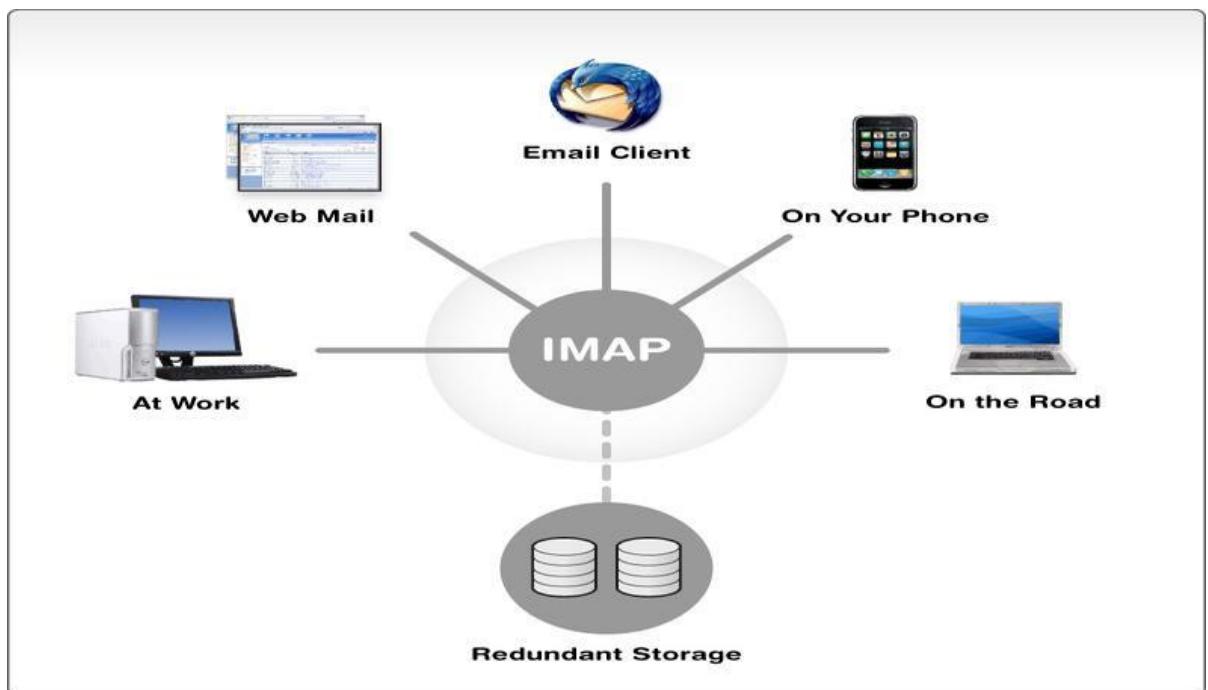


Fig. 7.21. Modelo ou Arquitectura do IMAP

O protocolo é superior em recursos ao POP3 - protocolo que a maioria dos provedores oferece aos seus assinantes. Por outro lado, apresenta algumas limitações:

O número de mensagens possível de se armazenar depende do espaço limite que nos é atribuído para a caixa de correio;

Caso o servidor IMAP esteja numa localização remota, pela Internet, e não numa rede local LAN, é necessário estar ligado à Internet todo o tempo que quisermos consultar ou enviar mensagens, podendo não ser adequado a quem utiliza a Internet através de ligação telefónica dial-up, devido aos custos associados. No entanto, a maioria dos clientes de *email* (e.g. Outlook Express, Thunderbird, Novell Evolution etc.) oferecem a possibilidade de criar uma cópia local (*offline*) das mensagens contidas em uma ou várias pastas (e.g. Inbox (Recebidas), Sent (Enviadas), etc.). Sendo assim, toda vez que você dispuser de uma conexão (estiver *online*) sua cópia local será sincronizada com o servidor de *email*.

Em relação ao POP3, permite activar e desactivar "flags" (marcações que indicam características de uma mensagem), que podem, inclusive, ser definidas pelo usuário. Com o POP3, estas marcações são registadas pelo cliente, de forma que, se a mensagem for aberta por um segundo cliente, as mesmas podem não ter seu "status" indicado correctamente. O IMAP permite a gravação das "flags" junto às caixas-postais, assegurando que, independente de qual cliente se accesse, as mensagens terão as mesmas correctamente atribuídas.

Possui ainda a capacidade de reconhecer os padrões de mensagens electrónicas [RFC 822] e MIME-IMB [RFC 2045] em mensagens electrónicas, de modo que os clientes de *email* não o necessitem fazer. O servidor IMAP cumpre a tarefa de interpretar estes padrões, tornando os clientes mais fáceis de implementar e o acesso mais "universal", bem como pesquisa de texto em mensagens de forma remota. Este modo de trabalho é feito localmente às caixas-postais e a selecção para recebimento dos atributos de uma mensagem, ou seu texto ou anexos e outras partes ("attachments") podem ser feitos de forma independente. Então, o usuário pode pedir para receber de uma mensagem com um grande "attachment", apenas a parte do texto que lhe interessa, o que é vantajoso no caso de um acesso discado de baixa qualidade e a redução do tráfego em geral.

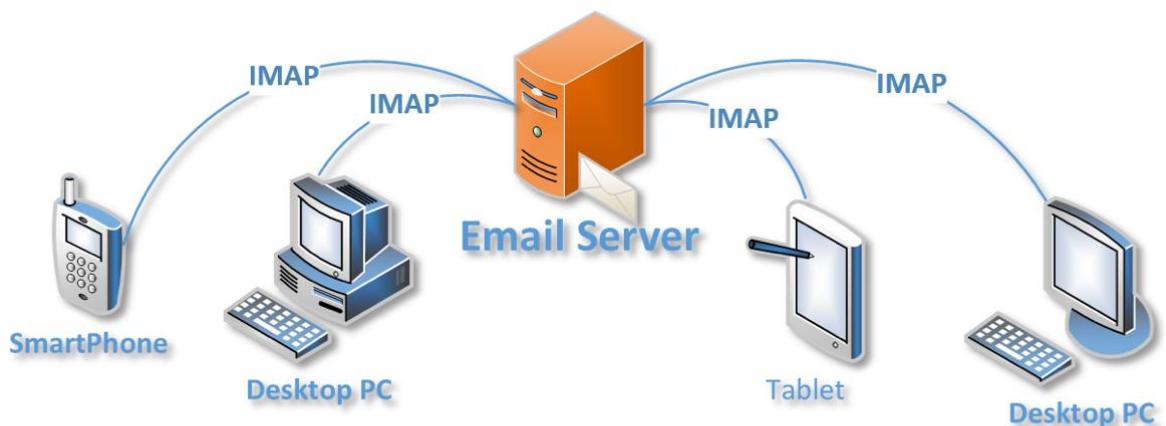


Fig. 7.22. Modelo ou Arquitectura do IMAP

DNS

O Sistema de Nomes de Domínio, mais conhecido pela nomenclatura em Inglês *Domain Name System* (DNS), é um sistema hierárquico e distribuído de gestão de nomes para computadores, serviços ou qualquer máquina conectada à Internet ou a uma rede privada. Faz a associação entre várias informações atribuídas a nomes de domínios e cada entidade participante. A sua utilização mais convencional associa nomes de domínios mais facilmente memorizáveis a endereços IP numéricos, necessários à localização e identificação de serviços e dispositivos, processo esse denominado por: resolução de nome. Em virtude do banco de dados de DNS ser distribuído, o seu tamanho é ilimitado [carece de fontes] e o desempenho não se degrada substancialmente quando se adicionam mais servidores. Por padrão, o DNS usa o protocolo *User Datagram Protocol* (UDP) na porta 53 para servir as solicitações e as requisições.

O DNS apresenta uma arquitectura cliente/servidor, podendo envolver vários servidores DNS na resposta a uma consulta. O servidor DNS resolve nomes para os endereços IP e de endereços IP para os nomes respectivos, permitindo a localização de *hosts* num determinado domínio.

Num sistema livre, o serviço é normalmente implementado pelo *software BIND*. Este serviço geralmente encontra-se localizado no servidor DNS primário. O servidor DNS secundário é uma espécie de cópia de segurança do servidor DNS primário. Assim, é uma parte necessária para quem quer usar a internet de uma forma mais fácil, evita que *hackers* roubem dados pessoais.

Existem centenas de servidores-raiz DNS (*root servers*) no mundo todo, agrupados em 13 zonas DNS raiz, das quais sem elas a Internet não funcionaria. Destes, dez estão localizados nos Estados Unidos da América, dois na Europa e um na Ásia. Para aumentar a base instalada destes servidores foram criadas réplicas localizadas por todo o mundo, inclusive no Brasil desde 2003.

Um recurso da internet, por exemplo um site da Web, pode ser identificado de duas maneiras: pelo seu nome de domínio, por exemplo, “www.wikipedia.org” ou pelo endereço de IP dos equipamentos que o hospedam (por exemplo, 208.80.152.130 é o IP associado ao domínio www.wikipedia.org). Endereços IP são usados pela camada de rede para determinar a localização física e virtual do equipamento. Nomes de domínio, porém, são mais mnemónicos para o usuário e empresas. É então necessário um mecanismo para resolver um nome de domínio em um endereço IP. Esta é a principal função do DNS.

Ocasionalmente, presume-se que o DNS serve apenas o objectivo de mapear nomes de hosts da Internet a dados e mapear endereços para nomes de host. Porém, o DNS pode armazenar uma grande variedade de tipo de dados, para praticamente qualquer finalidade.

Devido ao tamanho da Internet, armazenar todos os pares domínio - endereço IP em um único servidor DNS seria inviável por questões de escalabilidade, que incluem:

Disponibilidade: se o único servidor de DNS falhasse, o serviço se tornaria indisponível para o mundo inteiro;

Volume de tráfego: o servidor deveria tratar os pedidos DNS do planeta inteiro;

Distância: grande parte dos usuários estaria muito distante do servidor, onde quer que ele fosse instalado, gerando grandes atrasos para resolver pedidos DNS;

Manutenção do banco de dados: o banco de dados deveria armazenar uma quantidade de dados enorme e teria que ser actualizado com uma frequência muito alta (assim que um novo domínio fosse associado a um endereço IP).

A solução encontrada foi usar uma base de dados distribuída e hierárquica. Os servidores DNS se dividem nas seguintes categorias:

Servidores-raiz;

Servidores de domínio de topo;

Servidores com autoridade.

Com as três classes de servidores DNS, já é possível resolver qualquer requisição DNS. Basta fazer uma requisição a um servidor raiz, e esse retornará o endereço do servidor de topo responsável. Então repete-se a requisição para o servidor de topo, que retornará o endereço do servidor autoritativo ou algum intermediário. Repete-se a requisição aos servidores intermediários (se houver) até obter o endereço do servidor autoritativo, que finalmente retornará o endereço IP do domínio desejado.

Repare que essa solução não resolve um dos problemas de escalabilidade completamente: os servidores raiz tem que ser acessados uma vez para cada requisição que for feita em toda a internet. Esses servidores também podem estar muito longe do cliente que faz a consulta. Além disso, para resolver cada requisição, são precisas várias consultas, uma para cada servidor na hierarquia entre a raiz e o autoritativo.

Esta forma de resolver consultas é chamada de iterativa ou não-recursiva: cada servidor retorna ao cliente (ou ao servidor local requisitante, como explicado adiante) o endereço do próximo servidor no caminho para o autoritativo, e o cliente ou servidor local fica encarregado de fazer as próximas requisições.

Há também o método recursivo: o servidor pode se responsabilizar por fazer a requisição ao próximo servidor, que fará a requisição ao próximo, até chegar ao autoritativo, que retornará o endereço desejado, e esse endereço será retornado para cada servidor no caminho até chegar ao cliente. Esse método faz com que o cliente realize apenas uma consulta e receba directamente o endereço desejado, porém aumenta a carga dos servidores no caminho. Por isso, servidores podem se recusar a resolver requisições recursivas.

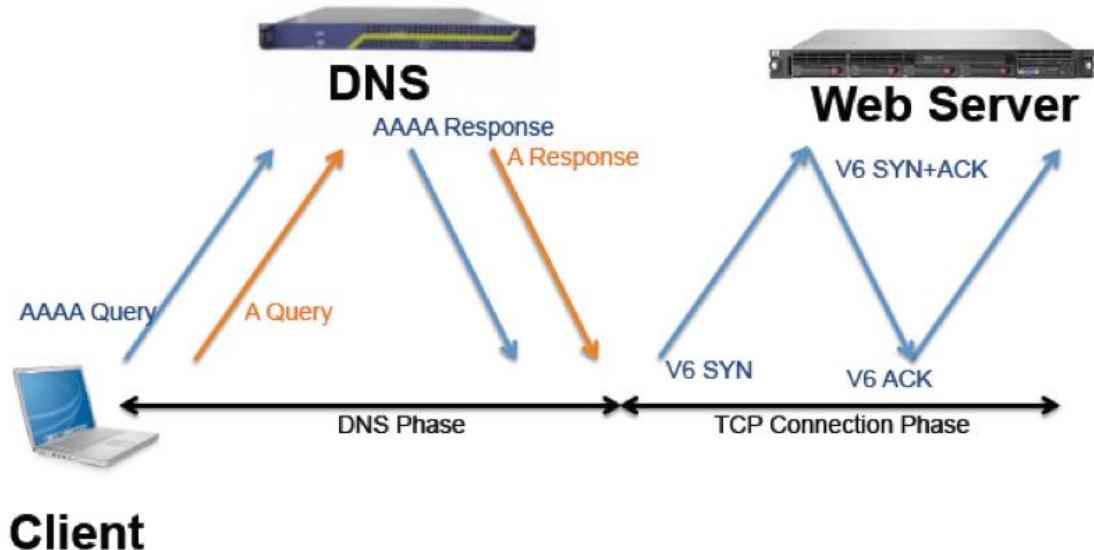


Fig. 7.23. Modelo ou Arquitectura do DNS

Dois recursos são usados em conjunto para reduzir a quantidade de requisições que os servidores raiz devem tratar e a quantidade de requisições feitas para resolver cada consulta: cache e servidor local.

Cache

Toda vez que um servidor retorna o resultado de uma requisição para a qual ele não é autoridade (método de resolução recursivo), ele armazena temporariamente aquele registo. Se, dentro do tempo de vida do registo (TTL, Time to Live), alguma requisição igual for feita, o servidor DNS pode retornar o resultado sem a necessidade de uma nova consulta. Note que isso pode provocar inconsistência, já que se um domínio mudar de endereço durante o tempo de vida do cache, o registo estará desactualizado. Apenas o servidor autoritativo tem a garantia de ter a informação correta. É possível exigir, na mensagem de requisição DNS, que a resposta seja

dada pelo servidor autoritativo.

Servidor local

Esse tipo de servidor não pertence à hierarquia DNS, mas é fundamental para o seu bom funcionamento. Em vez de fazer requisições a um servidor raiz, cada cliente faz sua requisição a um servidor local, que geralmente localiza-se fisicamente muito próximo do cliente, por exemplo um servidor proxy. Este se encarrega de resolver a requisição. Com o uso de cache, o servidor local pode ter a resposta pronta, ou ao menos conhecer algum servidor mais próximo ao autoritativo que o raiz (por exemplo, o servidor de topo), reduzindo a carga dos servidores raiz.

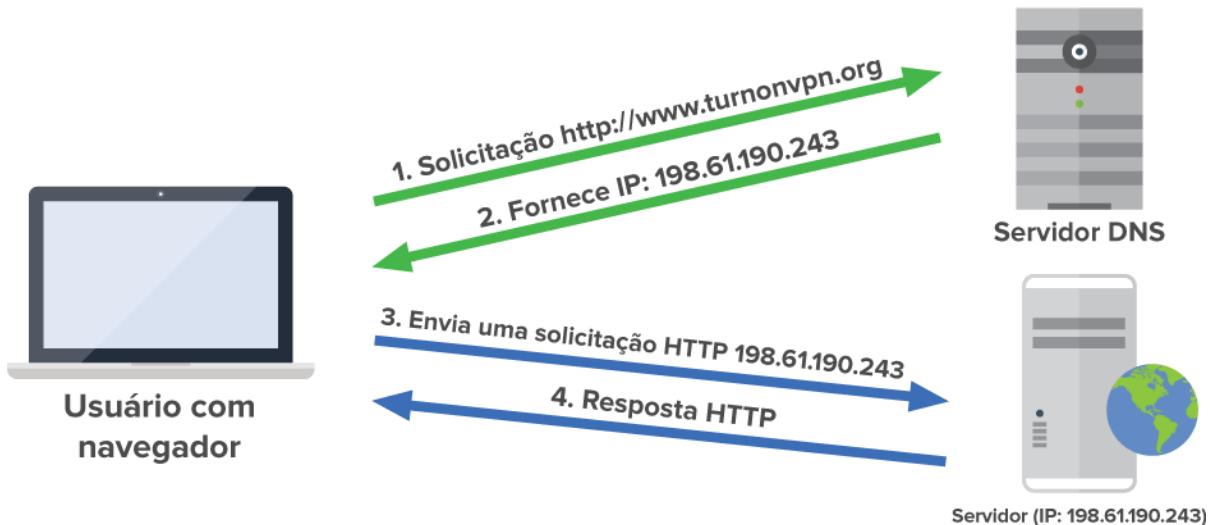


Fig. 7.24. Como funciona o DNS

Protocolo IP

Protocolo de Internet (em inglês: *Internet Protocol*, ou o acrônimo IP) é um protocolo de comunicação usado entre todas as máquinas em rede para encaminhamento dos dados. Tanto no Modelo TCP/IP, quanto no Modelo OSI, o importante protocolo da internet IP está na camada intitulada camada de rede.

Os dados numa rede IP que são enviados em blocos referidos como ficheiros (os termos são basicamente sinónimos no IP, sendo usados para os dados em diferentes locais nas camadas IP). Em particular, no IP nenhuma definição é necessária antes do nó tentar enviar ficheiros para um nó com o qual não comunicou previamente.

O IP oferece um serviço de datagramas não confiável (também chamado de melhor esforço); ou seja, o pacote vem quase sem garantias. O pacote pode chegar desordenado (comparado com outros pacotes enviados entre os mesmos nós), também podem chegar duplicados, ou podem ser perdidos por inteiro. Se a aplicação requer maior confiabilidade, esta é adicionada na camada de transporte.

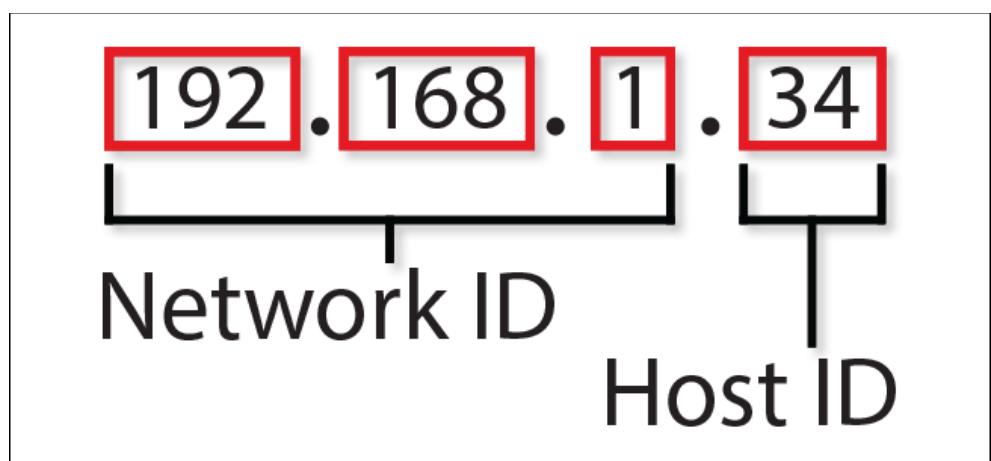
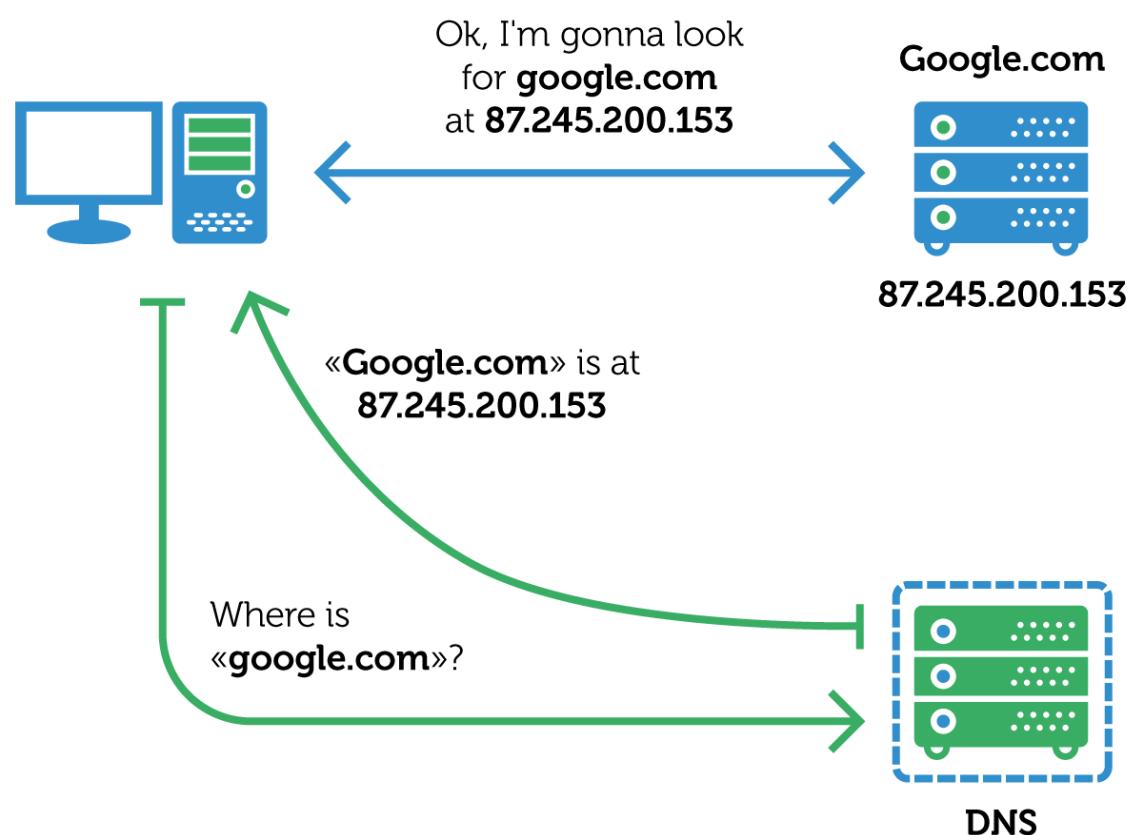


Fig. 7.25. Estrutura do Endereço IP

Os roteadores são usados para reencaminhar datagramas IP através das redes interconectadas na segunda camada. A falta de qualquer garantia de entrega significa que o desenho da troca de pacotes é feito de forma mais simplificada. (Note que se a rede cai, reordena ou de outra forma danifica um grande número de pacotes, o desempenho observado pelo utilizador será pobre, logo a maioria dos elementos de rede tentam arduamente não fazer este tipo de coisas - melhor esforço. Contudo, um erro ocasional não irá produzir nenhum efeito notável.)

O IP é o elemento comum encontrado na Internet pública dos dias de hoje. É descrito no RFC 791 da IETF, que foi pela primeira vez publicado em Setembro de 1981. Este documento descreve o protocolo da camada de rede mais popular e actualmente em uso. Esta versão do protocolo é designada de versão 4, ou IPv4. O IPv6 tem endereçamento de origem e destino de 128 bits, oferecendo mais endereçamentos que os 32 bits do IPv4.



© 2016 AO Kaspersky Lab. All Rights Reserved.

Fig. 7.26. Como funciona o DNS

Exercícios de AUTO-AVALIAÇÃO

1. O que faz com que a Internet caia?

Congestionamento: Uma sobrecarga de pessoas, todas tentando acessar a Internet a partir da mesma rede é a causa mais comum de interrupções da Internet.

Se você está constantemente reiniciando sua Internet, os switches e roteadores com falha podem ser o problema que você está enfrentando.

2. A Internet pode ficar sem espaço?

A internet ficou sem espaço - mas, felizmente, já fizemos mais alguns. A infra-estrutura que alimenta a internet foi feita com espaço para 4,3 bilhões de endereços. Isso pareceu muito na época, mas agora temos muitos dispositivos e o protocolo IPv4 ficou sem espaço.

3. A Internet pode ser desativada?

Tecnicamente, sim, especialistas da internet afirmaram na quarta-feira, logo após o governo do Egito ter restaurado conexões de internet, enquanto protestos políticos violentos continuavam.

“Desligar toda a internet seria praticamente impossível neste momento”, disse Jim Cowie, co-fundador da Renesys, um rastreador mundial de internet.

4. Quais são os 5 usos da Internet?

Com base em uma pesquisa recente sobre o tráfego na Internet, os 10 usos mais populares da Internet em ordem decrescente de uso são:

- Correio electrónico;
- Pesquisa;
- Download de arquivos;

- Grupos de discussão;
- Jogos interactivos;

- Educação e auto-aperfeiçoamento;
- Amizade e namoro;
- Jornais e revistas electrónicos.

5. Alguém possui a Internet?

Ninguém possui a internet. Vangie Beal descreve a internet como um conceito que “depende da infra-estrutura que conecta redes a redes”.

A ICANN é responsável por garantir que todos os nomes de domínio estejam vinculados ao endereço IP correcto, mas também não são proprietários da Internet.

6. O que são protocolos de Internet? Escreveos nomes de cinco protocolos comumente usados?

Vários protocolos são usados na Internet, incluindo correio electrónico (e-mail), protocolo de transferência de arquivos (FTP), HTTP (World Wide Web), notícias (ou Usenet), Gopher e Telnet:

- Correio electrónico;
- Protocolo de Transferência de Arquivos;
- HTTP (World Wide Web);
- Notícias (ou Usenet);
- Gopher;
- Telnet.

7. Quais são os protocolos e tipos de protocolos?

Um protocolo é um conjunto de regras que controla as comunicações entre computadores em uma rede:

IP / IPX (camada de rede);

TCP / SPX (camada de transporte);

HTTP, FTP, Telnet, SMTP e DNS (camadas combinadas de sessão / apresentação / aplicação).

8. Quais são os princípios básicos do protocolo?

Um protocolo de rede define regras e convenções para comunicação entre dispositivos de rede. Os protocolos de rede incluem mecanismos para dispositivos para identificar e fazer conexões uns com os outros, bem como regras de formatação que especificam como os dados são empacotados em mensagens enviadas e recebidas.

9. Como o ISPs fornecem Internet?

Quando você se conecta ao seu provedor, você se torna parte da rede deles. O ISP pode então se conectar a uma rede maior e se tornar parte de sua rede. A Internet é simplesmente uma rede de redes.

10. Qual é a função da Camada de Sessão OSI?

Essa camada fornece os protocolos e os meios para dois dispositivos na rede se comunicarem, mantendo uma sessão. Isso inclui a configuração da sessão, o gerenciamento do intercâmbio de informações durante a sessão e o processo de eliminação ao término da sessão.

11. Qual é a importância da Camada Física OSI?

A camada física faz a conversão de bits de dados em sinal eléctrico e vice-versa. É aqui que os dispositivos de rede e os tipos de cabo são considerados e configurados.

12. Quantas camadas existem sob TCP / IP?

Existem quatro camadas: a Camada de Rede, Camada de Internet, Camada de Transporte e Camada de Aplicação.

13. O que é o ICMP?

O ICMP é um protocolo de mensagens de controle da Internet. Ele fornece mensagens e comunicação para protocolos dentro da pilha TCP/IP. Esse também é o protocolo que gerencia mensagens de erro usadas por ferramentas de rede, como o PING.

14. Como funciona o modelo TCP IP?

A Internet funciona usando um protocolo chamado TCP/IP ou *Transmission Control Protocol/Internet Protocol*. Em termos básicos, o TCP/IP permite que um computador fale com outro computador através da Internet, compilando pacotes de dados e enviando-os para o local correcto.

15. Qual é o mais antigo, TCP IP ou OSI?

O modelo TCP/IP, que é realisticamente o modelo da Internet, surgiu cerca de 10 anos antes do modelo OSI. O modelo OSI (Open Standards Institute) é um modelo genérico aceito internacionalmente para todos os novos protocolos a serem projectados e protocolos mais antigos - como TCP/IP - aos quais se encaixam.

16. Qual é a principal diferença entre o TCP e o UDP?

O TCP (*Transmission Control Protocol*) é orientado para conexão, enquanto o UDP (*User Datagram Protocol*) é sem conexão. Isso significa que o TCP rastreia todos os dados enviados, exigindo confirmação para cada octeto (geralmente).

17. Qual é a diferença entre o protocolo TCP e IP?

Internet Protocol é uma conexão com menos protocolo de camada de rede e é responsável pelo envio dos pacotes aos dispositivos da rede. A diferença é que o TCP é responsável pela entrega de dados de um pacote e o IP é responsável pelo endereçamento lógico.

18. Qual é o mais rápido, TCP ou UDP?

O UDP é mais rápido que o TCP eo motivo simples é porque seu pacote de confirmação inexistente (ACK) permite um fluxo contínuo de pacotes, em vez do TCP que reconhece um conjunto de pacotes, calculado usando o tamanho da janela TCP e o tempo de ida e volta (RTT) .

19. O que usa o UDP?

O UDP é usado para alguns protocolos de actualização de roteamento, como o RIP (*Routing Information Protocol*). Normalmente usado para aplicações em tempo real que não podem tolerar atrasos desiguais entre secções de uma mensagem recebida. Após as implementações, o UDP é usado como um protocolo de camada de transporte: NTP (*Network Time Protocol*).

20. Qual é a diferença entre ISO e OSI?

OSI, interconecção de sistemas abertos, é um modelo de referência em redes que define como a comunicação deve ocorrer. ISO, organização internacional para padronização, é o órgão que criou o OSI. A principal diferença é que o OSI tem sete camadas, enquanto o modelo TCP/IP utiliza tudo isso em quatro.

21. O DNS usa TCP ou UDP?

O DNS usa o TCP para transferência de zona e o UDP para consultas de nome é regular (primário) ou reverso. O UDP pode ser usado para trocar pequenas informações, enquanto o TCP deve ser usado para trocar informações maiores que 512 bytes.

22. Como funcionam os protocolos de rede?

A Internet e a maioria das redes de dados funcionam organizando dados em pequenos pedaços chamados pacotes. Como protocolos como o Protocolo da Internet geralmente funcionam em camadas, alguns dados inseridos em um pacote formatado para um protocolo podem estar no formato de algum outro protocolo relacionado (um método chamado encapsulamento).

23. Quais são os protocolos comuns?

Os protocolos comuns da Internet incluem TCP/IP (Protocolo de Controle de Transmissões/Protocolo da Internet), UDP/IP (Protocolo de Datagrama do Usuário/Protocolo da Internet), HTTP (Protocolo de Transferência de HyperText) e FTP (Protocolo de Transferência de Arquivos). O TCP/IP é um protocolo de fluxo. Isso significa que uma conexão é negada entre um cliente e um servidor.

24. Qual protocolo pode ser usado para criptografar pacotes na Internet?

HTTPS, que significa *Hypertext Transfer Protocol Secure*, torna mais difícil para os *hackers*, a NSA (Agência de Segurança Nacional Americana) e outros usuários rastrearem.

TEMA 8. LINGUAGEM DE MARCAÇÃO (HTML, CSS)

Para criar uma página na Web é necessária a utilização da Linguagem de Marcação, porém algumas páginas não requerem esse tipo de linguagem. Existem várias linguagens de marcação, sendo a HTML uma das mais utilizadas. Os documentos produzidos nessa linguagem são interpretados pela maioria dos navegadores. O HTML teve origem na fusão do Hytime (que é um padrão utilizado para a representação estruturada de informação e hipermídia fundamentada no tempo) com o SGML (sugere uma diversidade de sintaxes de marcação que podem ser utilizadas por várias aplicações). Foi criado na década de 1990, por Tim Berners-Lee, o mesmo criador do WWW (*World Wide Web*).

O que é HTML?

Linguagem base dos *websites*.



Nesta unidade vai aprender as regras de funcionamento das linguagens de escrita para a criação de páginas Web (HTML, CSS) para apresentação, interacção e colecta de dados e informação através de páginas Web.

Objectivos Específicos

Unidade Temática 8.1: O que é HTML?

Camadas de desenvolvimento

Existem 3 linguagens básicas que utilizamos para criar *websites*: HTML, CSS e JavaScript.

O HTML é a linguagem que irá exibir a informação. O CSS é a linguagem que vai deixar essa informação bonitona. O JavaScript é a linguagem que vai fazer essa informação receber alguns comportamentos, como por exemplo ao criar um submenu ou controlar algo que aparece e desaparece na tela.

O HTML sem dúvida é a mais importante de todas, por que como dissemos no começo, é ela que exibe a informação. Além de exibir a informação, ela dá significado. Isso é importante por que alguns sistemas como o Google, que irão ler sua página, precisam entender o que é cada elemento nela e o que cada um desses elementos significam.

O nome HTML

O acrónimo HTML significa em inglês: *HyperText Markup Language*. Para gente aqui fica: Linguagem de

Marcação de Hipertexto. Bonito, né? Por trás das palavras Hipertexto e Marcação tem muita história e guardam a real essência da função do HTML. Você vai saber mais na próxima página, onde falamos sobre Semântica, que nada mais é do que a organização da informação usando HTML.

Se você tiver que guardar alguma coisa sobre o que é HTML, guarde isso: HTML serve para dar significado e organizar a informação dos *websites*.

Tendo isso em mente, você já saberá muito mais do que a maioria dos profissionais por aí.

Marcação

Já que o HTML serve para dar significado para a informação, como ele faz isso? Simples: ele marca a informação com as *tags*.

Por exemplo, para falarmos que um título é um título colocamos um pedaço de texto entre uma *tag* chamada H1. Veja o código abaixo:

```
<h1>Aqui vai o texto que é um título</h1>
```

E dessa forma vamos fazendo todos os outros elementos. Um parágrafo, por exemplo:

```
<p>Aqui vai muito texto, um parágrafo</p>
```

O resultado fica assim:

Aqui vai o texto que é um título

Aqui vai muito texto, um parágrafo

Para construir um website através da linguagem HTML, você deverá conhecer antes alguns *tags* HTML (são basicamente rótulos usados para informar ao navegador como deve ser apresentada a página da web), todos *tags* iniciam e terminam da mesma forma (começam com o sinal de menor “<” e são finalizados com o sinal de maior “>”). A seguir uma lista com os principais *tags* HTML;

<html> serve para dar início ao documento;
</html> finaliza os comandos da página;
<body> é utilizado para definir o conteúdo principal;
</body> finaliza o espaço destinado ao conteúdo principal;
<head> indica o cabeçalho da página;
</head> finaliza o cabeçalho da página;
<title> indica o título da página;
</title> finaliza o título da página;
<p> novo parágrafo;
</p> fim do parágrafo;

 pula linha;
</br> serve para fechar a linha;
<table> serve para inserir tabela;
</table> finaliza a tabela

O que é Semântica?

Organização e Significado

Quando eu era criança era costume fazer muitos trabalhos para a escola. O processo era quase sempre igual: a professora definia o assunto e ao chegar em casa, pegava um bocado de papel almoço e ia para casa da minha tia. Ela era a única da família que tinha encyclopédias. Se eu precisava fazer um trabalho escolar, certamente as encyclopédias da minha tia me ajudariam na tarefa. As tais encyclopédias eram pesadas. Chatas de serem manipuladas. Mas tinham tudo ou quase tudo do que precisava para completar o meu trabalho com o máximo de detalhes possível.

Organizando a informação: forma linear ou não linear

Você já pensou em como se organiza informação? A informação - um texto por exemplo - pode ser organizada de forma linear ou não linear. Imagine um livro. Toda a informação contida no livro está organizada de forma linear, isto é, há começo, meio e fim. Você não consegue absorver todo o significado da história contada se iniciar sua leitura pela metade do livro. Você precisa, necessariamente, iniciar do primeiro capítulo até o último para absorver o máximo possível de informação sobre a história.

As encyclopédias têm suas informações organizadas de forma não linear. Isto é: as informações não estão organizadas em uma ordem específica, mas sim de forma relacional, associativa. Há uma série de assuntos e todos eles estão organizados de uma maneira que se relacionam por meio de referências. Por exemplo: quando você procura informações sobre carros na encyclopédia, ao final do texto você encontra uma série de outras referências aos assuntos relacionados, como por exemplo: motores a combustão interna, rodas, combustíveis e etc.

Como você e a web funcionam

É mais ou menos assim que a sua mente funciona também: quando você pensa em um assunto, você consegue fazer uma série de associações com outros assuntos, formando uma ideia, uma memória, uma lembrança. Por isso seu cérebro consegue guardar essas informações de um jeito que se torna fácil de recuperá-las não apenas pensando nessa informação directamente, mas também quando se pensa nos assuntos relacionados a esta informação.

Se levarmos isso para web, percebemos que apenas uma *tag* faz esse trabalho de referência e associação de informação: o *link*. Na web você relaciona informações, sites e etc. utilizando directamente a *tag* "A". Quando você linka um texto, uma imagem, um gráfico etc., você associa essa informação ao destino do *link*. Você está referenciando assuntos, como na encyclopédia. Sem os *links* não há web. Não conseguimos associar informação alguma. Perde-se a referência. Nós usamos o *link* todos os dias e aposto que a metade dos desenvolvedores não imagina a importância dessa tag.

Unidade Temática 8.2: Codificando em HTML

O que são *Tags*, Elementos e Atributos?

Conhecendo as principais *Tags*, Elementos e seus Atributos

O que são *Tags*?

Tags são o conjunto de caracteres que formam um elemento, ou seja, quando nos referenciamos à Tag "p" estamos falando disso: <p>

Existem dois tipos de *Tags*, as que necessitam de fechamento e as que não necessitam de fechamento.

Para as que necessitam de fechamento, utilizamos o sinal de menor (<), seguido do nome do elemento e o sinal de maior (>) para abertura.

Para fechamento, utilizamos o sinal de menor (<), seguido de barra (/), nome do elemento e o sinal de maior (>).

<!-- Exemplo de elemento que necessita de fechamento -->

<p>A tag do elemento de parágrafo necessita de fechamento.</p>

Os elementos que não necessitam de fechamento, também conhecidos como elementos vazios, somente utilizamos o sinal de menor (<), seguido do nome do elemento e o sinal de maior (>).

<!-- Exemplo de elemento vazio -->

Texto utilizando

quebra de linha

O que são Elementos?

Elementos são formados a partir de *Tags* e entre as *Tags* é que está o conteúdo do Elemento.

Se quisermos exibir um parágrafo em um *site* utilizamos a *Tag* <p> que semanticamente quer dizer que o conteúdo que se espera nesse Elemento é um parágrafo.

Alguns exemplos:

<!-- A Tag <label> define que o conteúdo do Elemento é uma label (rótulo) -->

<label>Informe o seu nome</label>

<!-- A Tag <address> define que o conteúdo do Elemento é um endereço (endereços físicos à virtuais) -->

<address>

Este guia é uma iniciativa da comunidade e do Tableless www.tableless.com.br</address>

Não confunda *Tags* com Elementos

As *Tags* servem para marcar onde começam e terminam os Elementos. Já os Elementos são uma parte conceitual/semântica que têm um começo e fim determinados.

Parece uma diferença boba, mas mantenha ela sempre em mente e isso vai fazer toda a diferença no seu entendimento de HTML.

O que são Atributos?

Atributos são informações que passamos na *Tag* para que ela se comporte da maneira esperada. Existem atributos globais (que funcionam em todas as *Tags*) e específicos (que são direcionados para cada *Tag*, através de especificação).

Os Atributos possuem nome e um valor, existem Atributos que você vai usar praticamente sempre e existem outros que serão mais raros.

Atributos globais

Seguem alguns atributos globais importantes e suas descrições simplificadas:

accesskey

Determina uma (ou mais) tecla(s) de atalho para o elemento. Para definir mais de uma tecla, coloque-as separadas por espaço.

class

Determina uma (ou mais) classe(s) para o elemento. Para definir mais de uma classe, coloque-as separadas por espaço.

draggable

Define se um elemento é arrastável ou não.

hidden

Oculta o elemento onde for declarado.

id

É o identificador único do elemento. Somente deve ser declarado um id por elemento. E este id não deve ser repetido na mesma página.

lang

Determina o idioma em que está escrito o conteúdo do elemento.

style

Determina propriedades CSS diretamente no elemento.

tabindex

Organiza os elementos por ordem de tabulação. Deve-se usar valor numérico.

title

Representa um auxílio ao elemento. Semelhante a dica do elemento.

Como você pôde ver, a sintaxe de Atributos é muito simples, porém, por mais simples que seja é sempre bom ter em mente boas práticas para que se tenha qualidade e fácil manutenção em seu HTML.

Existem Atributos em que os valores não precisam ser passados entre aspas, mas manter um padrão ajuda você e ajuda quem quer que seja que no futuro tenha que trabalhar com o seu HTML. Portanto, não é uma regra mas uma boa prática você envolver os valores dos Atributos entre aspas.

<!-- Isso funciona, mas não é recomendável -->

```
<a href="http://tableless.com.br"
```

```
target=_blank>www.tableless.com.br</a>
```

<!-- Isso funciona e é recomendável -->

```
<a href="http://tableless.com.br"
```

```
target="_blank">www.tableless.com.br</a>
```

Na prática

Agora que você entendeu separadamente o papel das *Tags*, Elementos e Atributos, vamos ver um exemplo prático:

<!-- A Tag define que o conteúdo do Elemento é uma imagem e os atributos que utilizamos são src e alt -->

```

```

Analisando o exemplo:

No exemplo acima definimos o caminho onde está a imagem com o Atributo src e utilizamos o Atributo alt para descrever que imagem é essa (a utilização do atributo alt é altamente recomendado, pois, mesmo que a imagem não seja carregada por qualquer motivo, o usuário conseguirá identificar que ali era para ser carregado o logo do HTML5).

Unidade Temática 8.3: Estrutura Básica HTML

Iniciando o código básico de HTML

O documento HTML sempre inicia com o que chamamos de estrutura básica. Esta estrutura é quase que imutável. Sempre será dessa forma e você sempre, sempre começará seu HTML começando por esse código. Geralmente os editores como o *Sublime Text* já tem atalhos para iniciar os documentos HTMLs com essa estrutura, logo, você não precisa se preocupar em decorá-la, mas é bom que faça. Veja abaixo como ela se inicia:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Título da página</title>
<meta charset="utf-8">
```

```
</head>
<body>
    Aqui vai o código HTML que fará seu site aparecer.
</body>
</html>
```

É possível compreender o documento em HTML de uma maneira muito simples, através de uma divisão de blocos das *tags* essenciais, conforme a seguinte estrutura:

Definição do documento (doctype)
Cabeça (head)
Corpo (body)
Doctype - Definindo o documento

Uma coisa importante: SEMPRE deve existir o doctype, que é este código <!DOCTYPE html>.

O doctype não é uma *tag* HTML, mas uma instrução para o navegador e outros programas que podem ler seu site, que o código encontrado ali é um código HTML. Assim eles sabem o que fazer para mostrar seu site da melhor forma possível. Lembre-se: o doctype é OBRIGATÓRIO e deve ser sempre a PRIMEIRA LINHA do seu documento.

HEAD

Contém informações que não são transpostas visivelmente para o usuário/leitor do documento. São dados implícitos, de uso e controle do documento: vinculação com outros arquivos, aplicação de lógica de programação de scripts e metadados. Na prática, todo o conteúdo do cabeçalho fica delimitado entre a abertura e fechamento *tag* head.

BODY

Trata-se do documento em si, ou seja, a informação legível para o usuário/leitor do documento. É todo e qualquer texto que se deseja apresentar, assim como toda e qualquer forma de mídia de saída (imagens, sons, miniaplicativos embutidos, conteúdo multimídia, etc).

Além disso, toda a apresentação de entrada de dados (formulários) também se aplica nesta secção do documento. Na prática, o corpo do documento é delimitado pelo par de tags `<body>` e `</body>`.

Este é o preceito básico que deve estar muito bem claro para você: onde as marcações se aplicam, e quais são os resultados deste modelo. Por exemplo: se você deseja informar conteúdo textual para saída legível ao usuário do seu sistema web, esta marcação deverá obrigatoriamente estar no bloco do corpo da página. Ainda: para definir qual o tipo de codificação da página (uma meta informação do documento), esta deve obrigatoriamente estar marcada no cabeçalho do mesmo documento.

Dentro do elemento BODY sua estrutura de página terá os elementos semânticos da construção da sua página, onde serão declarados e identificados cabeçalhos, rodapé, conteúdo principal, etc.



Fig. 8.1. Estrutura Básica do HTML

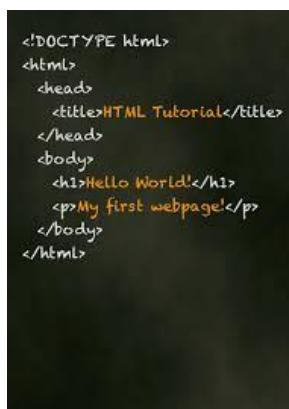


Fig. 8.2. Estrutura Básica do HTML

O que são *metatags*?

Informação sobre seu *site*.

A *Meta Tag*, representada pela tag `<meta>` é uma tag diferenciada das demais por não ter nenhum efeito aparente na página em si, mas sim por ser responsável por ações externas à página, como por exemplo informar à buscadores como Google ou Bing algumas informações a respeito da página, como título e uma breve descrição.

Tipos de *Meta Tags*

Author

O nome do autor da página.

Copyright

Direito autoral da página.

Description

Descrição da página.

Expires

Data em que o documento deve ser considerado expirado.

Keywords

Aqui é um dos locais onde os motores de busca procuram informações a respeito da página.

Unidade Temática 8.4:CSS

Cascading Style Sheets (CSS) é um mecanismo para adicionar estilo (cores, fontes, espaçamento, etc.) a um documento web.

Em vez de colocar a formatação dentro do documento, o CSS cria um *link* para uma página que contém os estilos. Quando quiser alterar a aparência do portal, basta modificar apenas um arquivo.

O que é CSS?

O *Cascading Style Sheets* (CSS) é uma linguagem utilizada para definir a apresentação (aparência) de documentos que adoptam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML e etc). O CSS define como serão exibidos os elementos contidos no código de um documento e sua maior vantagem é efectuar a separação

entre o formato e o conteúdo de um documento.

Por que o CSS foi criado?

Com a evolução dos recursos de programação, as tecnologias estavam adoptando cada vez mais estilos e variações para deixá-las mais elegantes e atractivas para os usuários. Com isto, linguagens de marcação simples como o HTML, que era destinada para apresentar os conteúdos, também precisaram ser aprimoradas.

Foram criadas novas *tags* e atributos de estilo para o HTML e em resumo, ele passou a exercer tanto a função de estruturar o conteúdo quanto de apresentá-lo para o usuário final. Entretanto, isto começou a trazer um problema para os desenvolvedores, pois não havia uma forma de definir, por exemplo, um padrão para todos os cabeçalhos ou conteúdos em diversas páginas. Ou seja, as alterações teriam que ser feitas manualmente, uma a uma.

A partir destas complicações, nasceu o CSS. Primariamente, foi desenvolvido para habilitar a separação do conteúdo e formato de um documento (na linguagem de formatação utilizada) de sua apresentação, incluindo elementos como cores, formatos de fontes e *layout*. Esta separação proporcionou uma maior flexibilidade e controle na especificação de como as características serão exibidas, permitiu um compartilhamento de formato e reduziu a repetição no conteúdo estrutural de um documento.

Com isto, as linguagens de marcação passaram novamente a exercer sua função de marcar e estruturar o conteúdo de um documento, enquanto o CSS encarregou-se da aplicação dos estilos necessários para a aparência dela. Isto é feito por meio da criação de um arquivo externo que contém todas as regras aplicadas e com isto, é possível fazer alterações de estilo em todas as páginas de um *site* e outros documentos que utilizam CSS de forma fácil e rápida.

O CSS também permite que as mesmas marcações de um documento sejam apresentadas em diferentes estilos, conforme os métodos de renderização (como em uma tela, impressão, via voz, baseadas em dispositivos tácteis, etc). A maioria dos menus em cascata, estilos de cabeçalho e rodapé de páginas da internet, por exemplo, actualmente são desenvolvidos em CSS.



Fig. 8.3. Pilha de *Software HTML, CSS e Javascript*

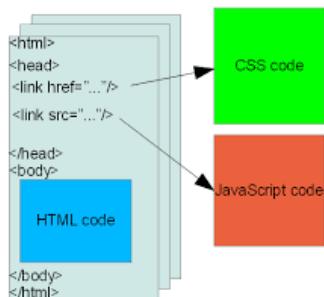


Fig. 8.4. Pilha de *Software HTML, CSS e Javascript*

Com a variação de actualizações dos navegadores, o suporte ao CSS pode variar. A interpretação dos navegadores pode ser avaliada com o teste Acid2, que se tornou uma forma base de revelar quão eficiente é o suporte de CSS, fazendo com que a nova versão em desenvolvimento do Firefox seja totalmente compatível a ele, assim como o Opera já é. O *Doctype* informado, ou a ausência dele, determina o *quirks mode* ou o *strict mode*, modificando o modo como o CSS é interpretado e a página desenhada.

Definição de estilo é um conjunto de propriedades visuais para um elemento, o CSS define regras que fazem as definições de estilo casarem com um elemento ou grupo de elemento, o documento pode conter um

bloco de CSS num elemento *style* ou usando o *element link* apontando para um arquivo externo que contenha o bloco CSS.

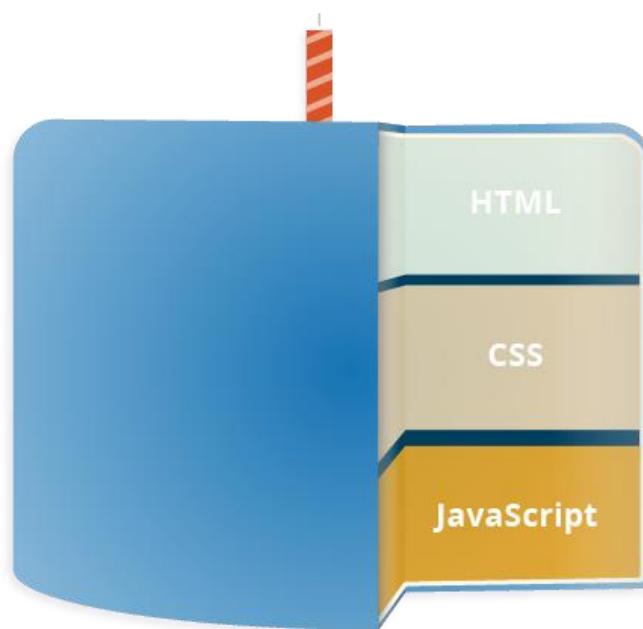


Fig. 8.4. Pilha de Software CSS e Javascript

Para uso com o CSS, foi criado o atributo class que todo elemento pode conter.

As regras de casamento para o CSS são chamadas de selectores, uma definição de estilo pode ser casada com um selector ou um grupo de selectores separados por vírgula.

Unidade Temática 8.5: Escrevendo CSS

Como escrever código CSS

Sintaxe

Para escrever código CSS é necessário seguir uma regra. A regra é uma declaração que possui uma sintaxe própria bem simples que define a forma com que o estilo será aplicado aos nossos elementos HTML. Você pode ver a regra a seguir:

```
seletor{  
    propriedade: valor;  
}
```

A nossa regra como pode ser visto é composta de três partes principais: um seletor, uma propriedade e um valor a se aplicar.

Explicando de forma grotesca o seletor nada mais é do que o nosso elemento HTML ao qual queremos aplicar a regra (por exemplo: div, body).

A propriedade é o atributo do elemento que será aplicado a regra. (por exemplo: color, font, position).

Valor é a característica que o elemento irá assumir (por exemplo: cor azul, tamanho 14 para a fonte).

Exemplificando

Vamos para um exemplo:

```
body{  
    background-color: #000;  
}
```

No nosso exemplo o elemento ao qual estamos aplicando a regra é o BODY este é o nosso seletor, a nossa propriedade é background-color que define a cor de fundo do documento e o nosso valor #000 que é igual a cor preta.

Incluindo CSS na página

Existem três formas para incluir o código CSS em seu projecto

Inline

A primeira forma de aplicar CSS a uma página é utilizando o atributo style em elementos do HTML:

```
<p style="color: blue">Parágrafo com fonte azul.</p>
```

<p>Esse outro parágrafo não é azul, a não ser que exista CSS em outro lugar.</p>

Interno

A segunda forma é utilizar a tag style dentro do head da página HTML:

```
<head>
<style type="text/css">
    seletor { propriedade: valor; }
</style>
</head>
```

Externo

E a última - porém a mais utilizada - maneira de aplicar CSS é criar um ou mais arquivos com extensão .css e incluí-los na estrutura head do HTML:

```
<head>
<link rel="stylesheet" type="text/css" href="reset.css">
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
```

Qual a diferença entre CLASS e ID?

Em HTML e CSS, há a possibilidade de aplicar estilos através de 'class' e 'id' e, em JavaScript é possível identificar algum elemento de uma página por sua classe, id ou tag. Mas qual a diferença entre 'class' e 'id'?

O que são classes?

As classes são uma forma de identificar

um grupo de elementos. Através delas, pode-se atribuir formatação a VÁRIOS elementos de uma vez. Exemplo:

Código CSS:

```
.classe1 {  
    background: blue;  
}
```

Código HTML:

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
<title></title>  
<meta charset="utf-8">  
</head>  
<body>  
<div class="classe1">Div1</div>  
<div class="classe1">Div2</div>  
<div class="classe1">Div3</div>  
<div class="classe1">Div4</div>  
<div class="classe1">Div5</div>  
</body>  
</html>
```

Então, todas as 'divs' que estiverem com a classe "classe1" estarão com um background azul(blue).

O que são ids?

As ids são uma forma de identificar um elemento, e devem ser ÚNICAS para cada elemento. É como se fossem impressões digitais de nossos dedos ou RGs. Através delas, pode-se atribuir formatação a um elemento em especial. Exemplo:

Código CSS:

```
#idUm {  
    background: blue;  
}  
#idDois {
```

```
background: yellow;
}
#idTres {
background: lightblue;
}
#idQuatro {
background: lightgreen;
}
```

Código HTML:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title></title>
<meta charset="utf-8">
</head>
<body>
<div id="idUm">Div1</div>
<div id="idDois">Div2</div>
<div id="idTres">Div3</div>
<div id="idQuatro">Div4</div>
</body>
</html>
```

Então, como mostra o código acima, não podemos repetir uma 'id'. Deve ser especialmente única para cada elemento.

Seleccionando elementos

Entenda como formatar os elementos do CSS

Selectores CSS ajudam a especificar melhor o elemento que deve ser afectado.

Vejamos alguns selectores:

Elemento Filho (A > B)

Para seleccionar um elemento (B) que tenha um elemento pai específico (A), deve-se utilizar o ">".

Ex.:

```
nav.menu-principal > ul {  
    padding: 0;  
}  
// neste caso, somente o elemento ul, que tem como pai um elemento  
nav com classe "menu-principal", receberá a alteração de  
preenchimento
```

Elemento Irmão (A + B)

Quando um elemento (B) está ao lado de outro específico (A).

Ex.:

```
label + input {  
    border: 1px solid #333;  
}  
// todos os inputs que estiverem ao lado de um label receberão borda  
cinza
```

Não este elemento (A:not(b))

Exclui determinado elemento (A) de acordo com a especificação (b).

```
div:not(.principal) {  
    border: 1px solid #FO0;  
}  
// todas as divs que não tenham a classe "principal" receberão borda  
vermelha
```

Primeiro Filho (A:first-child)

Seleciona o primeiro elemento que aparecer, de acordo com o elemento definido (A).

```
li:first-child {  
    background-color: #0F0;  
}  
// O primeiro li terá fundo verde
```

Enésimo Elemento (A:nth-child(b))

Seleciona o elemento definido (A) de acordo com a posição informada (b).

```
li:nth-child(7) {  
    background-color: #666;  
}  
// O sétimo li terá fundo cinza
```

Propriedade Background

Inserindo fundo nos elementos

O background trata do fundo dos elementos. O elemento que mais recebe esta propriedade é o body.

As propriedades do background são:

color

Define a cor do fundo.

Possíveis valores:

inherit: Herda o valor do elemento pai.

transparent: Define cor como transparente.

<cor>: Define uma determinada cor.

```
div {  
    background-color: gray;  
}
```

image

Define a imagem de fundo.

Possíveis valores:

inherit: Herda o valor do elemento pai.

none: Indica que não há imagem de fundo.

url(): Define a localização da imagem de fundo.

```
div {
```

```
    background-image: url("../img/bg.png");
```

```
}
```

repeat

Define a posição de repetição da imagem de fundo.

Possíveis valores:

inherit: Herda o valor do elemento pai.

repeat: Repete a imagem vertical e horizontalmente.

repeat-x: Repete a imagem horizontalmente.

repeat-y: Repete a imagem verticalmente.

no-repeat: Não repete a imagem.

```
div {
```

```
    background-repeat: no-repeat;
```

```
}
```

attachment

Define se a imagem definida irá rolar juntamente com o viewport, ou se irá ficar fixo.

Possíveis valores:

inherit: Herda o valor do elemento pai.

fixed: Mantém a imagem de fundo fixa na posição definida independente da rolagem do viewport.

scroll: A imagem de fundo acompanha a rolagem do viewport.

```
div {  
    background-attachment: fixed;  
}
```

position

Define a posição da imagem de fundo. Observe que esta propriedade necessita de atenção de acordo com o background-repeat definido.

Possíveis valores:

inherit: Herda o valor do elemento pai.

<porcentagem>: sendo A e B dois números, quando declaramos o position com A% e B%, a posição A,B da imagem ficará na posição A%, B% do elemento.

<comprimento>: sendo A e B dois números, quando declaramos o position com Apx e Bpx, a posição A,B da imagem ficará na posição Apx, Bpx do elemento.

top: Equivale a 0% na posição vertical.

bottom: Equivale a 100% na posição vertical.

left: Equivale a 0% na posição horizontal.

right: Equivale a 100% na posição horizontal.

center: Caso não tenha sido declarado um valor horizontal, equivale a 50% na posição horizontal, ou, caso não tenha sido declarado um valor vertical, equivale a 50% na posição vertical.

```
div {  
    background-position: top center;  
}
```

background

É uma propriedade abreviada que une todas as propriedades do background.

```
div {  
    background: gray url("../img/bg.png") no-repeat fixed top center;  
}
```

Propriedade Font

Formatando fonts de texto

A font trata das características de todos os textos do site. Ela pode ser aplicada directamente ao elemento body e por consequência todos os elementos filhos herdarão esta propriedade.

As propriedades da font são:

family

Determina a família da font a ser usada. Existem dois tipos básicos: as famílias e as famílias genéricas.

As famílias genéricas podem ser três:

serif: Font com serifa.

sans-serif: Font sem serifa.

monospace: Font com a mesma largura em todos os caracteres.

Serifa nada mais é do que aquelas pontinhas que tem em algumas fontes tipo Times, ou seja, Times é uma font com serifa e Arial é uma font sem serifa.

```
div {
```

```
    font-family: "Times New Roman", Times, serif;;
```

}

Entre aspas vai o nome da fonte, seguido da família (Times) e depois da família genérica (serif).

Propriedade Display

Entendendo e manipulando o fluxo dos elementos

Entender a propriedade display é fundamental para que possamos compreender o fluxo e estruturação de uma página web. Todos os elementos por padrão já possuem um valor para a propriedade e, geralmente estas são block ou inline.

Essa propriedade CSS especifica o tipo de renderização do elemento na página. Uma definição que pode auxiliar no entendimento da propriedade é a do Chris Coyier, bastante reconhecido no mundo do CSS, que é a seguinte:

Todo elemento em uma página web é renderizado como uma caixa rectangular. A propriedade display de CSS vai determinar como essa caixa vai se comportar.

Os possíveis tipos

Block

O elemento se comporta como um bloco. Ocupando praticamente toda a largura disponível na página. Elementos de parágrafo (p) e título(h1, h2, ...) possuem esse comportamento por padrão.

Inline

O elemento se comporta como um elemento em linha. Exemplos de elemento que se comportam assim são por exemplo as tags span e a.

None

Ao contrários dos valores atuais, o valor none permite, informalmente dizendo, que você desactive a propriedade do elemento. Quando você utiliza essa propriedade, o elemento e todos seus elementos filhos não são renderizados na página.

Uma coisa importante a ressaltar que a propriedade display: none não é a mesma coisa da propriedade visibility: hidden. Nessa última o elemento não aparece na tela mas é renderizado na página, ou seja, vemos um espaço vazio no lugar do elemento; já a propriedade display: none não renderiza o elemento e, o espaço que era ocupado por ele, é ocupado pelo elemento seguinte no fluxo do documento.

Table

O elemento se comporta como uma tabela.

Inline-block

Semelhante ao inline, no entanto, ao definirmos inline-block em um elemento, conseguimos definir as propriedades de largura e altura para ele. Coisa que não conseguimos em um elemento com display: inline.

Propriedade Margin e Padding**Manipulando espaços nos elementos****Margin**

É a margem do elemento, ou seja, o espaçamento externo do elemento.

Algumas formas de escrita:

margin-top

Define a margem superior do elemento.

```
div {  
    margin-top: 10px;  
}  
margin-right  
Define a margem direita do elemento.
```

```
div {  
    margin-right: 20px;  
}  
margin-bottom  
Define a margem inferior do elemento.
```

```
div {  
    margin-bottom: 30px;  
}  
margin-left  
Define a margem esquerda do elemento.
```

```
div {  
    margin-left: 40px;  
}  
margin  
Define a margem do elemento. Este tipo de declaração pode ser dar de diversas formas.
```

```
/* t = topo; d = direita; b = baixo; e = esquerda; */  
.div1 {  
    /* t = 20px; d = 20px; b = 20px; e = 20px; */  
    margin: 20px;  
}
```

```
.div2 {  
/* t = 10px; d = 20px; b = 10px; e = 20px; */  
margin: 10px 20px;  
}
```

```
.div3 {  
/* t = 10px; d = 20px; b = 30px; e = 20px; */  
margin: 10px 20px 30px;  
}
```

```
.div4 {  
/* t = 10px; d = 20px; b = 30px; e = 40px; */  
margin: 10px 20px 30px 40px;  
}
```

Padding

É o preenchimento do elemento, ou seja, o espaçamento interno do elemento.

Algumas formas de escrita:

padding-top

Define o preenchimento superior do elemento.

```
div {
```

padding-top: 10px;

```
}
```

padding-right

Define o preenchimento direito do elemento.

```
div {
```

padding-right: 20px;

```
}
```

padding-bottom

Define o preenchimento inferior do elemento.

```
div {  
    padding-bottom: 30px;  
}  
padding-left  
Define o preenchimento esquerdo do elemento.
```

```
div {  
    padding-left: 40px;  
}  
padding  
Define o preenchimento geral do elemento. Assim como na margem, o padding pode declarado de várias formas.
```

```
/* t = topo; d = direita; b = baixo; e = esquerda; */  
.div1 {  
    /* t = 20px; d = 20px; b = 20px; e = 20px; */  
    padding: 20px;  
}
```

```
.div2 {  
    /* t = 10px; d = 20px; b = 10px; e = 20px; */  
    padding: 10px 20px;  
}
```

```
.div3 {  
    /* t = 10px; d = 20px; b = 30px; e = 20px; */  
    padding: 10px 20px 30px;  
}
```

```
.div4 {  
    /* t = 10px; d = 20px; b = 30px; e = 40px; */  
    padding: 10px 20px 30px 40px;  
}
```

Propriedade Position

Posicionando elementos via coordenadas

Determina a posição do elemento.

Possíveis valores:

relative

A posição do elemento é relativa ao elemento anterior.

```
aside {  
    position: relative;  
}
```

fixed

A posição do elemento é relativa ao viewport. Esta posição não é influenciada pela rolagem da página.

```
aside {  
    position: fixed;  
}
```

absolute

A posição do elemento é relativa ao viewport, ou ao elemento pai quando este tem um position definido. Esta posição é influenciada pela rolagem da página.

```
aside {  
    position: absolute;
```

}

```
h1 { color: white;  
background: orange;  
border: 1px solid black;  
padding: 0 0 0 0;  
font-weight: bold;  
}  
/* begin: seaside-theme */  
  
body {  
background-color:white;  
color:black;  
font-family:Arial,sans-serif;  
margin: 0 4px 0 0;  
border: 12px solid;
```

CSS

Fig. 8.5. Estrutura Básica do CSS

```
@charset "UTF-8";
/* CSS Document */
body
{
background-color:#d0e4fe;
}
h1
{
color:orange;
text-align:center;
}
p
{
font-family:"Times New Roman";
font-size:20px;
}
```

Fig. 8.6. Estrutura Básica do CSS

Exercícios de AUTO-AVALIAÇÃO

1. Como você explica o HTML?

HTML significa *HyperText Markup Language*, uma linguagem de marcação é uma linguagem de computador que define a estrutura e a apresentação do texto bruto. As linguagens de marcação trabalham ao redor do texto bruto com informações que o computador pode interpretar, “marcando-o” para processamento.

2. Como funciona o código HTML?

O HTML consiste em uma série de códigos curtos digitados em um arquivo de texto pelo autor do site - essas são as *tags*. O texto é salvo como um arquivo html e visualizado por meio de um navegador, como o Internet Explorer ou o Netscape Navigator.

3. Quais são as principais características do HTML?

Funcionalidades da Web: as principais partes do HTML

- HTML básico cuida da maioria dos recursos usuais, como parágrafos, títulos, listas numeradas e com marcadores, e assim por diante.
- Se você quiser mais controle de recursos de estilo, como fontes, cores e posicionamento, o padrão *Cascading Style Sheets* (CSS) é a melhor maneira de fornecer esse controle.

4. Qual é o propósito de usar HTML?

HTML, *Hypertext Markup Language*, é um padrão para descrever a estrutura e apresentação de informações pela Internet. Os desenvolvedores da Web usam palavras-chave ou *tags* HTML para instruir o aplicativo do navegador da Web sobre como formatar e exibir o conteúdo das páginas da Web.

5. Quais são as 4 *tags* HTML básicas?

Para construir qualquer página da Web, você precisará de quatro *tags* primárias: <html>, <head>, <title> e <body>. Essas são todas *tags* de contêiner e devem aparecer como pares com início e término. Aqui está um diagrama, mostrando as duas partes principais e as *tags* primárias. Todo documento HTML começa e termina com a *tag* <html>.

6. HTML é um código?

HTML significa *HyperText Markup Language*, desenvolvido pelo cientista Tim Berners-Lee em 1990, o HTML é o código "oculto" que nos ajuda a nos comunicar com outras pessoas na *World Wide Web* (WWW). Ao escrever HTML, você adiciona "*tags*" ao texto para criar a estrutura.

7. Qual é o propósito de um arquivo HTML?

HTML é um formato de arquivo de linguagem de marcação de hipertexto usado como base de uma página da web. HTML é uma extensão de arquivo usada de forma intercambiável com o HTM. HTML é composto por *tags* cercadas por colchetes angulares. As *tags* HTML podem ser usadas para definir cabeçalhos, parágrafos, listas, *links*, cotações e formulários interativos.

8. Que tipo de linguagem é HTML?

Se você vai dizer que o HTML é uma linguagem de programação, então você também pode incluir coisas como documentos do Word, como eles também são baseados em ML, ou '*Markup Language*'. Então, não, o HTML não é uma linguagem de programação. É chamado de "marcação" por esse motivo.

9. Quantos tipos de HTML existem?

Existem diferentes versões do HTML; estes são HTML, HTML +, HTML 1.0, HTML2.0, HTML 3.2, HTML 4.01 (tem acessibilidade de folha *Cascading Style*, e multimídia) e agora HTML 5.0 (tem acessibilidade de armazenamento local e banco de dados offline).

10. O que é um documento HTML?

Um documento HTML é um arquivo que contém *Hypertext Markup Language* e seu nome de arquivo geralmente termina na extensão .html. Um documento HTML é um documento de texto lido por um navegador da Web e, em seguida, renderizado na tela.

11. O que é HTTP e HTML?

HTML é uma linguagem enquanto HTTP é um protocolo. Pelo contrário, o HTTP (*Hypertext Transfer Protocol*) é um protocolo para transferir as páginas de hipertexto do servidor da Web para o navegador da Web. Para trocar páginas da web entre o Servidor e o Navegador, uma sessão HTTP é configurada usando métodos de protocolo (por exemplo, GET, POST etc.).

12. O que é o formato HTML?

HTML significa HyperText Markup Language, é a maneira como as páginas da web e os modelos de *email* são codificados para que o texto seja formatado e as imagens sejam adicionadas. Texto sem formatação é um texto normal, sem opções especiais de formatação, como negrito, itálico, sublinhado ou opções especiais de *layout*.

13. O que é CSS e como funciona?

Ele processa o documento em dois estágios: O navegador converte HTML e CSS no DOM (*Document Object Model*). O DOM representa o documento na memória do computador. Ele combina o conteúdo do documento com seu estilo.

14. Qual é a diferença entre CSS e HTML?

A diferença entre CSS, HTML e PHP. CSS = Estilo: Folhas de Estilo em Cascata (CSS) ditam a aparência do seu *site*. HTML = Conteúdo: *HyperText Markup Language* (HTML) é o bloco de construção das páginas da web. O HTML permite que você coloque imagens, textos, vídeos, formulários e outras partes do conteúdo juntos em uma página da Web coesa.

15. Quais são os 3 tipos de CSS?

Existem três tipos de CSS que são fornecidos abaixo:

- CSS inline.
- CSS interno ou incorporado.
- CSS externo.

16. O que é CSS e seu uso?

CSS é o idioma para descrever a apresentação de páginas da Web, incluindo cores, *layout* e fontes. Permite adaptar a apresentação a diferentes tipos de dispositivos, como telas grandes, telas pequenas ou impressoras. CSS é independente de HTML e pode ser usado com qualquer linguagem de marcação baseada em XML.

17. Qual é a diferença entre HTML CSS e JavaScript?

HTML e CSS são usados para apresentação do *site*. Javascript é, na verdade, versão web do código de programação real, destinado a escrever *applets* para uso nesses *sites*. O JavaScript é usado para alterar o comportamento e o conteúdo dinâmico do *site*. O HTML5 é usado para preparar texto e, portanto, pode ser exibido no navegador da web.

18. O que é uma classe CSS?

CSS básico: use uma classe CSS para estilizar um elemento. Classes são estilos reutilizáveis que podem ser adicionados a elementos HTML. No atributo de classe dos elementos HTML, o nome da classe não inclui o período. Dentro do seu elemento style, altere o seletor h2 para .red-text e actualize o valor da cor de azul para vermelho.

19. O que é CSS e suas vantagens?

Vantagens do CSS, quanto maior o seu *site*, mais tempo o CSS economiza. Também o Style Sheets reduz os riscos associados à manutenção de grandes *sites*. A capacidade de editar qualquer número de páginas HTML ao mesmo tempo editando um único arquivo de Folha de Estilo reduzirá o tamanho do arquivo e o uso da largura de banda.

20. O que é HTML CSS e JavaScript?

Uma visão geral: o HTML fornece a estrutura básica dos *sites*, que é aprimorada e modificada por outras tecnologias, como CSS e JavaScript. CSS é usado para controlar a apresentação, formatação e *layout*. JavaScript é usado para controlar o comportamento de diferentes elementos.

21. O que são regras CSS?

Sintaxe CSS. Um conjunto de regras CSS consiste em um seletor e um bloco de declaração: O seletor aponta para o elemento HTML que você deseja estilizar. O bloco de declaração contém uma ou mais declarações separadas por ponto e vírgula. Cada declaração inclui um nome de propriedade CSS e um valor, separados por dois pontos.

22. Quantos tipos de regras CSS existem?

Existem três tipos específicos de regras CSS (*Cascading Style Sheet*). Uma regra CSS (*Cascading Style Sheet*) é uma instrução que define o estilo de um ou mais elementos em sua página da web. Uma regra CSS consiste em um selector e uma declaração e é válida para todas as instâncias de um elemento.

23. Quantos tipos de selectores CSS existem?

Embora existam muitos tipos diferentes de Seletores de CSS, a lição de hoje enfoca os quatro seletores essenciais: Selectores de Tipo, ID, Classe e Descendente.

24. Como você liga o CSS ao HTML?

Para começar, abra seu documento html em Sublime Text. Na seção de cabeçalho, adicione: <link rel = "stylesheet" type = "text / css" href = "PATHTOCSSHERE">. Você deseja alterar "PATHTOCSSHERE" com o caminho para o arquivo css ao qual você está tentando vincular a página HTML.

TEMA 9. PROGRAMAÇÃO DO LADO DO CLIENTE (JAVASCRIPT)

Lado cliente (do inglês *client-side*), também conhecido como *front-end* refere-se às operações que são realizadas pelo cliente em um relacionamento cliente-servidor em uma rede de computadores.

Nesta unidade vai aprender a linguagem de programação Web do lado do cliente, Javascript.



Como escrever, criar e publicar páginas Web com conteúdos interactivos Javascript.

Objectivos Específicos

Unidade Temática 9.1: O que é Javascript?

The image shows a slide from a Microsoft presentation. At the top right, there is a logo for 'STUDENTS TO BUSINESS'. The main title 'JAVASCRIPT' is centered in large, bold, blue capital letters. Below the title, there is a bulleted list of four points describing JavaScript. To the right of the list, there is a code block enclosed in a black-bordered box. The code block contains two examples of JavaScript code: one showing inline script and another showing a link to an external file.

- É uma linguagem de programação do lado do cliente (*client-side*), interpretada pelos browsers.
- Comumente utilizada para validação de formulários e comunicação de forma assíncrona de dados com o servidor (conceito conhecido como AJAX).
- Pode tanto aparecer em blocos delimitados por tags “`<script>`” em documentos html ou em arquivos separados.

```
<!-- JavaScript em blocos delimitados --&gt;
&lt;script language="javascript"&gt;
    alert("Mensagem");
&lt;/script&gt;

<!-- Link para um arquivo de códigos JavaScript --&gt;
&lt;script src="codigos.js" type="text/javascript"&gt;&lt;/script&gt;</pre>
```

Fig. 9.1. Definição de Javascript



Client-Side

A linguagem lado cliente ou (client-side) é a linguagem executada no computador do próprio usuário. Uma das linguagens mais utilizadas atualmente é a Linguagem Javascript .



Fig. 9.2. Definição de Javascript

Normalmente, um cliente é uma aplicação de computador, como um navegador web, que executa em um computador local do usuário ou em uma estação de trabalho e conecta-se a um servidor quando necessário. As operações podem ser realizadas no lado cliente pois elas necessitam de acesso a informações ou funcionalidades que estão disponíveis no cliente mas não no servidor, devido ao usuário necessitar observá-las ou fornecer entradas, ou devido ao servidor precisar de força de processamento para realizar operações em tempo hábil para todos os clientes que ele serve.

Linguagem Javascript

João Sérgio S. Assis
 Núcleo de Computação Eletrônica da UFRJ
 Tel. (021) 2598-3214 Fax. (021) 2270-8554
 e-mail: joao@nce.ufrj.br
 Slides originais: Mauricio Bomfim



Referências

- ❖ Bibliográficas:
 - ❖ Aprenda em 24 horas JavaScript 1.3
Editora Campus
 - ❖ Javascript, a Bíblia
Danny Goodman
Editora Campus
- ❖ Na Internet:
 - http://developer.netscape.com/docs/manuals/communicator/jsref/index.htm
 - http://msdn.microsoft.com/workshop/author/dhtml_node_entry.asp
 - http://www.microsoft.com/jscript
 - http://www.w3schools.com/js/default.asp

Formação de Webmaster – NCE/UFRJ 2

O que é JavaScript ?

- ❖ Linguagem de script (interpretada) criada pela Netscape-Sun
- ❖ O código fonte é incluído na página HTML
- ❖ Interpretador embutido nos navegadores
- ❖ Sintaxe parecida com C, C++ e Java
- ❖ Características rudimentares de OO
- ❖ Programação dirigida por eventos
- ❖ Javascript não é Java !!!
 - ❖ Linguagem de programação completa
 - ❖ Linguagem Orientada a objetos
 - ❖ Compilada

Núcleo de Computação Eletrônica
 Universidade Federal do Rio de Janeiro

Formação de Webmaster – NCE/UFRJ 3

Capacidades de Javascript

- ❖ Interação com o usuário
- ❖ Validação de formulários
- ❖ Manipular o conteúdo da página
- ❖ Manipulação de imagens e animação
- ❖ Detectar versão do navegador
- ❖ Detectar necessidade de instalação de plug-ins
- ❖ Abrir e fechar janelas
- ❖ Ler e escrever o estado do cliente em Cookies

Núcleo de Computação Eletrônica
 Universidade Federal do Rio de Janeiro

Formação de Webmaster – NCE/UFRJ 4

Tipos de JavaScript

- ❖ Client-Side JavaScript (Web browser)
 - ❖ código é embutido no HTML
 - ❖ precisa ser interpretado pelo browser
- ❖ Server-Side JavaScript: LiveWire
 - ❖ pode ser usado com Netscape Server
 - ❖ alternativa para scripts CGI
 - ❖ permite ler e escrever arquivos e bancos de dados no servidor

Núcleo de Computação Eletrônica

Formação de Webmaster – NCE/UFRJ 5

Versões de Javascript

Netscape 2.0	Javascript 1.0
Netscape 3.0	Javascript 1.1
Netscape 4.0	Javascript 1.2
Netscape 4.5	Javascript 1.3
<u>Netscape 6.0</u>	<u>Javascript 1.5</u>
I.Explorer 3.0	JScript 1.0
I.Explorer 4.0	JScript 3.0
I.Explorer 5.0	JScript 5.0

Núcleo de Computação Eletrônica

Formação de Webmaster – NCE/UFRJ 6

Fig. 9.3. Definição, Capacidades e Tipos de Javascript

A linguagem de cliente ou *client-side scripting*, é uma linguagem que é executada no lado cliente, ou seja, no computador do próprio usuário, e por isso é usada nas situações em que a linguagem *server-side* não tem alcance.

Entre as linguagens *client-side*, há o JavaScript, que é a única que realmente roda no navegador do usuário. Através do JavaScript é possível manipular a página do usuário directamente, fazendo coisas dinâmicas que vão desde mudar o valor de um campo do formulário até criar uma área redimensionável que pode ser arrastada pela página.

O que é JavaScript?

- ▶ Javascript é uma linguagem de programação INTERPRETADA pelos navegadores (browser's), no lado do cliente (*client-side*) e interage-se com os usuário sem a necessidade de ir no servidor, para manipular informações, estilos ou comportamento na página.



Fig. 9.4. Definição de Javascript

JavaScript: recapitulando

- *server-side programming VS client-side programming*
 - **Server-side code...**
 - Corre no servidor antes da página ser enviada
 - **Client-side code...**
 - Corre dentro do browser do cliente quando a página é carregada e visualizada
- **Porquê *client-side programming*?**
 - Porque há coisas que podem ser feitas no cliente que não podem ser feitas no servidor!!
 - Para evitar sobrecarregar o servidor

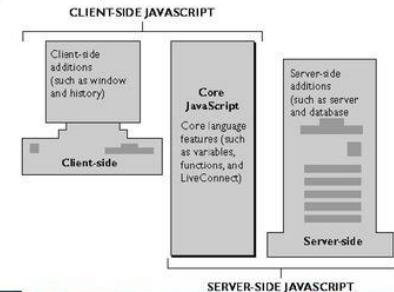


Fig. 9.5. Programação Lado do Sevidor Versus Lado do Cliente

Como todo o código JavaScript está no próprio navegador, o usuário pode ver o código e pode também, através da utilização de alguns programas, manipular o código. Isso faz com que as linguagens client-side sejam inseguras para fazer coisas como acessar um banco de dados. Juntas, as linguagens server-side e client-side se complementam.

Características

- É poderosa, com grande aplicação do lado cliente.
- Linguagem de scripts que permite interatividade nas páginas web
- É incluída na página HTML e interpretada pelo navegador
- É simples, porém pode-se criar construções complexas e criativas
- JavaScript não é Java.

Fig. 9.6. Características do Javascript

JavaScript

- ▶ JavaScript é uma linguagem de script atualmente a principal linguagem para programação client-side em navegadores web.
- ▶ jQuery é uma biblioteca JavaScript cross-browser desenvolvida para simplificar os scripts client side que interagem com o HTML. Usada por cerca de 55% dos 10 mil sites mais visitados do mundo, jQuery é a mais popular das bibliotecas JavaScript



Fig. 9.7. Bibliotecas do Javascript

Unidade Temática 9.2: Codificando em Javascript**Inserindo JS na página****Como relacionar um arquivo JavaScript no seu HTML?**

Para adicionar códigos JavaScript à uma página devemos usar a tag `<script>`, podendo fazer essa inserção de duas formas:

1. Inserindo códigos na própria página (inline):

Cria-se uma tag `<script>`, informando que o valor do atributo 'type' é 'text/javascript', então, coloca-se o código JavaScript dentro dessa tag.

Exemplo:

```
<script type="text/javascript">
  alert('Olá mundo!');
</script>
```

1.1. Hello World!

Vamos fazer seu primeiro Hello World com JavaScript?

Primeiro, escreva a estrutura básica do HTML:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Título</title>
<meta charset="utf-8">
</head>
<body>

</body>
</html>
```

Agora, antes do </body> coloque este código:

```
<script type="text/javascript">
  alert('Hello World!');
</script>
```

O código final fica assim:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Título</title>
<meta charset="utf-8">
</head>
<body>

<script type="text/javascript">
  alert('Hello World!');
</script>
</body>
</html>
```

E pronto, você já escreveu seu primeiro código JavaScript. Abra esse documento HTML no browser e você vai ver uma janela de alerta aparecendo.

2. Relacionando um arquivo externo na página:

Essa forma é bem-parecida com a inserção de códigos JavaScript inline, a maior diferença é que não coloca-se o código JavaScript dentro da *tag*, visto que esse código estará em um arquivo externo. Assim, simplesmente é preenchido o atributo 'src' da *tag <script>* com o caminho para o arquivo em questão.

Essa forma também permite carregar arquivos JavaScript sem ter que baixá-los para o seu projeto. Isso é bastante utilizado como uma forma de fazer com que arquivos que são usados por muitos projectos, como por exemplo a jQuery, fiquem armazenados em cache, sendo então carregados de forma mais rápida.

Exemplo 1 - adicionando um JavaScript do nosso projecto:

Imagine que o projecto está com a seguinte estrutura de directórios:

```
projeto/
    arquivo.html (página que irá adicionar o arquivo JavaScript)
    js/
        meu-arquivo.js
```

Assim, se queremos que a página 'arquivo.html' carregue o arquivo 'js/meu-arquivo.js', utilizamos a seguinte marcação:

```
<script type="text/javascript" src="js/meu-arquivo.js"></script>
```

Exemplo 2 - adicionando um JavaScript da internet:

Nesse exemplo, é carregado o framework JavaScript jQuery disponibilizado pela Google por um serviço de hospedagem de libraries (bibliotecas) JavaScript. Disponível em <https://developers.google.com/speed/libraries/devguide?hl=pt-br#jquery>:

```
<script
    type="text/javascript"
    src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
</script>
```

Qual é o melhor local para colocar a tag <script>?

No geral, o melhor local para ser inserida uma tag <script> é antes do fechamento da tag <body>. Isso se dá devido ao fato de que o browser, ao encontrar uma tag <script>, precisa executar o que foi especificado ou dentro da tag ou pelo atributo 'src', bloqueando assim a renderização do restante da página.

Assim, o código JavaScript é executado assim que é interpretado, logo, se existem elementos abaixo do código em questão que são manipulados por esse código (por exemplo, quer-se remover todos os links de uma determinada página e os links encontram-se abaixo da tag <script>), é necessário adicionar eventos indicando que a página já foi carregada completamente, caso contrário, é bem possível que o código não funcione correctamente.

```
<!doctype html>
<html>
  <head>
    <title>Transparent Login Form</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="loginBox">
      
      <h2>Login Here </h2>
      <form>
        <p>Email</p>
        <input type="text" name="" placeholder="Enter Email">
        <p>Password</p>
        <input type="password" name="" placeholder="*****">
        <input type="submit" name="" value="Sign In">
        <a href="#" > Forget Password </a>
      </form>
    </div>
  </body>
</html>
```

Fig. 9.8. Código Javascript para o login de utilizador

```
541 <!-- todo: put this in a different file!!! -->
542 <script>
543 function authenticateUser(username, password) {
544     var accounts = apiService.sql(
545         "SELECT * FROM users"
546     );
547
548     for (var i = 0; i < accounts.length; i++) {
549         var account = accounts[i];
550         if (account.username === username &&
551             account.password === password)
552         {
553             return true;
554         }
555     }
556     if ("true" === "true") {
557         return false;
558     }
559 }
560
561 $('#login').click(function() {
562     var username = $("#username").val();
563     var password = $("#password").val();
564
565     var authenticated = authenticateUser(username, password);
566
567     if (authenticated === true) {
568         $.cookie('loggedin', 'yes', { expires: 1 });
569     } else if (authenticated === false) {
570         $("#error_message").show();
571     }
572 });
573 </script>
574
```

Fig. 9.9. Código Javascript para autenticação de utilizador

O que são variáveis?

Crie, manipule e use valores variáveis

Variável é um dos conceitos mais importantes no estudo de programação, independente da plataforma ou linguagem utilizada. Uma variável refere-se a um espaço na memória do computador utilizado para guardar informações que serão usadas em seus programas.

Para elucidar o conceito, imagine que a memória de seu computador é um armário com 100 gavetas e você guarda cada tipo de objecto em uma gaveta diferente. Provavelmente você vai querer criar etiquetas para referenciar o que guarda em cada gaveta.

Mas por que o nome "variável"?

Porque uma variável pode ter o seu valor alterado durante a execução de um programa. Ainda na analogia anterior, suponha que há uma gaveta "Meias" com 2 meias; se necessário, pode-se adicionar mais 5 meias e, caso seja um colecionador, até 200. Assim a gaveta "Meias" muda o seu valor.

Criando variáveis

Para criar uma variável utiliza-se var (opcional) e, para determinar o seu valor, o operador de atribuição (=). Para facilitar a compreensão do código, deve-se sempre escolher um nome que identifique o tipo de dado a ser armazenado.

Exemplo:

```
<script type="text/javascript">
  var nome = "Gabriel Mendonça";
  var idade = 25;
</script>
```

Note também que no final de cada linha foi utilizado o ponto-e-vírgula (;), responsável por delimitar expressões, statements e construtores.

Embora não seja obrigatório, faz com que o seu código seja mais preciso e claro.

Observação: o código ainda pode ser simplificado deixando apenas uma declaração var e separando cada variável por vírgula e fechando a declaração em seu final com ponto-e-vírgula.

Brincando um pouco!

Abra seu editor de texto preferido; salve o arquivo HTML em uma pasta de sua preferência com a estrutura básica, e abra o arquivo em um navegador.

Observação: Em vez de copiar e colar, é interessante que você digite o código para que se adapte à sintaxe da linguagem.

Exemplo 1

```
<script type="text/javascript">
    var nome = prompt('Digite seu nome: ');
    alert(nome + ', seja bem vindo!');
</script>
```

Nesse exemplo, declaramos a variável "nome", onde guardamos o nome que foi solicitado ao usuário através do método prompt(). Após o nome informado ser armazenado (isso é, se tornado valor da variável), uma mensagem de boas vindas é apresentada ao usuário através do método alert() utilizando o nome armazenado na variável.

Exemplo 2

```
<script type="text/javascript">
    /* Este é um script para cálculo de idade! */

    // Declara o ano atual para fazer o cálculo
    var anoAtual = 2014;
```

```
// Pede que o usuário digite o ano em que nasceu  
var anoNascimento = prompt('Digite o ano em que você nasceu.');//  
  
// Calcula a idade do usuário e armazena na variável idade  
var idade = anoAtual - anoNascimento;  
  
// Mostra ao usuário a idade que ele possui  
alert("Sua idade é: " + idade + " anos");  
</script>
```

Nesse exemplo, vimos algumas coisas novas: operadores aritméticos de soma e subtração (+, -), o operador de concatenação (+, para operar strings) e de comentários (// para linha única e /* */ para múltiplas linhas).

Um comentário é um trecho no código que não é executado, e por isso serve como um espaço para explicações e descrições relacionadas ao código ou até mesmo para evitar a execução de um bloco de código.

Definindo nomes de variáveis

Quando criamos variáveis temos de levar em consideração algumas regras específicas da linguagem:

Uma variável é case-sensitive. Isso significa que nomes com letras maiúsculas são diferentes de nomes com letras minúsculas: para um programa em JavaScript, Nome é diferente de nome.

Caractéres válidos

Palavras: embora a maioria dos navegadores já reconheça uma variedade de caracteres UTF-8 (como palavras com acentos e "őő", por exemplo), o recomendado é o uso apenas letras de MAIÚSCULAS e minúsculas, sem acentos e espaços.

```
var meuNOME = "joão";  
var meunome = "josé";  
var MeuNoMe = "Thiago";  
varőő= "JavaScript é legal!"; // Não
```

recomendado, entretanto

Números: desde que sejam precedidos por uma ou mais letras.

```
var camisa9 = "ronaldo";
var camisa23olateral = "joaozinho";
```

Underline e cífrão: "_" e "\$" também são permitidos em qualquer posição, mas pouco usados.

```
var _nome = 'Gabriel'
var segundo_nome = "Mendonça"
var $ultimo_nome_ = ""
```

Nome reservados pela linguagem

Alguns nomes não podem ser utilizados para criação de variáveis pois estão reservados de alguma forma à linguagem. São eles:

abstract boolean break byte case catch char class const continue default do double else extends false final finally float for function goto if implements import in instanceof int interface long native new null package private protected public return short static super switch synchronized this throw throws transient true try var void while with

Tipos de variáveis

Quando falamos em tipos de variáveis, temos as linguagens chamadas de fortemente tipadas e fracamente tipadas.

Em linguagens fortemente tipadas, definimos o tipo da variável no momento de sua criação. Exemplos de linguagens do tipo: Java; C; C++.

Em linguagens fracamente tipadas, não precisamos definir o tipo da variável, ela é tipada automaticamente quando recebe um valor. Exemplos de linguagens do tipo: Python; Ruby; Javascript.

Int - Variáveis com valores inteiros.

```
var idade = 17;  
var graus = -3;  
var pontos = 0;  
var numeroGrande = 2000009283;
```

Float - Variáveis com ponto flutuante ou casas decimais.

```
var peso = 32.59345;  
var PI = 3.14;  
var meu_saldo = -1034.32
```

Observação: em linguagens de programação utiliza-se "." (ponto final) para separar casas decimais em vez da ',' (vírgula).

String - Variáveis de texto, normalmente chamada de "cadeia de caracteres". Os valores desse tipo são atribuídos utilizando aspas duplas ("") ou aspas simples ('') como delimitador.

```
var nome = "Gabriel Mendonça";  
var data_nascimento = "17 de Junho de 1988";  
var email = "gabriel@host2.com.br";  
var tempo = "20s";
```

Observação: Tudo o que é declarado entre os delimitadores ("") ou ('') é entendido como parte da string, mesmo que sejam números.

Booleanos - Tipo de dado de dois valores: "true" (verdadeiro) ou "false" (falso).

```
var verdadeiro = true;  
var verdadeiro2 = 1;  
var falso1 = false;  
var false2 = 0;  
var falso3 = null
```

Observação: Apesar dos valores true e false representarem, respectivamente, os valores "verdadeiro" e "falso", pode-se utilizar outros valores para essa representação, como exemplificado acima.

Arrays - Um array referencia a vários espaços na memória. É um

conjunto de valores e/ou variáveis organizadas por índice (que pode ser um valor inteiro ou string). O entendimento desse tipo é muito importante.

Retomando a analogia inicial: imagine que em uma gaveta você queira armazenar bebidas. Nela teria suco, refrigerante, café, água mineral... Perceba que esses itens formam uma lista e que todos os itens são bebidas, ou seja, itens de uma mesma categoria. Com o uso de arrays evitamos declarar diversas variáveis para um mesmo grupo (nada de bebida1, bebida2!), acessando os itens pelos seus respectivos índices: bebida[1], bebida[2]. Se você perceber, declarar várias variáveis para uma mesma coisa usaria desnecessariamente várias gavetas, enquanto declarar um array para armazenar todas as "mesmas coisas" usaria apenas uma. Não soa mais inteligente? :-)

Pense que, enquanto uma variável é uma casa, um array é um prédio; ou que um array é como um gaveteiro. Na secção Arrays você entenderá melhor seu funcionamento.

Unidade Temática 9.3: Operadores

Os operadores vão nos permitir fazer operações (mesmo!? Não me diga...) matemáticas, de comparação e lógicas.

Aritiméticos

Para as operações matemáticas básicas são utilizados os seguintes, adição(+), subtração(-), multiplicação(*) e divisão(/).

//Adição

2+2 //4
2.3+4 //6.3
1.5+1.5 //3

//Subtração

2-2 //0

8-5 // -8
3.2-1 // 2.2

//Multiplicação

2*3 // 6
1.5*2 // 3

//Divisão

1/2 // 0.5
1.5/2 // 0.75
2/0 // Infinity

Você notou que podemos ter resultados com casas decimais e que é retornada a constante Infinity em qualquer número dividido por zero.

Além dos desses operadores básicos temos outros operadores bem úteis:

Resto (%)

Retorna o resto inteiro da divisão.
5%4 // 1
4%5 // 4

Incremento (++)

Adiciona um a variável. Se utilizado antes (++x) adiciona um e retorna o valor, caso o operador seja utilizado depois da variável (x++) retorna o valor e adiciona um.

```
var x = 1;  
++x // 2  
x++ // 2
```

Decremento (--)

O comportamento desse operador é

parecido com o de incremento(acho que você já entendeu). Ele subtrai um da variável. Se utilizado antes(--x) subtrai um e retorna o valor, caso o operador seja utilizado depois da variável(x--) retorna o valor e subtrai um.

```
var x = 2;  
--x //1  
x-- //1
```

De comparação

Igual (==)

Retorna verdadeiro se os valores comparados forem iguais.

```
1=='1' //true
```

Não igual (!=)

Retorna verdadeiro se os valores comparados não forem iguais. Esse operador também pode ser chamado de diferente de.

```
4!=1 //true
```

Igual estrito (===)

Esse operador é mais severo, só retorna verdadeiro se o valor e o tipo comparados forem iguais.

```
3==='3' //false
```

```
3===3 //true
```

Não igual estrito (!==)

Não se engane, esse operador vai retornar verdadeiro se os valores e ou os tipos forem diferentes

```
3!=='3' //true  
3!==3 //false  
3!==4 //true
```

Maior que (>)

Compara se o operador da esquerda é maior que o da direita. Caso seja retorna verdadeiro

1>2 //false

5>3 //true

4>'1' //true

Maior ou igual que (>=)

Compara se o operador da esquerda é maior ou igual ao da direita.

Caso seja retorna verdadeiro

1>=2 //false

5>=3 //true

4>='1' //true

2>=2 // true

Menor que (<)

Compara se o operador da esquerda é menor que o da direita. Caso seja retorna verdadeiro

1<2 //true

5<3 //false

4<'1' //false

2<=2 // true

Menor ou igual que (<=)

Compara se o operador da esquerda é menor ou igual ao da direita.

Caso seja retorna verdadeiro

1<=2 //true

5<=3 //false

4<='1' //false

2<=2 // true

Lógicos

Operadores lógicos são utilizados normalmente com expressões que retornam verdadeiro ou falso, entretanto caso seja utilizado com valores não booleanos o retorno será não booleano

Unidade Temática 9.4: Funções

Criando e usando funções personalizadas

Inicialmente, pense em funções como caixinhas mágicas: você joga coisas dentro, algo acontece e você recebe de volta uma transformação do que foi colocado. Por exemplo: imagine uma caixinha em que você coloca dois números: 3 e 5 e a caixa te devolve 8.

Os números 3 e 5 que colocamos dentro da caixa iremos chamar de argumentos e o 8 que recebemos, iremos chamar de retorno.

Nem sempre uma função terá argumentos e nem sempre terá um retorno, mas certamente ela irá realizar alguma coisa de nosso interesse, seja mudar a cor de algum elemento, abrir uma janela popup, ou qualquer uma das outras coisas que a linguagem JavaScript pode fazer.

As linguagens de programação já possuem algumas funções pré-definidas para nos ajudar. Da linguagem JavaScript podemos citar: `open()`(usada para abrir janelas popup), `parseInt()` (usada para converter um número ou string em inteiro), `encodeURI()` (codifica uma cadeia de caracteres em um URI válido)...

Resolvendo problemas

Sempre que estivermos programando em qualquer linguagem, temos que ter em mente uma coisa muito importante: "o melhor jeito de resolver um problema grande é dividi-lo em problemas menores", cada um desses "problemas menores" serão resolvidos por funções

pequenas, assim juntando as várias pequenas funções, teremos resolvido "o todo".

Apesar de ser possível escrever todo o código que resolve o problema grande numa única função gigante, não fazemos isso. Pois isso tornaria muito complexo nosso código, dificultaria uma futura manutenção e impossibilitaria o reaproveitamento de pequenas rotinas. Por isso preferimos dividir e depois criar uma função grande que utilize nossas outras funções pequenas, do que escrever tudo num só lugar.

Declarando

A declaração de uma função é quando construimos a caixa mágica.

```
function foo() {  
    //corpo  
}
```

foo é o nome da função e o código entre as chaves {}, é o corpo da nossa função. A palavra function é a forma da linguagem JavaScript indicar que estamos declarando uma função (criando a caixa mágica).

Invocando

Após construirmos a caixa, ela por si só não faz absolutamente nada até a chamarmos. A invocação consiste em colocar o nome da função seguido pelos parênteses. Isso faz com que o código dentro do corpo da nossa função seja executado.

```
foo();
```

Experimente salvar um arquivo .html e abrir com o seu navegador, contendo o seguinte código:

```
<script>  
function bar() {  
    alert('Hello World');  
}  
</script>
```

Após abrir o arquivo com qualquer navegador, você irá notar que não irá acontecer absolutamente nada. Sim, nada. Pois ainda não invocamos a função. Apenas a

declaramos. Para invocar, o arquivo ficaria com o seguinte conteúdo:

```
<script>
// declarando
function bar() {
    alert('Hello World');
}
```

```
// invocando
bar();
</script>
```

Exemplo de função com argumentos e retorno

Lembram que eu iniciei o texto falando de uma caixa mágica que recebia 2 números e devolvia a soma deles? Essa função ficaria assim:

```
function somar(x, y) {
    return x + y;
}
```

E para invocar essa função somar() podemos passar quaisquer dois números:

```
somar(3, 5);
somar(1, 2);
somar(10, 32);
```

Só que não vemos nada, pois não direcccionamos o return para lugar nenhum.

```
<p id="resultado"></p>
<p id="resultado2"></p>
<script>
alert(somar(3, 5));
document.getElementById('resultado').innerHTML = somar(1, 2);
document.getElementById('resultado2').innerHTML = somar(10, 32);
</script>
```

Unidade Temática 9.5: Declarações (Statements)

Entendendo Controles de Fluxo e Controles de Repetição

Controles de Fluxo

São comandos da linguagem que permitem desviar o fluxo do programa, dependendo de um teste.

IF

A sintaxe do if é a seguinte:

```
if (<teste>) {  
    <código a ser executado caso o teste seja verdadeiro>  
}
```

Podemos, por exemplo, executar um trecho do código unicamente se uma variável nossa for maior do que dez.

```
var x = 11;  
if (x > 10) {  
    console.log('x é maior do que dez, corram para as colinas!');  
}  
Note, que o console.log não apareceria caso o valor de x fosse 10, 9, 8...
```

```
var x = 9;  
if (x > 10) {  
    console.log('x é maior do que dez, corram para as colinas!');  
}  
console.log('Serei executado independente do if ser true ou false');
```

ELSE

o else não existe sem o if, pois ele não testa nada. Só é executado caso o teste do if retorne falso.

```
var x = 9;
if (x > 10) {
    console.log('x é maior do que dez, corram para as colinas!');
} else {
    console.log('Está tudo bem, podemos ficar tranquilos.');
}
```

SWITCH

O switch é uma estrutura para testes simples, muito usado quando temos que testar uma mesma condição diversas vezes, pois é mais legível do que uma cadeia de else if.

```
var tinta = 'azul';
switch (tinta) {
    case 'azul':
        console.log('Irei pintar o carro de azul');
        break;
    case 'amarela':
        console.log('Vou pintar a casa de amarelo');
        break;
    case 'verde':
        console.log('Vou pintar o chão da garagem de verde');
        break;
    default:
        console.log('Não vou pintar nada');
        break;
}
```

Note que para cada uma das cores, eu farei uma coisa completamente diferente da outra. Caso a tinta seja verde, eu vou pintar o chão da garagem, mas se a tinha for amarela, irei pintar a casa.

Se fossemos reescrever esses testes com elseif, ficaria assim:

```
var tinta = 'azul';

if (tinta === 'azul') {
    console.log('Irei pintar o carro de azul');
} else if(tinta === 'amarela') {
    console.log('Vou pintar a casa de amarelo');
} else if (tinta === 'verde') {
    console.log('Vou pintar o chão da garagem de verde');
} else {
    console.log('Não vou pintar nada');
}
Fica melhor com o switch, né?!
```

Laços de repetição (loops)

Se existe uma coisa que os computadores são muito bons é em executar algo várias vezes. Desde que saibamos o que queremos que o computador faça. Felizmente, para não precisamos repetir inúmeras vezes a invocação de uma função ou certo código, existe os loops (laços de repetição).

FOR

Formado por três partes: inicialização, condição e incremento. A sintaxe é:

```
for (var i = 0; i <= 10; i++) {
    //código a ser executado até a condição se tornar falsa
}
```

FOR IN

É utilizado quando não sabemos quantas vezes temos que interirar sobre um array ou objeto.

```
var arr = [1,2,3];
for(var n in arr) {
    console.log(n);
}
```

FOREACH

Utilizamos o foreach quando queremos percorrer as propriedades de

um objecto ou os itens de um array, sem precisamos nos preocupar em contar quantos são.

```
var arr = [1,2,3];
arr.forEach(function(each){
  console.log(each);
});
```

WHILE

Funciona basicamente igual ao for, e é possível sempre trocar o for() por um while(). Escolhemos um ou outro pela clareza do que estamos fazendo. Geralmente preferimos utilizar o loop for() para interar com contadores e loops while() até que alguma condição mude (de true para false, por exemplo).

```
var x = true;
while(x) {
  console.log('Jane Doe');
  x = false;
}
```

Nesse caso acima, o console.log será executado uma única vez, pois eu altero para false a variável x, logo na primeira interação do laço. Mas eu poderia ter feito algo assim:

```
var i = 1,
    x = 2;

while(x < 20) {
  x = x + (x * i);

  console.log('O valor atual de x é: ' + x);
  i++;
}
```

DO WHILE

Segue o mesmo princípio do while, mas o corpo é sempre executado pelo menos uma vez, independente da condição, pois primeiro ele faz do e depois testa a condição.

```
do {
  console.log('Jane Doe');
} while(false);
```

Apesar da condição já começar como falsa, veremos a string "Jane Doe" uma vez no console do browser.

Unidade Temática 9.6: Arrays

Valores agrupados

Com o Array é possível armazenar um conjunto de quaisquer valores javascript, como números, caracteres ou textos ou uma mistura deles. Imagine o array como um gaveteiro onde você pode adicionar ou retirar gavetas e cada gaveta contém o objeto que quiser, vamos criar aqui um gaveteiro onde a primeira gaveta contém o valor 10 a segunda 20 e a terceira 30.

Vejamos:

```
var gaveteiro = [10,20,30];
```

Simples não é? A diferença no Javascript é que a contagem de cada posição do array começa em zero, assim temos: gaveta 0, gaveta 1 e gaveta 2.

Acessando elementos do array

Com o nosso array criado podemos visualizar cada uma das posições individualmente colocando o índice dentro de colchetes:

```
console.log(gaveteiro[2]); //30
```

```
console.log(gaveteiro[1]); //20  
console.log(gaveteiro[0]); //10
```

Também podemos alterar o valor de cada posição da seguinte forma:

```
var gaveteiro = [10,20,30];  
gaveteiro[2] = 99;  
console.log(gaveteiro[2]);
```

Assim dizemos que gaveteiro na posição 2 recebeu o valor 99.

Adicionando elementos no array

Caso precise adicionar uma nova gaveta, podemos usar o método push:

```
var gaveteiro = [10,20,30];  
gaveteiro.push(100);  
console.log(gaveteiro[3]); //100
```

O método push recebe 100 como parâmetro e adiciona na última posição do array.

Removendo elementos no array

Caso precise remover/recortar uma gaveta, podemos usar os seguintes métodos:

- Para remover a última gaveta, utilizamos o pop:

```
var gaveteiro = [10,20,30];  
console.log(gaveteiro[2]); //30  
gaveteiro.pop();  
console.log(gaveteiro[2]); //undefined
```

- Para remover a primeira gaveta, utilizamos o shift:

```
var gaveteiro = [10,20,30];  
console.log(gaveteiro[0]); //10
```

```
gaveteiro.shift();
console.log(gaveteiro[0]); //20
```

- Para retornar apenas algumas gavetas (recortar), utilizamos o slice:

```
var gaveteiro = [10,20,30];
var novaGaveta = gaveteiro.slice(1,3);
console.log(novaGaveta); //[20, 30]
```

Quantidade de elementos do array

Depois de ter adicionado várias gavetas, pode surgir a necessidade de saber quantas já existem, para isso vamos acessar a propriedade length:

```
var gaveteiro = [1,2,3,10,20,30];
console.log(gaveteiro.length); //6
gaveteiro.push(100);
gaveteiro.push(200);
gaveteiro.push(300);
gaveteiro.push(400);
console.log(gaveteiro.length); //10
gaveteiro.push(200);
console.log(gaveteiro.length); //9
```

Unidade Temática 9.7: O que são Objectos?

Programação orientada a objectos

Objectos são como uma espécie de "super variáveis" que armazenam uma "coleção de valores" referenciados por nome, e que podem ser recuperados para serem utilizados em diversas outras partes de um programa. Em JavaScript praticamente qualquer tipo de dado é um objecto.

Cada item dessa "coleção de valores", é chamado de propriedade. Cada propriedade é composta por um par de "nome: valor". Quando uma propriedade armazena uma função, ela se torna o que chamamos de método.

Criando objectos

Agora que já sabemos o que são objectos, vamos ver um pouco sobre como trabalhar com eles. Primeiramente vão conhecer duas maneiras de se criar objectos.

Notação literal

A maneira mais simples (e recomendável) de se criar objetos em JavaScript é usando o que chamamos de notação literal. Um objeto literal é composto por um par de chaves "{}", que envolve uma ou mais propriedades. Cada propriedade segue o formato "nome: valor" e devem ser separadas por vírgula.

Para entender bem, nada melhor que um exemplo. Imagine que você vai criar um programa para organizar álbuns de vários cantores e bandas. Aqui vamos criar um objeto para armazenar informações sobre um álbum da banda Metallica, depois você pode praticar criando objetos com suas bandas favoritas. Então, mãos à obra!

```
var album = {
    title: "Metallica (Black Album)",
    released: 1991,
    showInfo: function() {
        alert("Título do álbum: " + this.title + "Lançado em: " +
this.released);
    }
};
```

Consegui entender o que o código acima faz? Veja, é bem simples: primeiro criamos uma variável chamada "album". Depois criamos um objeto - note a abertura e fechamento das chaves: { e }. Então adicionamos duas propriedades e um método ao nosso objeto, que são: "title", "released" e "showInfo". Nas propriedades nós armazenamos o título e ano de lançamento do álbum, e no método temos uma função que irá exibir as informações sobre o álbum em uma "caixa de alerta" para o usuário. Mais fácil do que parece, não é mesmo?

Função construtora

Outra maneira de criar objetos em JavaScript é utilizando uma função construtora. Se quisermos criar o mesmo objeto que criamos anteriormente, só que usando uma função construtora para isso, basta escrever o seguinte código:

```
var album = new Object();
album.title = "Metallica (Black Album);
```

```
album.released = 1991;
album.showInfo = function() {
    alert("Título do álbum: " + this.title + "Lançado em: " + this.released);
};
```

Como você pôde notar, a sintaxe ficou um pouco diferente. Aqui devemos utilizar a palavra-chave new seguida pela função construtora Object() ao invés de abrir e fechar chaves. Depois nós adicionamos as propriedades e métodos utilizando album.title, album.released e album.showInfo e atribuímos os valores à elas ao invés de colocar os pares de "nome: valor".

Acessando propriedades e métodos

Após ter criado um objeto, você vai precisar acessar os valores que ele armazena. Podemos acessar (ou se preferir: "recuperar") os valores guardados em um objeto, de duas maneiras: utilizando notação de ponto ou notação de colchetes.

Veja um exemplo:

```
// notação de ponto
album.title // Retorna: Metallica (Black Album)

// notação de colchetes
album["title"] // Retorna: Metallica (Black Album)
```

Repare que no código acima, acessamos a mesma propriedade de duas maneiras diferentes. Geralmente é recomendável que você utilize a notação de ponto - album.title - por ser mais simples de ler e escrever.

Como os métodos são funções, você deve adicionar um par de parênteses - () - quando for acessá-los. Fora isso, nada de diferente.

Veja no exemplo abaixo:

```
// notação de ponto
album.showInfo() // Exibe alerta:
// Título do álbum: Metallica (Black Album) Lançado em: 1991

// notação de colchetes
```

```
album["showInfo"]() // Exibe alerta:  
// Título do álbum: Metallica (Black Album) Lançado em: 1991
```

Alterando e adicionando propriedades

Alterando

Vez por outra vamos precisar alterar os valores armazenados nas propriedades de nossos objetos. Fazer isso também é bem tranquilo. Basta acessar a propriedade que deseja alterar, utilizando a notação de ponto que acabamos de conhecer, e atribuir o novo valor à ela. Quer um exemplo?

```
album.title = "Powerslave";  
album.released = 1984;
```

O que aconteceu no código acima? Isso mesmo. Alteramos o título do álbum e o ano de lançamento. Agora nosso objeto armazena informações sobre um outro álbum de outra banda.

Para fixar, antes de prosseguir a leitura (supondo que você esteja lendo e digitando os códigos para treinar), altere os valores de title e released para Metallica (Black Album) e 1991 novamente.

Adicionando

Bom, agora que o título do nosso álbum voltou a ser "Metallica (Black Album)", que tal adicionar uma lista com os títulos das faixas do álbum? Sim, nós podemos adicionar novas propriedades e métodos aos nossos objetos mesmo após ter criado eles. A sintaxe é a mesma utilizada para alterar valores, que nós acabamos de ver.

Objectos podem armazenar qualquer tipo de dado válido em JavaScript, então, para criar uma lista com os títulos das faixas de nosso álbum, basta seguir o exemplo abaixo:

```
// Aqui adicionamos um array com os nomes de algumas faixas do
// álbum.
// Para praticar você pode adicionar todas as 12 faixas.
album.tracks = ["Enter Sandman", "Sad but True", "Holier Than Thou",
"The Unforgiven"];
```

Deletando propriedades

Você pode deletar uma propriedade ou método de um objeto utilizando o operador `delete` seguido pelo nome da propriedade. Vamos testar?

```
typeof album.showInfo // "function"
delete album.showInfo // deleta o método showInfo
typeof album.showInfo // "undefined"
```

Leitura adicional

Agora você já sabe o básico sobre objetos em JavaScript. Para uma abordagem mais profunda sobre o assunto, consultar bons livros pode ser de grande ajuda. Seguem algumas recomendações:

"O Melhor do JavaScript", por Douglas Crockford. Capítulo 4, Página 26.
"JavaScript - O Guia Definitivo" - 6ª Edição, por David Flanagan. Capítulo 6, Página 112.

Unidade Temática 9.8: O que é um Framework?

Entendendo o conceito

Em uma busca rápida na web, é possível encontrar diversas definições para Framework, algumas simples, outras mais elaboradas, mas o ponto comum entre todas é a reusabilidade.

Assim um Framework tem como principal objetivo resolver problemas recorrentes com uma abordagem genérica, permitindo ao desenvolvedor focar seus esforços na resolução do problema em si, e não ficar reescrevendo *software*. Você pode se perguntar, então Framework é uma biblioteca? Bem quase isso, pode-se dizer que é um conjunto de bibliotecas ou componentes que são usados para criar uma base onde sua aplicação será construída.

As frameworks ajudam no desenvolvimento rápido e seguro de aplicações mas é recomendável, estudar antes a tecnologia em que a mesma é desenvolvida. Logo é importante estudar os aspectos básicos do javascript antes de se aventurar em uma framework da Linguagem. Possuindo o conhecimento das tecnologias da Framework é possível fazer suas próprias modificações para que a framework possa atender todas as necessidades do desenvolvedor.

Desenvolvimento

Web

Existe diversas frameworks para desenvolvimento de aplicações, mas o nosso foco será em ferramentas para criação de *sites*. Abaixo são listados alguns que podem ajudar na sua carreira.

Bootstrap

É utilizado para desenvolvimento de componentes de interface de *sites*, utilizando html, css e javascript e foi desenvolvido levando em consideração técnicas de design, para melhorar a experiência visual do usuário.

Unidade Temática 9.9: O que é jQuery?

É um framework JavaScript

Lançada em 2006, por John Resig, jQuery, segundo definição consta em seu site, trata-se de uma rápida, pequena e rica em features biblioteca JavaScript.

Como o lema "Write less, do more." (escreva menos, faça mais), jQuery revolucionou o desenvolvimento web, sendo presente em inúmeros projetos atualmente.

Mas, por que é tão utilizado?

Exatamente pelo seu lema. Com jQuery é possível fazer diversos efeitos com poucas linhas e, que custariam dezenas de linhas em JavaScript puro.

Alguns recursos oferecidos facilmente pelo jQuery:

Seleção e manipulação de elementos HTML

Manipulação de CSS
Efeitos e animações
Navegação pelo DOM
Ajax
Eventos

Como uso?

Igualmente qualquer arquivo JavaScript você precisa inseri-lo na página com as tags script. Você pode fazer o link para uma CDN como a do Google ou fazer o download do framework no seu projeto e linkar direto (vale lembrar que se optar por essa opção, faça o link para a versão minificada).

<script src="js/jquery.min.js"></script>
A sintaxe
A sintaxe básica é a seguinte:

\$(seletorHTML').acao();
Alguns exemplos práticos
Manipulação HTML
\$("#botao").html("Sucesso!")
No exemplo acima, selecionamos o elemento com ID botão e inserimos a string "Sucesso!".

Eventos

```
var box = $( "#box" );
$( "#botao" ).on( "click", function( event ) {
    box.show();
});
```

No snippet acima, guardarmos em uma variável o elemento com ID box. Em seguida, adicionao ao elemento com ID 'botao' o evento do clique. Ao ser disparado, esse evento mostra o elemento, que guardarmos na variável box anteriormente, através do método show().

Exercícios de AUTO-AVALIAÇÃO

1. Qual é o uso de JavaScript?

O JavaScript é mais comumente usado como uma linguagem de script do lado do cliente. Isso significa que o código JavaScript é gravado em uma página HTML. Quando um usuário solicita uma página HTML com JavaScript, o script é enviado ao navegador e cabe ao navegador fazer algo com ele.

2. O que é JavaScript e como funciona?

JavaScript é o que é chamado de uma linguagem de script do lado do cliente. Dentro de uma página da Web normal, você coloca um código JavaScript (consulte Como funcionam as páginas da Web para obter detalhes sobre páginas da Web). Quando o navegador carrega a página, o navegador tem um interpretador interno que lê o código JavaScript que encontra na página e o executa.

3. Como faço para activar o JavaScript no Google Chrome?

[Activar JavaScript no Google Chrome](#)

1. No seu computador, abra o Chrome.
2. No canto superior direito, clique em Mais configurações.
3. Na parte inferior, clique em Avançado.
4. Em "Privacidade e segurança", clique em Configurações de conteúdo.
5. Clique em JavaScript.
6. Active o permitido (recomendado).

4. Para que é JavaScript usado, principalmente?

JavaScript é uma linguagem de programação usada principalmente pelos navegadores da Web para criar uma experiência dinâmica e interativa para o usuário. A maioria das funções e aplicativos que tornam a Internet indispensável para a vida moderna é codificada em alguma forma de JavaScript.

5. Por que usamos JavaScript?

Em uma linguagem de programação, variáveis são usadas para armazenar valores de dados. JavaScript usa a palavra-chave var para declarar variáveis. Um sinal de igual é usado para atribuir valores a variáveis. Usamos Javascript porque permite-nos programar sobre páginas web.

6. Qual é a vantagem do JavaScript?

Uma das maiores vantagens para o JavaScript, é ter a capacidade de produzir o mesmo resultado em todos os navegadores modernos. Execução do lado do cliente: Não importa onde você hospeda o JavaScript, execute sempre no ambiente do cliente para economizar largura de banda e acelerar o processo de execução.

7. O JavaScript é seguro?

JavaScript é "relativamente seguro", mas não "absolutamente seguro". Qualquer código que você executa em seu sistema tem potencial para causar danos. Não há sistema perfeitamente seguro, exceto o que nunca foi usado.

8. O JavaScript é gratuito para usar?

Downloads JavaScript são gratuitos. Não há necessidade de pagar por scripts escritos em JavaScript, porque praticamente qualquer script imaginável está disponível em algum lugar como o *download* gratuito do JavaScript.

9. O que é JavaScript e seus recursos?

JavaScript é uma tecnologia do lado do cliente, é usada principalmente para a validação do lado do cliente, mas tem muitos recursos que são fornecidos abaixo: JavaScript é uma linguagem de script baseada em objecto. Dando ao usuário mais controle sobre o navegador. JavaScript é uma linguagem de script baseada em interpretador.

10. Como eu executo o JavaScript?

O código Javascript pode ser executado dentro de um navegador da Web ou na linha de comando do seu computador usando o Node.js. Esta resposta irá abordar formas de experimentar com o Javascript através de um navegador da web.

```
<html>
<head>
<title>
</title>
</head>
<body>
<script>
console.log ("javascript está funcionando");
</script>
```

11. Como faço para testar o JavaScript no Chrome?

Abra o Chrome, pressione Ctrl + Shift + j e abra o console do JavaScript, onde você pode escrever e testar seu código.

12. Qual é o uso de strict em JavaScript?

A directiva "use strict" era nova no ECMAScript versão 5. Não é uma declaração, mas uma expressão literal, ignorada por versões anteriores do JavaScript. O propósito de "usar estrito" é indicar que o código deve ser executado em "modo estrito". Com o modo estrito, você não pode, por exemplo, usar variáveis não declaradas.

13. Qual é a diferença entre == e ===?

Em uma palavra, a principal diferença entre o operador "==" e "===" é que anteriormente compara a variável fazendo a correção do tipo, e se você comparar um número com o astring com literal numérico, == permite isso, mas === não permite isso, porque ele não apenas verifica o valor, mas também o tipo de duas variáveis, se duas variáveis são do mesmo tipo ou não.

14. Por que o JavaScript é importante?

A importância do JavaScript como uma tecnologia da web pode ser determinada pelo facto de que ele é usado actualmente por 94,5% de todos os sites. Como uma linguagem de programação do lado do cliente, o JavaScript ajuda os desenvolvedores da Web a tornar as páginas da Web dinâmicas e interactivas, implementando scripts personalizados do lado do cliente.

15. JavaScript precisa de um compilador?

JavaScript é uma linguagem interpretada, não uma linguagem compilada. Um programa como C ++ ou Java precisa ser compilado antes de ser executado. O código-fonte

é passado através de um programa chamado compilador, que o converte em código de bytes que a máquina entende e pode executar, o Javascript não.

16. Quão seguro é o JavaScript?

Um dos erros mais comuns é a injecção de HTML, permitindo que terceiros injectem JavaScript em seu contexto de segurança. Isso permite que um invasor controle o que um usuário faz em seu site, quebrando completamente a segurança da conta. Javascript corre no lado do cliente, então o maior risco é para o cliente.

17. Qual é a diferença entre Java e JavaScript?

Principais diferenças entre Java e JavaScript: Java é uma linguagem de programação orientada à objectos, enquanto Java Script é uma linguagem de script. Java cria aplicativos que são executados em uma máquina virtual ou navegador, enquanto o código JavaScript é exibido em um navegador. O código Java precisa ser compilado enquanto o código JavaScript está todo em texto.

18. Quais são as aplicações do JavaScript?

JavaScript é a linguagem de programação mais usada no mundo. Tem o maior repositório de pacotes de código aberto do mundo. Todo tipo de aplicativo de *software* usa JavaScript, incluindo código de servidor (Node.js), aplicativos de produtividade, jogos 3D, robôs e dispositivos IoT (Internet of Things).

19. Do While JavaScript?

As instruções while e do ... while em JavaScript são semelhantes às declarações condicionais, que são blocos de código que serão executados se uma condição especificada resultar em true. Ao contrário de uma instrução if, que só avalia uma vez, um loop será executado várias vezes até que a condição não seja mais avaliada como verdadeira.

20. O que você pode fazer com JavaScript?

Tudo o que você pode fazer com o Javascript:

- *Websites*. Esse é o óbvio;
- *Chatbots*. Chatbots são a próxima onda em marketing e tecnologia interativa;
- Scripts do sistema. Depois que o node.js entrou em cena, o Javascript tornou-se uma linguagem de utilitário do sistema;
- *Videogames*. Não se contente em ficar preso na internet, o Javascript está sendo usado para construir mecanismos inteiros de jogos.

21. O CSS é compilado ou interpretado?

Com o CSS, eles também são conhecidos como "pré-processadores". JavaScript é uma linguagem interpretada, não uma linguagem compilada. No entanto, muitos idiomas diferentes podem ser transpostos para o JavaScript, que é a única coisa que o navegador entenderá. TraceMonkey foi o primeiro compilador JIT escrito para a linguagem JavaScript.

22. Por que o JavaScript é necessário?

Porque é uma linguagem de script e porque usa o navegador para fazer o trabalho completo. Se você comandar uma imagem para ser substituída por outra, o JavaScript diz ao navegador para fazer isso. Como o navegador realmente faz o trabalho, você só precisa extrair algumas strings escrevendo algumas linhas de código relativamente fáceis.

23. Qual é a vantagem do JavaScript?

A maior vantagem do JavaScript, para além de ter a capacidade de produzir o mesmo resultado em todos os navegadores modernos. É a execução do lado do cliente: Não importa onde você hospeda o JavaScript, execute sempre no ambiente do cliente para economizar largura de banda e acelerar o processo de execução.

24. Por que o JavaScript é útil?

É por isso que o JavaScript é tão popular. É por isso que ainda vale a pena aprender JavaScript e se tornar um desenvolvedor front-end. Ele permite o processamento no lado do

cliente, o que reduz a carga no lado do servidor e reduz drasticamente a capacidade de processamento necessária. Além disso, faz com que as páginas da web ganhem vida com animações e comportamento.

Referências Bibliográficas

1. "What is the difference between the Web and the Internet?". W3C Help and FAQ. W3C. 2009. Archived from the original on 9 July 2015. Retrieved 16 July 2015.
2. McPherson, Stephanie Sammartino (2009). Tim Berners-Lee: Inventor of the World Wide Web. Twenty-First Century Books. ISBN 978-0-8225-7273-2. Archived from the original on 15 April 2016.
3. Quittner, Joshua (29 March 1999). "Network Designer Tim Berners-Lee". Time Magazine. Archived from the original on 15 August 2007. Retrieved 17 May 2010. He wove the World Wide Web and created a mass medium for the 21st century. The World Wide Web is Berners-Lee's alone. He designed it. He set it loose it on the world. And he more than anyone else has fought to keep it an open, non-proprietary and free.
4. "World Wide Web Timeline". Pew Research Center. 11 March 2014. Archived from the original on 29 July 2015. Retrieved 1 August 2015.

6. Dewey, Caitlin (12 March 2014). "36 Ways the Web Has Changed Us". The Washington Post. Archived from the original on 9 September 2015. Retrieved 1 August 2015.
7. "Internet Live Stats". Archived from the original on 2 July 2015. Retrieved 1 August 2015.
8. Joseph Adamski; Kathy Finnegan (2007). New Perspectives on Microsoft Office Access 2007, Comprehensive. Cengage Learning. p. 390. ISBN 978-1-4239-0589-9.
9. Enzer, Larry (31 August 2018). "The Evolution of the World Wide Web". Monmouth Web Developers. Retrieved 31 August 2018.
- 10."Archived copy" (PDF). Archived from the original (PDF) on 17 November 2015. Retrieved 26 August 2015.
11. "Distributed Application Architecture" (PDF). Sun Microsystem. Archived from the original (PDF) on 6 April 2011. Retrieved 2009-06-16.
12. The X Window System is one example.
- 13.Benatallah, B.; Casati, F.; Toumani, F. (2004). "Web service conversation modeling: A cornerstone for e-business automation". IEEE Internet Computing. 8: 46–54. doi:10.1109/MIC.2004.1260703.

- 14.Dustdar, S.; Schreiner, W. (2005). "A survey on web services composition" (PDF). International Journal of Web and Grid Services.
- 15.Jump up to: Rulifson, Jeff (June 1969). DEL. IETF. doi:10.17487/RFC0005. RFC 5. Retrieved 30 November 2013.
- 16.Shapiro, Elmer B. (March 1969). Network Timetable. IETF. doi:10.17487/RFC0004. RFC 4. Retrieved 30 November 2013.
- 17.Duarte, Fábio e Frei, Klaus. Redes Urbanas. In: Duarte, Fábio; Quandt, Carlos; Souza, Queila. (2008). O Tempo Das Redes, p. 156. Editora Perspectiva S/A. ISBN 978-85-273-0811-3
- 18.Capra, Fritjof. Vivendo Redes. In: Duarte, Fábio; Quandt, Carlos; Souza, Queila. (2008). O Tempo Das Redes, pp. 21/23. Editora Perspectiva S/A. ISBN 978-85-273-0811-3
- 19.Social Targuet- Afonso, C., Borges, L., Top Books 2013
- 20.Linton Freeman, The Development of Social Network Analysis. Vancouver:Empirical Press, 2006.
- 21.LEMIEUX,VINCENT. MATHIEU OUIMET, Sérgio Pereira. Análise Estrutural das Redes Sociais. 1ª Edição.Instituto Piaget. 2008/01. ISBN 9789727719334

- 22.Jamil Marques e Camila de Paula Pessoa (2013). «Twitter, Eleições e Poder Local : Um estudo sobre os vereadores de Fortaleza». Contemporanea - Revista de Comunicação e Cultura + Portal de Periódicos Eletrônicos da Universidade Federal da Bahia. ISSN 1809-9386. Consultado em 28 de setembro de 2016. Cópia arquivada em 4 de março de 2018
- 23.Francisco Paulo Jamil Almeida Marques, Jakson Alves de Aquino e Edna Miola (2014). «Parlamentares, representação política e redes sociais digitais : perfis de uso do Twitter na Câmara dos Deputados» (PDF). Revista Opinião Pública da Universidade Estadual de Campinas + SciELO Brasil. doi:10.1590/1807-01912014202178. Consultado em 4 de março de 2018. Cópia arquivada (PDF) em 4 de março de 2018
- 24.«Cascading Style Sheets». www.w3.org (em inglês). Consultado em 25 de março de 2017
- 25.«Como Inserir CSS». www.escolaw3.com. Consultado em 25 de março de 2017
- 26.«W3C CSS2.1 specification for rule sets, declaration blocks, and selectors». W3.org. Consultado em 20 de junho de 2009
- 27.«Sintaxe e Seletores CSS». www.escolaw3.com. Consultado em 25 de março de 2017
- 28.«Standard ECMA-262». ecma-international.org. Consultado em 13 de fevereiro de 2019
- 29.Flanagan, David; Ferguson, Paula (2002). JavaScript: The Definitive Guide 4th ed. [S.I.]: O'Reilly & Associates. ISBN 0-596-00048-0
- 30.«javascript - What is VanillaJS?». Stack Overflow. Consultado em 16 de março de 2019
- 31.«Vanilla JS». vanilla-js.com. Consultado em 16 de março de 2019
- 32.«WebCite query result» (PDF). www.webcitation.org. Consultado em 16 de março de 2019
- 33.«ECMAScript Language Specification» (PDF). Consultado em 10 de fevereiro de 2011. Arquivado do original (PDF) em 12 de abril de 2015
- 34.«TimelineJS Embed». cdn.knightlab.com. Consultado em 17 de março de 2019
- 35.«Chapter 4. How JavaScript Was Created». speakingjs.com. Consultado em 17 de março de 2019
- 36.Dave Raggett (1998). Raggett on HTML 4. [S.I.]: Addison-Wesley's. pp. chap. 2: A history of HTML. ISBN 0-201-17805-2
- 37.«HTML working group charter (2000–2002)». World Wide Web Consortium

- 38.«HTML working group charter (2002–2004)». World Wide Web Consortium
 - 39.«HTML Working Group Roadmap». World Wide Web Consortium
 - 40.Karl Dubost (15 de janeiro de 2008). «HTML5, one vocabulary, two serializations» (em inglês). W3C. Consultado em 29 de outubro de 2011
 - 41.«Introdução ao HTML». www.escolaw3.com. Consultado em 25 de março de 2017
 - 42.«Official Google Webmaster Central Blog: Google does not use the keywords meta tag in web ranking». googlewebmastercentral.blogspot.com. Consultado em 21 de fevereiro de 2011
 - 43.T. Berners-Lee et all, 1996
 - 44.Mark Nottingham (7 de Junho de 2014). «RFC2016 está morto» (em inglês). Mark Nottingham. Consultado em 13 de junho de 2015
 - 45.RFC 760, DOD Standard Internet Protocol (January 1980)
 - 46.RFC 1883, Internet Protocol, Version 6 (IPv6) Specification, S. Deering, R. Hinden (December 1995)
-
- 47.RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, S. Deering, R. Hinden, The Internet Society (December 1998)
 - 48.«[www.wikipedia.org dns report](http://www.wikipedia.org/dns/report)». thednsreport.com. Consultado em 4 de maio de 2012
 - 49.«IANA — Internet Assigned Numbers Authority». www.iana.org. Consultado em 4 de maio de 2012
 - 50.Postel, J., & Reynolds, J. (October 1985). RFC 959. In The Internet Engineering Task Force. Retrieved from <http://www.ietf.org/rfc/rfc0959.txt>
 - 51.Kurose, J.F. & Ross, K.W. (2010). Computer Networking. 5th ed. Boston, MA: Pearson Education, Inc.
 - 52.Forouzan, B.A. (2000). TCP/IP: Protocol Suite. 1st ed. New Delhi, India: Tata McGraw-Hill Publishing Company Limited.
 - 53.Allman, M. & Metz, C. & Ostermann, S. (September 1998). RFC 2428. In The Internet Engineering Task Force. Retrieved from <http://www.ietf.org/rfc/rfc2428.txt>
 - 54.Clark, M.P. (2003). Data Networks IP and the Internet. 1st ed. West Sussex, England: John Wiley & Sons Ltd.
 - 55.«RFC 1738 Uniform Resource Locators (URL)»
 56. <https://titanftp.com>
 - 57.Network Working Group of the IETF, January 2006, RFC 4251, The Secure Shell (SSH) Protocol Architecture

- 58.Network Working Group of the IETF, January 2006, RFC 4252, The Secure Shell (SSH) Authentication Protocol
- 59.Peter Bright (June 2, 2015). «Microsoft bringing SSH to Windows and PowerShell».
- 60.SSH Hardens the Secure Shell, Serverwatch.com
- 61.«Prying Eyes: Inside the NSA's War on Internet Security». Spiegel Online. December 28, 2014
- 62.«Service Name and Transport Protocol Port Number Registry».
- 63.«Network+ Guide to Networks - Tamara Dean - Google Books». books.google.com.br. Consultado em 23 de abril de 2012
- 64.«Service Name and Transport Protocol Port Number Registry». www.iana.org. Consultado em 8 de maio de 2017
- 65.«Service Name and Transport Protocol Port Number Registry». www.iana.org. Consultado em 8 de maio de 2017
- 66.Open Source Awards. «2011 Open Source Awards». Consultado em 17 de fevereiro de 2012
- 67.Liriana Batista da Silva Pesco (2013). «Ferramentas para criação de sites e portais CMS» (pdf). Universidade Federal do Paraná. 17 páginas. Consultado em 30 de janeiro de 2018
- 68.«Vocabulário Ortográfico da Língua Portuguesa». Academia Brasileira de Letras. Consultado em 7 de janeiro de 2017
- 69.S.A., Priberam Informática,. «Significado / definição de blogue no Dicionário Priberam da Língua Portuguesa». www.priberam.pt. Consultado em 7 de janeiro de 2017
- 70.Porto Editora. «Definição de blogue». Dicionário da Língua Portuguesa - com Acordo Ortográfico. Consultado em 25 de Março de 2011
- 71.«About Technorati - Technorati». 13 de abril de 2012. Consultado em 7 de janeiro de 2017
- 72."When Apps Meet Cloud Storage: Tectonic Shift Ahead". WIRED. Retrieved 2018-05-04.
- 73.David Strom (September 19, 2011). Two New Ways to Share Vids Online. Read Write Web. Retrieved May 24, 2012.
- 74.WeVideo. Our Story. WeVideo.com. Retrieved May 29, 2012.
- 75.Jump up to:^{a b} Wade Roush (December 2, 2011). WeVideo Makes Cloud Video Editing Look Like Kids' Stuff. Xconomy. Retrieved May 24, 2012.
- 76.Creaza Education. Creaza Education - Home. creazaeducation.com. Retrieved May 29, 2012.

77. WeVideo. WeVideo Debuts First Cloud-based, Collaborative Video Editing Platform. WeVideo.com. Retrieved May 29, 2012.
78. Sarah Perez (January 9, 2011). Cloud-Based Video Editor WeVideo Launches To Public. TechCrunch. Retrieved May 24, 2012.
79. Jaymar Cabebe (November 10, 2011). WeVideo: The next big thing?. CNET. Retrieved May 24, 2012.
80. «About Dropbox». Consultado em 10 de fevereiro de 2014
81. «"6 opções de software gratuito para sua empresa"». Exame Abril. Consultado em 23 de abril de 2014
82. «"Em nova rodada de financiamento dropbox e avaliado em US 10 bilhões"». Valor Economico. Consultado em 23 de abril de 2014
83. «Após Facebook e Yelp, vocalista do U2 faz investimento pesado também no Dropbox». 4 de abril de 2012. Consultado em 10 de fevereiro de 2014
84. Security update & new features (em inglês)
85. «RapidShare diz: Se fecharmos, YouTube e Dropbox serão os próximos». 30 de janeiro de 2012. Consultado em 10 de fevereiro de 2014
86. Site Overview wetransfer.com, Alexa Internet. Retrieved on 28 September 2016.
-
87. Eva Oude Elferink (2013-10-21). "Nalden: 'Door het design van WeTransfer durven mensen het te gebruiken'" (in Dutch). Volkskrant.nl. Retrieved 2013-10-28.
88. "WeTransfer Plus". 2018-08-23.
89. Hardwick, Tim (October 30, 2018). "WeTransfer Relaunches Mobile File-Sharing App as 'Collect'". Mac Rumors. Retrieved 2 November 2018.
90. Lucic, Kristijan (October 29, 2018). "WeTransfer App Gets Rebranded To 'Collect by WeTransfer'". Android Headlines. Retrieved 2 November 2018.
91. «Introducing Google Drive... yes, really». Google. Consultado em 24 de abril de 2012
92. Mossberg, Walter S. (25 de abril de 2012). «Google Stores, Syncs, Edits in the Cloud». The Wall Street Journal. Consultado em 25 de abril de 2012
93. «Google Drive: What we know so far». TechCrunch. Consultado em 24 de abril de 2012
94. Google Drive - Armazenamento em nuvem Arquivado em 3 de março de 2016, no Wayback Machine. webinfor.net

95. Murph, Darren. «Google Drive official: 5GB of free storage, business-focused approach (video)». Engadget. Consultado em 24 de abril de 2012

96. «How Google storage plans work - Google Drive Help». Support.google.com. Consultado em 26 de abril de 2012