

Universidade Virtual Africana

INFORMÁTICA APLICADA: ITI 4305

FUNDAMENTOS DAS TECNOLOGIAS WEB E PROGRAMAÇÃO DE INTERNET

Flavio Semedo

Prefácio

A Universidade Virtual Africana (AVU) orgulha-se de participar do aumento do acesso à educação nos países africanos através da produção de materiais de aprendizagem de qualidade. Também estamos orgulhosos de contribuir com o conhecimento global, pois nossos Recursos Educacionais Abertos são acessados principalmente de fora do continente africano.

Este módulo foi desenvolvido como parte de um diploma e programa de graduação em Ciências da Computação Aplicada, em colaboração com 18 instituições parceiras africanas de 16 países. Um total de 156 módulos foram desenvolvidos ou traduzidos para garantir disponibilidade em inglês, francês e português. Esses módulos também foram disponibilizados como recursos de educação aberta (OER) em oer.avu.org.

Em nome da Universidade Virtual Africana e nosso patrono, nossas instituições parceiras, o Banco Africano de Desenvolvimento, convido você a usar este módulo em sua instituição, para sua própria educação, compartilhá-lo o mais amplamente possível e participar ativamente da AVU Comunidades de prática de seu interesse. Estamos empenhados em estar na linha de frente do desenvolvimento e compartilhamento de recursos educacionais abertos.

A Universidade Virtual Africana (UVA) é uma Organização Pan-Africana Intergovernamental criada por carta com o mandato de aumentar significativamente o acesso a educação e treinamento superior de qualidade através do uso inovador de tecnologias de comunicação de informação. Uma Carta, que estabelece a UVA como Organização Intergovernamental, foi assinada até agora por dezenove (19) Governos Africanos - Quênia, Senegal, Mauritânia, Mali, Costa do Marfim, Tanzânia, Moçambique, República Democrática do Congo, Benin, Gana, República da Guiné, Burkina Faso, Níger, Sudão do Sul, Sudão, Gâmbia, Guiné-Bissau, Etiópia e Cabo Verde.

As seguintes instituições participaram do Programa de Informática Aplicada: (1) Université d'Abomey Calavi em Benin; (2) Université de Ougadougou em Burkina Faso; (3) Université Lumière de Bujumbura no Burundi; (4) Universidade de Douala nos Camarões; (5) Universidade de Nouakchott na Mauritânia; (6) Université Gaston Berger no Senegal; (7) Universidade das Ciências, Técnicas e Tecnologias de Bamako no Mali (8) Instituto de Administração e Administração Pública do Gana; (9) Universidade de Ciência e Tecnologia Kwame Nkrumah em Gana; (10) Universidade Kenyatta no Quênia; (11) Universidade Egerton no Quênia; (12) Universidade de Addis Abeba na Etiópia (13) Universidade do Ruanda; (14) Universidade de Dar es Salaam na Tanzânia; (15) Université Abdou Moumouni de Niamey no Níger; (16) Université Cheikh Anta Diop no Senegal; (17) Universidade Pedagógica em Moçambique; E (18) A Universidade da Gâmbia na Gâmbia.

Bakary Diallo

O Reitor

Universidade Virtual Africana

Créditos de Produção

Autor

Flavio Semedo

Par revisor(a)

Célio Sengo

UVA - Coordenação Académica

Dr. Marilena Cabral

Coordenador Geral Programa de Informática Aplicada

Prof Tim Mwololo Waema

Coordenador do módulo

Victor Odumuyiwa

Designers Instrucionais

Elizabeth Mbasu

Benta Ochola

Diana Tuel

Equipa Multimédia

Sidney McGregor

Michal Abigael Koyier

Barry Savala

Mercy Tabi Ojwang

Edwin Kiprono

Josiah Mutsogu

Kelvin Muriithi

Kefa Murimi

Victor Oluoch Otieno

Gerisson Mulongo

Direitos de Autor

Este documento é publicado sob as condições do Creative Commons

[Http://en.wikipedia.org/wiki/Creative_Commons](http://en.wikipedia.org/wiki/Creative_Commons)

Atribuição <http://creativecommons.org/licenses/by/2.5/>



O Modelo do Módulo é copyright da Universidade Virtual Africana, licenciado sob uma licença Creative Commons Attribution-ShareAlike 4.0 International. CC-BY, SA

Apoiado por



Projeto Multinacional II da UVA financiado pelo Banco Africano de Desenvolvimento..

Tabela de conteúdo

Prefácio	2
Créditos de Produção	3
Direitos de Autor	4
Descrição Geral do Curso	7
Bem-vindo(a) a Fundamento das Tecnologias Web e Programação de Internet . . .	7
Pré-requisitos	7
Materiais	7
Objetivos do Curso	7
Objetivos específicos:	8
Unidades.	8
Unidade 0: Introdução do curso e requisitos necessários	8
Unidade 1: Entender os fundamentos da Internet e World Wide Web	8
Unidade 2: Desenvolvimento e Programação de Páginas Web	8
Unidade 3: Programação Web avançada do lado do Servidor (Server Side)	8
Unidade 4: Hospedagem de páginas Web	8
Unidade 2	12
Unidade 3	12
Unidade 4	12
Unidade 0. Diagnóstico	14
Introdução à Unidade	14
Objetivos da Unidade	14
Avaliação da Unidade	14
Unidade 1. Fundamentos da internet e da World Wide Web	17
Introdução à Unidade	17
Objetivos da Unidade	17
Atividade 1 - A internet e a WWW (perspetiva histórica)	18
Perguntas:	19

Atividade 2 - Instalação e Configuração do Ambiente de Desenvolvimento	21
Perguntas:	22
Atividade 3 - Introdução às Tecnologias Web	23
Perguntas de autoavaliação	37
Perguntas de autoavaliação:	45
Atividade 4 - Cores em HTML/CSS3	46
Pergunta de autoavaliação:	54
Atividade 5- Formatação de elementos em HTML5 com CSS3	63
Unidade 2. Programação Web avançado do lado do servidor	68
Introdução	68
Objetivos da Unidade	68
Atividade 1. Noções básicas de programação com PHP	69
Atividade 2 - Sintaxe básicos com PHP	70
Atividade 3 - Expressões e Estrutura de controlo com PHP	75
Atividade 4 – Criando um projeto com PHP	81
Perguntas de autoavaliação	81
Perguntas de autoavaliação:	83
Atividade 5 – Manipular dados na base de dados com PHP	85
Perguntas de autoavaliação:	91
Unidade 3: Hospedagem de sites na Web	93
Objetivos da Unidade	93
Atividade 1 - Registo de nomes de domínio	94
Atividade 2 - Registrar o seu domínio com 1and1.com	100
Perguntas de autoavaliação	106

Descrição Geral do Curso

Bem-vindo(a) a Fundamento das Tecnologias Web e Programação de Internet

Este curso centraliza-se na programação para Web. Permite aos(as) estudantes adquirir conhecimentos e desenvolver habilidades em diferentes categorias de linguagem Web, assim como projetar, desenvolver e implementar aplicações Web modernas. Este é um módulo muito importante porque a Web é hoje o centro da economia digital.

Ter experiência em desenvolvimento para Web é uma vantagem importante frente à concorrência no mercado do trabalho centrado no uso das tecnologias.

Pré-requisitos

Para iniciar os treinamentos, o(a) estudante deve ter conhecimento mínimo de informática com domínio de um sistema operativo (Windows, Linux ou OS X) ou qualquer outro sistema operativo.

O (a) estudante deverá ter noções básicas de diretórios e ficheiros no contexto de um sistema operativo.

E por último, deve ter conhecimentos de fundamentos de linguagens de programação e os princípios básicos de programação: variáveis, atribuição, loops, estruturas de controle (if, else...).

O conhecimento dos editores de texto é uma grande vantagem para a aprendizagem deste módulo, mas realçamos que não é um requisito obrigatório.

Materiais

Os materiais necessários para completar este curso incluem:

- Um computador com um sistema operativo Windows, Linux ou MAC.
- Conexão à internet

Objetivos do Curso

Esta unidade curricular tem por objetivo capacitar os alunos para adquirir competências necessárias para projetar, desenvolver e implementar aplicações modernas centradas na Web.

Objetivos específicos:

No fim da unidade curricular, o(a) estudante deve ser capaz de:

Demonstrar o conhecimento sobre as principais tecnologias Web e o modelo três camadas (3 - tier) para o desenvolvimento de aplicações Web dinâmicas;

Desenvolver aplicações Web robustas usando os componentes Client Side (interface do utilizador) ;

Desenvolver aplicações seguras a custos mais baixos usando scripts do lado do servidor;

Compreender o processo para a hospedagem de sites na Web

Unidades

Unidade 0: Introdução do curso e requisitos necessários

Introdução às tecnologias Web e requisitos de instalação de ambientes de desenvolvimentos

Unidade 1: Entender os fundamentos da Internet e World Wide Web

Este capítulo irá ajudá-lo(a) a conhecer os conceitos e tecnologias que estão em jogo para o bom funcionamento da internet e World Wide Web.

Unidade 2: Desenvolvimento e Programação de Páginas Web

Este capítulo tem como objetivo ajudá-lo a praticar as linguagens básicas para o lado do cliente web. As linguagens estudadas são HTML, CSS, JavaScript e XML.

Unidade 3: Programação Web avançada do lado do Servidor (Server Side)

Este capítulo irá ajudá-lo(a) a desenvolver aplicações Web complexas que tiram partido de linguagens de programação do lado do servidor. PHP é usada como um exemplo devido à sua popularidade.

Unidade 4: Hospedagem de páginas Web

Este capítulo tem como objetivo ajudá-lo(a) a entender, passo a passo, o alojamento de uma aplicação na Web. Essas etapas incluem: a compra de um nome de domínio, escolher um anfitrião, o uso de painel, transferir ficheiros, e testar a aplicação.

Avaliação

Em cada unidade encontram-se incluídos instrumentos de avaliação formativa a fim de verificar o progresso do(a)s estudantes.

No final de cada módulo são apresentados instrumentos de avaliação sumativa, tais como testes e trabalhos finais, que compreendem os conhecimentos adquiridos e as competências desenvolvidas ao estudar este módulo.

A implementação dos instrumentos de avaliação sumativa fica ao critério da instituição que oferece o curso. A estratégia de avaliação sugerida é a seguinte:

	Título de Atividade	Nota (%)	Descrição
1	Avaliação sumativa	60%	Pontuação: 20 valores. Duração: mínima 03:00h. Esta avaliação pode ser de preferência um projeto para avaliar as habilidades e competências adquiridas.
2	Avaliação formativa - unidade 1	10%	Pontuação: 20 valores. Avaliação por meio de mini projetos on-line ou compostos pela avaliação contínua em curso.
3	Avaliação formativa - unidade 2	10%	Pontuação: 20 valores. Avaliação por meio de mini projetos on-line ou composta pela avaliação contínua em curso.
4	Avaliação formativa - unidade 3	10%	Pontuação: 20 valores. Avaliação por meio de mini projetos on-line ou composta pela avaliação contínua em curso.
5	Avaliação formativa - unidade 4	10%	Pontuação: 20 valores. Avaliação por meio de mini projetos on-line ou compostos pela avaliação contínua em curso.

Descrição de avaliação de notas:

Nota 0 - 20	Descrição
17-20	Muito Bom: o estudante foi além dos objetivos do curso
14-16	Bom. o estudante atingiu os objetivos do cursos
10-13	Suficiente: os objetivos foram parcialmente alcançados. Atividades corretivas são necessárias para melhor aprender as lições e desenvolver competências.
< 10	Insuficiente: o estudante não atingiu os objetivos mínimos do módulo

Calendarização

Unidade	Temas e Atividades	Estimativa do tempo
Sessão 1	Introdução da unidade 0. Objetivos e competências do curso Pré-requisitos Apresentação do curso Apresentação dos conteúdos Trabalhos para serem desenvolvidos a prazo e o sistema de pontuação Recursos disponíveis para o ensino e aprendizagem	2:00H
Sessão 2	Introdução da unidade 1 Leitura e pesquisa dos conteúdos apresentados da unidade 1 Realizar as atividades da unidade 1 Planeamento contínuo e supervisão das atividades da unidade 1	2:00H

Descrição Geral do Curso

Sessão 3	<p>Introdução da unidade 2</p> <p>Os alunos devem aplicar as atividades de aprendizagem da unidade 2</p> <p>Leitura e pesquisa dos conteúdos apresentados da unidade 2</p> <p>Realizar as atividades da unidade 2</p>	2:00H
Sessão 4	<p>Leitura e pesquisa dos conteúdos apresentados da unidade 2</p> <p>Realizar as atividades da unidade 2</p> <p>Planeamento contínuo e supervisão das atividades da unidade 2</p>	2:00H
Sessão 5	<p>Leitura e pesquisa dos conteúdos apresentados da unidade 2</p> <p>Realizar as atividades da unidade 2</p> <p>Planeamento contínuo e supervisão das atividades da unidade 2</p>	2:00H
Sessão 6	<p>Introdução da unidade 3</p> <p>Os alunos devem aplicar as atividades de aprendizagem da unidade 3.</p> <p>Leitura e pesquisa dos conteúdos apresentados da unidade 3</p> <p>Realizar as atividades da unidade 2</p> <p>Planeamento contínuo e supervisão das atividades da unidade 3</p>	2:00H
Sessão 7	<p>Introdução da unidade 4</p> <p>Leitura e pesquisa dos conteúdos apresentados da unidade</p> <p>Planeamento contínuo e supervisão das atividades da unidade 4</p> <p>Planeamento do exame final do semestre</p>	2:00H

Leituras e outros Recursos

As leituras e outros recursos deste curso são:

Unidade 2

Leituras e outros recursos obrigatórios:

Título: HTML5 e CSS3, domine a Web do futuro

Autor: Lucas Mazza

Título: HTML5 e CSS3, Web Design Responsivo, e A Web Mobile

Autor: Sergio Lopes, Lucas Mazza e Tércio Zemel

Unidade 3

Leituras e outros recursos obrigatórios:

Título: Desenvolvendo Aplicações com PHP e MySQL

Autor: Daniel dos Santos

Título: Desenvolvimento Web com PHP e MySQL

Autor: Evaldo Bento

Título: PHP Cookbook, Solutions & Examples for PHP programmers

Autores: David Sklar, Adam Trachtenberg

Unidade 4

Leituras e outros recursos obrigatórios:

Título: Escolha o seu .com, guia definitivo de como escolher o seu domínio

Autor: Ricardo Monteiro

Título: Web Hosting for Dummies

Autor: Peter Pollock

OUTRAS REFERENCIAS

- <http://www.w3resource.com/php-exercises/php-basic-exercises.php>
- http://www.tutorialspoint.com/php/php_arrays.htm
- https://php.net/manual/pt_BR/language.variables.scope.php
- http://www.w3schools.com/php/php_variables.asp
- <https://www.caelum.com.br/curso-php-mysql/>
- <http://www.htmlprogressivo.net/p/curso-de-html5-online-completo-e-gratis.html>
- <http://www.w3c.br/pub/Materiais/PublicacoesW3C/guia-xhtml-w3cbr.pdf>
- Site: <http://www.htmlprogressivo.net/p/curso-de-css-cs-cascading-style-sheets.html>
- <http://pt-br.html.net/tutorials/>

Unidade 0. Diagnóstico

Introdução à Unidade

O propósito desta unidade é verificar a compreensão do conhecimento que possui relacionado com este curso.

Objetivos da Unidade

Após a conclusão desta unidade, deverá ser capaz de:

Conhecer as operações básicas de um sistema operativo (criar diretório, ficheiros e navegar pelo sistema de ficheiros)

Demonstrar o seu conhecimento dos conceitos básicos de programação (variáveis, atribuições e estruturas condicionais)

Demonstrar a sua capacidade de pesquisa na Web usando um motor de busca como o Google

TERMOS-CHAVE

Sistema Operativo: é um programa ou conjunto de programas cuja a função é gerir os recursos de sistema, fornecendo uma interface entre o computador e o utilizador http://pt.wikipedia.org/wiki/Sistema_operativo

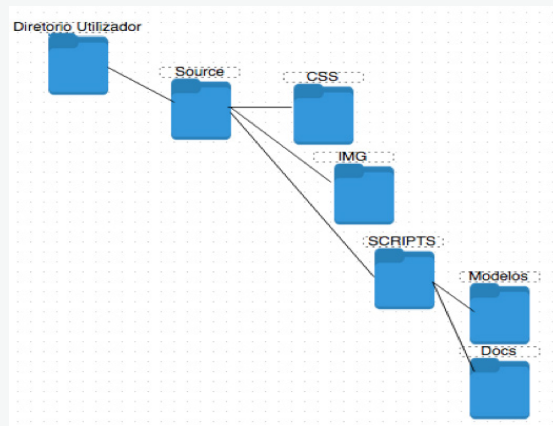
Ficheiro ou arquivo: é um recurso para armazenamento de informação que está disponível a um programa de computador e é normalmente baseado em algum tipo de armazenamento durável http://pt.wikipedia.org/wiki/Arquivo_de_computador

Programação: é um processo de escrita, teste e manutenção de um programa de computador http://pt.wikipedia.org/wiki/Programa%3%A7%C3%A3o_de_computadores

Avaliação da Unidade

Verifique a sua compreensão!

- 1) Dentro do diretório do utilizador do Windows, crie uma estrutura de diretório e subdiretórios.



2) Quais são os caminhos absolutos dos diferentes diretórios?

Diretórios	Caminho Absoluto
Source	
CSS	
IMG	
Script	
Modelos	
Vídeos	

3) Quais são os caminhos relativos de acordo com o diretório HOME?

Diretórios	Caminho Relativo
Source	
CSS	
IMG	
Script	
Modelos	
Texto	

4) Baixe um editor de texto (Notepad ++) e crie o seguinte arquivo texto TEXTO/texto.txt

Pesquisa na Web:

Ler o documento proposto por Tim Berners-Lee

História da Web

Pesquisar 50 (Cinquenta) sites mais famosos da África

1) Escreva um programa com base em uma linguagem de programação para exibir o seguinte:

*	*	
**	**	
***	***	*
****	****	
*****	*****	**

	*****	***

6) É possível criar um ficheiro ou pasta através de um terminal (linha de comando)?

- A. Sim
- B. Não

Leituras e Outros Recursos

As leituras e outros recursos desta unidade encontram-se na lista de "Leituras e Outros Recursos do curso".

Unidade 1. Fundamentos da internet e da World Wide Web

Introdução à Unidade

Esta unidade irá ajudar o(a) estudante a compreender os conceitos básicos da internet e a WWW, bem como as diferentes tecnologias envolvidas no desenvolvimento de aplicações WEB.

Objetivos da Unidade

Após a conclusão desta unidade deverá ser capaz de:

Definir os conceitos básicos da Internet e a WWW (World Wide Web).

Identificar os componentes da Web e entender o seu papel (HTML, HTTP, URL, endereçamento de recursos).

Saber a diferença entre os vários tipos de recursos da Web (Página estática, dinâmica e Ativos).

Distinguir a arquitetura das aplicações Web (Arquitetura Cliente/Servidor).

Aplicar exemplos de clientes e servidores Web (Web services).

TERMOS-CHAVE

Internet: É um sistema de redes global interligados que utilizam um conjunto de protocolos padrão de internet (TCP/IP).

WWW: World Wide Web

Páginas Web: Sistemas de informática projetados para a utilização através de um navegador, através de internet ou aplicativos desenvolvidos utilizando tecnologias Web (HTML/CSS/Javascript)

Cliente/Servidor: Aplicações distribuídas que distribuí 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.

Atividades de Aprendizagem

Actividades de leitura

- A internet e a WWW (perspectiva histórica).

- Introdução às tecnologias Web
- Arquitectura das aplicações Web
- Introdução a protocolos de internet (TCP/IP) e a WWW (HTTP)
- Linguagens de programação Web

Actividades Práticas

Instalação do ambiente de desenvolvimento Web (LAMP/WAMP). Analisar as transações do protocolo HTTP utilizando navegadores modernos (Chrome, Firefox, IE). Analisar o código fonte das páginas Web

Actividades de Pesquisa

Domínios de aplicação das tecnologias Web

Atividade 1 - A internet e a WWW (perspetiva histórica)

Introdução

Esta atividade pretende que o (a) estudante domine os conceitos básicos da internet e da WWW.

Objetivos específicos de aprendizagem e ensino

Descrever a história da internet e as tecnologias emergentes

Apontar a origem da WWW

A internet é uma rede global, que consiste em milhares de redes independentes de computadores, de empresas privadas, entidades governamentais e instituições científicas e educativas. Surgiu por volta de 1970, designado por Arpanet, com o objetivo principal de partilhar informações entre grupo de instituições científicas e universidades desenvolvidas no projeto. Esta rede global, torna viável para os computadores enviar mensagens entre si. Uma das ações mais importantes realizadas na Web é pesquisar informações variadas para diferentes fins. Para auxiliar esta tarefa de procurar informações nas páginas Web o utilizador tem à sua disposição os chamados "motores de pesquisa", "motores de busca" ou search engines. Por exemplo, a Google (www.google.com), a Yahoo (www.yahoo.com), entre outros motores de pesquisa.

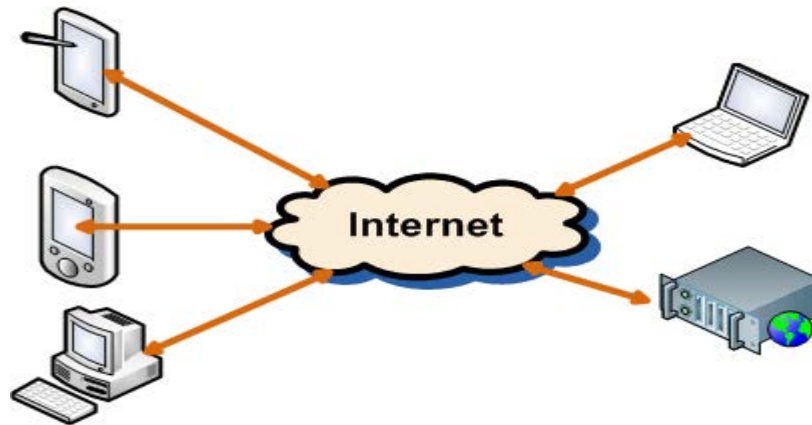


Figura 1.1: Rede global

A Web ou a WWW

A World Wide Web (WWW), foi criada por Tim Berners Lee em 1989, com o objetivo de, trocar as informações entre os pesquisadores. A WWW é o programa utilizado para obter informações solicitadas pelos utilizadores. Progressivamente, foi-se alimentando com a introdução de novas aplicações Web, não só para ambientes de pesquisas, como também para fins comerciais disponíveis na internet.

Pode-se dizer que atualmente, a Web é o universo de informações sem fronteiras com várias características. Exemplo, interface gráfica consistente, integração de um conjunto de tecnologias e documentos, etc.

Provedor de Serviço de Internet (ISP)

Um ISP é simplesmente uma empresa que fornece o acesso à Internet. Cada computador que está conectado à Internet faz parte de uma rede, inclusive um em sua casa. Quando a conexão é estabelecida com o seu ISP, isto significa que a sua máquina se torna parte de uma rede maior. A Internet é uma rede de redes.

Perguntas:

P 1: Qual é a relação existente entre a WWW e a internet? Exemplifique?

R:

P 2: O que é um navegador Web (browser)?

R:

P 3: Cite três exemplos de atividades que podem ser desenvolvida na internet?

R:

Passo 1 : Leia o material de leitura do curso intitulado

Passo 2 :

P 1: Na lista seguinte, indique quais são os componentes da Web?

Servidor Web

Servidor do Email

HTTP

P 2: Indique a linguagem utilizada para o lado do servidor Web?

PHP

HTML

Javascript

CSS

P 3: Quais são as linguagens no lado do cliente?

PHP

HTML

Javascript

CSS

P 4: Defina um servidor HTTP?

Servidor Web

Servidor de Email

Servidor FTP (transferência de arquivos)

P 5: Quais são os protocolos utilizados para a transferência de arquivos na Web?

WWW

HTTP

HTML

XML

FTP

P 6: Faça uma lista de tipos de páginas Web.

P 7: Explique a diferença entre páginas estáticas e dinâmicas.

R:

P 8: Quais são as linguagens indicadas na lista abaixo que podem ser usadas para criar páginas estáticas.

(X)HTML

XML

CSS

Javascript

PHP

Flash

P 9: Qual é o papel da linguagem HTML na arquitetura Web?

R:

P 10: Qual é o papel da linguagem PHP e ASPNET na arquitetura Web?

R:

P 11: A WWW funciona em modo Cliente-Servidor? Explique?

R:

Atividade 2 - Instalação e Configuração do Ambiente de Desenvolvimento

Introdução

Para desenvolver aplicações Web, o/a estudante precisa de um ambiente de desenvolvimento funcional. O ambiente de desenvolvimento é um conjunto de ferramentas que permite desenvolver aplicações Web.

Objetivos da atividade

No fim da atividade, o/a estudante será capaz de:

Identificar e usar correctamente as ferramentas de desenvolvimento para aplicações Web;

Instalar o servidor Web (recomenda-se o servidor WAMP);

Instalar os componentes:

Instalar configurar os servidor Apache (incluído no pacote servidor WAMP para Windows);

Instalar e configurar o servidor de base de dados - MySQL (incluído no pacote servidor WAMP para Windows).

Perguntas:

Parte 1:

Leia as páginas seguintes, e responda às questões:

<http://pt.wikipedia.org/wiki/WAMP>

<http://pt.wikipedia.org/wiki/LAMP>

P 1: Defina e explique o funcionamento da arquitetura WAMP/LAMP ?

R:

P 2: Quais são os componentes embutidos nos servidores WAMP/LAMP ?

R:

P 3: Qual é a linguagem de programação utilizada no lado do servidor WAMP/LAMP?

R:

P 4: Quais são os componentes embutidos no pacote de software?

PHP

Apache

MySQL

PHPMyAdmin

Parte 2:

Baixar e instalar o servidor WAMP

O servidor WAMP é um ambiente de desenvolvimento web para windows.

Para desenvolver e depurar as aplicações para Web baseados em PHP e MySQL é necessário instalar no seu computador um servidor local.



Figura 1.2. Página para download e instalação do servidor WAMP

Atividade 3 - Introdução às Tecnologias Web

Introdução

Nesta actividade irá aprender a usar as tecnologias Web. Vai aprender também a importância das tecnologias Web no quotidiano.

Objetivos da atividade:

Definir e aplicar os conceitos da arquitetura cliente/servidor;

Identificar as linguagens de programação de páginas Web, tipos de servidores Web e entender o protocolo HTTP;

Protocolo Web: HTTP

O protocolo HTTP permite o transporte de recursos na Web. Os clientes (browser), envia uma solicitação HTTP, para aceder os recursos do servidor, e o servidor Web responde às solicitações através da emissão de respostas HTTP. O HTTP é a sigla Protocolo de Transferência de Hipertexto (Hypertext Transfer Protocol), é um protocolo de comunicação utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos. Esta atividade apresenta uma visão geral das transações do protocolo HTTP utilizando um navegador Web (Google Chrome/ Firefox). Para desenvolver e depurar aplicações Web, é importante ter bom entendimento de como o protocolo HTTP funciona e os servidores Web.

Exemplo de transações HTTP:

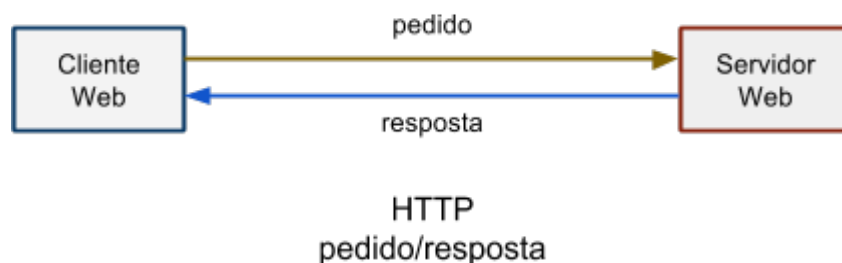


Figura 1.3. Transações HTTP

As arquiteturas de aplicações Web são comumente modeladas com base em arquitetura cliente-servidor. O HTTP é implementado em dois programas: um programa do lado do servidor e um programa do lado de cliente.

Como mostrada na figura anterior o cliente web solicita informações a um servidor, o servidor em geral responde com as informações solicitadas.

Parte 1: Ferramentas de desenvolvimento Web Firefox - instalar o plugin Firebug

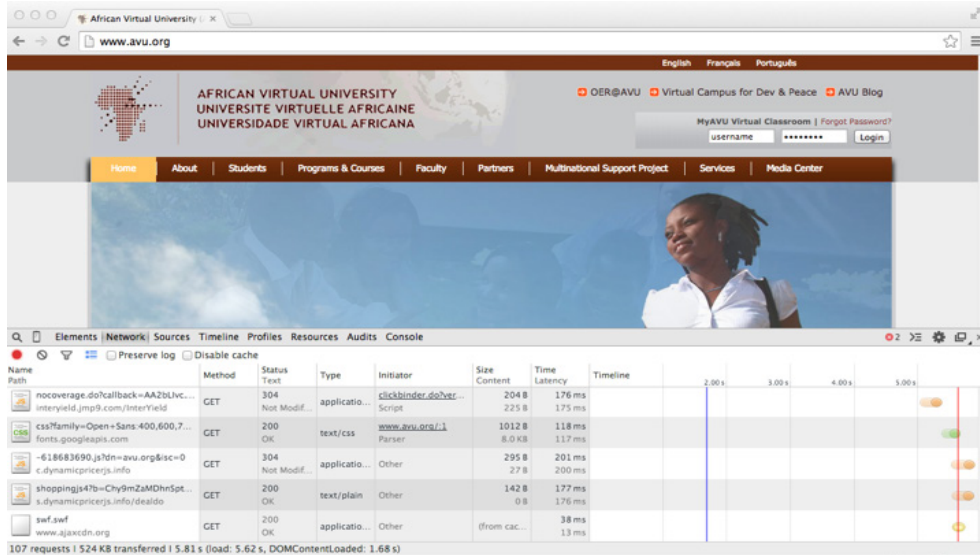


Figura 1.4: Página AVU

Vamos supor que a página consiste em uma base HTML. O HTTP define como um cliente Web requisita uma página Web e um servidor web transfere a página para o cliente.

Considere a figura:



Figura 1.5: Protocolo HTTP pedido/resposta

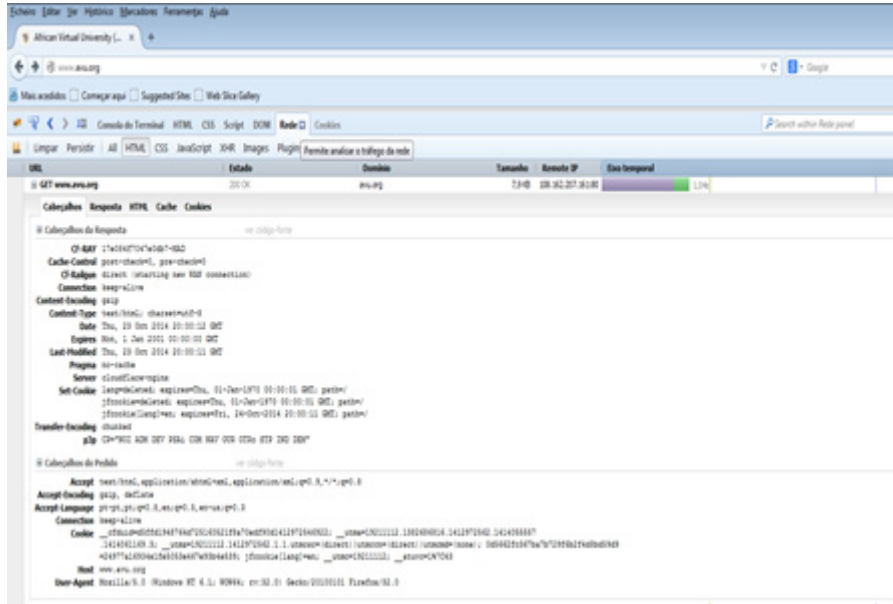


Figura 1.6: Cabeçalho de pedido e resposta (www.avu.org)

Podemos ver que, há um conjunto de informações trocadas através do protocolo HTTP, e que não são apresentadas através dum navegador Web. Entender a comunicação entre o cliente Web e o servidor Web, é a chave para entender o protocolo HTTP e trabalhar com os serviços Web.

Etapa 1: Faça um estudo de 4 páginas Web e mostre a relação entre o e-commerce e a WWW?

Etapa 2: Encontre 10 áreas de negócio que utilizem intensamente a WWW em suas atividades?

Campo de atividade	Descrição

Etapa 3: Mostre quatro (4) tipos de perfis profissionais relacionados com a Web?

O modelo de comunicação: Cliente/Servidor

O serviço Web é baseado na arquitetura Cliente/Servidor, ou seja, o cliente solicita um serviço e o servidor, por sua vez, responde à solicitação. A figura seguinte mostra a comunicação entre o cliente e o servidor.

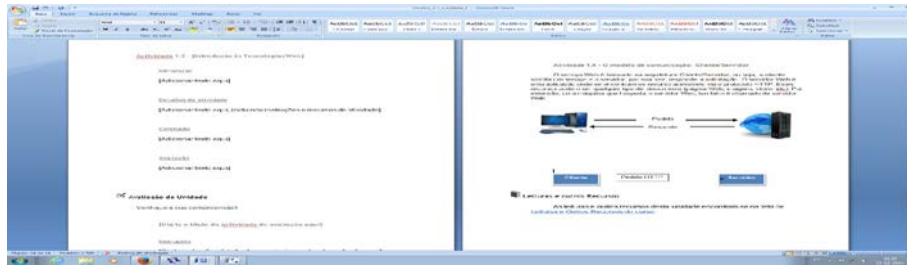


Figura 1.7: Modelo Cliente/Servidor

Este é o modelo que predomina nas redes informáticas. 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.

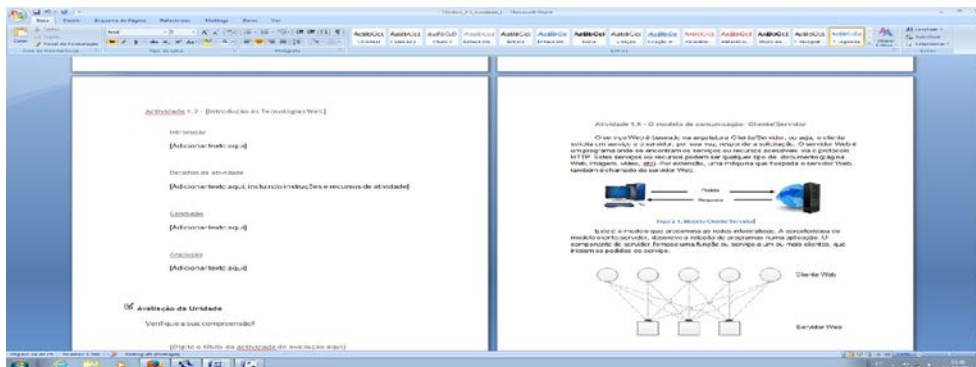


Figura 1.8: Pontos de comunicação entre o Cliente e o Servidor

Um servidor web é um programa que armazena recursos acessíveis via protocolo HTTP. Esses recursos podem ser arquivos de qualquer tipo (Página Web, imagem, vídeo, software ...) armazenados no sistema de ficheiros da máquina que hospeda o servidor ou a informação. Por extensão, uma máquina que hospeda o servidor Web também é chamado de servidor Web.

A implementação do protocolo HTTP é o elemento mais importante para o servidor HTTP que é o sinónimo de servidor. O servidor Web não interpreta os recursos que gere, fornece apenas o acesso a estes recursos serviço. Um outro aspecto importante sobre os servidores Web, é que estes servidores não têm nenhuma interação direta com os seres humanos, precisam usar um cliente capaz de recuperar recursos alojados num servidor Web.

Exemplos de servidores Web:

Apache2: É o servidor mais popular na Web. Cerca de 60% dos sites são executados sob Apache2. É de uso aberto, ou seja, é praticamente utilizável sem restrições proprietárias;

Microsoft IIS: servidor Web da Microsoft. É usado principalmente para aplicações baseadas em tecnologias da Microsoft, como ASP.NET;

NGINX: Este servidor está se tornando cada vez mais popular, por ser um servidor web rápido, leve, e com inúmeras possibilidades de configuração para melhor performance;

Google WebServer: Este é um servidor Web com base no Apache 2, que foi modificado por engenheiros do Google para suas necessidades. Essas mudanças não foram divulgadas. O cliente Web é um programa capaz de recuperar os recursos em servidores Web através de emissão de solicitações HTTP. Historicamente, o sucesso dos primeiros recursos Web são os documentos das páginas Web interligados entre si através de links (ligação). Um navegador Web é um cliente Web, que exibe páginas da web e permite ao utilizador navegar de uma página para outra, usando hiperlinks. O navegador Web tem duas funções importantes: recuperação de recursos e exibição de páginas Web. Existem também, clientes Web que não são navegadores Web, são utilizados principalmente entre programas em que, a única característica interessante é o acesso aos recursos da Web. No contexto do desenvolvimento web, não especificado, o cliente Web é o browser (navegador). A Figura 8 mostra de forma simplificada a arquitetura de um navegador web. É constituída por: um motor HTTP para interagir com o servidor web para obter o recursos. Este é o motor que constrói solicitações HTTP e processa as respostas;

Interpretadores HTML, CSS, JavaScript que permite exibir a página web.

Plugins: permite executar corretamente as páginas Web dinâmicas que são baseados em código binário. Protocolos: A utilização de protocolo HTTP para comunicação entre o cliente e o servidor e o FTP para transferência de ficheiros entre outros apresentados na figura.

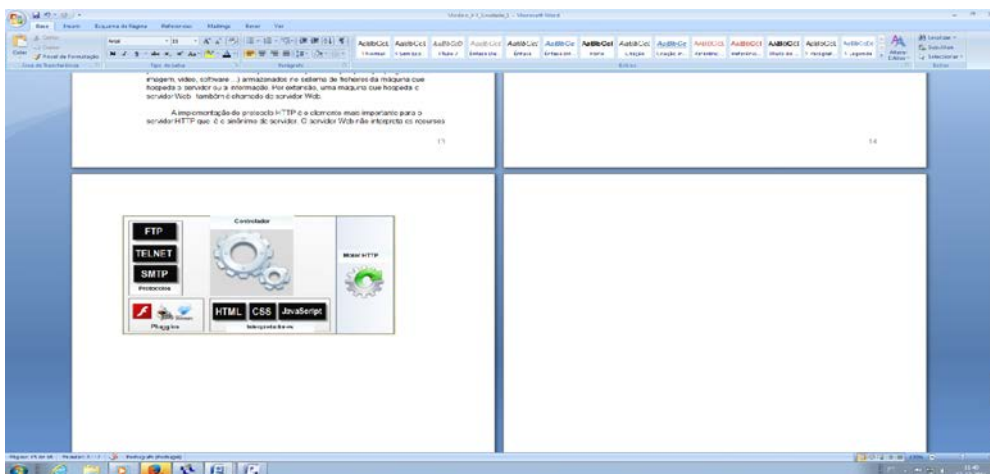


Figura 1.9: Arquitetura de um navegador Web

A Web permite o acesso a recursos que podem ser páginas web, imagens, vídeos, PDFs, Excel .

Localizador Universal de Recursos (URL)

Os recursos disponibilizados na Web têm um único endereço para o acesso a qualquer página Web. Os endereços utilizados são chamados de URLs (Localizador Universal de Recursos).

Exemplo:

<http://www.avu.org/Service/content-development.html>

Este endereço identifica:

- O protocolo de acesso ao recurso (HTTP);
- O servidor remoto a ser contactado `www.avu.org`;
- Caminho do diretório até ao recurso `/Service/`;
- o recurso ou documento Web obtido `/content-development.htm`.

URL (Uniform Resource Locator) são nomes padrão dos recursos de internet em geral (incluindo os recursos da web em particular). Eles indicam a localização destes recursos (o servidor que os contém) e como interagir com eles para o obter (protocolo padrão). Estes são os endereços que são utilizados para localizar exclusivamente cada recurso disponível na Web.

Veja o exemplo a seguir, solicitação de um recurso através de um navegador para o servidor remoto da AVU:

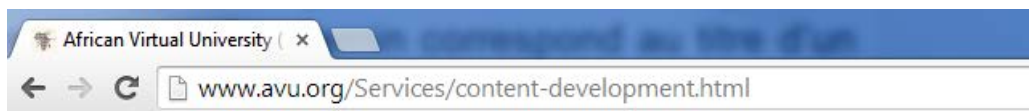
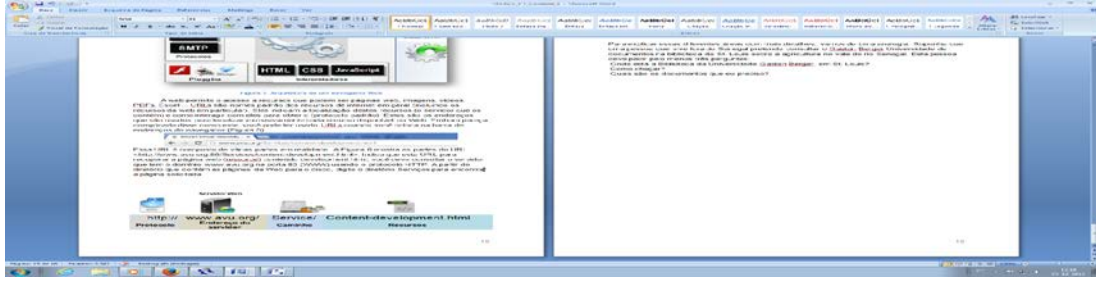


Figura 1.10: Barra de endereço

A URL na barra de endereço, é composto por várias partes. A Figura 1.10 mostra as partes do URL `<http://www.avu.org:80/Services/content-development.html>`. Indica que esta URL que, para recuperar a página web (recurso) `content-development.html`, deve consultar o servidor que tem o domínio `www.avu.org`, porta 80 (WWW) usando o protocolo HTTP. A partir do diretório que contém as páginas da Web, digite o caminho `"/Service/"` para encontrar a página solicitada.



Mais geralmente, a sintaxe simplificada de URL é:

```
<protocolo>://<servidor>  
:<porta>/<caminho>;<parâmetros>?<pedido>#<etiqueta>
```

Para melhor compreensão, suponhamos que um indivíduo que vive num determinado país queira solicitar um documento para a sua pesquisa, numa biblioteca situada num outro país. Este indivíduo terá que, no mínimo efetuar as seguintes questões:

O endereço da biblioteca? Qual é o caminho para os recursos disponibilizados? Quais é o documento que precisa para a sua pesquisa.

Websites

Websites pode ser definida como um conjunto de recursos na Web cujo URLs têm o mesmo valor no <protocolo>:// <server>: <porta> /. Em outras palavras, um website é um diretório de recursos eletrônicos em um servidor Web.

Categorias de página Web

Existem duas categorias principais de páginas web: páginas estáticas e páginas dinâmicas (ver Figura 10).

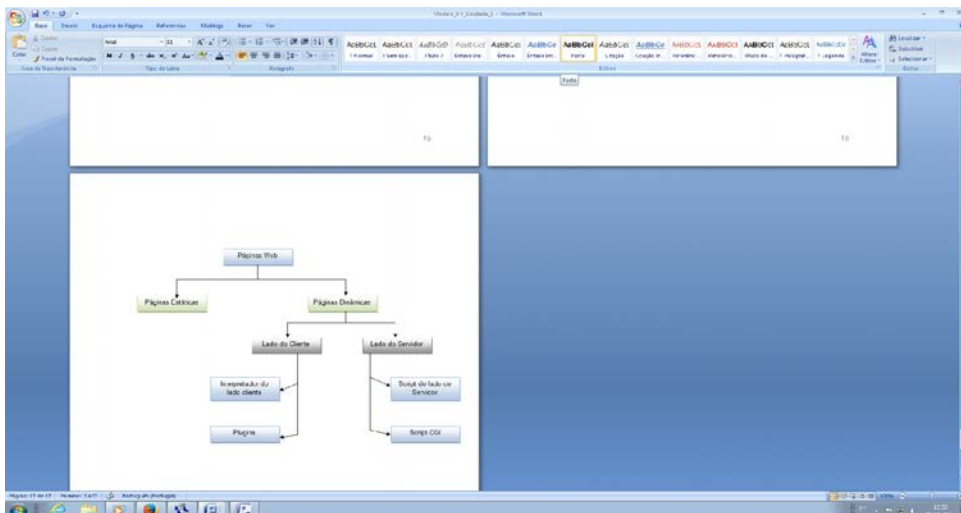


Figura 1.11: Taxonomia de páginas Web

Páginas estáticas têm conteúdo e formatação fixa (código HTML) no momento da sua criação. A aparência e o conteúdo dessas páginas é o mesmo para todos os clientes, a qualquer tempo. Para alterar a página, o desenvolvedor Web deve editar o código-fonte da página que terá a mesma aparência e conteúdo para todos os clientes. Eles têm a vantagem, por ser simples e segura (sem troca de dados com o utilizador), e facilmente visível para os motores de busca.

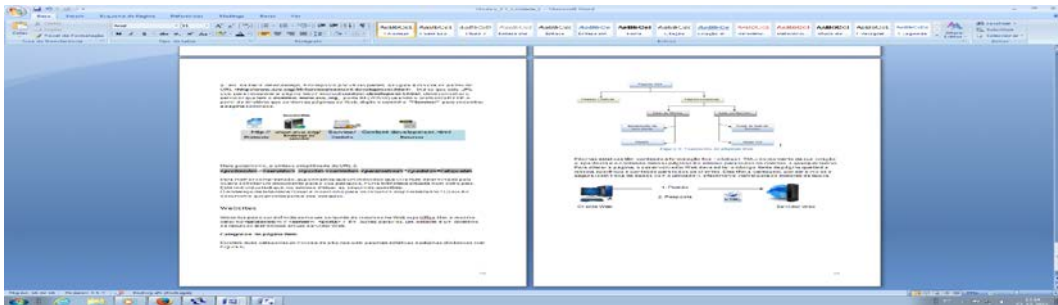


Figura 1.12.

As linguagens utilizadas para páginas estáticas servem para estruturar os conteúdos na página Web. Por exemplo, colocar um texto a negrito, sublinhado, a cor de fundo, posição do layout, etc. Exemplos de linguagens: HTML, CSS, X(HTML), XML. Estas linguagens não são de programação, mas sim, são linguagens de marcação.

Uma página dinâmica em geral é a referência do uso de linguagens de programação server-side, tal como PHP, ASP, JSP, ColdFusion, ASPNET, entre outras, no desenvolvimento de um site ou aplicações para intranet e extranet.

Existem dois tipos de categorias de página dinâmicas:

- Páginas Web dinâmicas do lado do cliente;
- Páginas Web dinâmicas do lado do servidor.

As linguagens Web

A termo aplicação Web é utilizado principalmente para descrever a utilização das linguagens para a criação de conteúdos web dinâmico.

O ecossistema de desenvolvimento web é composta de linguagens de marcação e linguagens de programação.

As linguagens de marcação (Markup) são utilizados para escrever páginas Web, para colocar os hiperlinks e definir as suas aparências(layout) nos navegadores. As principais linguagens de marcação são:

HTML (HyperText Markup Language): Atualmente, é muito difícil para um desenvolvedor web suceder sem ter o domínio da linguagem Web;

CSS (Cascading Style Sheet): utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento;

XML (Extensible Markup Language): esta linguagem é amplamente utilizado para troca de dados na internet.

As linguagens do lado do servidor web permitem colocar o código interpretável em HTML e assim gerar o conteúdo da página web com o processamento no servidor. Estas línguas são do tipo server side script (script do lado do servidor) ou linguagens CGI. As linguagens de script do lado do servidor mais populares são:

PHP: Esta linguagem cuja sintaxe é baseada na linguagem em C, é, certamente, a linguagem mais popular da Web. A evidência empírica deste sucesso é o grande número de CMS escrito em PHP. Exemplo: Joomla, Drupal, Wordpress, Magento, ... Sua popularidade é devido à sua facilidade de aprendizagem para iniciantes.

JSP/Servlet: É uma linguagem muito popular no desenvolvimento de sistemas relativamente complexos;

ASP.NET: É uma linguagem da Microsoft. É uma linguagem que oferece uma série de componentes de interface (Widget). Facilita o desenvolvimento quando o ambiente de implementação é a Microsoft;

Python, Ruby e outros: Python e Ruby são linguagens que se encontram em crescimento através do uso dos seus respetivos frameworks: Django e Ruby on Rails.

Atividades de Aprendizagem: Formatação de páginas Web com CSS

Esta secção tem como objetivo, aprender a desenvolver a primeira folha de estilo de um documento. CSS3 é uma linguagem para estilos em uma página Web. Isso significa que, estabelece regras de formatação a ser aplicadas a partes específicas do documento (texto, fontes, cores, imagens de fundo, posicionamento, etc). Podemos aplicar diferentes tipos de estilos para tipos de elementos específicos em uma página HTML.

Começaremos por abordar os seletores CSS, em seguida, gestão de cores, textos, tabelas, links e botões, etc.

Atividade 1 - Sintaxe e seletores CSS3

Introdução

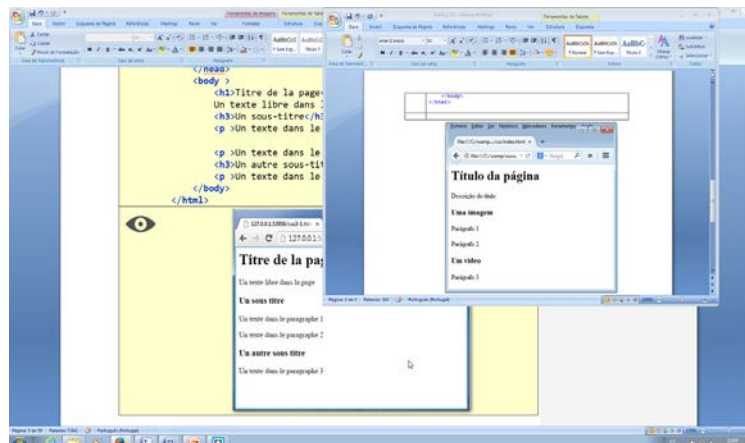
O uso de CSS no desenvolvimento de páginas Web é um avanço significativo no mundo do Web Design. O CSS permite controlar melhor o layout com a criação de diferentes folhas de estilo, é aplicado para tipos diferentes de mídias (impressora, tela, etc). A lista a seguir mostra algumas vantagens em ter a apresentação separada dos conteúdos: **Simplifica a atualização de conteúdos da página Web;**

O design visual do site pode ser feito sem afetar o conteúdo. Se quiser mudar o design do site, só tem que mudar o CSS correspondente.

Primeira abordagem com CSS3

Nesta parte, vamos ver o que inicialmente parece com código CSS3. Para isso, considere o seguinte código:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body >
    <h1>Título da página</h1>
    Descrição do título
    <h3>Uma imagem</h3>
    <p >Parágrafo 1</p>
    <p >Parágrafo 2</p>
    <h3>Um video</h3>
    <p >Parágrafo 3</p>
  </body>
</html>
```



O primeiro passo é saber como integrar CSS em uma aplicação Web, a fim de especificar a formatação. Há três formas de aplicar CSS a um documento HTML:

CSS externo - Colocar o código em um ficheiro CSS (com a extensão .css) e indicar no documento HTML que o caminho para este ficheiro;

CSS externo - Incorporar o código CSS diretamente no documento HTML , utilizando a tag <style> dentro da tag <head>;

CSS inline - incorporar o estilo CSS diretamente no corpo da tag HTML, utilizando um atributo style e o valor do atributo como propriedades CSS.

A primeira forma (CSS externo), é criar um ficheiro com o código CSS separado, depois declará-la dentro do documento HTML. A tag utilizada é o <link>

Ver o exemplo:

```
<head>
    ...
    <link rel="stylesheet" type="text/css" media="all"
href="nomeficheiro.css" />
    ...
</head>
```

O caminho do ficheiro CSS é indicado pelo atributo HREF que assume tipo valor de URL. É possível anexar vários ficheiros CSS no mesmo documento utilizando vários elementos link.

```
<head>
    ...
    <link rel="stylesheet" type="text/css" media="all"
href="nomeficheiro.css" />
    <link rel="stylesheet" type="text/css" media="all"
href="nomeficheiro3.css" />
    ...
</head>
```

Para ilustrar melhor este método CSS externo, veja o exemplo:

```
<! DOCTYPE html>
<html>
```

```

<head>
  <meta charset="utf-8">
  <link type="text/css" rel="stylesheet" media="all" href="css/
estilo.css"/>
</head>
<body >
  <h1>Titulo da página</h1>
  Descrição do título
  <h3>Uma imagem</h3>
  <p >Parágrafo 1</p>
  <p >Parágrafo 2</p>
  <h3>Um video</h3>
  <p >Parágrafo 3</p>
</body>
</html>

```

```

body {
  background-color: brown; /* cor de fundo */
  color: white; /* cor branca do texto*/
}

h3 {
  color: blue; /* cor do texto entre h3 */
}

```

Neste exemplo, o código HTML tem um uma ligação para um ficheiro CSS externo, com a tag LINK. O ficheiro CSS contém instruções com efeitos visuais, como alterar a cor de fundo e texto em toda a página, em seguida, cor <h3>. O resultado é dado pela seguinte figura:

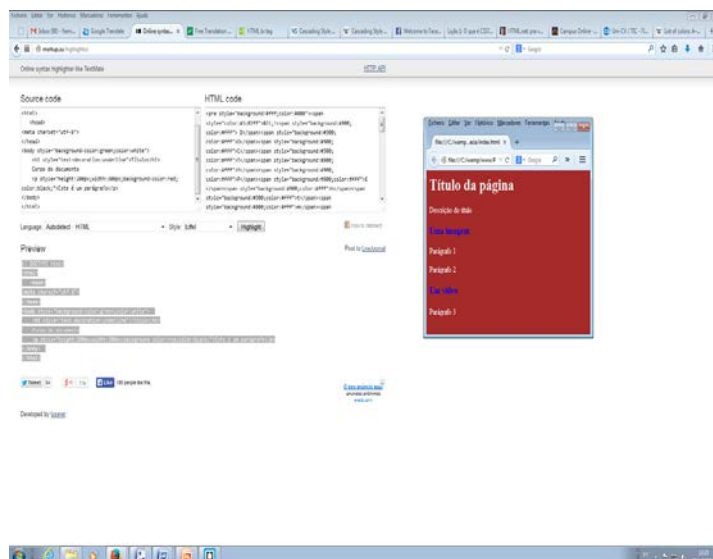



Figura 2.1: Resultado da aplicação de estilo CSS no documento HTML

A segunda forma (CSS interno) de adicionar o código CSS é colocá-lo entre o `<style> ... </style>`. Neste método, o código CSS encontra inserido no mesmo documento HTML entre o `<head>`;

```
 <style type="text/css">
/* Codigo CSS */
</style>
```

Este método é ilustrado pelo seguinte código, com um resultado equivalente ao do exemplo anterior (brown, white, blue).

```
 <! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Título da pagina</title>
    <style>
      body {
        background-color: brown;
        color: white;
      }
      h3 {
        color: blue;
      }
    </style>
  </head>
  <body > ... </body>
</html>
```

O terceiro método é o estilo atributo que é comum a todos os elementos HTML. Esse atributo é utilizado para formatar elementos HTML, especificando linha (isto é na tag de abertura) propriedades como cor, altura, largura, tipo de borda, em outras palavras colocar um estilo sobre o elemento ... O código a seguir é o exemplo de como aplicar o estilo:



```
<! DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

  </head>

  <body
style="background-color:green;color:white">

    <h1 style="text-
decoration:underline">Título</
h1>

    Corpo do documento

    <p style="height:200px;width:300px;ba
ckground-color:red;color:black;">Este é um
parágrafo</p>

  </body>

</html>
```

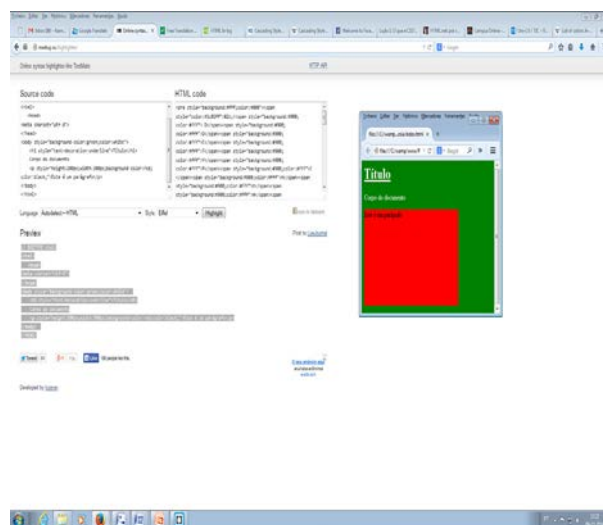


Figura 2.2

Como no exemplo anterior, este código altera a cor da página (background-color:green) e a cor do texto (color:white). Estas propriedades são separadas por um ponto-e- vírgula no valor do atributo style. Em seguida, o título <h1>, entretanto, tem um estilo que especifica que o texto é sublinhado. A cor do texto é herdada do estilo aplicado <body> (<h1 style="text-decoration:underline">).

Finalmente, o estilo de parágrafo indica que o tamanho do parágrafo deve ter a altura de 200pixels e largura 300pixels (height:200px;width:300px;) e a cor de fundo deve ser vermelha e a cor do texto preta (background-color:red;color:black;).

Realço que este método não é recomendado, porque pode sobrecarregar a página HTML.

Perguntas de autoavaliação

P 1: Qual é o função do CSS no design de uma página Web?

P 2: Liste três formas de inserir o código CSS em uma página web.

P 3: Quais são os elementos HTML que podem conter o código CSS?

Inserir código CSS dentro do código HTML

Exemplo de código HTML e CSS3

```
<! DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Universidade Virtual Africana</title>
</head>
<body>
  <table>
    <caption> Exemplos de países </caption>
    <thead>
      <tr>
        <th>Continente</th><th>Países</th>
        <th>Capital</th><th>População</th>
      </tr>
    </thead>
    <tfoot>
      <tr>
```

```
<th> </th><th> </th><th>Total</th><th>318 269 000</th>
</tr>
</tfoot>
<tbody>
<tr>
<td>África</td><td>África do sul</td>
<td>Pretoria</td><td>51 770 560</td>
</tr>
<tr>
<td>Europa</td><td>França</td>
<td>Paris</td><td>66 600 000</td>
</tr>
<tr>
<td>América</td><td>Estado Unidos</td>
<td>Washington</td><td>318 269 000</td>
</tr>
<tr>
<td>Asia</td><td>India</td>
<td>News Dehli</td><td>318 269 000</td>
</tr>
</tbody>
</body>
</html>
table{
width: 100%;
text-align: left;
border-collapse: collapse;
border: 1px gray solid ;*/
}
th {
color: #ffffff;
background-color: #555555;
}
tr:nth-child(odd) {
background-color: #dfdfdf;
}
}
```

Inserir o código CSS no código HTML

Crie um ficheiro CSS externo e adicionando as instruções, faça a ligação com o documento HTML.

Utilize a tag <style> para formatar o documento.

Explorando os elementos de estilo em uma página Web

Neste secção, aprenderá como utilizar as ferramenta de desenvolvimento Web do Google Chrome para ver o CSS em uma página Web.

Execute o Google Chrome e abra a página da Wikipédia sobre África clicando no link: <http://pt.wikipedia.org/wiki/África>.

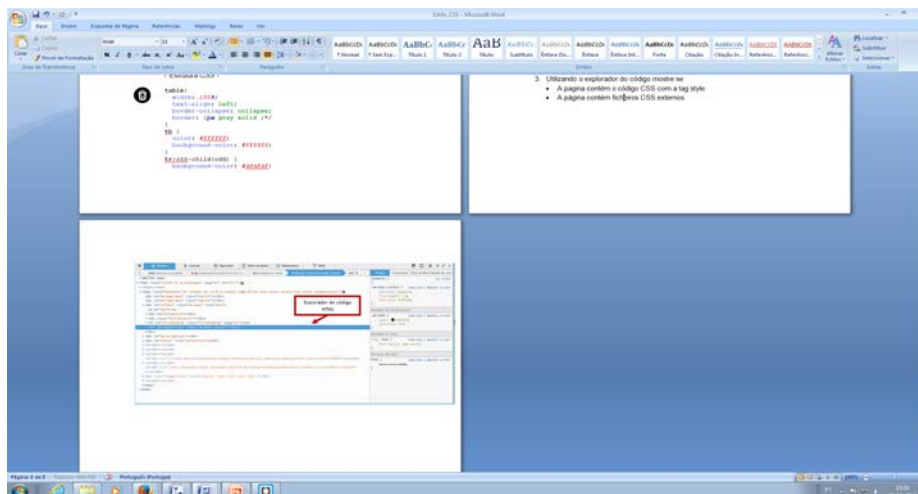
Clique botão direito -> inspecionar elemento em uma área em branco da página para visualizar o código fonte.



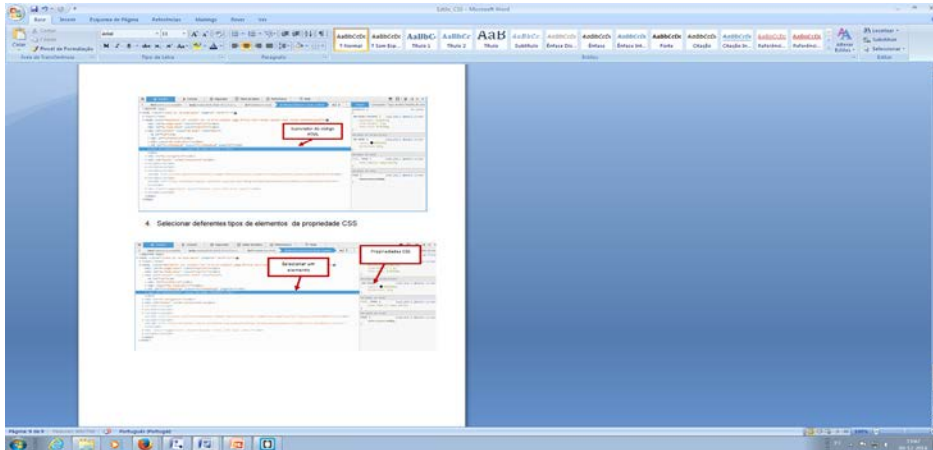
Utilizando o explorador do código mostre se:

A página contém o código CSS com a tag style

A página contém ficheiros CSS externos



Selecionar diferentes tipos de elementos da propriedade CSS



Seletores CSS

A folha de estilo CSS é simplesmente um conjunto de regras que especificam as propriedades de formatação (margens internas e externas, layout, altura, largura, ...). A regra CSS é composta por um seletor que identifica os elementos HTML seguido por um bloco de instruções delimitadas por chaves ({...}). A forma geral de regras CSS é dado pela seguinte código:



```
seletor {  
    /*bloco de instruções*/  
    propriedade1:valor;  
    propriedade2:valor;  
    ...  
}
```

A semântica de uma regra CSS é que as propriedades especificadas na regra se aplicam a todos os elementos HTML que correspondem ao seletor padrão.

Seletor Universal

O seletor representado por um asterisco (*) seleciona todas as tags do documento HTML. Se quisermos mudar a fonte de todo o documento, podemos utilizar o seguinte código:


```
* {  
    font-family: serif;  
}
```

Tipo de seletor

Seletores funcionam como ponte de ligação de um documento HTML com a folha de estilo CSS. Para aplicar uma regra ou um estilo todos os cabeçalhos H2 em um documento HTML, podemos utilizar a tag h2 como seletor. Isto significa que todos os cabeçalhos de nível 2 serão alteradas segundo as regras aplicadas.

```
h2 {  
  
    font-family: serif;  
}
```

Seletor de classe

O seletor de classe normalmente se aplica a um grupo de elementos . É especificado em CSS com um ponto (.), e pode afeta um ou mais elementos HTML com o uso da mesma classe. Sintaxe CSS é como se segue:

```
.nome_de_classe {  
    /* declaração(s) */  
}  
  
elemento.nome_de_classe {  
    /* declaração */  
}
```

Em um documento HTML, referimo-nos a este tipo de classe (por exemplo, o elemento P):

```
<p class="nome_de_classe">...</p>
```

Qualquer elemento em uma página HTML pode utilizar esta classe:

```
<p class="nome_de_classe">...</p>  
  
<ul class="nome_de_classe">  
    <li>...</li>
```

```
<li>...</li>
</ul>
```

Além de aplicar regras comuns a um conjunto de elementos com o mesmo nome da classe, também pode-se seleccionar um elemento específico com e aplicar estilos deferentes.

```
.nome_de_classe {
    color: gray;
}

p.nome_de_classe {
    font-style: italic;
}
```

Os elementos da classe são exibidos com a cor cinza (color: gray;) e os parágrafos a itálico (font-style: italic;).

Seletor ID

O atributo ID pode ser aplicado apenas a um único elemento do código HTML. A principal diferença é que somente pode ser aplicado a um único elemento. Isto significa que não pode existir mais de que 1 elemento com a mesma identificação. Exemplos de elementos estruturais de uma documento HTML: cabeçalho, logotipo, rodapé, secções, etc.

O seletor ID é especificado em CSS com um cardinal (#), a sintaxe é a seguinte:

```
#nome_de_id {
    /* declaração*/
}
```

Ou

```
elemento#nome_de_id {
    /* declaração*/
}
```

Em um documento HTML, se identificador nome_id refere-se a um tipo de elemento div, deve-se escrever apenas uma única vez na página:

<div id="nome_id">...</div>

Existem outros tipos de seletores, ver a tabela seguinte:

Padrão	Significado	Exemplo
*	Corresponde com qualquer elemento	* { color:blue; }
E	Corresponde com qualquer elemento E (ou seja, um elemento do tipo E).	p { color: red; background-color: green; }
E F	Corresponde com qualquer elemento F que é um descendente do elemento E.	div p { display : block margin-left : 20px ; }
E > F	Corresponde com qualquer elemento F como um filho do elemento E.	div > p { display : block margin-left : 20px ; }
E:first-child	Corresponde com um elemento E como o primeiro filho de seu elemento pai.	section:first-child { font-size : 80em ; }

<p>E:link E:visited</p>	<p>Corresponde com um elemento E que é uma ligação não visitado (: link) ou (: visited).</p>	<pre>a:link { font-color : blue; text-decoration: none; } a:visited { font-color : orange; }</pre>
<p>E:active E:hover E:focus</p>	<p>Corresponde com um elemento E, e ocorre durante certas ações do utilizador</p>	
<p>E + F</p>	<p>Corresponde com qualquer elemento F imediatamente precedido por um elemento E.</p>	<pre>div + p { display : block margin-left : 20px ; }</pre>
<p>E[foo]</p>	<p>Corresponde com qualquer elemento E com o atributo "foo"</p>	
<p>E[foo="warning"]</p>	<p>Corresponde com qualquer elemento E cujo atributo "foo" tem exatamente o valor de "warning".</p>	
<p>E[foo~="warning"]</p>	<p>Corresponde com qualquer elemento E cujo atributo "foo" definida como uma lista de valores separados por espaços em branco e uma delas é "warning".</p>	

E[lang="en"]	Corresponde com qualquer elemento E cujo atributo "lang" definida como uma lista de valores separados por traços, a lista que começa "en".	
DIV.warning	O mesmo significado com DIV[class~="warning"].	
E#meuid	Corresponde com qualquer elemento E cujo ID é "meuid"	

Perguntas de autoavaliação:

1. Explique sucintamente o papel de seletores.
2. Qual é a estrutura básica de um seletor em um elemento? Dê um exemplo
3. Qual é a estrutura básica de um seletor ID? Dê um exemplo.
4. Qual é a estrutura básica de um seletor de classe? Dê um exemplo.

Seletor ID

Este exercício tem como objetivo aprender, pelo menos, quatro categorias de seletores CSS básicos:

```
<div id="e1">
  <p id="e2" class="c1">
    <ul id="e3">
      <li id="e4"> <a id="e5" href="1.html">Este</a></li>
      <li id="e6"> <a id="e7"href="2.html">Oeste</a></li>
      <li id="e8"> <a id="e8"href="3.html">Norte</a></li>
      <li id="e9"> <a id="e9" href="4.html">Norte</a></li>
    </ul>
  </p>
  <p id="e10"class="c2">
```

```
A <a id="e11" href="http://pt.wikipedia.org/wiki/África"> África </a>
é o terceiro continente mais extenso (atrás da Ásia e da América) com
cerca de 30 milhões de quilômetros quadrados, cobrindo 20,3 % da área
total da terra firme do planeta.
```

```
</p>
```

```
</div>
```

Para cada um dos seletores apresentados na lista seguinte mostre os elementos selecionados.

```
#e9 {...}
```

```
.c1 {...}
```

```
p {...}
```

```
p li {...}
```

```
p > li {...}
```

```
p.c1 {...}
```

```
p > a
```

```
p > li {...}
```

```
ul > a {...}
```

```
ul a {...}
```

Atividade 4 - Cores em HTML/CSS3

Introdução

Esta atividade têm como objetivo entender como aplicar cores em HTML e CSS.

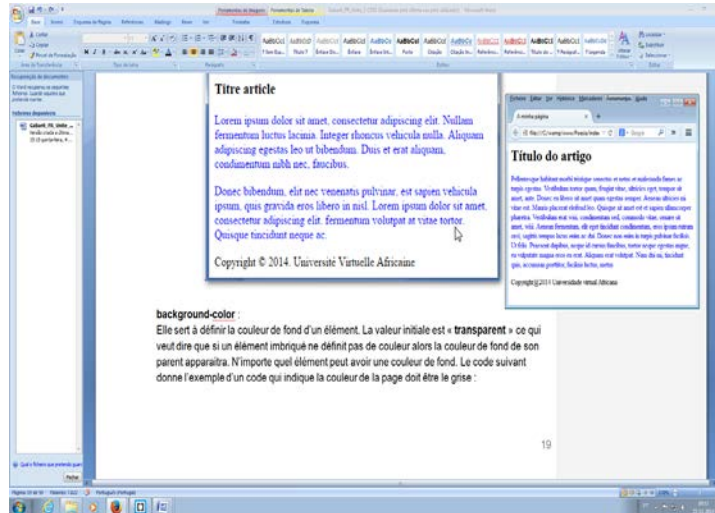
Principais propriedades de cores em CSS

CSS3 possui um conjunto de propriedades para definir cores em uma página HTML.

Abordaremos nesta atividade alguns exemplos:

Color: esta propriedade define a cor de um texto exibido em um elemento. Geralmente, por padrão as cores são definidas a preto e pode ser mudada com a utilização desta propriedade. O exemplo seguinte ilustra a utilização desta propriedade:


	<pre>p { color:blue; }</pre>
---	----------------------------------

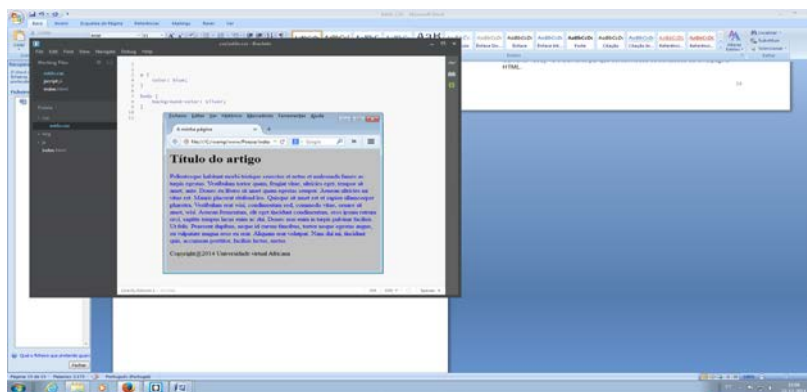


background-color :
Elle sert à définir la couleur de fond d'un élément. La valeur initiale est « transparent » ce qui veut dire que si un élément imbriqué ne définit pas de couleur alors la couleur de fond de son parent apparaîtra. N'importe quel élément peut avoir une couleur de fond. Le code suivant donne l'exemple d'un code qui indique la couleur de la page doit être la grise :

Background-color: é utilizado para definir a cor de fundo de um elemento. O elemento <body> é o elemento pai que contém todos os conteúdos de uma página HTML.

O código seguinte mostra um exemplo de um código que indica a cor da página deve ser cinzento:

	<pre>body { background-color: silver; }</pre>
---	---



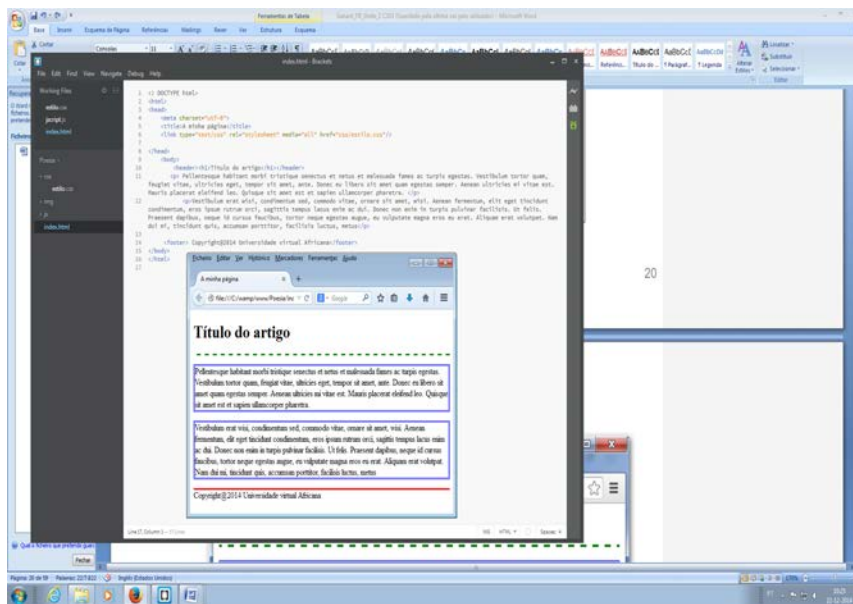
border-color, border-bottom-color, border-top-color, border-left-color, border-right-color: Estas diferentes propriedades são utilizadas para definir a cor da borda como um todo ou individualmente e definir bordas superior, inferior, esquerda e direita. Estas propriedades são geralmente utilizadas em conjunto com border-style, border-bottom-style, border-left-style, border-right-style, border-top-style. Essas propriedades definem o tipo de borda para os elementos. O código a seguir mostra um exemplo da utilização destas propriedades:

```

p {
    border-style: double;
    border-color: blue ;
}

header {
    border-bottom-style: dashed;
    border-bottom-color: green;
}

footer {
    border-top-style: solid;
    border-top-color: red;
}
    
```



Navegadores admitem três métodos para especificar a cor de um objeto na página. O nome da cor, utilizando o hexadecimal e o formato RGB.


Métodos	Exemplo	Descrição
Nome da Cor	background-color: silver;	A cor é designada pelo nome.

Cores em hexadecimal	background-color: #808080;	A cor é identificado pelo seu código hexadecimal (6 dígitos). Os dois primeiros dígitos representam o valor de vermelho, os dois seguintes o verde e os dois últimos azul.
Formato RGB	background-color: RGB(128,128,128);	A cor é definido por três números decimais de 0 a 255, indicam a taxa de níveis de vermelho, verde e azul.







Nome da cor: A forma mais fácil e mais conveniente para selecionar uma cor é digitar o nome em inglês. Dezasseis cores chamados "standard" são reconhecidas da mesma forma por todos os navegadores. A tabela a seguir mostra as dezasseis cores utilizadas:


Aqua	Black	Blue	Fuchsia	Gray	Green	Lime	Maroon
Navy	Olive	Purple	Red	Silver	Teal	White	Yellow

Notação RGB: As cores são exibidos através da combinação de três cores: vermelho (red), verde (green) e azul (blue). Em CSS as cores são exibidos com a seguinte forma:


	rgb (vermelho, verde, azul) ;
---	-------------------------------

Esta notação é chamado RGB (Red, Green, Blue). Nesta forma os valores de vermelho, verde e azul são expressos em valores entre 0 e 255 ou entre 0% e 100%. A listagem seguinte é exemplos de cores com valores RGB:


Valor RGB	Cores
rgb (255, 0, 0)	
rgb (0, 255, 0)	
rgb (0, 0, 255)	
rgb (175, 238, 238)	
rgb (255, 222, 173)	
rgb (255, 105, 180)	

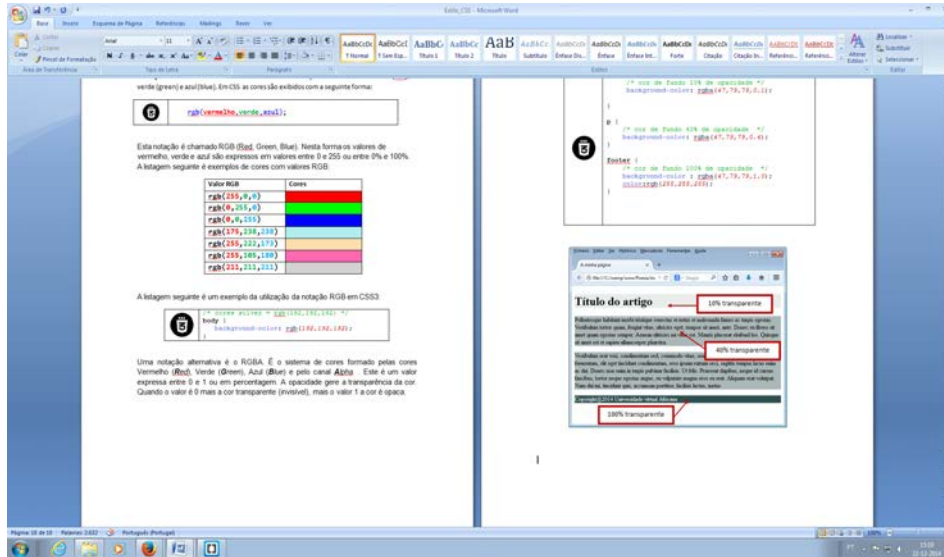
rgb(211,211,211)	
------------------	--

A listagem seguinte é um exemplo da utilização da notação RGB em CSS3:

	<pre>/* cores silver = rgb(192,192,192) */ body { background-color: rgb(192,192,192); }</pre>
---	---

Uma notação alternativa é o RGBA. É o sistema de cores formado pelas cores Vermelho (Red), Verde (Green), Azul (Blue) e pelo canal Alpha. . Este é um valor que se expressa entre 0 e 1 ou em percentagem. A opacidade gere a transparência da cor. Quando o valor é 0 mais a cor transparente (invisível), mas quando o valor é 1 a cor é opaca.

	<pre>header { /* cor de fundo 10% de opacidade */ background-color: rgba(47,79,79,0.1); } p { /* cor de fundo 40% de opacidade */ background-color: rgba(47,79,79,0.4); } footer { /* cor de fundo 100% de opacidade */ background-color : rgba(47,79,79,1.0); color:rgb(255,255,255); }</pre>
---	--



Notação Hexadecimal: Notação hexadecimal é uma forma compacta de representar os componentes RGB que expressam cada byte com dois dígitos hexadecimais. O formato geral é:



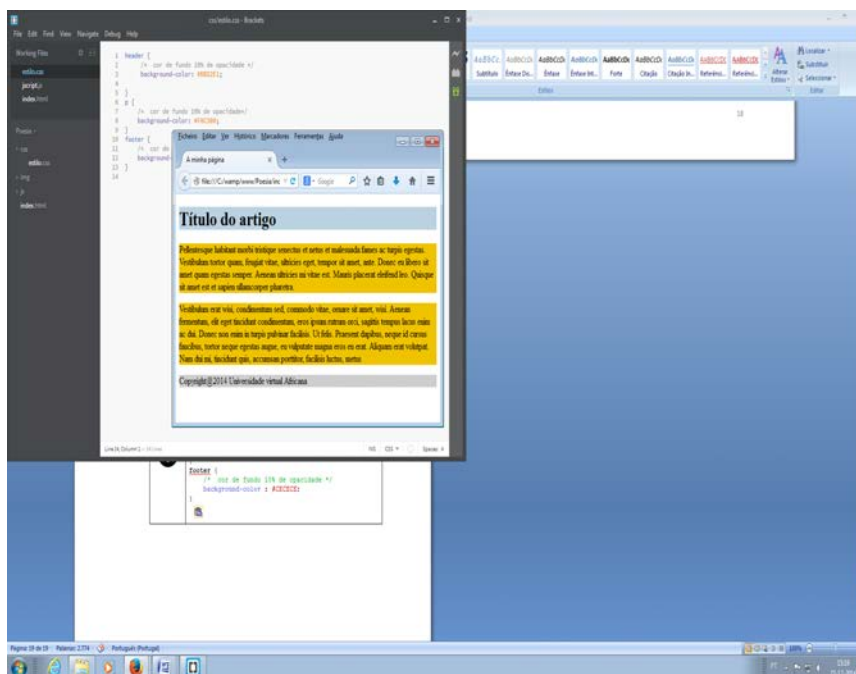
Neste formato, o valor da cor começa com um # seguido de três dígitos hexadecimais pares representando respetivamente vermelho, verde e azul. A tabela a seguir mostra as 16 cores básicas de HTML expressos com a notação hexadecimal.

#00FFFF Aqua	#000000 Black	#0000FF Blue	#FF00FF Fuchsia	#008000 Green	#00FF00 Lime	#800000 Maroon
#000080 Navy	#808000 Olive	#800080 Purple	#FF0000 Red	#008080 Teal	FFFFFF White	FFFF00 Yellow

O código a seguir ilustra o uso da notação hexadecimal em código CSS:



```
header {  
  
    /* cor de fundo 10% de opacidade */  
    background-color: #BBD2E1;  
  
}  
  
p {  
  
    /* cor de fundo 10% de opacidade*/  
    background-color: #F0C300;  
  
}  
  
footer {  
  
    /* cor de fundo 10% de opacidade */  
    background-color : #CECECE;  
  
}
```



Gradientes em CSS3

CSS3 fornece uma funcionalidade para criar gradientes de cor: linear e radial.

Neste exercício, veremos que são gradientes lineares, a transição gradual de uma cor para outra. Muitas vezes, a cor inicia ou termina com branco ou preto.

Para criar um gradiente em CSS3 devemos seguir a seguinte sintaxe:

```
element {  
...  
background : linear-gradient(direção,  
cor_inicial, cor_final);  
}
```



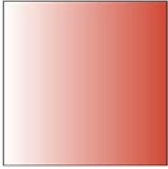



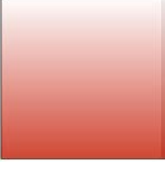

Nesta sintaxe:

direção (direction) indica o ponto de interrupção da cor final e direção de progressão do gradiente. Exemplo:

to top ; to right ; to left ; to bottom ;

to top left ; to top right ; to bottom left ; to bottom right.

A cor de início e fim são as duas cores que serão utilizadas para criar o gradiente.

Direção	Resultado	Direção	Resultado
to top		to top left	
to right		to top right	
to left		to bottom left	
to bottom		to bottom right	



```
div{  
  
    width: 200px;  
  
    height: 200px;  
  
    background: linear-gradient(to  
top, #fff, #d14836);  
  
}
```

Pergunta de autoavaliação:

- 1) Como mudar a cor da borda de um elemento?
- 2) Como mudar a cor da borda à esquerda?
- 3) Qual é a propriedade que muda a cor de um texto e cor de fundo?
- 4) Descreva as seguintes propriedades:

Propriedades CSS	Descrição
color	
border	
background	
background-color	
border-top-color	
border-bottom-color	
border-left-color	
border-right-color	
opacity	

- 5) Explique sucintamente os diferentes formatos de especificação de cor.
- 6) Preencha os espaços em branco na tabela a seguir:

Formato RGB	Formato Hexadecimal
#000080	
	RGB(0,191,255)
#191970	
	RGB(72,61,139)
#800080	
	RGB(139,69,19)

7) Explique as cores seguintes CSS:

Código CSS	Descrever o código CSS
<pre>html { color: #000; background: #FFF; }</pre>	
<pre>.rating { border-top-width: 1px; border-top-style: solid; border-top-color: rgb(227, 227, 227); }</pre>	
<pre>p{ background: transparent; }</pre>	
<pre>table { border-color: gray; }</pre>	

Mudar cores de um elemento da página

Considere o seguinte código HTML:

```
<body>

  <h1>A continente Africano</h1>

  <p>

    A África é o terceiro continente mais extenso (atrás da Ásia e da América) com cerca de 30 milhões de quilômetros quadrados, cobrindo 20,3 % da área total da terra firme do planeta. É o segundo continente mais populoso da Terra (atrás da Ásia) com cerca de um bilhão de pessoas (estimativa para 2005 1 ), representando cerca de um sétimo da população mundial, e 54 países independentes.

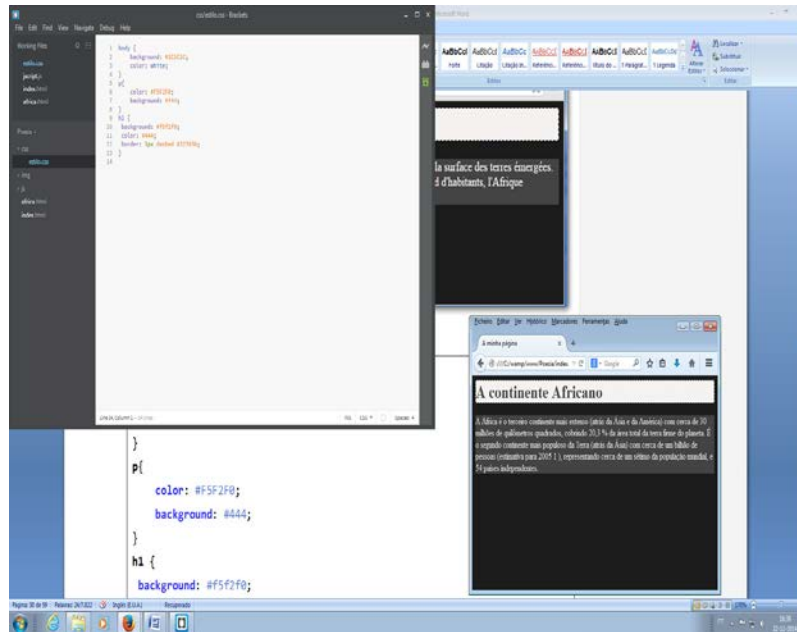
  </p>

</body>
```

Aplique o código CSS indicado na tabela abaixo para modificar as cores nos seguintes elementos e mostre os resultado.

Elemento	Propriedades - valores
body	Cor de fundo : #1C1C1C Cor de texto : blanc
p	Cor de fundo : #F5F2F0 Cor de texto : #040404
h1	Cor de fundo: #f5f2f0 Cor de texto: #040404 borda: 1px dashed #32393b;

Resultado:



Resposta:

```
body {  
    background: #1C1C1C;  
    color: white;  
}  
  
p{  
    color: #F5F2F0;  
    background: #444;  
}  
  
h1 {  
    background: #f5f2f0;  
    color: #444;  
    border: 1px dashed #32393b;  
}
```

Manipular gradientes em CSS3

1) Neste exercício, veremos alguns exemplos de uso de gradientes. Considere o seguinte código HTML:

```
<body>

  <h1>A continente Africano</h1>

  <button> Botão Google </button>

</body>
```

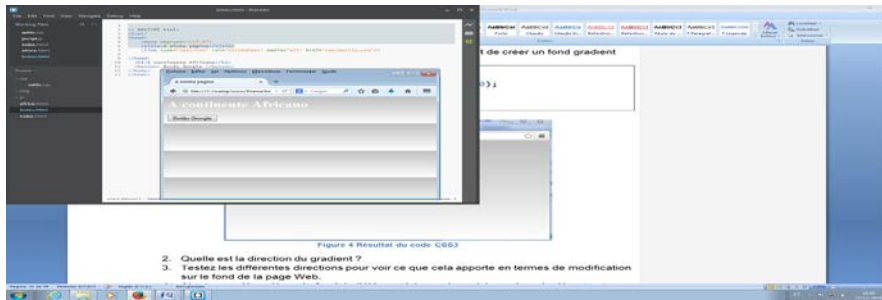
Aplicar para o seguinte código HTML as propriedades CSS para criar um fundo gradiente:

```
background: linear-gradient(to top, #FFFFFF, #C0C0C0);

color: white;

}
```

Resultado do código CSS3:



- 2) O que é a direção do gradiente em CSS?
- 3) Teste as diferentes direções para ver o resultado em termos de mudanças na parte inferior da página web.
- 4) Adicionar uma cor gradiente no fundo do elemento h1 para o qual a cor inicial é # F4A460.
- 5) Considere o seguinte código CSS:

```
button{

    width: 200px;

    height: 27px;

    background: linear-gradient(to
top,#dd4b39,#d14836);

    color: #FFF;

    text-shadow: 0 1px rgba(255,255,255,0.81);

    text-transform: uppercase;

    border-radius: 2px;

    font-size: 11px;

    font-weight: bold;

    text-align: center;

    margin-right: 16px;

    border: 1px solid #D3D3D3;

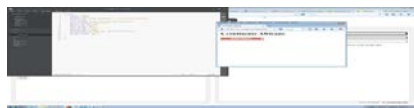
    box-shadow: 0px 0px 5px rgba(10,10,10,0.5);

}
```

Quais são as propriedades relacionadas com a cor?

Qual é a propriedade que permite aplicar gradiente no texto?

Insira o código CSS no código HTML acima e explique as mudanças que foram feitas para a aparência do botão:



Criando um modelo com CSS

O objetivo desta atividade é criar um cabeçalho, corpo e rodapé. Considere o seguinte código HTML: <!DOCTYPE html>

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">

<title>Poesia Africana</title>

<link type="text/css" rel="stylesheet" media="all" href="css/
estilo.css"/>

</head>

<body>

  <div class="box">

    <header>

      <h1 class="justify"> Poesia Africana</h1>

      <nav id="menu">

        <ul>

          <li><a href="index.html"> Sobre África </a>|</li>

          <li><a href="#">Poesia </a>|</li>

          <li><a href="#">Autores </a>|</li>

          <li><a href="contact.html">Contatos</a></li>

        </ul>

      </nav>

    </header>

    <aside>

      <h3>Informações úteis</h3>

      <p>Inserir o tipo de conteúdo</p>

    </aside>

    <section>

      <h2>Benvindo a nossa página</h2>

      <article>

        <h3>Título do artigo</h3>

        <p>

          Lorem ipsum dolor sit amet, consectetur adipiscing
```

elit. Nullam fermentum luctus lacinia. Integer rhoncus vehicula nulla. Aliquam adipiscing egestas leo ut bibendum. Duis et erat aliquam, condimentum nibh nec, faucibus.

```
</p>
```

```
<p>
```

Donec bibendum, elit nec venenatis pulvinar, est sapien vehicula ipsum, quis gravida eros libero in nisl. Lorem ipsum dolor sit amet, consectetur adipiscing elit. fermentum volutpat at vitae tortor. Quisque tincidunt neque ac.

```
</p>
```

```
</article>
```

```
</section>
```

```
<footer>
```

```
Copyright © 2014. Universidade Virtual Africana
```

```
</footer>
```

```
</div>
```

```
</body>
```

```
</html>
```

Crie um ficheiro HTML com o código acima indicado.

Crie um ficheiro CSS e faça a ligação com o ficheiro HTML utilizando o elemento `<link>`.

Adicione o código seguinte no ficheiro CSS depois e explique a função de cada propriedade utilizada.

```
body {  
  
    background: rgba(204, 204, 204, 0.12);  
  
    color: #666666;  
  
    background-size: 100%;  
  
    font-family: Helvetica,Arial,Sans-Serif;  
  
}
```

Adicione esta classe no ficheiro CSS:

```
.box {  
  
    box-shadow: 0px 0px 30px rgba(0, 0, 0, 0.12);  
  
    margin: auto;  
  
    width: 100%;  
  
    max-width: 980px !important;  
  
    margin-top: 20px;  
  
    padding-top: 20px;  
  
    padding-right: 0px;  
  
  
    margin-bottom: 0px;  
  
    background-color: #FFFFFF;  
  
    color : #E3601A;  
  
  
    /* concepção*/  
  
    border: 1px solid #cccccc;  
  
}
```

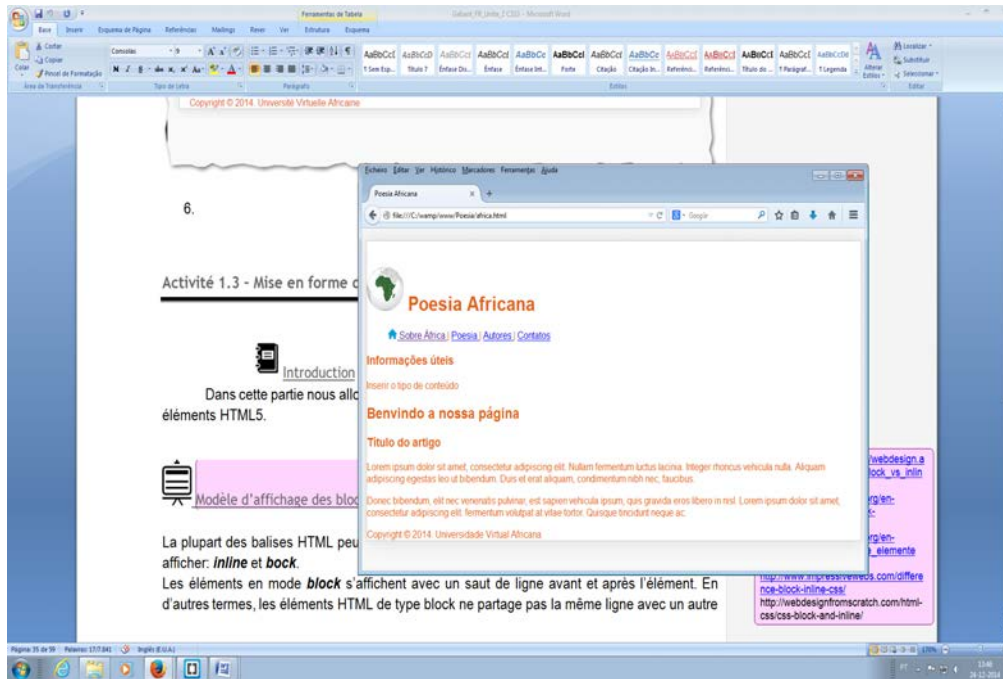
Crie uma tag div cuja o valor da classe é box que contém todo o conteúdo do corpo (body).

```
<! DOCTYPE html>  
  
<html>  
  
<head>  
  
    <meta charset="utf-8">  
  
    <title>Poesia Africana</title>  
  
</head>  
  
<body>  
  
    <div class="box">
```

...

```

</div>
</body>
</html>
    
```



Atividade 5- Formatação de elementos em HTML5 com CSS3

Introdução

Nesta atividade, aplicaremos as propriedades de CSS3 para formatar elementos HTML5.

Modelo de formatação CSS: blocos

A maioria das tags em HTML podem ser classificadas em duas categorias de acordo com a forma de display: inline e block. A tabela seguinte apresenta uma lista completa de elementos que estão no modo padrão em bloco.

<dd>	<address>	<figcaption>	<hr>
<div>	<article>	<figure>	<noscript>
<dl>	<aside>	<footer>	<canvas>
<fieldset>	<audio>	<form>	<header>
<hgroup>	<blockquote>	<1>, <2>, <h3>, <h4>, <h5>, <h6>	

O elemento HTML no modo de bloco mais utilizados é o elemento <div>. O código seguinte ilustra a exibição dos blocos.

```
<! DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <title>Africa</title>

</head>

<body>

    <h1> UVA: Ambição, Missão, Valores e Objetivos</h1>

<h2> Visão</h2>

<p>

    "To be the Leading Pan-African Open, Distance and eLearning Network"

    </p>

<h2>Missão</h2>

<p>

    "To facilitate the use of effective Open, Distance and eLearning in
    African institutions of tertiary education"

    </p>

<h2>Valores fundamentais</h2>

<div>

    The AVU will facilitate increased access to tertiary education in
    Africa professionally, passionately and efficiently. The AVU will
    achieve its mission in partnership with member countries, partner
    institutions and partners that support its mission. <p>The
    AVU recognizes and encourage staff to be honest, trustworthy and
    respectful.</p>

</div>

</body>

</html>

p {
```

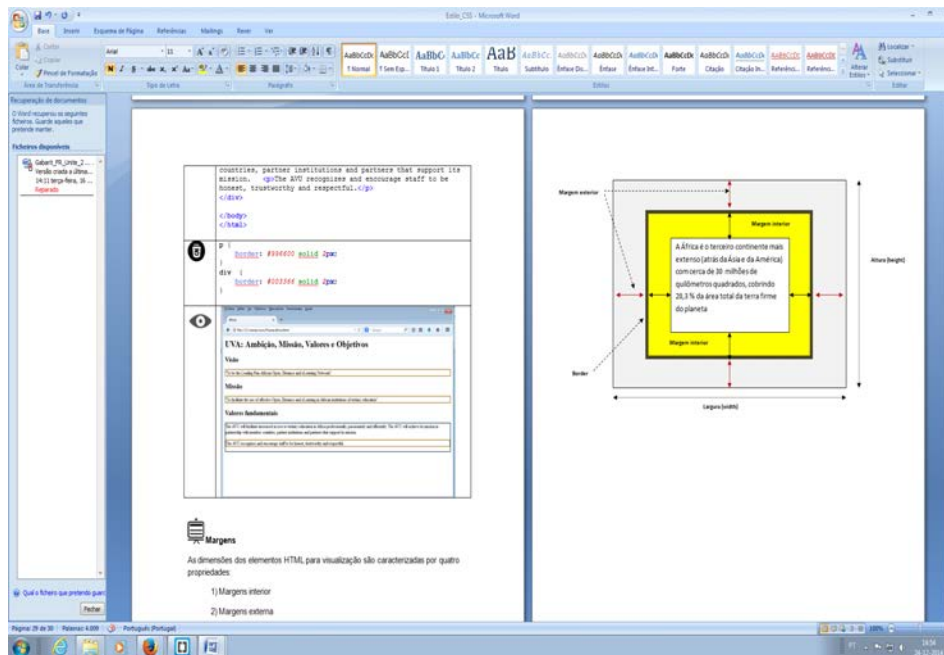


```
border: #996600 solid 2px;
}
div {
border: #003366 solid 2px;
}
```

Formatação de Margens

As dimensões dos elementos HTML para visualização são caracterizadas por quatro propriedades:

- 1) Margens interior
- 2) Margens externa
- 3) Altura
- 4) Largura



O código seguinte mostra um exemplo de definição das dimensões de um elemento.



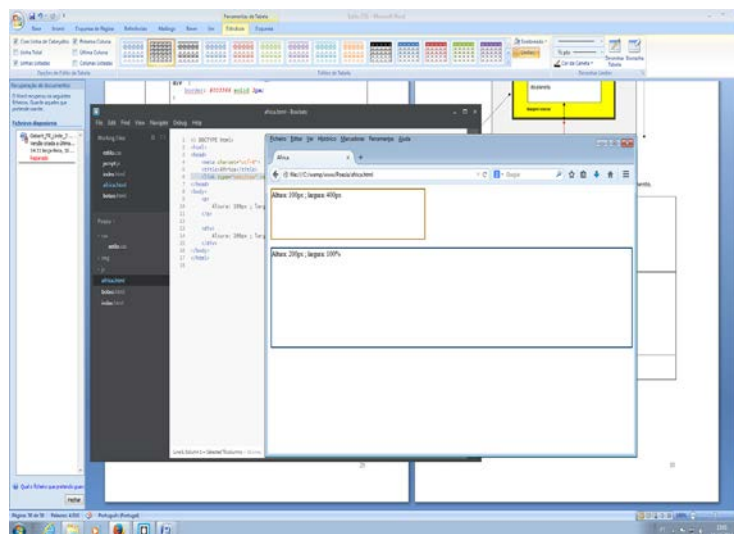
```
<body>
  <p>
    Altura: 100px ; largura: 400px
  </p>

  <div>
    Altura: 200px ; largura: 100%
  </div>
</body>
```



```
p {
  border: #996600 solid 2px;
  height: 100px;
  width: 400px;
}

div {
  border: #003366 solid 2px;
  height: 200px;
  width: 100%;
}
```



As margens externas são definidas pelas propriedades CSS começando com margin:

margin	Define a margem externa, parte superior, inferior e esquerda, direita
margin-bottom	define a margem externa para baixo
margin-left	define a margem esquerda exterior
margin-right	define a margem direita exterior
margin-top	define a margem externa no topo

As margens internas são definidas pelas propriedades CSS começando com padding:

padding	Define a margem superior interno, inferior, esquerda e direita
padding-bottom	define o preenchimento na parte inferior
padding-left	define o preenchimento esquerda
padding-right	define a margem direita interior
padding-top	define o preenchimento no top

Perguntas de autoavaliação:

- 1) Dê alguns exemplos de propriedades para especificar a formatação de um tag CSS.
- 2) Com base nos exemplos de propriedades selecionados crie um ficheiro HTML e CSS e exib os efeitos em um navegador Web.

Unidade 2. Programação Web avançado do lado do servidor

Introdução

Esta unidade tem como objetivo ajudá-lo(a) a compreender e desenvolver aplicações Web dinâmicas que tiram partido de linguagens de programação do lado do servidor. PHP (inicialmente o acrónimo Personal Home Page, atualmente definido como HyperText Preprocessor) é utilizada como um exemplo neste módulo devido à sua popularidade. A diferença da linguagem PHP em relação as outras linguagens é que o código é executado do lado do servidor, sendo enviado para o cliente somente o código HTML.

Objetivos da Unidade

Após a conclusão desta unidade, pretende-se que o (a) estudante seja capaz de:

Definir e aplicar o conceito da linguagem do lado do servidor, o caso do PHP:

- Identificar as vantagens e desvantagens do PHP;
- Instalar e configurar o ambiente de desenvolvimento Web;
- Conhecer a sintaxe da linguagem PHP, tipos e precedentes, etc.;
- Conhecer os identificadores da linguagem, variáveis e array;
- Reconhecer a estrutura de controlo com PHP: estrutura condicional e de repetição;
- Desenvolver aplicações Web utilizando o server side script;
- Criar scripts de interação com a base de dados (acesso, inserção, exclusão, atualização e listagem dos dados);

Saber utilizar corretamente sessões para autenticar utilizadores.

TERMOS-CHAVE

PHP:HyperText Preprocessor

MySQL: Sistema de Gestão de Base de Dados.

Página Web: Páginas em formato HTML e com ligações de hipertexto que permitem a navegação de uma página, ou sessão para outra.

Razões da escolha da linguagem com PHP e o MySQL

O MySQL é um sistema de gestão de base de dados (SGBD) relacional. Permite armazenar e manipular informações de forma eficiente e seguro. O PHP sendo uma linguagem interpretado no lado do servidor Web é um produto open source (código aberto), o utilizador pode usar e distribuir sem nenhum custo.

Atividades de Aprendizagem

Atividade 1. Noções básicas de programação com PHP

Introdução

Esta atividade tem como objetivo primário, entender os conceitos básicos e como funciona a linguagem PHP. É uma linguagem script utilizada para programar as páginas Web que geram conteúdos dinâmicos.

Objetivos

Conhecer a sintaxe básica da linguagem em PHP;

Criar uma aplicação em PHP;

Aplicar os conceitos de programação do lado do servidor.

HyperText Preprocessor (PHP)

PHP é uma linguagem simples com raízes em C e Perl. Possui suporte para orientação ao objeto e está disponível para muitos sistemas operativos, tais como, o Windows, Linux, etc.

Por padrão, os documentos PHP são definidos com a extensão "nomeficheiro.php" Quando um servidor Web executa um arquivo solicitado com esta extensão, automaticamente interpreta-o como um arquivo PHP.

Nas unidades anteriormente apresentadas, foi instalado o editor Brackets para criar páginas Web (ver a unidade 2). Nesta unidade, instalaremos o pacote WAMP. O PHP, o MySQL e servidor o Apache são instalados automaticamente com a instalação do pacote WAMP.

Programação do lado do Servidor (Server Side Programming)

O script do lado do servidor é uma tecnologia em que o utilizador (cliente) faz uma solicitação e é realizada por meio de execução de um script directamente do servidor Web. Depois de o servidor Web, processar o pedido, será enviado ao cliente apenas o conteúdo através da linguagem de marcação. Para compreender melhor o funcionamento da linguagem, suposemos o seguinte cenário: Suponha que que digitou no navegador (browser) o endereço <http://exemplo.com/index.php> como mostrado na figura 3.1:

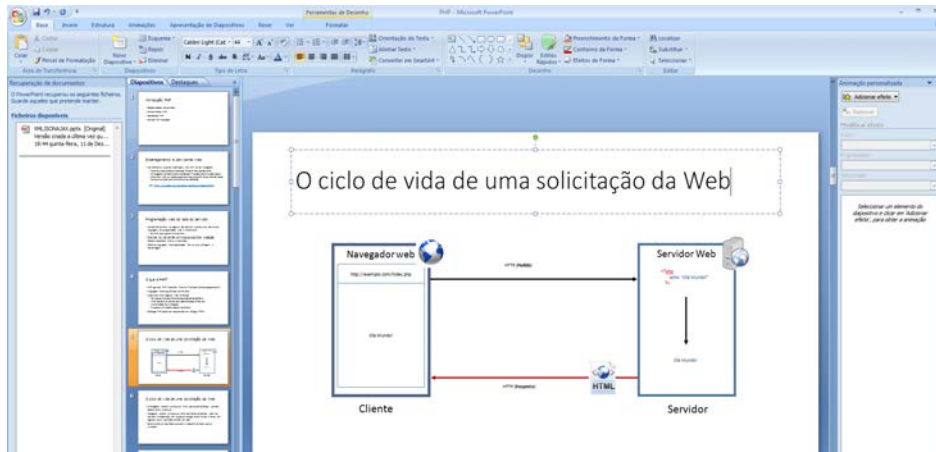


Figura 3.1: Funcionamento do PHP

O servidor faz a leitura do arquivo PHP solicitado e executa as instruções. Após a interpretação das instruções o servidor envia os resultados ao cliente. Com base na figura 3.1, é apresentado ao cliente somente o conteúdo, sem as instruções PHP. Neste caso, o cliente somente poderá ver o código fonte da página (HTML) se inspecionar o código fonte através do navegador.

Estilos de tags no PHP

```
<?php //inicia o script  
    Instruções;  
?> // termina o script
```

Para que o servidor interprete o código PHP, deve iniciar o script com a tag `<?php` e terminá-lo com a tag `?>`

Atividade 2 - Sintaxe básicos com PHP

Introdução

Nesta atividade abordaremos as regras básicas para uma estrutura com PHP. Pode-se embutir o código HTML com o PHP. O que for do PHP será executado pelo lado do servidor e o script PHP não é mostrado pelo navegador. O código HTML será enviado para o navegador Web, que será exibido do lado do cliente. Ver o exemplo:

```
<html>

  <head>

    <title>Aprender PHP</title>

  </head>

  <body>

    <?php

      echo "Programar com PHP!";

    ?>

  </body>

</html>
```



Comentários com PHP

Há varias formas de definir um comentário com PHP:

```
// Este é um comentário simples
```

```
/* Este é um comentário múltipla linhas */
```

Exemplo de comentário com múltiplas linhas com PHP:

```
<?php

  /* Este é o exemplo de comentário
   com múltiplas linhas que não será
   interpretada durante a execução do programa */

?>
```

Ponto e vírgula

Com PHP todas as intrusões terminam com um ponto-e-vírgula.

Exemplo: `$var = 10; /* a declaração termina com um ponto-virgula*/`

Variáveis e tipos de variáveis com PHP

As variáveis são representadas com PHP por um símbolo \$ e uma string, que deve iniciar por uma letra ou caracteres. O PHP é case sensitive, ou seja, definir uma variável com \$var é diferente com a variável \$VAR.

Exemplo de tipos de variáveis suportados com PHP:

Inteiro (integer);

Strings;

Vetores (Array);

Objetos.

Declaração de uma variável:

```
<?php
    $var = 10; /* var é o nome atribuído e 10 é o valor*/
    $string = "Ola mundo!";
    $array = array("Um", "Dois", "Três");
?>
```

Variável global

Com PHP, assim como outras linguagens de programação são declaradas variáveis globais, o que significa que elas estão sempre acessíveis, independente do escopo.

Por exemplo:

```
<?php
    $x = 4;
    $y = 8;
    function exemplo() {
        global $x, $y;
        $x = $x+ $y;
    }
    exemplo();
    echo $x;
?>
```


Variáveis de impressão echo e print

Em PHP, existem duas formas básicas para exibir os dados: echo e print. São mais ou menos diferentes. O echo não tem valor de retorno enquanto que o print tem um valor de retorno para que possa ser utilizado em expressões.

Exemplo: echo "Aprender PHP!";

```
print "Aprender PHP!";
```

Operadores com PHP

No quadro a seguir, são enumerados os principais operadores com PHP: os operadores matemáticos, operadores de comparação e operadores lógicos.

Operadores Matemáticos	Descrição	Exemplo
+	Adição	<code>\$var + 2</code>
-	Subtração	<code>\$var - 7</code>
*	Multiplicação	<code>\$var * 5</code>
/	Divisão	<code>\$var / 3</code>
%	Módulo (resto da divisão)	<code>\$var % 9</code>
++	Incremento	<code>++\$var</code>
--	Decremento	<code>--\$var</code>
Operadores de comparação	Descrição	Exemplo
==	É igual a	<code>\$var == 6</code>
!=	Não é igual a	<code>\$var != 11</code>
<	É maior que	<code>\$var < 2</code>
>	É menor que	<code>\$var > 7</code>
<=	É menor igual a	<code>\$var <= 4</code>
>=	É maior igual a	<code>\$var >= 12</code>
Operadores lógicos	Descrição	Exemplo
&&	E	<code>\$var == 3 && \$cont == 5</code>

And	E (low-precedence)	<code>\$var == 3 and \$cont == 5</code>
	Or	<code>\$var == 3 \$cont == 5</code>
Or	Or (low-precedence)	<code>\$var == 3 or \$cont == 5</code>
!	Negação	<code>!(\$var == \$cont)</code>
Xor	Exclusivo or	<code>\$var xor \$cont</code>

Perguntas de autoavaliação:

P 1: Indique quatro tipos de linguagens do lado do servidor que conhece.

R:

P 2: Qual é a diferença entre a linguagem PHP e o JavaScript?

R:

P 3: Entre as opções apresentadas, selecione a opção correta de uma variável com PHP:

#cont

Var

&var

\$var

P 4: Qual é a diferença entre uma variável local e uma global?

R:

P 5: Qual é a diferença entre o comando echo e o print?

R:

P 6: Escreva um script com PHP para exibir as seguintes sequências de caracteres como resultado.

"Fundamentos da internet."

"Programação Web"

P 7: Coloque a variável seguinte na sessão de título <h1>, como uma âncora de texto dentro do documento HTML.

Exemplo: Programar com PHP.

P 8: Escreva um algoritmo para exibir um conjunto de caracteres e valores dentro de uma tabela.

P 9: Escreva um algoritmo que muda a cor do primeiro caracter de uma palavra.

Exemplo: Aprender PHP

Resultado: Aprender PHP

P 10: Escreva um algoritmo com PHP que escreva a data em que um ficheiro foi modificado.

P 11: Dado o seguinte algoritmo:

```
<?php
    $ficheiro = "conexao.php"
    $linha = count(file($ficheiro));
    echo "Sao $linha linhas no ficheiro";
?>
```

Identifique o erro no algoritmo.

Qual é o resultado após a compilação?

Atividade 3 - Expressões e Estrutura de controlo com PHP

Introdução

Uma expressão é a combinação de valores, variáveis, operadores e funções.

Por exemplo: `$var = 10;` // significa que é atribuída a variável `$var` o valor 10

Podemos dizer que o "10" é uma expressão com o valor 10. Qualquer script PHP é construído por uma série de instruções. Uma instrução pode ser uma atribuição, uma chamada de função, um 'loop', uma instrução condicional, ou mesmo uma instrução que não faz nada (um comando vazio). A implementação de estruturas de controlo permite gerir a ordem como as instruções serão executadas em um programa.

Comando de seleção: if e switch

Os comandos de seleções, tem como objetivo selecionar um determinado bloco de comandos a partir de avaliação de uma expressão. O comando if é um dos mais importantes em uma linguagem de programação. Este comando permite a execução condicional de fragmentos de código.

```
if(condição){ ...instrução;}
```

Ver o exemplo:

```
<?php
    $x = 5;
    $y = 3;

    if($x > $y){
        echo "X e maior que Y";
    }
?>
```

if..else...

Executa o primeiro bloco de comando se a expressão for verdadeira, e o else (se não) caso a expressão for outra coisa.

```
    if(condição){ ... se verdadeiro a condição; }else{...se falso a
condição;
    }
```

Ver o exemplo seguinte:

```
<?php
    $x = 5;
    $y = 3;

    if($x > $y){
        echo "X e maior que Y";
    }else{
        echo "X e menor que Y";
    }
?>
```

If...elseif...else

Este comando permite analisar várias condições. Para cada elseif, uma nova expressão deve ser analisada, até que o bloco else seja executada.

```
if(condição) {  
    se verdadeiro a condição;  
elseif(condição) {  
    se verdadeiro a condição;  
} else {  
    se falso a condição  
}
```

Ver o exemplo seguinte:

```
<?php  
  
    $x = 5;  
  
    $y = 3;  
  
    $z = 1;  
  
    if($x > $y) {  
        echo "X e maior que Y";  
    }elseif($x > $z) {  
        echo "X e maior que Z";  
    }else{  
        echo "X e menor que Y e Z";  
    }  
  
?>
```

Comando switch...case

O comando switch funciona de forma semelhante a vários if. A expressão é comparada com cada uma das cláusulas case até que sejam iguais. Caso aconteça, será executada o bloco do comando correspondente até que seja interrompida com o comando break. Se a expressão não coincidir é executado o comando default.

```
switch(expressão) {  
    case "valor1": instrução;  
    break;  
    case "valor2": instrução;  
    break;  
    default: instrução;  
}
```

Ver o exemplo seguinte:

```
<?php  
    $exp ="x";  
    switch($exp) {  
        case "x": echo "Voce venceu!";  
            break;  
        case "y": echo "Voce perdeu!";  
            break;  
        default: echo "Operacao falhou";  
    }  
?>
```

Comando de repetição: While, Do-While e For

O comando de repetição while é do tipo mais simples em PHP. A condição é executada no início da iteração. Caso for verdadeira, repete o bloco de comandos, e se for falsa, termina a repetição.

```
while(condição) {...instrução;}
```

Ver o exemplo seguinte:

```
<?php  
    $var = 1;  
    while ($var <= 5) {
```

```
        echo $var++;  
    }  
?>
```

Do-While

A diferença entre o do-while com o while é o facto de que, a condição é executada no final do bloco do comando, o que significa que a instrução será executada pelo menos uma vez.

```
do{...instrução;} while(condição);
```

Ver o exemplo seguinte:

```
<?php  
  
    $x = 0;  
  
    do{  
  
        echo $x;  
  
    }while ($x > 0);  
  
?>
```

Repetição For

É utilizada quando sabemos o número exato de iterações.

```
for(início; condição; incremento){... instrução }
```

Veja o exemplo seguinte:

```
<?php  
  
    for($var=1; $var<10; $var++) {  
  
        echo "$var <br>";  
  
    }  
  
?>
```

Funções em PHP

As funções em PHP, assim como, outras linguagens de programação tem o mesmo princípio. Em PHP, não é preciso declarar os tipos de variáveis. Uma função é um trecho de código que recebe valores de entrada, processa e retorna-os.

```
function (parâmetros) {  
    instrução;  
}
```

Ver o exemplo:

```
<?php  
  
function olamundo() {  
    $n1 = 4;  $n2 = 5;  
  
    $soma = $n1 + $n2;  
  
    echo "Soma de ".$n1. " e ".$n2." = ".$soma;  
}  
  
echo olamundo();  
  
?>
```

Neste exemplo o retorno da função será o valor 9.

Array (Vetores)

Array são estruturas de dados que armazenam um ou mais tipos valores em um único valor.

Considere as seguinte palavras: Futebol, Golfe, Voleibol, Natação com o nome de array: desporto.

```
array[índice] = {elemento};
```

Índice	1	2	3	4
Elemento	Futebol	Golfe	Voleibol	Natação

Ver o exemplo:

```
<?php  
  
$arr[0] = "Futebol";  
  
$arr[1] = "Golfe";  
  
$arr[2] = "Voleibol";
```



```
$arr[3] = "Natação";  
  
echo $arr[0] . " | " . $arr[1] . " | " . $arr[2] . " | " .  
$arr[3];  
  
?>
```

Perguntas de autoavaliação

P 1: Mostre como os comandos if e while interpretam expressões condicionais de diferentes tipos de dados.

R:

P 2: Escreva um algoritmo "Adivinha o número". O utilizador insere um número:

Se o número escolhido for igual ao número aleatório escreva "Acertou!"

Se o número é maior que o número aleatório, escreva "É maior"

Se não escreva "É menor".

P 3: Crie um algoritmo utilizando o comando de repetição para adicionar números inteiros entre 0 a 100 e exiba o total da soma.

R:

P 4: Crie uma função para calcular o fatorial de um número inteiro não negativo. A função recebe o número como um argumento.

R:

P 5: Crie uma função que verifica se uma cadeia de caracteres estão em minúscula.

R:

P 6: Escreva um algoritmo que insere um item em qualquer posição de um vetor.

RESUMO DE ACTIVIDADE

Nesta atividade, aprendemos a noção básica do PHP. Na sequência, vimos a sintaxe básica do PHP, assim como, a estrutura condicional e de repetição. Os conceitos abordados nesta atividade são de extrema importância para desenvolver páginas Web dinâmicas.

Atividade 4 – Criando um projeto com PHP

Introdução

Esta atividade, tem como objetivo principal instalar e configurar o ambiente de desenvolvimento de aplicações para Web.

Em seguida, será executada o primeiro programa, que servirá para testar a instalação e configuração do ambiente. O processo de instalação e configuração será no ambiente do Windows.

Objetivos

Instalar o ambiente de desenvolvimento (PHP e MySQL);

Configurar a variável de ambiente;

Criar um programa "Olá Mundo" para testar o ambiente de desenvolvimento.

Requisitos necessários

O PHP é a linguagem do lado do servidor. Neste sentido, é preciso:

- Um servidor para processar o PHP;
- Um servidor HTTP para estabelecer comunicação entre o cliente e o servidor;
- Um sistema de gestão de base dados (SGBD), o MySQL;
- Um interpretador PHP para rodar o código PHP, para chamada de sistemas (System calls), consultas a base de dados e conteúdos dinâmicas para páginas Web.

Etapa 1: Instalar e configurar o ambiente de desenvolvimento

O pacote WAMP é uma das soluções desenvolvidas pela comunidade Open Source para o desenvolvimento de aplicações Web dinâmicas. Foi desenvolvido o apache, MySQL e PHP, incorporado no pacote WAMP para ambientes Windows.

Faça o download do pacote no site <http://www.wampserver.com/en/>. Para mais informações de instalação, consulte o site.

Para executar aplicações do lado do cliente é preciso um navegador Web (Chrome/Firefox). Após a instalação do pacote WAMP, abra o navegador e digite <http://localhost>. Em seguida, é apresentado a seguinte página, clique no menu phpMyAdmin.

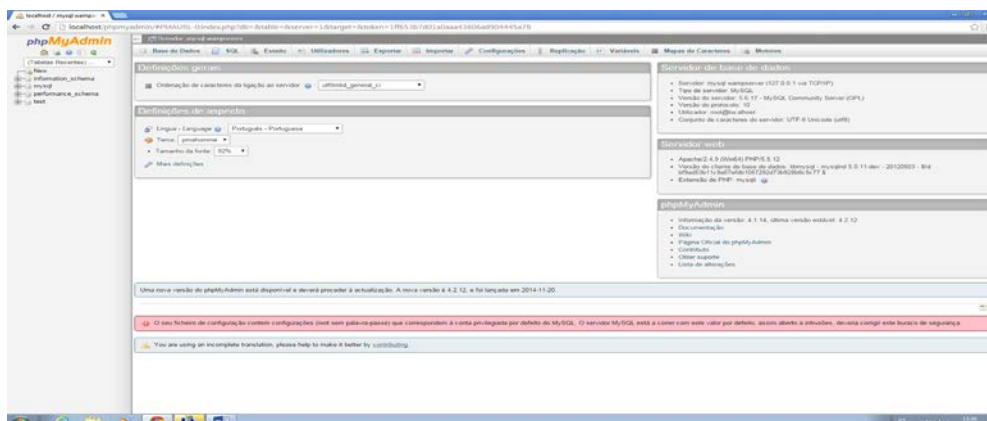


Figura 3.3: Página principal do phpMyAdmin

Com a instalação do pacote WAMP, podemos notar um aviso na página principal do phpMyAdmin. Isto porque, o phpMyAdmin notifica o utilizador que a base de dados não tem a senha para o utilizador principal (root), que é, de facto, uma porta aberta para intrusos.

Sendo a instalação bem sucedida, deverá ser ativado o painel de controlo do WAMP. A figura exhibe o painel de controlo com o Apache e MySQL. Para ativar os serviços deverá seleccionar o menu "Put online".

Um outro processo a ser executado, é configuração da variável de ambiente PATH.

Clique com o botão direito em Computador e selecione Propriedades no menu de contexto. Na janela que se abre, selecione a guia Avançado e clique no botão variáveis de Ambiente. É aberto uma lista de todas as variáveis de Ambiente. Na parte inferior, variáveis de Sistema, selecione e edite o caminho C:\wamp\bin. Todas as entradas devem ser separadas por um ponto-e-vírgula.

Perguntas de autoavaliação:

P 1: Qual é a principal diferença entre o WAMP e o LAMP?

R:

P 2: Quais são os quatro (4) componentes mínimos necessário para criar uma página Web dinâmica?

R:

P 3: PHP e JavaScript são as duas das principais linguagens de programação que geram conteúdos dinâmicos em uma página Web. Qual é a principal diferença, e porquê do uso das duas linguagens?

R:

P 4: Qual é o elemento utilizado para fazer com que o PHP interprete o código do programa? E qual é a forma abreviada do elemento?

R:

P 5: Qual é o símbolo utilizado para introduzir variáveis com PHP?

R:

P 6: É permitido utilizar espaços nos nomes das variáveis?

R:

P 7: Como criar uma variável acessível a todas as partes de um programa com PHP?

R:

P 8: Porque a repetição for é considerada a mais poderosa que a repetição while?

R:

P 9: Qual é o benefício em utilizar funções?

R:

P 10: Qual é o significado de escopo com PHP?

R:

P 11: Como incorporar um arquivo PHP em outro arquivo?

R:

P 12: Implemente o algoritmo que faz a soma da nota (N1, N2, N3) duma disciplina, e efetue a média da mesma. Em caso da média ser maior ou igual a dez (10), é apresentado a mensagem "Aprovado", senão "Reprovado".

R:

P 13: Crie um algoritmo que imprima a seguinte estrutura:

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
****
```

```
***
```

```
**
```

```
*
```

P 14: Crie um algoritmo que armazena 10 números inteiros em um array e imprime uma mensagem de todos os números pares.

R:

Etapa 2: Antes de iniciarmos o projeto com PHP, precisamos criar uma pasta dentro do diretório C:\wamp\www\ com o nome "MeuPrograma".

Abra o navegador e digite `http://localhost/meuprograma/`. É exibida a página o projeto MeuPrograma sendo servida pelo PHP.

Etapa 3: Para testarmos a instalação do PHP, vamos implementar um simples programa, porém muito útil.

Abra o editor Brackets e crie um programa com PHP que imprime "Ola Mundo!"

```
<?php
    echo "<h1>". "Ola Mundo!".</h1>
?>
```

Guarde o documento dentro da pasta `c:\wamp\www\MeuPrograma/` com o nome `olamundo.php`

Utilizando o navegador Chrome/Firefox execute o projeto. O resultado, mostra como o PHP funciona. O script é executado do lado do servidor ficando disponível apenas o conteúdo da página.

RESUMO DE ACTIVIDADE

Nesta atividade, instalamos os requisitos necessários para o desenvolvimento de uma aplicação Web dinâmica, criamos um simples programa "Ola Mundo!" para testar o ambiente de desenvolvimento.

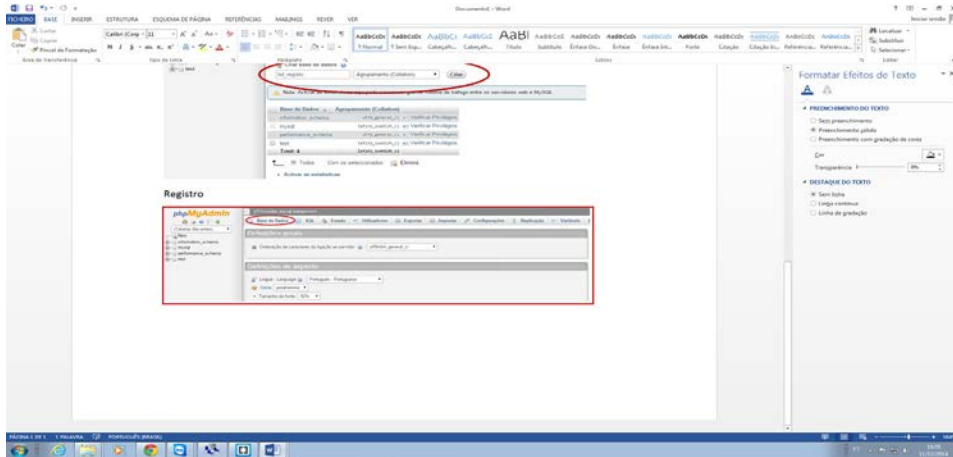
Atividade 5 – Manipular dados na base de dados com PHP

Introdução

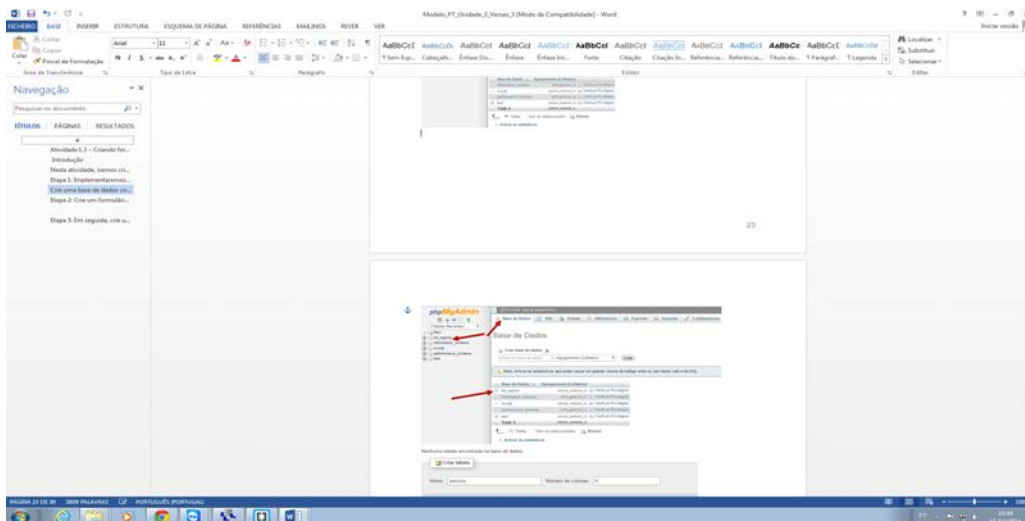
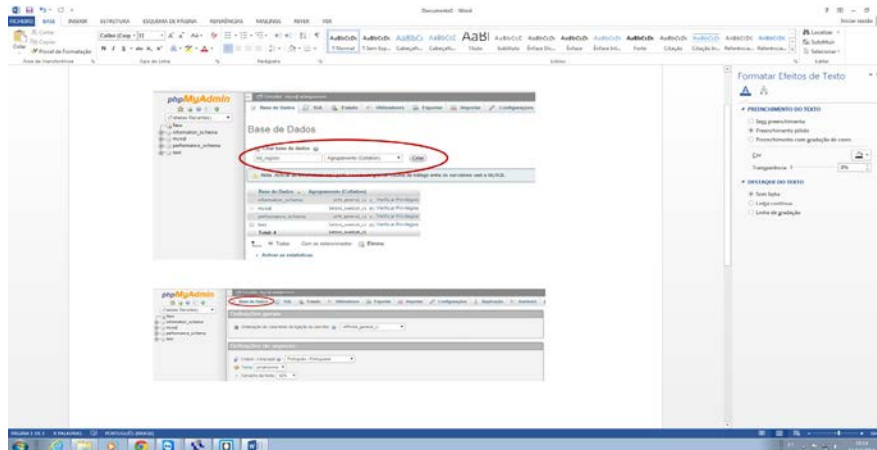
Nesta atividade, iremos criar o primeiro exemplo para manipulação de dados com PHP. Abordaremos as seguintes queries SQL: inserir (insert), selecionar (select), eliminar (delete) e atualizar (update) dados numa tabela MySQL utilizando PHP.

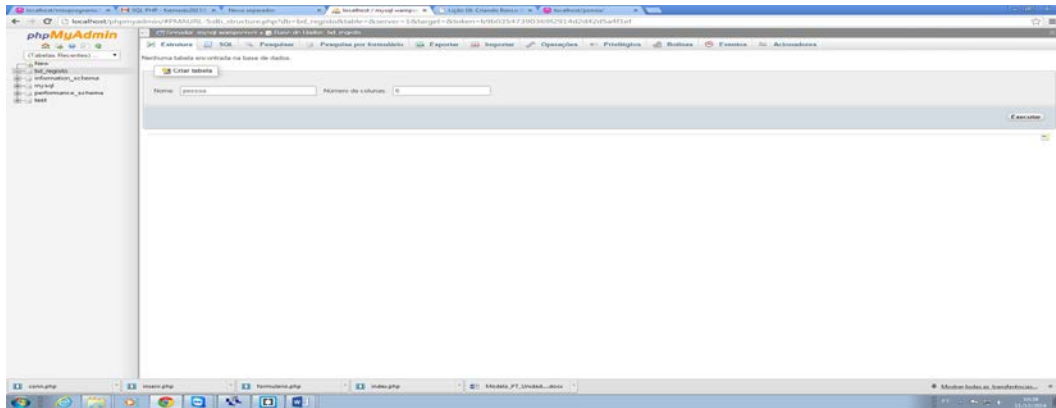
Etapa 1: Nesta secção, vamos criar uma simples base de dados e tabelas com phpMyAdmin. Base de dados é um conjunto de informações/dados organizados de forma a facilitar a manipulação dos dados. Nesta etapa, criaremos a base de dados com a ferramenta phpMyAdmin da MySQL, que é fácil e intuitiva.

Crie uma base de dados com o nome `bd_registro` utilizando phpMyAdmin e de seguida crie uma tabela com o nome `pessoa`.



Estando conectado ao servidor e com acesso à base de dados, digite o nome da base de dado.





Em seguida, crie a tabela pessoa com a seguinte estrutura:

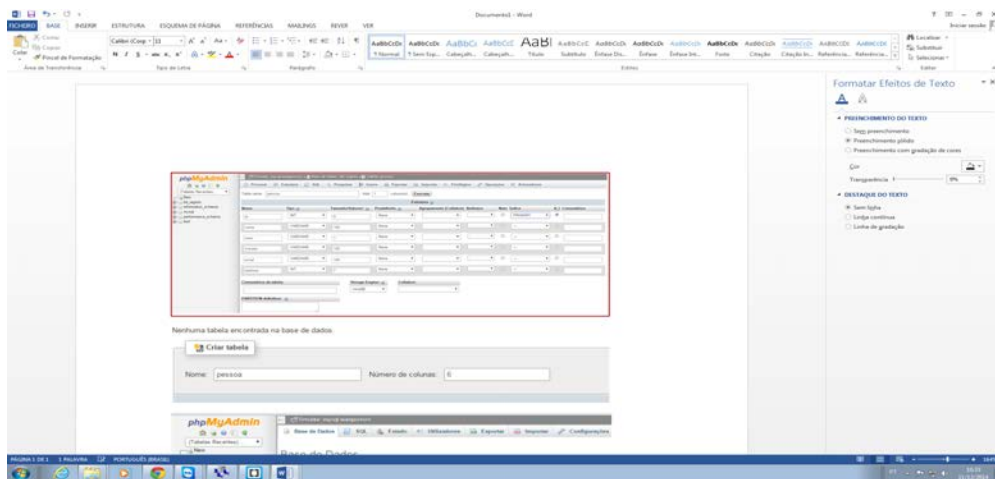


Figura 3.7: Estrutura da tabela pessoa

Etapa 2: Nesta etapa veremos com criar base de dados e tabelas phpMyAdmin

A manipulação de dados é efetuada através da função `mysql_query()`. Esta função permite efetuar consultas (queries) em uma base de dados.

As consultas são escritas com a linguagem SQL (Strutred Query Language). SQL é a linguagem de pesquisa declarativa padrão para uma base de dados relacional.

Query SQL para criar uma base de dados

```
CREATE DATABASE nome_bd
```

Exemplo com PHP:

```
<?php
```

```
mysql_connect ("localhost", "root", ""); or die (mysql_error());  
mysql_query ("CREATE DATABASE bd_registro") or die (mysql_error());  
mysql_close();  
?>
```

Neste trecho de código, criamos a base de dados com o nome bd_registro utilizando a função mysql_query ()

Etapa 3: Crie um formulário com o nome index.html para a inserção dos dados na tabela cadastro. Neste exemplo, criaremos dois (2) tipos ficheiros: o index.html e o inserir.php.

Exemplo prático: index.html

```
<html>  
<head></head>  
<body>  
    <h1>Inserir dados:</h1>  
    <form method="post" action="inserir.php">  
        <pre>  
            Nome:    <input type="text" name="nome"/> <br/>  
            Sexo:    <input type="text" name="sexo"/> <br/>  
            Morada:  <input type="text" name="morada"/> <br/>  
            Email:   <input type="text" name="email"/> <br/>  
            Telefone:<input type="text" name="telefone"/> <br/>  
                    <input type="submit" name="submit" value =  
"Inserir"/><br/>  
        </pre>  
    </form>  
</body>  
</html>
```

Etapa 4: Em seguida, crie um ficheiro inserir.php para a conexão com o servidor de base de dados MySQL utilizando a função mysql_connect(). A função contém os seguintes parâmetros:

Um servidor (localhost)

Nome utilizador e senha para conexão (login).

Exemplo da sintaxe: `mysql_connect (servidor, utilizador, senha);`

Conectando a um servidor local:

```
mysql_connect ("localhost", "root", "") or die (mysql_error());
```

O parâmetro servidor indica o local onde se encontra a base de dados, e os parâmetros utilizador e senha que dão acesso à base de dados.

m seguida , é utilizada a função `mysql_select_db()`, para selecionar a base de dados (entidade). Foi criada uma tabela com o nome de pessoa com seis (6) campos como mostrado na figura 3.7.

Exemplo:

Para definir o método de acesso aos formulários são utilizados dois tipos de métodos. O método GET em que os dados são enviados juntamente com o nome da página para o envio de dados. Exemplo: <http://localhost/meuprograma/atualizar.php?id=3>

? - significa o início da cadeia de variáveis

= - separa as variáveis dos seus valores

O método POST envia os dados por meio da mensagem enviada ao servidor.

Exemplo: `http://localhost/meuprograma/atualizar.php`

Inserir dados na tabela pessoa com o PHP:

```
<?php
//Conectar a base de dados
mysql_connect("localhost", "root", "");
mysql_select_db("bd_registro");
    $nome = $_POST["nome"];
    $sexo = $_POST["sexo"];
    $morada = $_POST["morada"];
    $email= $_POST["email"];
    $telefone = $_POST["telefone"];
//Inserir dados
```

```
$dados = "INSERT INTO pessoa (nome, sexo, morada, email,
telefone) VALUES ('$nome', '$sexo', '$morada', '$email',
'$telefone)";

$resultado = mysql_query($dados);

if($resultado){
    echo("<br>Os dados foram inseridos com sucesso!");
}
else{
    echo ("<br>A Os dados não foram inseridos!");
}

?>
```

Etapa 5: Tratamentos de resultado de consulta utilizando a query SELECT

Ao executar uma consulta SQL, através do comando `mysql_query`, o identificador do resultado deve ser armazenado numa variável que pode ser de diversas formas para apresentação de resultados. Na atividade que se segue, utilizaremos o comando `mysql_fetch_row`. Crie um ficheiro `selecionar.php` dentro do diretório `...\MeuPrograma`

Abra o Brackets e digite o seguinte de código:

```
<?php

//Conectar a base de dados
mysql_connect("localhost", "root", "");
mysql_select_db("bd_registo");

//Consulta
$query = "SELECT * FROM pessoa ORDER BY id";
$resultado = mysql_query($query);

//Apresentar a consulta numa tabela
echo("<h1>". "Resultado da consulta: " . "</h1>");
echo("<table border=3>");

echo("<tr><td>ID</td><td>NOME</td><td>SEXO</td><td>MORADA</td><td>EMAIL</td><td>TELEFONE</td></tr>");

while($linhas = mysql_fetch_row($resultado)){
    echo("<tr><td>$linhas[0]</td><td>$linhas[1]</td><td>$linhas[2]</td><td>$linhas[3]</td><td>$linhas[4]</td></tr>");
}
```

```
td><td>$linhas[5]</td></tr>");  
  
    }  
  
    echo ("</table>");  
  
?>
```

Nota-se, a utilização da função `mysql_fetch_row($resultado)`, que retorna o número de linhas de um resultado. A variável `resultado` é o identificador da memória de resultados, obtido como retorno da função `mysql_query`. Os resultados são colocados em um array, utilizando a variável `linhas[índice]`.

Código para eliminar os dados na tabela cadastro:

```
<?php  
  
    $eliminar = "DELETE FROM pessoa WHERE id='$id'";  
  
?>
```

Código atualizar os dados na tabela cadastro:

```
<?php /  
  
    $atualizar = "UPDATE pessoa SET nome='$nome', sexo='$sexo',  
morada='$morada', email='$email', telefone='$telefone' WHERE id=  
$id'";  
  
?>
```

RESUMO DE ATIVIDADE

Nesta atividade, criamos uma página interativa para inserir, listar atualizar e eliminar os dados.

Perguntas de autoavaliação:

P 1: Qual é o objetivo do ponto-e-vírgula nas consultas MySQL?

R:

P 2: Qual é o comando utilizado para mostrar as informações disponíveis numa base de dados?

R:

P 3: Qual é a função padrão para se conectar com uma base de dados MySQL e a extensão MySQLi? Explique?

R:

P 4: Porque razão utilizar método POST é melhor do que o método GET?

R:

P 4: Crie uma aplicação com o nome biblioteca que faz a listagem de todos registos de um livro utilizando a seguinte estrutura:

Exemplo de resultado:

Código: 2361

Título: Programação para Web

Autor: João Barbosa

Data da edição: 05-11-2014

Editora: XXX

P 5: Crie um formulário que permite inserir, editar e eliminar registos na base de dados da aplicação biblioteca.

Unidade 3: Hospedagem de sites na Web

Para que o seu site seja acessível em qualquer parte do mundo deve ser alojado em um servidor remoto Web com um endereço IP (Internet Protocol) único. A aquisição desses endereços IP públicos são muito caros, por isso recorre-se ao alojamento compartilhado em grandes servidores. Os endereços IP são também difíceis de manusear, nomes de domínios como facebook.com, uva.org são utilizados para fazer referência a websites. Este capítulo tem como objetivo ajudá-lo a entender nomes de domínio, princípios de hospedagem Web e implementação de aplicações em um servidor de hospedagem Web.

Objetivos da Unidade

No final desta unidade, o(a) estudante deve ser capaz de:

Descrever o funcionamento da Internet;

Compreender o funcionamento entre um navegador Web e um Servidor de hospedagem Web;

Indicar a relação entre o DNS e a internet;

Introduzir a hospedagem na Web

Nomes do domínio

Selecionar um servidor de hospedagem

Aplicar o protocolo de transferência de ficheiros

TERMOS-CHAVE

Hospedagem de site: Serviço que possibilita a pessoas ou empresas com sistemas online guardar informações, imagens, vídeo, ou qualquer conteúdo acessível por Web. <http://www.oitobitdigital.com.br/servico/hospedagem-de-sites/>

DNS: Domain Name Server é um sistema de gestão de nomes hierárquico e distribuído para computadores, serviços ou qualquer recurso conectado à Internet ou em uma rede privada. http://pt.wikipedia.org/wiki/Domain_Name_System

Atividade 1 - Registo de nomes de domínio

Nomes de domínio

Para aceder o site da UVA é preciso digitar o endereço `www.avu.org` sem ter que identificar o endereço IP do servidor onde o site se encontra hospedado.

Nesse contexto, o nome de domínio é o endereço `www.avu.org` utilizado como o nome que faz referência ao site da UVA.



Ter um nome de domínio para seu site ou aplicação oferece uma série de vantagens:

Um site pode trocar de servidor de hospedagem sem mudar o nome de domínio. Isto pode acontecer sem que o visitante perceba. O que muda é o endereço IP para qual o nome de domínio se refere. Google, por exemplo, muda muitas vezes seus servidores sem que os utilizadores percebam.

Para uma empresa, um nome de domínio aumenta a credibilidade. Hoje em dia, se alguém precisa de informação, geralmente vai buscar em seu site.

Um nome de domínio relevante permite que as pessoas se lembrem mais facilmente o endereço de um website. Por exemplo `uva.org` é muito fácil de lembrar.

Nesta parte, vamos ver as etapas para a aquisição e aplicação de um nome de domínio.

Estrutura de um nome de domínio

Especificamente, um nome de domínio é composto por um conjunto de etiquetas (palavras-chave) separados por pontos (.). Como exemplo, `www.uva.org` A estrutura geral do nome de domínio é a seguinte:

SubDomínio.Domínio.Domínio_nivel_superior

A tabela a seguir ilustra exemplos de nomes de domínios:

	<code>www.google.com</code>	<code>www.governo.cv</code>	<code>www.uva.org</code>
Domínio de nível superior	<code>com</code>	<code>cv</code>	<code>org</code>
Domínio	<code>google</code>	<code>governo</code>	<code>uva</code>
Subdomínio	<code>www</code>	<code>www</code>	<code>www</code>

O domínio de topo (Top Level Domain em Inglês ou TLD) é uma extensão que permite nomes de domínio de grupos por país ou uso. É o rótulo mais à direita do nome de domínio. Existem duas categorias de TLDs: código genérico e do país.

Um TLD genérico (em Inglês, generic Top-Level Domain ou gTLD) é um domínio de nível superior destinado a hospedar subdomínios com uma outra característica comum do que geográfica. Por exemplo, o domínio .com é destinado a hospedar tipo de subdomínios comercial.

Em um nome de domínio, o domínio de nível superior é a última etiqueta de um nome de domínio (Exemplo: pt.wikipedia.org, o domínio de nível superior é .org).

Domínio de nível superior genérico	Descrição
.com	Comercial
.org	Organização
.net	Para redes
.info	Para serviços de informações
.biz	Negócio
.pro	Para sites profissionais Qualificados

Um domínio de nível superior com base em códigos ISO permite associar um Web site a um país. É utilizado para dar impressão de proximidade em relação a um país. CcTLDs (country code Top Level Domain) são TLDs em que as duas letras referem-se especificamente a uma parte (local) de um país em particular, um estado soberano ou território autônomo para uso em serviço à sua comunidade. CcTLDs são derivadas do código ISO 3166-1 alpha-2 de um país. A lista ccTDL se encontra disponível no seguinte endereço <http://icannwiki.com/CcTLD>.

Country Code Top level domain	Países
.cv	Cabo Verde
.bj	Benin
.ca	Canada
.bf	Burkina Faso
.cm	Rep. Camarões
.sn	Senegal

A extensão do domínio governo.cv a ccTDL .cv corresponde a Cabo Verde. Todos esses TDLs são reservados e geridos pela ICANN, que é um órgão da ONU (Organizações das Nações Unidas).As áreas representam parte reservada pelas empresas e indivíduos sobre o nome de domínio. Reservar um nome de domínio é escolher um nome de domínio e extensão TLD. O domínio africa.com, o nome de domínio é a africa e o TLD é .com.

Um subdomínio é tipicamente um nome atribuído a um servidor que executa o esquema de nomeação SubDomínio.Domínio.Domínio_nivel_superior

Provavelmente o subdomínio mais conhecido e o mais utilizado é o “www”. É uma convenção chamada servidores Web em um domínio com o nome www. No entanto, não é obrigatório, por exemplo, o nome da página Wikipédia em português é pt.wikipedia.org que não contém WWW.

Escolher um nome de domínio

A escolha de um nome de domínio é muito importante porque deve ser ao mesmo tempo simples e deve corresponder ao conteúdo do site. Nem sempre é fácil escolher o nome de domínio, porque deverá ser fácil de lembrar, e resumir o conteúdo do seu site.

Há um conjunto de critérios que são considerados na escolha de um nome de domínio:

Comprimento: um nome de domínio deve ser curto e simples. É importante evitar siglas que são difíceis de pronunciar.

Pontuação: Costuma-se colocar palavras separadas no nome de domínio. Neste caso, às vezes é possível colocar o hífen entre palavra. É recomendável evitar a utilização de hífen nos nomes do domínio.

Singular/Plural: É possível que a palavra-chave seja uma palavra que pode ser colocada no singular ou no plural.

Comprar um nome de domínio

Para adquirir e utilizar um nome de domínio é necessário efetuar a compra. A organização responsável pela compra de nome de domínio é ICANN (o regulador de nomes de domínio). No entanto, é raro comprar o nome de domínio diretamente com a ICANN. No caso de Cabo Verde, a ANAC (Agência Nacional de Comunicações) é o responsável pelo registo de nome de domínio com extensão .cv. Cada servidor de hospedagem e registo de domínio tem os seus próprios preços, e variam por ano. Exemplos:

www.anac.cv

www.1and1.com

ww.hostmonster.com

www.ovh.com

etc.

A figura seguinte ilustra a página para o pedido de registo de nome de domínio com extensão .cv.



Hospedagem de Sites na Web

Para disponibilizar um site na Web, deverá ser armazenado em um computador com um servidor Web e um endereço IP público. É muito caro comprar e alugar um servidor Web e um IP público para cada indivíduo. O princípio de alojamento é comprar um espaço em um servidor que é gerido e mantido por uma empresa especializada: provedor de hospedagem Web. Estes provedores possuem Datacenters que lhes permitem armazenar milhares de websites.

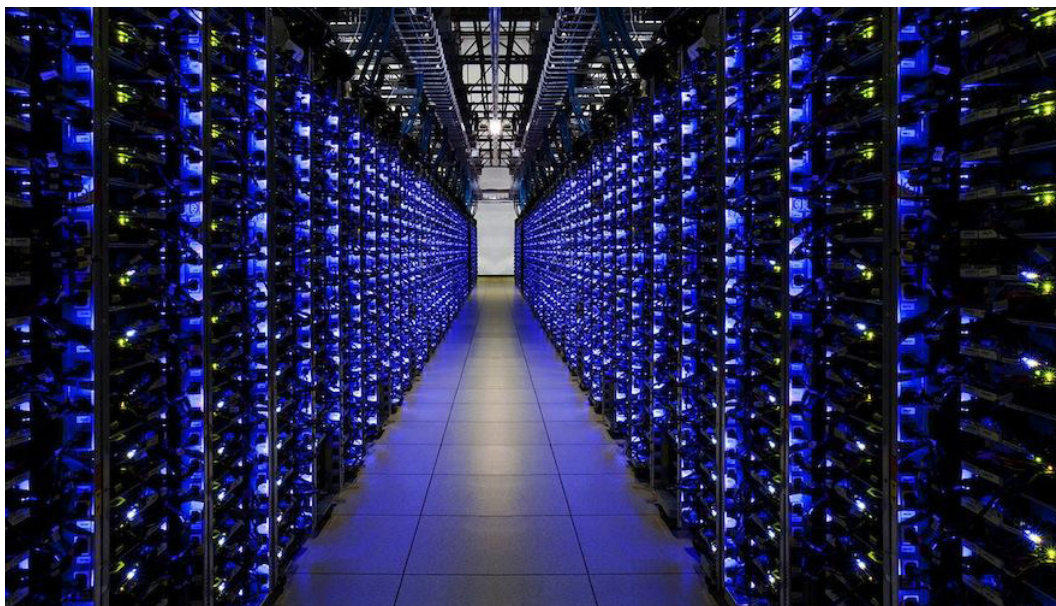
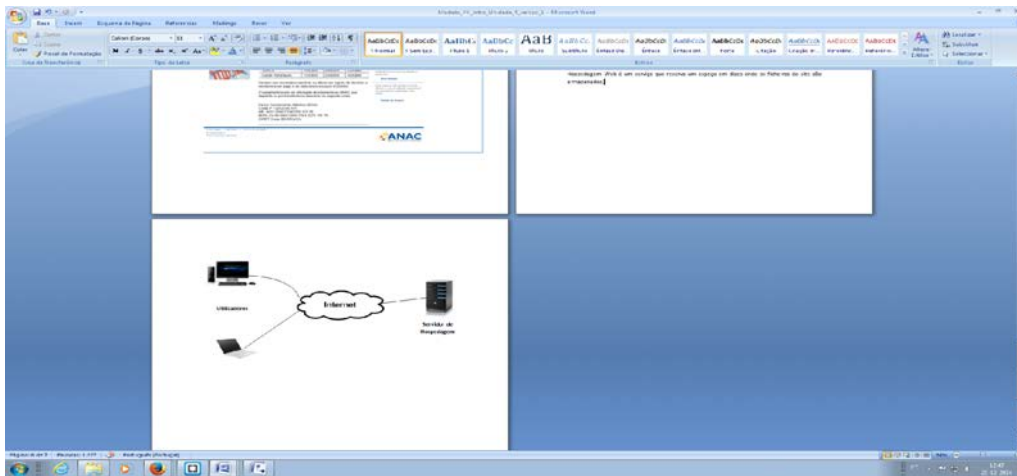


Figura 1: Datacenter do google (<http://www.trendsnow.net/2012/10/inside-google-data-center.html/inside-google-data-center-03>)

Hospedagem Web é um serviço que reserva um espaço, em disco, onde os ficheiros do site são armazenados.

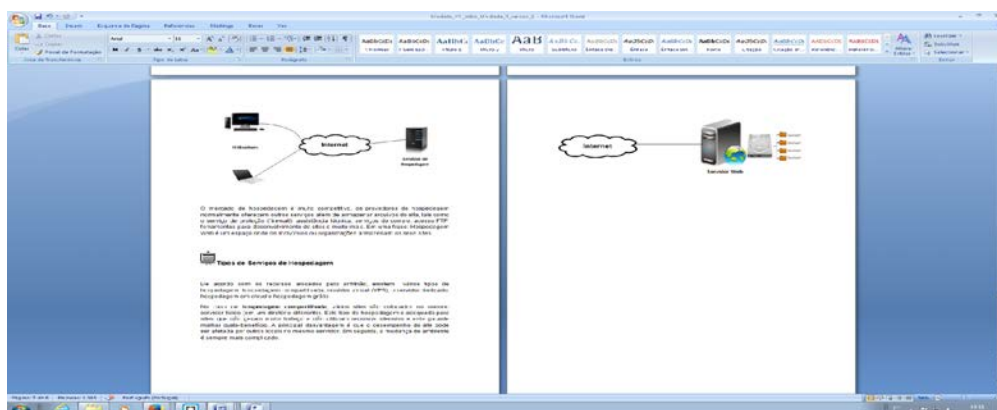


O mercado de hospedagem é muito competitivo, os provedores de hospedagem normalmente oferecem outros serviços além de armazenar arquivos do site, tais como o serviço de proteção (firewall), assistência técnica, serviços de correio, acesso FTP, ferramentas para desenvolvimento de sites e muito mais. Em uma frase: Hospedagem Web é um espaço onde os indivíduos ou organizações armazenam os seus sites.

Tipos de Serviços de Hospedagem

De acordo com os recursos alocados pelo anfitrião, existem vários tipos de hospedagem: hospedagem compartilhada, servidor privado virtual (VPS), o servidor dedicado, hospedagem em cloud e hospedagem grátis.

No caso de hospedagem compartilhada, vários sites são colocados no mesmo servidor físico (em um diretório diferente). Este tipo de hospedagem é adequado para sites que não geram muito tráfego e não utilizam recursos intensivo e este garante melhor custo-benefício. A principal desvantagem é que o desempenho do site pode ser afetada por outros locais no mesmo servidor. Em seguida, a mudança de ambiente é sempre mais complicada.



Exemplo de anfitriões web que oferecem serviços de hospedagem compartilhada:

Dreamhost

Host Gator

1&1

<http://www.justhost.com>

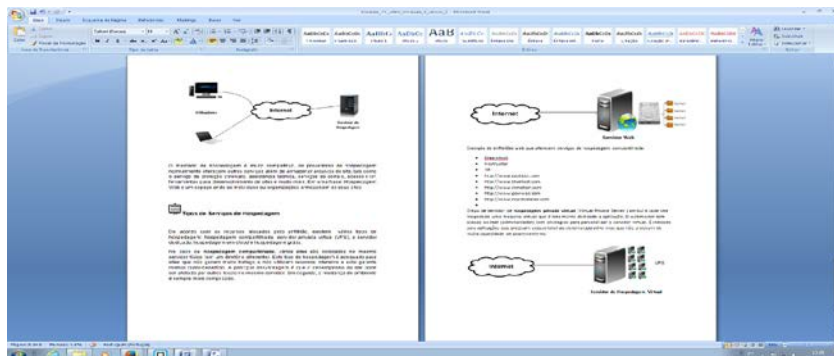
<http://www.bluehost.com>

<http://www.inmotion.com>

<http://www.powweb.com>

<http://www.hostmonster.com>

O tipo de servidor de hospedagem privada virtual (Virtual Private Server) atribuí a cada site hospedado uma máquina virtual que é totalmente dedicada à aplicação. O webmaster tem acesso ao root (administrador) com privilégios para personalizar o servidor virtual. É indicado para aplicações que precisam acesso total ao sistema operativo mas que não precisam de muita capacidade de processamento.



Exemplo de anfitriões web:

<http://www.slicehost.com>

<http://www.linode.com>

<http://www.asmallorange.com>

<http://www.prgmr.com>

<http://www.vr.org>

Quando um servidor dedicado é alugado para hospedar um site, o webmaster tem o controlo total sobre a configuração do servidor físico, bem como os seus recursos. Os recursos não são compartilhados com qualquer outro site e é o administrador que faz a configuração. Este tipo de hospedagem é adequado quando o desempenho do site é crítico.

Este é o tipo de hospedagem com custos elevados.

Exemplo de anfitriões web que oferecem serviços de hospedagem:

<http://www.rackspace.com>

<http://www.serverbeach.com>

<http://www.theplanet.com>

<http://www.verio.com>

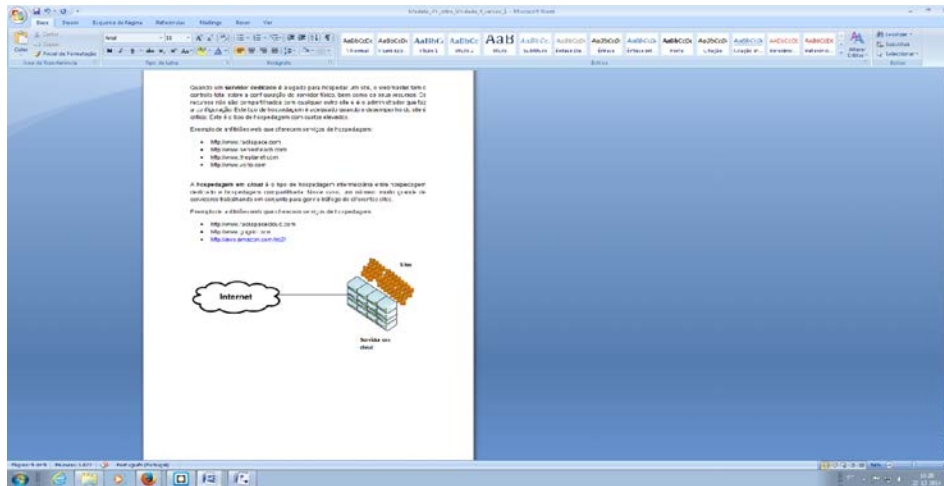
A hospedagem em cloud é o tipo de hospedagem intermediária entre hospedagem dedicada e hospedagem compartilhada. Neste caso, um número muito grande de servidores trabalhando em conjunto para gerir o tráfego de diferentes sites.

Exemplo de anfitriões web que oferecem serviços de hospedagem:

<http://www.rackspacecloud.com>

<http://www.gogrid.com>

<http://aws.amazon.com/ec2/>



Atividade 2 - Registrar o seu domínio com 1and1.com

Introdução

Objetivos

Escolher o provedor de hospedagem

Registrar o domínio

Criar uma conta para e efetuar a hospedagem

Hospedagem é um serviço que permite a uma entidade privada ou pública alojar conteúdos na Web. Existem muitos servidores no mercado com suporte a PHP e MySQL. Quando se desenvolve um projeto Web, devemos considerar a localização para hospedar a nossa aplicação. Este local é fornecido pelo provedor de hospedagem.

Para mais informações sobre o registo e plano de hospedagem consulte a página <http://www.dominiohost.com.br/hospedagem.html>.

Para escolher um provedor de hospedagem, devemos ter em consideração três pontos essenciais:

- O custo da Hospedagem
- Qualidade de serviço
- Atendimento e suporte

Há muitos tipos de servidores de hospedagem disponíveis na Web. Nesta atividade apresentaremos a hospedagem utilizando o provedor 1and1. A escolha deve ser do cliente, tendo em consideração as suas necessidades e recursos disponíveis.

Etapa 1: Registrar e hospedar o domínio com o provedor 1and1.com

Para hospedar o seu site na Web é preciso registrar o domínio através de um nome que permite localizar a sua aplicação (site).

Para registrar o domínio acede à página 1and1.com. Nota-se que, o processo para registo e hospedagem de páginas Web é semelhante a qualquer serviço de hospedagem na Web.

Acesse o menu Domain Name e clique no submenu Domain Name Registration

Em seguida é exibida a página para verificar se o domínio existe. Insira o nome do domínio e clique check. Verá a figura seguinte:

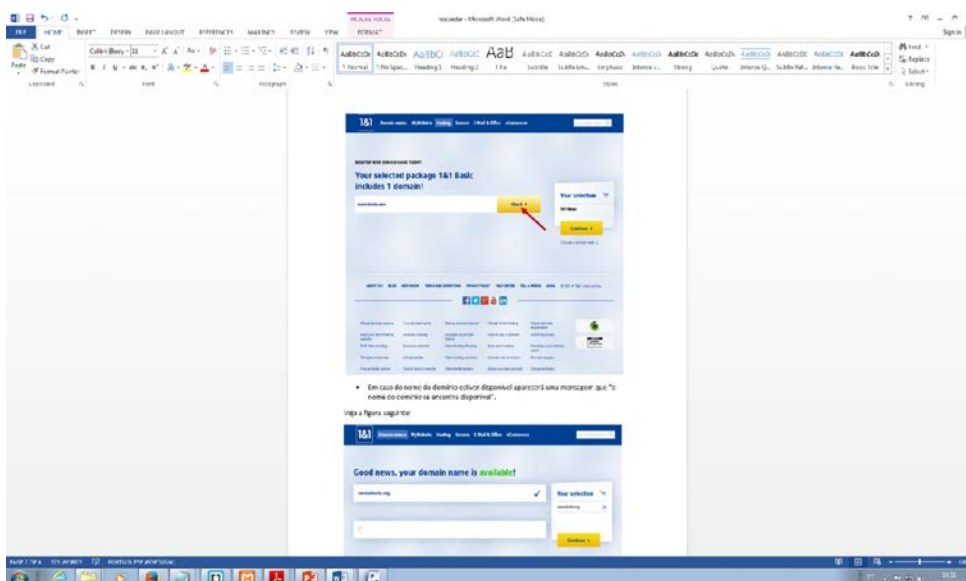


Figura 4.2: Painel de verificação do nome do domínio

Se o nome do domínio estiver disponível aparecerá uma mensagem que “o nome do domínio se encontra disponível”.

Ver a figura seguinte:

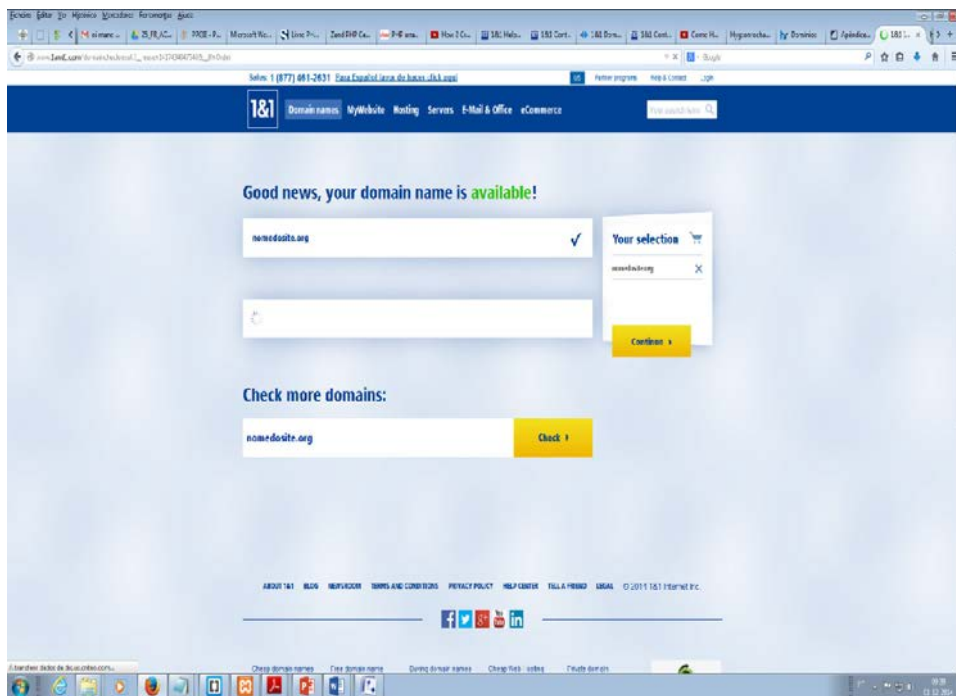


Figura 4.3: Exemplo da mensagem após da verificação do nome do domínio

Em seguida é exibida a página para efetuar o registo, onde deverá escolher o plano e efetuar o pagamento Online.

Enviar os ficheiros para o servidor de hospedagem.

Efetue login no 1and1 para aceder o painel de controlo usando o seu ID de cliente ou o nome de domínio e a senha;

Carregue o diretório da aplicação dentro da WebSpaceExplorer;

Configure a base de dados MySQL;

Configure PHP.

Clique no WebSpaceExplorer para carregar o diretório da aplicação Web utilizando o protocolo FTP. Podemos compactar a pasta do projeto, e descompactar dentro do WebSpaceExplorer.

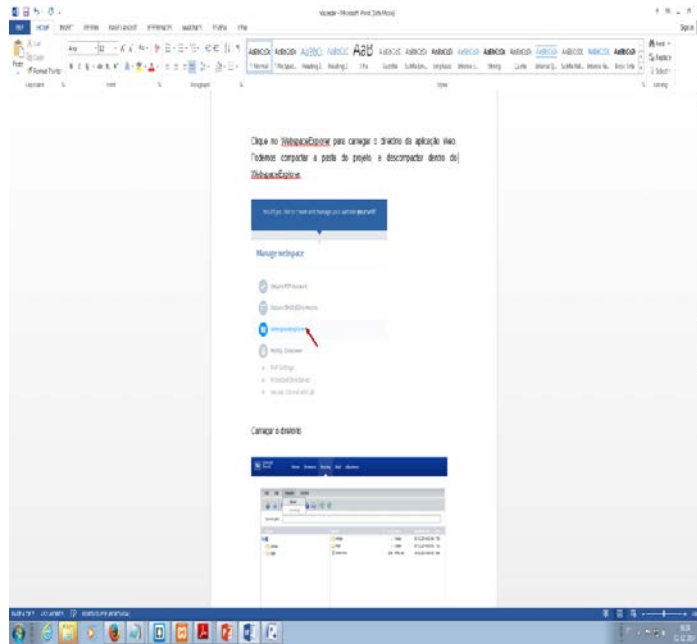


Figura 4.5: WebspacerExplorer

Carregue o diretório da aplicação Web

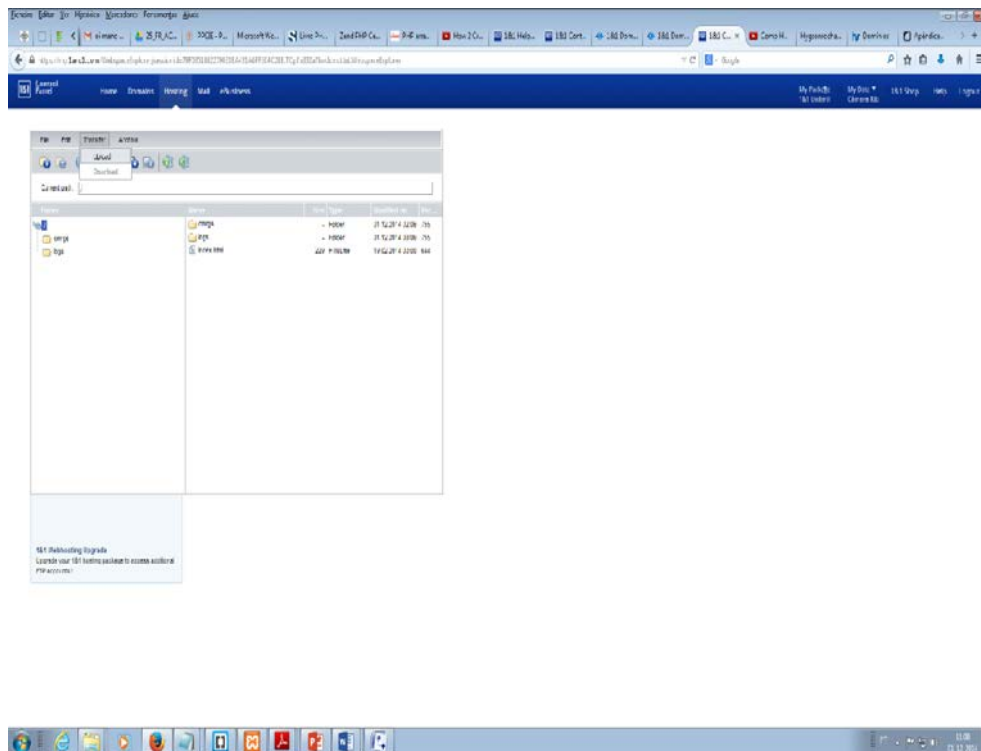


Figura 4.6: Painel de gestão da WebspacerExplorer

Descompactar a o diretório dentro do WebspacerExplorer.

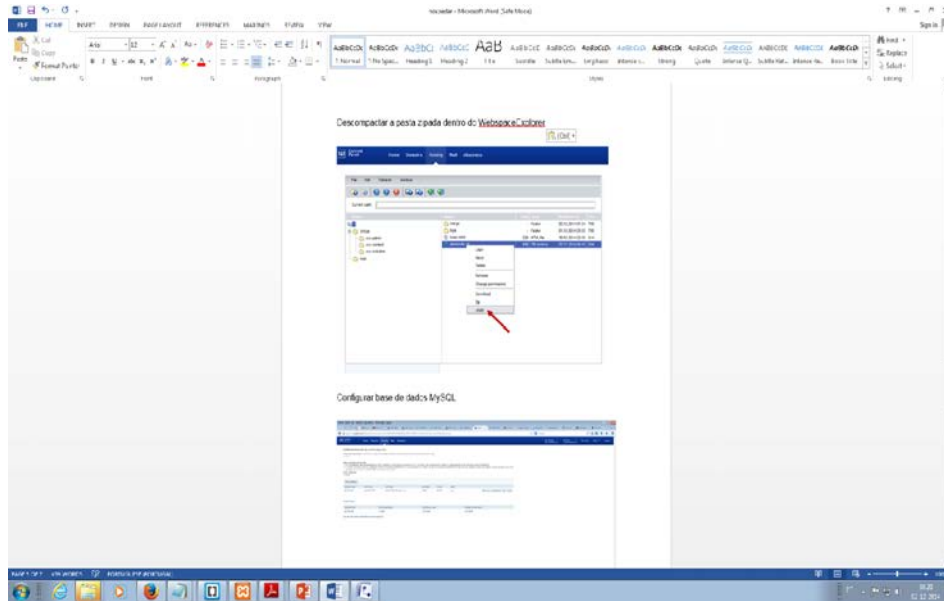


Figura 4.7: Dissipar o ficheiro dentro do Webspacer Explorer

Aponte o nome de domínio para o diretório da aplicação. Clique na opção Domain Center e abrirá a página com o nome de domínio. Veja o exemplo:

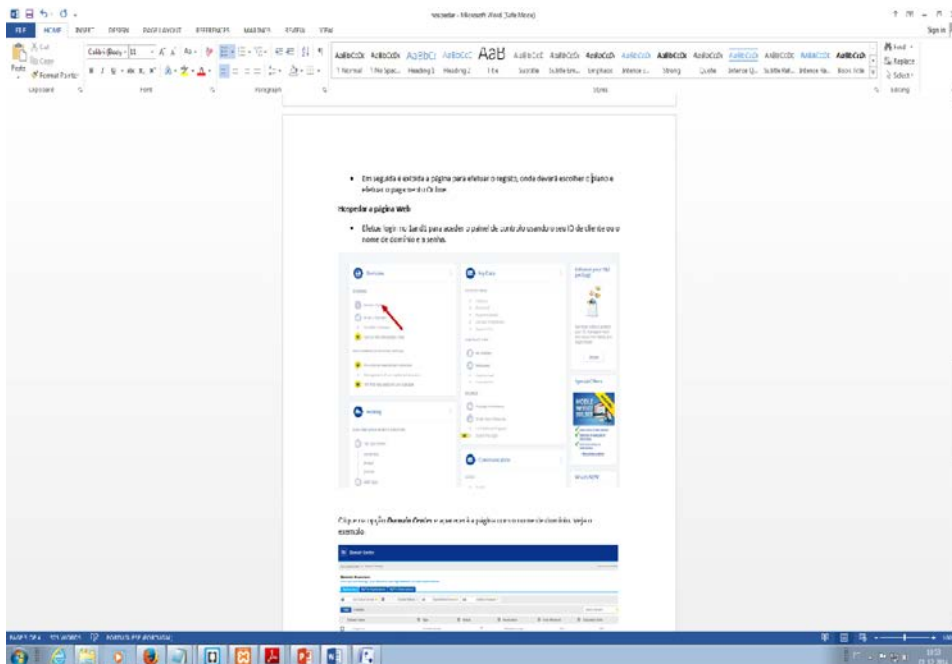


Figura 4.10: Domain Center

Unidade 3: Hospedagem de sites na Web

Na figura abaixo, podemos adicionar um novo domínio, ou pesquisar um domínio existente para hospedar a nossa página Web.

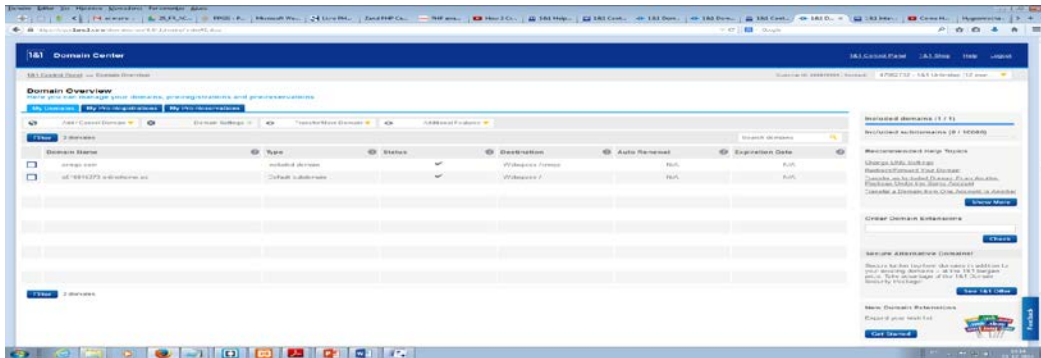


Figura 4.11: Configuração geral

A janela seguinte exibe as configurações básicas para configuração do domínio para a hospedagem da página.

Veja a figura seguinte:

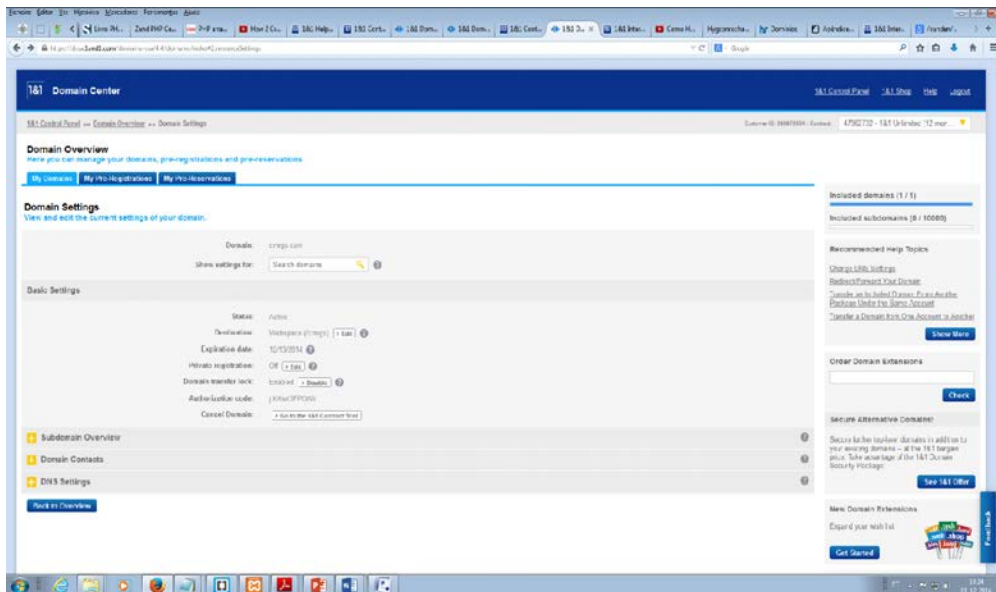


Figura 4.12: Configuração geral

Em seguida, devemos configurar a Webspaces, apontando o nome do domínio (nomedosite.com) para o local (diretório) onde se encontra alojado da sua aplicação Web.

Resumo da atividade: Nesta atividade estudamos como registrar e hospedar uma aplicação na Web, tendo como exemplo prático o provedor 1and1.

Perguntas de autoavaliação

P 1: Quais são os passos necessários para disponibilizar uma aplicação na Web?

P 2: Cite no mínimo três tipos de recursos para o plano de hospedagem?

P 3: Qual é o protocolo utilizado para transferência de ficheiros?

HTTP

FTP

P 4: Quais são a base de dados mais utilizados na Web?

P 5: Selecione o protocolo utilizado para o envio de correio eletrónico:

HTTP

SMP

FTP

P 6: Qual é a diferença entre um servidor DNS e um servidor de email?

Sede da Universidade Virtual africana

The African Virtual University
Headquarters

Cape Office Park

Ring Road Kilimani

PO Box 25405-00603

Nairobi, Kenya

Tel: +254 20 25283333

contact@avu.org

oer@avu.org

Escritório Regional da Universidade Virtual Africana em Dakar

Université Virtuelle Africaine

Bureau Régional de l'Afrique de l'Ouest

Sicap Liberté VI Extension

Villa No.8 VDN

B.P. 50609 Dakar, Sénégal

Tel: +221 338670324

bureauregional@avu.org