

UNIDADE 3 – ARQUIVOS DE INTERNET E MULTIMÍDIA

MÓDULO 1 – ARQUIVOS DE E-MAIL

01

1 - FERRAMENTAS PARA GERENCIAMENTO DE E-MAILS

Olá, seja bem-vindo a mais um módulo de estudos. Na unidade passada vimos conceitos básicos sobre suítes de escritório e arquivos digitalizados. Neste módulo, iremos aprender características dos e-mails e os tipos de arquivos relacionados aos e-mails.

Hoje em dia, enviar e receber *e-mails* faz parte do trabalho de qualquer organização. É praticamente impossível uma organização, mesmo de pequeno porte, não utilizar e-mails para comunicação.

O baixo custo de uso, a facilidade de implementação, alta velocidade de entrega, múltiplos destinatários em um único e-mail, acesso fácil e comum, entre outras possibilidades técnica envolvidas são algumas das vantagens de se utilizar o e-mail como sistema de comunicação. Entretanto, o uso de e-mails também tem algumas desvantagens: não há certeza de que o destinatário receberá a mensagem, o tempo em que o destinatário recebe a mensagem pode ser muito além do necessário (mensagem atrasada), a disponibilidade de acesso à Internet em determinados locais pode ser difícil ou até mesmo impossível, acesso indevido por pessoas não autorizadas de mensagens não criptografadas.



Normalmente, há duas formas de uma pessoa enviar e receber *e-mail*: ou por um **software Web/Internet** ou por um **software instalado no seu computador**. Alguns *softwares* oferecem os dois ambientes ao mesmo tempo. O sistema de mensagens da Microsoft, composto pelo servidor Exchange, oferece uma interface Web com as mesmas funcionalidades do *software* Office Ms Outlook, que é instalado localmente. O Ms Outlook faz parte do pacote Office da Microsoft e é hoje o mais famoso dos gerenciadores de *e-mail* corporativo.

02

Quando utilizamos serviços públicos de *e-mail*, como o Gmail ou o Hotmail, normalmente utilizamos as próprias interfaces WEB que esses programas oferecem. Por meio desses *softwares* web é possível realizar todas as operações e configurações do seu *e-mail* e da sua conta de *e-mail*.

Compose Mail

Inbox (3)

Starred

Sent Mail

Drafts (2)

Hiking (3)

Urgent!

12 more...

Chat

Search, add, or invite:

Hiking Fan
Set status here ▾

Call phone

Arielle

Emily

Jason

Michael

Paul

SEARCH MAIL

SEARCH THE WEB

1 - 15 of 15

Jason Cornwell > Please return my stapler - Hi, You seem to have taken my stapler. Please, 1:10 pm

Paul McDonald > Fun Hike Yesterday! - Thanks for the great hike yesterday, it was awesome 1:06 pm

Arielle Reinstein > July 4th weekend - Hi there! I heard you'll be around this weekend and I'd like to 1:06 pm

JS Bach > Tonhalle concert Friday - Hey man, there's a great concert this Friday evening 1:06 pm

Christine Chiu > Hi Hiking, Looking for opinion on my diet/fitness app - Hi Hiking, I bumped into 1:06 pm

Yan Tseytin (2), Draft > Hey there! - I heard you found a great place to go hiking. Let me know when 1:06 pm

Kenneth, me (2) > Group dinner? - Sushi sounds great! On Fri, Mar 25, 2011 at 10:06 AM, Kenneth 1:06 pm

Kenneth, me (2) > Long time! - Hey Ken! Things have been really good! And lunch sounds great! 1:06 pm

Michael Bolognino > This weekend - Hi there. Let's meet up at 8PM tonight for burgers and then I 1:06 pm

Arielle Reinstein > Dipsea trail - When it stops raining I really want to hike the Dipsea Trail again 1:06 pm

Jason Toff > How are you? - Hey there, We haven't spoken in a while. How are you? Wow! 1:06 pm

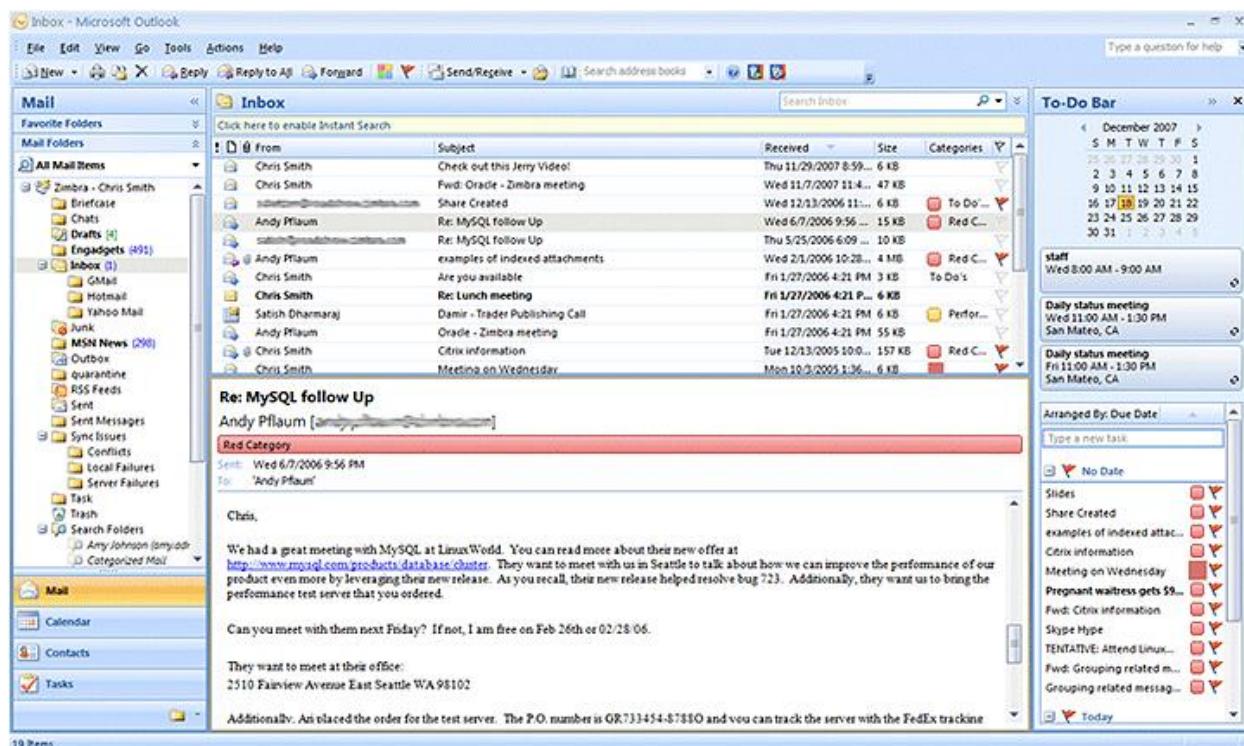
Jr Wikane > VW Auction in Tacoma - Hi, I was doing a search on Google for VWs in Tac 1:06 pm

Giovana Vieira > New unicomm from /R101 R102,5507 at 5:10 AM - Unicomm from /R101 R102,5507 1:06 pm

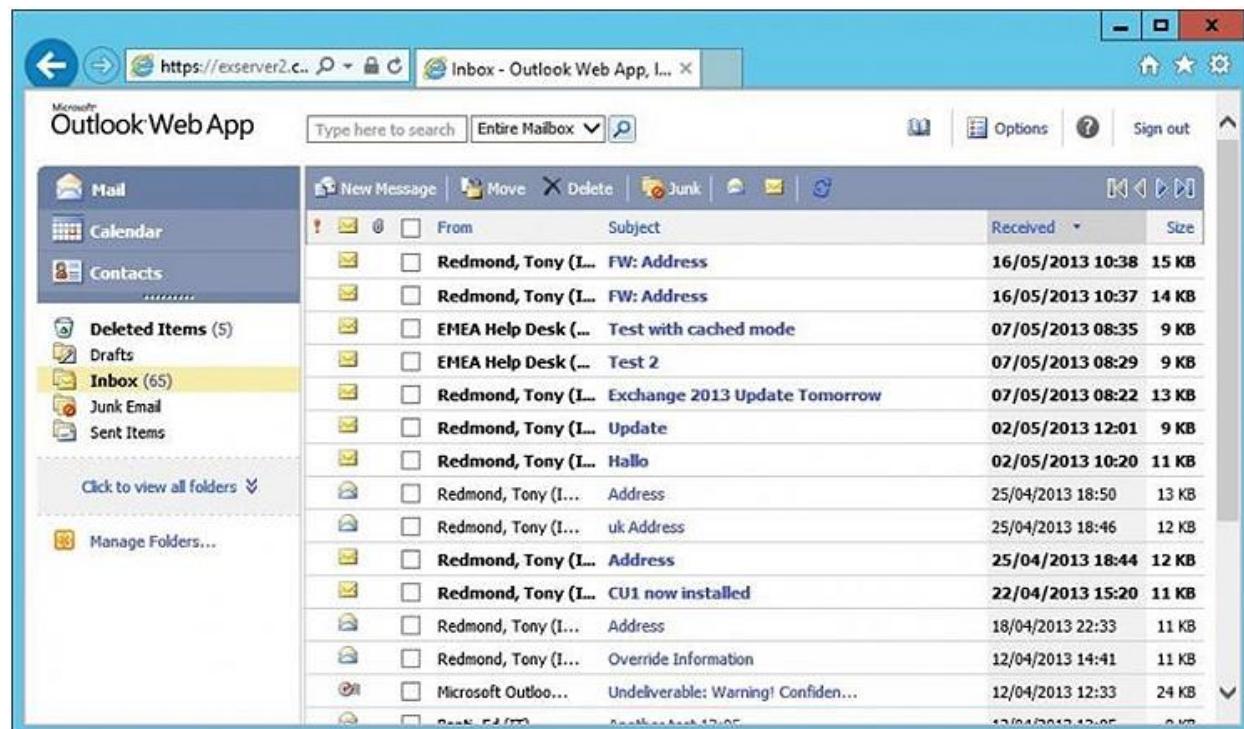
Interface do Gmail

03

Por outro lado, há também diversos *softwares para gerenciar e-mail*, pagos e gratuitos, como o Ms Outlook, Ms Outlook Express, Thunderbird, e outros. Geralmente, os *softwares desktop* não permitem realizar configurações específicas da sua caixa de *e-mail*, mas são apenas uma forma alternativa para poder receber e enviar *e-mails*. Note, abaixo, que as interfaces desktop do Ms Outlook e web (OWA – Outlook Web App) são bastante semelhantes:



Interface desktop do Outlook



Interface do Outlook Web App (OWA)

04

Além disso, quando você está acessando um software WEB, você está acessando a sua caixa postal no servidor de *e-mails*. Já quando você utiliza um *software desktop*, geralmente você está efetuando o *download* dessas mensagens para o seu computador (mas pode também acessar as mensagens no servidor). Veja no quadro abaixo algumas vantagens e desvantagens de cada uma das tecnologias:

	Vantagens	Desvantagens
Sistema WEB	<ul style="list-style-type: none"> • Você pode acessar suas mensagens de qualquer micro ou dispositivo eletrônico. • Permite acessar e configurar todos os parâmetros da sua caixa postal. • Qualquer navegador de Internet que você utilizar para acessar suas mensagens, você sempre verá as mesmas mensagens. • Não precisa instalar <i>software</i> algum, basta um navegador Web. 	<ul style="list-style-type: none"> • Precisa necessariamente de uma conexão Internet para acessar as mensagens. • Seu espaço disponível para armazenamento é limitado pelo tamanho da sua caixa postal.
Sistema Local	<ul style="list-style-type: none"> • Após o <i>download</i> das mensagens, você pode acessá-las sem necessitar estar conectado na Internet. • Você pode criar <i>e-mails</i> enquanto desconectado da Internet e elas serão automaticamente enviadas após nova conexão. • Seu espaço disponível para armazenamento é limitado pelo tamanho do seu HD. 	<ul style="list-style-type: none"> • Não permite realizar configurações de caixa postal. • Precisa sincronizar mensagens para garantir que as mensagens apresentadas são as mesmas do servidor. • Precisa ter um <i>software</i> instalado no computador, tablet ou <i>smartphone</i>.

Comparativo de vantagens e desvantagens de interfaces para gerenciamento de *e-mail*.

05

2 - ARQUIVOS RELACIONADOS A *E-MAILS*

Geralmente, todas as ferramentas de gerenciamento de *e-mail*, seja ela WEB ou desktop, oferecem um meio para salvar *e-mails* no formato de um arquivo no seu disco rígido. Isso é muito interessante, pois permite fazer *backups* e cópias de *e-mails* importantes.

As ferramentas WEB como o Gmail e o Hotmail permitem salvar mensagens no formato HTML. Já as ferramentas desktop geralmente utilizam sistemas de arquivo específicos para salvar *e-mails*. Outras ainda oferecem mais de um formato de arquivo.

O Ms Outlook, por exemplo, trabalha com dois tipos de arquivos:

- PST** → É uma espécie de banco de dados com todas as suas mensagens salvas localmente.
- MSG** → É uma mensagem salva em um arquivo. Um arquivo MSG só pode conter um único e-mail; entretanto, um e-mail pode ter outros e-mails como anexo.

O Outlook também permite salvar *e-mails* nos formatos XML, EML (que é uma espécie de XML), HTML, PDF, XPS, TXT e outros.

O Ms Outlook pode utilizar vários arquivos PST para gerenciar seus *e-mails*. Se dois computadores possuem caixas postais PST locais e apontam para a mesma conta de *e-mail* no servidor, mesmo assim, é bem provável que o conteúdo dos dois arquivos PST seja diferente, pois não há sincronismo entre o arquivo PST e o conteúdo do servidor.

Outras ferramentas como o Lotus Notes e o Firebird possuem conceitos similares com versões diferentes de arquivo para caixa postal e mensagem de *e-mail*.

Arquivos PST

Você pode ter um arquivo PST para cada ano, organizando assim seus *e-mails* antigos por ano. O conteúdo de um arquivo PST geralmente não está sincronizado com os *e-mails* constantes no servidor de *e-mails*. O mais comum é utilizarmos o arquivo PST para guardar localmente as informações recebidas e enviadas pelo servidor. É normal que o usuário move as mensagens do servidor (que tem espaço de caixa postal geralmente pequeno) para o arquivo PST (que tem espaço muito maior para abrigar *e-mails*). Dentro do arquivo PST pode haver milhares de *e-mails* enviados e recebidos, bem como outras informações do Ms Outlook, como seus contatos ou agendamentos.

06

3 - PROGRAMAS QUE ENVIAM E-MAILS

Há muitas formas de os sistemas alertarem ou comunicarem informações a seus usuários. Destas formas, enviar *e-mails* é a mais comum delas. Hoje em dia, é muito comum encontrarmos programas que enviam *e-mails*. Normalmente, esses *e-mails* contêm mensagens de alerta sobre determinada questão. Veja alguns exemplos de uso prático de programas que enviam *e-mails*:

- **Para sinalizar ao administrador do sistema que:**
 - **Ocorreu um erro no sistema;**
 - **Um novo cadastro de usuário está pendente de aprovação;**

- **Parte ou todo o sistema não está funcional.**
- **Para sinalizar aos usuários do sistema que:**
 - **Há alguma informação importante para ler;**
 - **Há uma atividade pendente associada ao usuário;**
 - **Há algo pendente de aprovação pelo usuário;**
 - **Ajuda para recuperar senha ou login;**
 - **Uma nova informação foi cadastrada no sistema e ela parece ser útil ao usuário;**
 - **Informar / comunicar algo relacionado a um sistema.**

Muitas dessas mensagens são automáticas e não temos a opção de simplesmente não aceitá-las ou ignorá-las visto que o bom andamento do programa depende de uma ação sua.

As regras de negócio dos sistemas ditam a necessidades de eventuais comunicações com os usuários, como as citadas anteriormente. Cabe ao programador implementar o sistema de comunicação para que os usuários sejam devidamente comunicados. A arquitetura do sistema e os recursos técnicos disponíveis na organização onde o sistema funcionará definirão características tecnológicas de como o sistema deverá funcionar.

Para um programa ser capaz de enviar *e-mail* é necessário que o programa tenha um serviço de *e-mail* associado a ele. Isso pode ser feito de diversas formas, vamos ver algumas delas:

07

3.1 - Acessando um serviço de *e-mail* da Internet

O seu sistema pode utilizar serviços de *e-mail* disponíveis na Internet para enviar *e-mails*. Para isso, você precisaria, inicialmente, criar uma conta de *e-mail* em um provedor de *e-mail*, como o Hotmail, o Gmail ou outro qualquer.

Depois disso, você deve definir uma tecnologia (protocolo) de envio e recebimento de mensagens, que podem ser **IMAP** e **POP** para recebimento de mensagens e o **SMTP** para envio de mensagens.

IMAP/
SMTP

O protocolo IMAP/SMTP é um protocolo on-line no qual, o seu programa se conecta ao servidor de e-mail, realiza o sincronismo das mensagens (enviadas e recebidas) e após esse processo, mantém a conexão ativa para que as alterações e mensagens novas sejam atualizadas quase que em tempo real. Esse protocolo tem acesso a todas as pastas de sua conta e deixa o status das mensagens igual tanto no servidor como no software (status como “enviada”, “a enviar”, “recebida”, “lida” etc.). Esse tipo de configuração é recomendado para quem precisa do máximo de controle possível sobre as mensagens recebidas e enviadas.

POP/S
MTP

O protocolo POP/SMTP é um protocolo off-line no qual, o seu programa se conecta ao servidor, realiza o envio e o recebimento das mensagens, e após esse processo, finaliza a conexão. Esse protocolo tem acesso apenas à caixa de entrada, não conseguindo baixar, ou ao menos visualizar nenhuma outra pasta de sua conta. Esse é um protocolo mais simples, se comparado ao IMAP.

08

Enquanto que os protocolos IMAP e POP recebem *e-mails*, o protocolo SMTP é o que envia *e-mails*. Por isso os protocolos IMAP e POP aparecem sempre associados ao SMTP. O SMTP sozinho não consegue receber mensagens (somente enviar), da mesma forma que os protocolos IMAP e POP não conseguem enviar mensagens (somente receber). Veja um exemplo.

Para completar, o IMAP/SMTP oferece uma experiência geral mais estável. Enquanto o POP tende a perder mensagens ou fazer o download da mesma mensagem várias vezes, o protocolo IMAP evita isso por meio da capacidade de sincronização dupla entre seu sistema e o provedor de e-mail na Web.

A título de curiosidade, se você tem um celular smartphone ou um tablet que se conecta a uma conta de *e-mail* como o Gmail ou o Hotmail, há 99% de certeza que essa conexão é feita utilizando o protocolo IMAP/SMTP.

É importante que o seu sistema também seja capaz de receber informações, pois muitas vezes é necessário um relacionamento de dupla via (enviar e receber); de toda a forma, se seu sistema for apenas enviar mensagens, basta então definir os parâmetros SMTP, visto que o POP e o IMAP estarão desabilitados.

Exemplo

Quando um sistema “A” envia um *e-mail* para o destinatário “B”, ele usa o protocolo SMTP para enviar. Feito isso, a mensagem ficará no servidor de *e-mails* do destinatário “B”; quando este destinatário for consultar esse *e-mail*, ele utilizará os protocolos POP ou IMAP. Dessa forma, o envio sempre usa SMTP e o recebimento POP ou IMAP.

09

Definida a tecnologia, agora é hora de consultar no seu **provedor de acesso** as configurações para envio e recebimento de mensagens. Quando utilizamos servidores públicos como o Gmail ou o Hotmail, basta

pesquisar na Internet para encontrar as informações de configuração desejáveis. Veja nessas páginas do Gmail e do Hotmail as configurações dos protocolos POP, IMAP e SMTP:

- Servidor de recebimento de e-mails (IMAP) - Requer SSL
 - imap.gmail.com
 - Porta: 993
 - Requer SSL: sim
- Servidor de e-mails enviados (SMTP) - Requer TLS
 - smtp.gmail.com
 - Porta: 465 ou 587
 - Requer SSL: sim
 - Requer autenticação: sim
 - Usar as mesmas configurações do servidor de recebimento de e-mails:
- Nome completo ou Nome para exibição: [seu nome]
- Nome da conta ou Nome de usuário: seu endereço completo do Gmail (nomedeusuário@gmail.com). Usuários do Google Apps devem digitar nomedeusuário@seu_domínio.com
- Endereço de e-mail: seu endereço completo do Gmail (nomedeusuário@gmail.com) Usuários do Google Apps devem digitar nomedeusuário@seu_domínio.com
- Senha: sua senha do Gmail

Configurações do Gmail para uso do IMAP/SMTP.

Servidor de recebimento de e-mails (POP3) - requer SSL:	pop.gmail.com Usar SSL: sim Porta: 995
Servidor de e-mails enviados (SMTP) - requer TLS ou SSL:	smtp.gmail.com Usar autenticação: sim Porta para TLS/STARTTLS: 587 Porta para SSL: 465
Tempos limite do servidor	(Mais que um minuto, recomendamos cinco)
Nome completo ou Nome para exibição:	[seu nome]
Nome da conta ou Nome de usuário:	seu endereço de e-mail completo (incluindo @gmail.com ou @seu_domínio.com)
Endereço de e-mail:	seu endereço de e-mail (nomedeusuário@gmail.com ou nomedeusuário@seu_domínio.com)
Senha:	sua senha do Gmail

Configurações do Gmail para uso do POP/SMTP

- | | |
|---|--|
| <ul style="list-style-type: none"> • Servidor de entrada (IMAP) <ul style="list-style-type: none"> • Endereço do servidor: imap-mail.outlook.com • Porta: 993 • Conexão criptografada: SSL • Servidor de saída (SMTP) <ul style="list-style-type: none"> • Endereço do servidor: smtp-mail.outlook.com • Porta: 25 (ou 587 se a 25 estiver bloqueada) • Autenticação: Sim • Conexão criptografada: TLS • Nome de usuário: seu endereço de email • Senha: sua senha | <ul style="list-style-type: none"> • Servidor de entrada (POP3) <ul style="list-style-type: none"> • Endereço do servidor: pop-mail.outlook.com • Porta: 995 • Conexão criptografada: SSL • Servidor de saída (SMTP) <ul style="list-style-type: none"> • Endereço do servidor: smtp-mail.outlook.com • Porta: 25 (ou 587 se a 25 estiver bloqueada) • Autenticação: Sim • Conexão criptografada: TLS • Nome de usuário: seu endereço de email • Senha: sua senha |
|---|--|

Configurações do Gmail para uso do POP/SMTP

Configurações do Gmail para uso do POP/SMTP

Os parâmetros de configuração provavelmente deverão ficar configurados num arquivo XML ou INI do seu aplicativo. Nunca coloque informações de configuração no código fonte, pois desta forma sempre que houver uma mudança de configuração, você precisará recompilar seu código fonte.

10

Vamos supor que você tenha criado uma conta de e-mail denominada `sigad@gmail.com`, com a senha de acesso “`senha#1234`”. Como seria uma configuração de um arquivo INI para comportar todos os

parâmetros de uma conexão do tipo IMAP/SMTP com o Gmail? Uma sugestão seria a que você vê abaixo.

```

1 ; Estas são as configurações do serviço de e-mail entre o sistema SIGAD e o Gmail:
2
3 [ServidorDeEmail_GERAL]
4 ContaEmail = sigad@gmail.com
5 Senha = senha#1234d
6
7 [ServidorDeEmail_IMAP]
8 ServidorEmail = imap.gmail.com
9 Porta = 993
10 SSL = SIM
11
12 [ServidorDeEmail_SMTP]
13 ServidorEmail = smtp.gmail.com
14 Porta = 465
15 TLS = SIM

```

SIGAD_Config.INI

É claro que, como esse arquivo contém a senha de acesso ao Gmail, é muito importante que ele fique em uma pasta que somente o sistema tem acesso em consultá-lo, nenhum usuário do sistema ou outras pessoas (exceto o administrador da rede) deve ter acesso à pasta onde está esse arquivo de configuração.

11

Da mesma forma, vamos agora pensar que seu sistema utiliza o Hotmail como provedor. Suponha que você tenha configurado o e-mail `sigad@hotmail.com` com a senha “`senha#4321`”, e tenha escolhido o protocolo POP/SMPT. Como seria um arquivo de configuração hipotético no formato XML? Uma sugestão você vê a seguir:

```

1 <ConfiguracoesDeEmail>
2   <ServidorDeEmail_GERAL>
3     <ContaEmail>sigad@hotmail.com</ContaEmail>
4     <Senha>senha#4321</Senha>
5   </ServidorDeEmail_GERAL>
6   <ServidorDeEmail_POP>
7     <ServidorEmail>pop-mail.outlook.com</ServidorEmail>
8     <Porta>995</Porta>
9     <SSL>SIM</SSL>
10  </ServidorDeEmail_IMAP>
11  <ServidorDeEmail_SMTP>
12    <ServidorEmail>smtp-mail.outlook.com</ServidorEmail>
13    <Porta>25</Porta>
14    <TLS>SIM</TLS>
15  </ServidorDeEmail_SMTP>
16 </ConfiguracoesDeEmail>

```

SIGAD_config.XML

É importantíssimo lembrar que na rede em que seu sistema funciona é necessário que as portas de comunicação configuradas para POP, SMTP e IMAP estejam desbloqueadas. Verifique junto ao administrador da sua rede se essas portas estão disponíveis e abertas para troca de e-mails.

3.2 - Outras funcionalidades possíveis ligadas a *e-mail*

Além da função básica de enviar um *e-mail* para uma pessoa, há ainda outras funcionalidades possíveis de se fazer ao programar um código de envio de e-mails utilizando o protocolo SMTP.

Entre as principais funcionalidades, destacamos as possibilidades de:

- Enviar *e-mails* para uma ou várias pessoas ao mesmo tempo;
- Enviar e-mails utilizando as opções de cópia e cópia oculta;
- Anexar um ou vários arquivos;
- Utilizar nomes de pessoas ao invés de endereços de e-mail e validar os nomes no servidor de *e-mail* ou domínio da sua empresa;
- Criar texto de *e-mails* em formato RTF ou HTML, com as possibilidades de inserir gráficos, tabelas, texto formatado etc.;
- Inserir assinaturas digitais (certificado digital);
- Criptografar o conteúdo do *e-mail*.
- E outros.

3.3 - Acessando um serviço de e-mail da sua Rede Corporativa

Outra forma de envio de mensagens é utilizar serviços de e-mail da própria rede corporativa. Quando a empresa possui um computador “servidor de *e-mail*”, ao invés de você um provedor externo, o correto é utilizar o próprio provedor de mensagens.

Um dos serviços de *e-mail* mais comuns que vemos nas empresas é o Microsoft Exchange. O Exchange é uma plataforma de *e-mail* que possui diversos recursos de compartilhamento e sincronização de conteúdo. O Exchange utiliza os protocolos POP ou IMAP e o SMTP.

Além do Microsoft Exchange, outras opções de **servidores de *e-mail* famosos** são:

- Mercury Mail;
- Apache James;
- IBM Lotus Domino;
- Eudora;
- Oracle Communications Messaging Server.

O link a seguir, além das opções já citadas acima, descreve e compara cerca de 50 outros sistemas servidores de *e-mail*: http://en.wikipedia.org/wiki/Comparison_of_mail_servers

Mercury Mail

O Mercury Mail vem junto com o Apache gratuitamente.

Apache James

O Apache James é outro sistema gratuito do Apache.

IBM Lotus Domino

O IBM Lotus Domino é um dos mais antigos sistemas de *e-mail* que existe, criado na década de 80 e utilizado ainda nos dias de hoje.

Eudora

O Eudora também é um sistema com muitos anos de mercado.

Oracle Communications Messaging Server

Oracle Communications Messaging Server é sistema da Oracle com repositório 100% baseado em banco de dados (acessível via comandos SQL).

14

Tanto o Exchange como outros *softwares* de *e-mail* semelhantes oferecem mais de uma forma de um programa poder acessar o serviço, utilizando como base MAPI/POP/SMTP. São **formas alternativas de acesso**:

- Chamando diretamente como um serviço do sistema operacional;
- Acessando o sistema de forma semelhante a um banco de dados, realizando operações SQL de consulta e de cadastro de mensagens;
- Por meio de componentes do sistema operacional;
- Por meio de componentes do serviço de *e-mail*;
- Por meio de componentes da linguagem de programação.

Para o seu programa se **autenticar** no servidor, geralmente há as seguintes alternativas:

- Por meio de um usuário e senha do serviço de *e-mail*;
- Por meio de um usuário e senha do banco de dados;
- Por meio de autenticação de rede;
- Por meio de autenticação LDAP.

Dependendo de como você estruturar seu código fonte para enviar *e-mails*, incluindo os componentes e tecnologias que você escolher, podem ser necessárias algumas outras informações acerca do servidor de *e-mail* da sua organização, como, por exemplo:

- O nome do computador servidor ou o IP do computador servidor; exemplo: SrvMailExchange ou 10.1.100.128;
- O nome do domínio/usuário que enviará mensagens ou o e-mail corporativo desse usuário; exemplo: EmpresaXPTO/sigad ou sigad@empresaxpto.com.br;

Não há uma resposta para qual é o melhor método de acesso, a resposta a esta questão é uma decisão tecnológica que passa sobre as características da sua organização, questões de segurança, maturidade da equipe, necessidades do sistema e linguagem de programação disponível.

15

3.4 - Como um programa envia uma mensagem de *e-mail* para o Exchange

Conforme falamos, há muitos componentes disponíveis para tal tarefa, os mais comuns são:

- **Protocolo SMTP**, com o componente específico para a linguagem de programação que você utiliza. Todos os servidores de e-mail suportam o protocolo SMTP.
- **SendMail**, que é o componente dos mais antigos da Microsoft para envio de mensagens. O SendMail utiliza o protocolo SMTP.
- **CDO.Message** ou **SMTP Client** que são opções de componentes da Microsoft.
- **Microsoft.Office.Interop.Outlook** ou **Outlook.Application**, que são componentes do Outlook (você precisaria ter o Ms Outlook instalado), e que tanto podem ser executados a partir do seu programa como também por Visual Basic Script, que funciona a partir de um script no sistema operacional ou nos programas do Office.
- **System.Web.Mail** ou **System.Net.Mail**, que são componentes do sistema operacional Windows para conexão direta com o Exchange.

- **Exchange WebDAV**, que utiliza o protocolo XML para conexão e troca de mensagens. Utiliza ainda o componente Microsoft XML HTTP para gerenciar a conexão com o Exchange.
- **ExchangeClient** ou **Exchange Web Service**, que são componentes do próprio Exchange Server que utilizam SOAP.

16

Vamos ver três **exemplos hipotéticos em VB.NET** para analisarmos a complexidade do código fonte de cada um deles. Os dois primeiros exemplos utilizam um servidor de Exchange da organização, o terceiro utiliza o Gmail. Não é intenção destes exemplos ensinar você a programar, mas tão somente apresentar opções e analisar a complexidade técnica de cada método.

Exemplo 1 – utilizando System.Net.Mail:

```

1  'Cria objeto com dados do e-mail.
2  Dim objEmail As New System.Net.Mail.MailMessage()
3
4  'Define o Campo From do e-mail.
5  objEmail.From = New System.Net.Mail.MailAddress("SIGAD <sigad@empresaxpto.com.br>")
6
7  'Define os destinatários do e-mail.
8  objEmail.To.Add("Fulano <fulano@empresaxpto.com.br>")
9
10 'Define o titulo do e-mail.
11 objEmail.Subject = "[SIGAD] Você recebeu um alerta!"
12
13 'Define o corpo do e-mail.
14 objEmail.Body = "Atenção, você recebeu uma mensagem de alerta. " & _
15     "Acesse o sistema SIGAD para ver pendências atribuídas no seu nome!"
16
17 'Cria objeto com os dados do SMTP
18 Dim objSmtp As New System.Net.Mail.SmtpClient
19
20 'Alocamos o endereço do host para enviar os e-mails, localhost(recomendado)
21 objSmtp.Host = "ServidorExchangeXPTO"
22
23 'Enviamos o e-mail através do método .Send()
24 objSmtp.Send(objEmail)
25
26 'Excluimos o objeto de e-mail da memória
27 objEmail.Dispose()

```

Exemplo 2 – utilizando ExchangeClient:

```

1 ' Cria uma instância do ExchangeClient
2 Dim client As ExchangeClient = New ExchangeClient("http://EmpresaXPTO/exchange/sigad", _
3     "sigad", "senha#1234", "empresaxpto")
4
5 ' Cria uma instância do objeto MailMessage
6 Dim msg As MailMessage()
7
8 ' Define quem envia a mensagem.
9 msg.From = "SIGAD <sigad@empresaxpto.com.br >"
10
11 ' Define quem recebe a mensagem.
12 msg.To = " Fulano <fulano@empresaxpto.com.br>"
13
14 ' Define o assunto:
15
16 msg.Subject = "[SIGAD] Você recebeu um alerta!"
17
18 ' Define a mensagem:
19 msg.HtmlBody = " Atenção, você recebeu uma mensagem de alerta. " & _
20     "Acesse o sistema SIGAD para ver pendências atribuídas no seu nome! "
21
22 ' Envia a mensagem
23 client.Send(msg)

```

Exemplo 3 – utilizando diretamente o SMTP presente no componente Mail do Visual Basic:

```

1 Dim oSmtp As New EASendMailObjLib.Mail
2 ' O Gmail pede um código de licença, caso a conta seja gratuita, use "TryIt".
3 oSmtp.LicenseCode = "TryIt"
4
5 ' Remetente:
6 oSmtp.FromAddr = "sigad@gmail.com"
7
8 ' Destinatário
9 oSmtp.AddRecipientEx "fulano@empresaxpto.com.br", 0
10
11 ' Assunto
12 oSmtp.Subject = "[SIGAD] Você recebeu um alerta"
13
14 ' Mensagem
15 oSmtp.BodyText = "Atenção, você recebeu uma mensagem de alerta." & _
16     "Acesse o sistema SIGAD para ver pendências atribuídas no seu nome!"
17
18 ' Servidor SMTP do GMAIL
19 oSmtp.ServerAddr = "smtp.gmail.com"
20
21 ' Porta de acesso
22 oSmtp.ServerPort = 465
23
24 ' Configuração SSL/TLS
25 oSmtp.SSL_init
26
27 ' autenticação
28 oSmtp.UserName = "sigad@gmail.com"
29 oSmtp.Password = "senha#1234"
30
31 oSmtp.SendMail()

```

Viu como os métodos são parecidos? Escolher um ou outro é uma opção do seu projeto. Entretanto, por questões de compatibilidade, dê preferência à escolha de componentes que utilizam diretamente o protocolo SMTP como primeira opção de tecnologia. É a forma mais simples, prática, e que certamente terá menos incompatibilidade em relação às outras técnicas.

17

RESUMO

Neste módulo, aprendemos:

- a. Que e-mails podem ser gerenciados por ferramentas Web ou softwares instalados localmente. Em que entre essas duas opções há vantagens e desvantagens de cada uma delas.
- b. Que muitos sistemas enviam informações por e-mail para seus usuários. Portanto, é muito importante que você saiba como escrever uma funcionalidade em um programa que envie e receba e-mails.
- c. Que sistemas enviam informações para usuários comuns e administradores de sistemas com os mais variados motivos.
- d. Que seu programa tanto pode usar um servidor da organização para enviar e-mails como utilizar um servidor público.
- e. Que as tecnologias de protocolo mais comuns são IMAP, POP e SMTP.
- f. Que há softwares que disponibilizam tanto interface WEB quanto interface desktop para acessar e gerenciar e-mails.
- g. Que é possível salvar e-mails em arquivos a fim de fazer backups ou cópias das informações.
- h. Que os sistemas precisam armazenar as informações de configuração de e-mails em arquivos como arquivos INI ou XML.
- i. Que há várias formas de um programa enviar e receber e-mails e que a escolha da tecnologia mais indicada é uma decisão do projeto e da estrutura tecnológica de onde vai funcionar o sistema.

UNIDADE 3 – ARQUIVOS DE INTERNET E MULTIMÍDIA

MÓDULO 2 – ARQUIVOS HTML

01

1 - O QUE É HTML

Olá, seja bem-vindo a mais um módulo de estudos. No módulo passado vimos conceitos básicos sobre a tecnologia utilizada para envio e recebimento de e-mails.

Neste módulo, iremos aprender características sobre os arquivos HTML.

Atenção: para você mesmo fazer testes e verificar os exemplos citados neste módulo, utilizaremos um software livre para edição de arquivos HTML. Apesar de o “Notepad++” permitir a edição de arquivos HTML, o melhor seria um programa específico para edição de arquivos HTML, como o “KompoZer”, disponível gratuitamente neste endereço.

O termo HTML vem do inglês HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto. O HTML é uma linguagem de marcação (assim como o XML também é) destinada a produzir páginas na Web.

Toda vez que acessamos um site, não importa qual, o que vemos no nosso navegador é o resultado de informações em formato HTML.

```

25 </head>
26 <body text="#000000"
27 bgcolor="#FFFFFF">
28 <table width="1000"
29   <tr>
30     <td width="200"
31     <td valign="top"
32       <div align="center"
33         <h1>Boa
34           <h2>Cap
35             <h3>Entertainment</a>
36           | <a href=
37

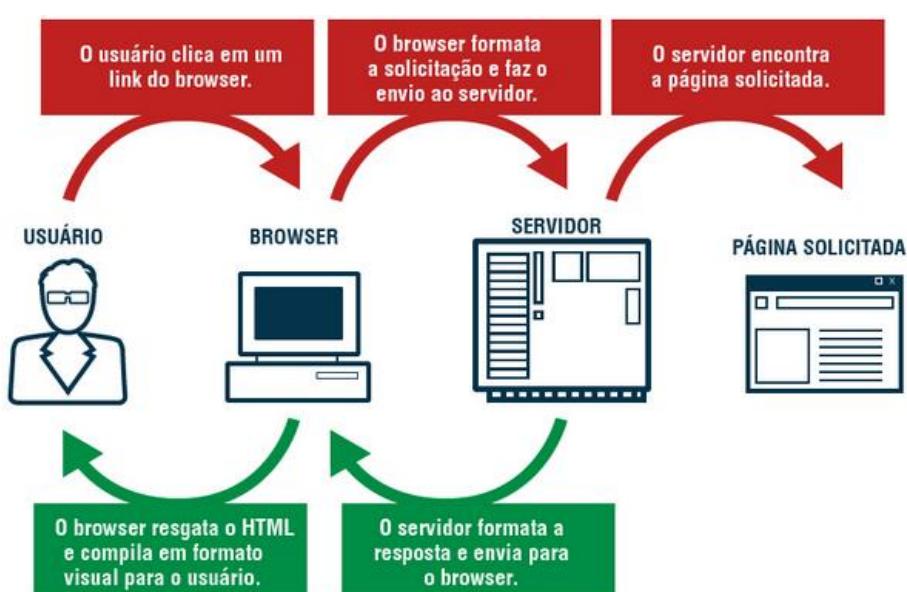
```

Trecho de código HTML

02

Quando acessamos um site, este site não envia para nosso navegador um arquivo para ser aberto, mas sim informações que são transmitidas pela rede de dados e entre no nosso computador por meio de um protocolo denominado HTTP.

Já dentro do computador, o sistema operacional direciona as informações para o *software* navegador (que interpreta o conteúdo HTML vindo pelo protocolo HTTP), que então apresenta o conteúdo na tela.



O conteúdo original poderia não ser um arquivo HTML, mas sim um programa Web que gerou um código HTML para o usuário final. Mesmo assim, a página que vemos na tela pode ter seu conteúdo salvo em um arquivo local para futuro acesso. A forma nativa de salvar o conteúdo é o formato HTML. Entretanto, também podemos salvar o conteúdo de uma página de Internet em formato PDF por exemplo. Porém, somente o formato HTML preserva todas as informações originais. Exemplo.

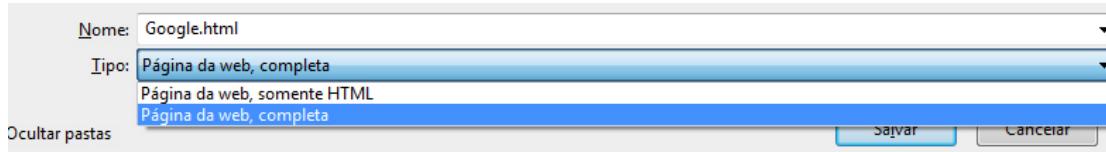
Exemplo

Se você salvar um arquivo no formato PDF ele terá as dimensões exatas de uma página impressa (formato A4 retrato, por exemplo), já em formato HTML o conteúdo pode se adequar a qualquer tamanho de tela, formato ou resolução.

03

Importante saber que **o HTML só contém o conteúdo textual e a formatação da página**. Imagens, arquivos multimídia, controles (como componentes, estilos etc.) não podem ser salvos dentro do arquivo HTML. Dessa forma, o que o navegador pode fazer ao salvar uma página que contém essas outras informações é criar, além do arquivo HTML, uma pasta no mesmo local do arquivo HTML e lá efetuar o *download* das diversas informações complementares.

Vamos fazer um teste. Acesse no seu navegador a página do Google. Localize a opção para salvar a página (normalmente o atalho é CTRL+S). Observe que ao salvar duas opções serão apresentadas no tipo de arquivo: a) página da web, somente HTML e b) página da web, completa.



Duas opções para salvar o arquivo: somente HTML e completa

Salve o arquivo no seu desktop ou em outro lugar de sua preferência utilizando a primeira opção. Salve novamente, mudando o nome do arquivo e escolhendo a segunda opção. Observe agora, onde você salvou esses dois arquivos, que deverá conter dois arquivos e uma pasta, similar à imagem abaixo:



Arquivos e pasta salvos em formato HTML.

04

Veja que quando você optou por salvar apenas o HTML, só um arquivo foi criado. Já, ao optar pela opção “completa”, também foi criada uma pasta com arquivos. Vamos dar uma olhada no que tem dentro desta pasta:

Nome	Data de modificação	Tipo	Tamanho
be07e20879b03cf1.js	04/11/2014 09:35	Arquivo de script ...	54 KB
cb=gapi.loaded_0	04/11/2014 09:35	Arquivo LOADED_0	137 KB
rs=ACT90oGw7fZ2eTtGIsbc2ajqeYGq4BhpKw	04/11/2014 09:35	Arquivo	366 KB
rs=ACT90oGw7fZ2eTtGIsbc2ajqeYGq4BhpKw(1)	04/11/2014 09:35	Arquivo	127 KB
rs=AltRSTNT00OfQdiIPYIJLIDiqcfbJvPbgA	04/11/2014 09:35	Arquivo	14 KB
rs=AltRSTPjMQJR5VYoEGS7-LmRLn5bNa9E6A	04/11/2014 09:35	Arquivo	158 KB
rs=AltRSTPjMQJR5VYoEGS7-LmRLn5bNa9E6A(1)	04/11/2014 09:35	Arquivo	26 KB
tia.png	04/11/2014 09:35	IrfanView PNG File	1 KB

Arquivos contidos na pasta “google2_files”

Perceba que a pasta contém uma série de arquivos, como um arquivo em JavaScript, outros arquivos de código de programação, e um arquivo de imagem. São essas informações que transformam a página do Google em algo dinâmico, que permite você escrever e realizar pesquisas, a clicar em botões e ações serem executadas.

Essas informações não são originalmente HTML, mas sim componentes a parte que transformam a página em uma página com conteúdo de programação embutido nela.

05

2 - MARCAÇÃO (TAGS) E ESTRUTURA BÁSICA DO ARQUIVO HTML

Todo conteúdo HTML possui marcações (também chamadas de **tags** ou **etiquetas**). Essas marcações são feitas pelos sinais “maior que” e “menor que” (“<” e “>”). Cada tipo de marcação representa um comando diferente. Por exemplo, caso você queira que uma frase seja formatada em negrito, dentro do HTML haverá algo como “este é um exemplo de texto em negrito ”. A marca “” indica que a partir dali, todo o conteúdo à direita estará em negrito; a marca “” indica que ali termina a formatação em negrito.

Há vários tipos de marcações, para texto em negrito, itálico e sublinhado, para o tamanho da letra, para alinhamento do texto, para criação de listas e tabelas, e muito mais.

Outro comando simples é o direcionamento a um link. Veja o exemplo:

Clique aqui e você será direcionado para o Google.

Observe nesse pequeno trecho o seguinte conteúdo:

- <a> e posteriormente indica a abertura e o fechamento de uma instrução do tipo hyperlink.
- href é um atributo que o hyperlink será para uma página web.
- =http://www.google.com> indica para qual url será direcionado.
- “Clique aqui e você será direcionado para o Google” é o texto que aparecerá formatado para hyperlink (provavelmente em azul e sublinhado).

- indica o encerramento do comando.

Os marcadores são necessários para definir comandos e delimitar partes do conteúdo HTML que obedecerão a esses comandos. Logo a seguir iremos aprender e exercitar alguns comandos HTML.

06

A primeira linha do arquivo HTML se chamada **DOCTYPE**. O Doctype avisa aos navegadores, robôs de busca, leitores de tela e outras coisas que tipo de documento é aquele. Existem outros códigos que podemos carregar, por exemplo XML. Por isso o Doctype avisa aos outros sistemas para que ele saiba como se comportar ao ler o código.

Depois começamos com a tag (marcação) HTML. Isso quer dizer que tudo o que estiver entre as tags

```

1.  <!DOCTYPE html>
2.
3.  <html lang="pt-br">
4.  <head>
5.  <meta charset="utf-8">
6.  <title>Título da página</title>
7.  </head>
8.  <body>
9.  ... aqui vai todo o código HTML que faz seu site...
10. </body>
11. </html>

```

<html></html> é escrito em HTML. Ao lado da palavra HTML tem um atributo chamado “lang”, que indica qual o idioma do texto. Esse atributo facilita a tradução por sistemas tradutores e o ajuste de caracteres acentuados.

Logo após a tag <html> temos a tag <head>. Na tag “head” nós indicamos o título do documento e a tabela de caracteres que o

browser deve usar para apresentar seu texto. De acordo com os padrões de idioma, os caracteres podem possuir formatos especiais, daí a necessidade de especificar o código do idioma.

A tag <title> é muito importante. É com ela que você indica o título do documento. O Google e outros sistemas de busca utilizam essa tag para indicar em suas buscas o título da página. Isso é muito importante para que o documento apareça bem representado nas buscas.

Logo depois da tag de fechamento </head> começamos a tag <body>. Dentro deste elemento é que vamos escrever todo o código HTML do conteúdo da página Web.

07

3 - EDIÇÃO DE DOCUMENTOS HTML

Um dos principais problemas no desenvolvimento para internet é a mistura dos diversos códigos. Nós não usamos apenas o HTML para fazer sites. Além do HTML, utilizamos ainda:

- o **CSS**, que é uma linguagem para configurarmos o visual das páginas,
- o **Javascript**, que vai cuidar do comportamento da página (por exemplo, o que acontece quando o usuário clica em um botão).

Há também as linguagens de programação, como PHP, Python, Ruby, ASP, C# etc. Essas linguagens fazem tudo funcionar. Elas fazem os cálculos nos servidores e dão a resposta para o navegador do usuário.

Para que os códigos não se misturem, nós os separamos em diversas camadas. Para ficar mais fácil de entender, imagine que o HTML é sempre o esqueleto do site. É com ele que vamos fazer toda a estrutura de código, onde iremos dizer o que é um título, o que é um parágrafo, uma imagem etc.

O CSS será a parte externa do corpo. É o que deixará o esqueleto bonito. É com o CSS que iremos dar cor para o título, configurar o tamanho do texto, largura das colunas etc. Marcações como `<script></script>`, e outros delimitam o código HTML de outros códigos.

08

Os documentos em HTML são arquivos de texto simples que podem ser criados e editados em qualquer editor de textos comum, como o Bloco de Notas do Windows, ou o TextEdit, do Macintosh. Para facilitar a produção de documentos, existem editores HTML específicos, com recursos sofisticados, que facilitam a realização de tarefas repetitivas, inserção de objetos, elaboração de tabelas e outros recursos. Basicamente, os softwares editores dividem-se em três tipos:

Editor de texto comum	→	Não insere e não controla nenhum código ou marcação. Tudo deve ser feito de modo manual pelo usuário (muito difícil de se usar na prática).
Editor de texto fonte	→	Inserem automaticamente as etiquetas, orientando a inserção de atributos e marcações. Auxilia a criação dos códigos, mas o resultado final só aparece quando salvamos o arquivo e abrimos seu conteúdo em um navegador.
Editores WYSIWYG	→	Oferecem ambiente de edição com um "esboço" resultado final das marcações, exatamente como apresentado em um navegador (o mais fácil de trabalhar de todos).

WYSIWYG é o acrônimo da expressão em inglês "**What You See Is What You Get**", cuja tradução remete a algo como "**O que você vê é o que você obtém**". Significa a capacidade de um programa de computador de permitir que um documento, enquanto manipulado na tela, tenha a mesma aparência de sua utilização, usualmente sendo considerada final a forma impressa. O uso inicial do termo foi relacionado a editores de texto, agora, porém, é aplicado a qualquer tipo de programa, como, por exemplo, programas de *web design*. Dois exemplos clássicos de editores WYSIWYG são o Writer e o Microsoft Word, nos quais o documento é mostrado na tela da mesma forma que será impresso.

09

3.1 - Código limpo versus código poluído

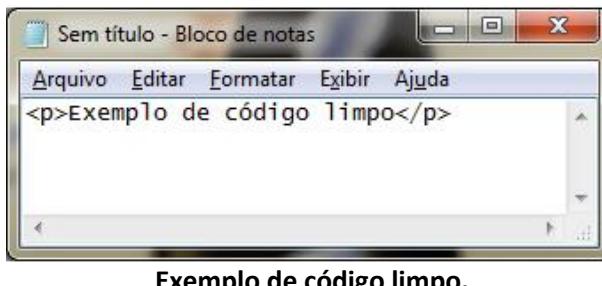
Quando uma pessoa escreve o código HTML por meio de um editor de texto comum ou editor de texto fonte, ele tem total controle sobre as marcações HTML que são inseridas no documento. Já quando ele utiliza um editor WYSIWYG, ele não tem, a princípio, controle sobre quais informações o programa está inserindo no código HTML. Portanto, é possível que o programa esteja inserindo muito mais informação que o necessário, e de certa forma poluindo o código HTML.

Sendo assim, é primordial que você, ao utilizar um *software* com capacidade WYSIWYG, utilize um bom *software* e observe sempre o código HTML que ele está criando, para evitar um código poluído. O Ms Word é um *software* WYSIWYG, entretanto, o código HTML que ele gera é extremamente poluído.

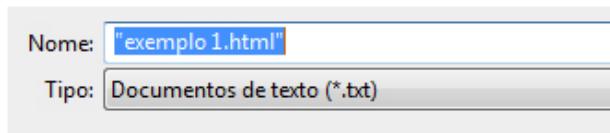
Vamos ver um exemplo de código HTML limpo. Abra o bloco de notas do Windows ou um editor de texto puro (como o Notepad++), escreva “

Exemplo de código limpo

”, salve o arquivo como “exemplo 1.html” (utilize as aspas para salvar, caso você esteja utilizando o bloco de notas do Windows).



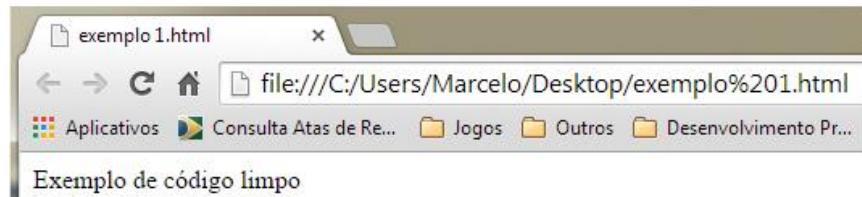
Exemplo de código limpo.



Use aspas duplas para salvar o documento no bloco de notas.

10

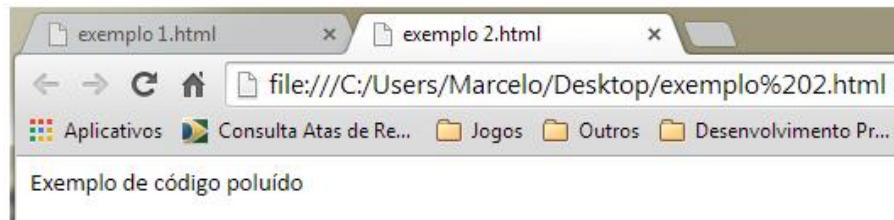
Localize o arquivo salvo, clique duas vezes e veja o conteúdo dele no formato HTML de dentro do seu navegador:



Arquivo “exemplo 1.html” sendo mostrado no navegador.

Veja como o navegador exclui as marcações `<p>` e `</p>` e mostra o conteúdo final da tela.

Vamos agora ver um exemplo de código HTML poluído. Abra Ms Word, escreva “Exemplo de código poluído”, salve o arquivo como “exemplo 2.html” (em tipo de arquivo, escolha “página da Web HTML”). Localize o arquivo salvo, clique duas vezes e veja o conteúdo dele no formato HTML de dentro do seu navegador:

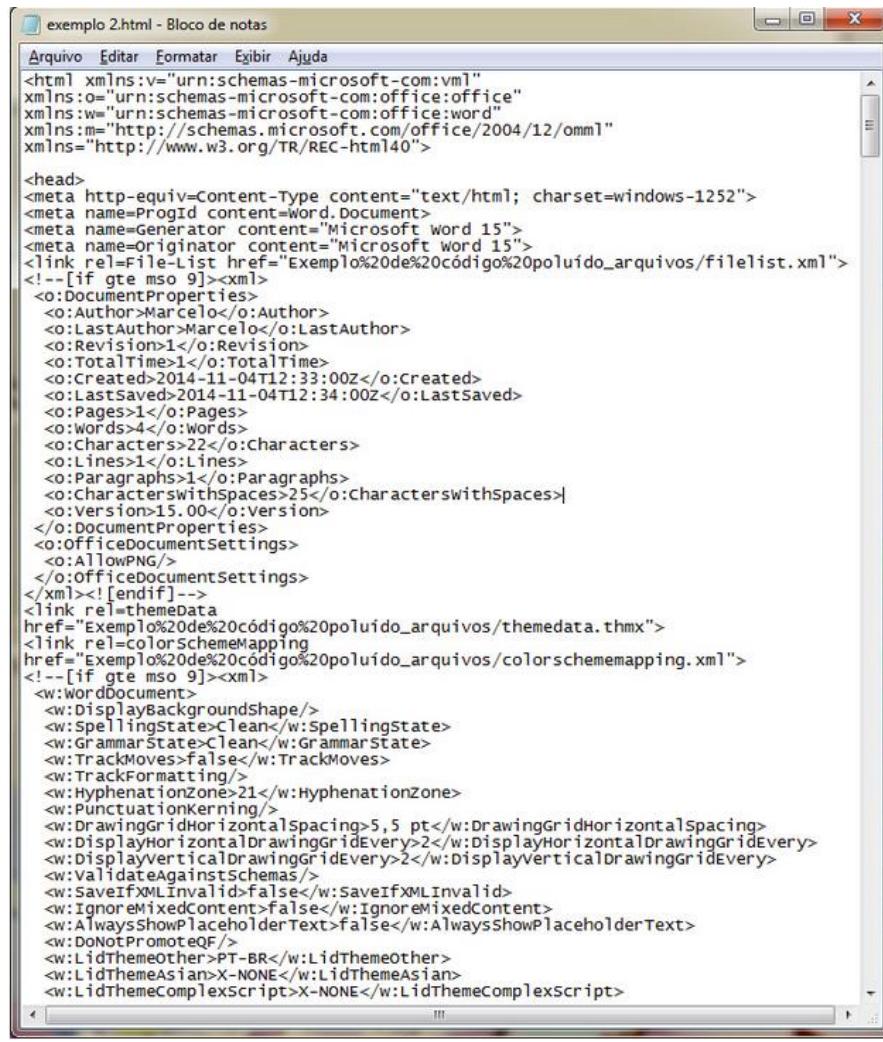


Arquivo “exemplo 2.html” sendo mostrado no navegador.

Observe que o resultado no arquivo “exemplo 2.html” é o mesmo do “exemplo 1.html”. Agora, qual será o conteúdo do arquivo “exemplo 2.html”? Veja a seguir.

11

Abra o bloco de notas ou outro editor de texto puro, localize o arquivo “exemplo 2.html” e abra-o. Veja o que tem dentro dele! Algo semelhante ao da imagem abaixo:



```

<html xmlns:v="urn:schemas-microsoft-com:vm"
      xmlns:o="urn:schemas-microsoft-com:office:office"
      xmlns:w="urn:schemas-microsoft-com:office:word"
      xmlns:m="http://schemas.microsoft.com/office/2004/12/omml"
      xmlns="http://www.w3.org/TR/REC-html140">

<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-1252">
<meta name=ProgId content="Word.Document">
<meta name=Generator content="Microsoft Word 15">
<meta name=originator content="Microsoft Word 15">
<link rel=File-List href="Exemplo%20de%20poluído_arquivos/filelist.xml">
<!--[if gte ms 9]>xml>
<o:DocumentProperties>
  <o:Author>Marcelo</o:Author>
  <o:LastAuthor>Marcelo</o:LastAuthor>
  <o:Revision>1</o:Revision>
  <o:TotalTime>1</o:TotalTime>
  <o:Created>2014-11-04T12:33:00Z</o:Created>
  <o:LastSaved>2014-11-04T12:34:00Z</o:LastSaved>
  <o:Pages>1</o:Pages>
  <o:Words>4</o:Words>
  <o:Characters>22</o:Characters>
  <o:Lines>1</o:Lines>
  <o:Paragraphs>1</o:Paragraphs>
  <o:Characterswithspaces>25</o:Characterswithspaces>
  <o:Version>15.00</o:Version>
</o:DocumentProperties>
<o:OfficeDocumentSettings>
  <o:AllowPNG/>
</o:OfficeDocumentSettings>
</xml><!--[endif]-->
<link rel=themedata href="Exemplo%20de%20poluído_arquivos/themadata.thmx">
<link rel=colorschemeMapping href="Exemplo%20de%20poluído_arquivos/colorschememapping.xml">
<!--[if gte ms 9]>xml>
<w:WordDocument>
  <w:DisplayBackgroundShape/>
  <w:SpellingState>Clean</w:SpellingState>
  <w:GrammarState>Clean</w:GrammarState>
  <w:TrackMoves>false</w:TrackMoves>
  <w:TrackFormatting/>
  <w:HyphenationZone>21</w:HyphenationZone>
  <w:PunctuationKerning/>
  <w:DrawingGridHorizontalSpacing>5,5 pt</w:DrawingGridHorizontalSpacing>
  <w:DisplayHorizontalDrawingGridEvery>2</w:DisplayHorizontalDrawingGridEvery>
  <w:DisplayVerticalDrawingGridEvery>2</w:DisplayVerticalDrawingGridEvery>
  <w:ValidateAgainstSchemas/>
  <w:SaveIfXMLInvalid>false</w:SaveIfXMLInvalid>
  <w:IgnoreMixedContent>false</w:IgnoreMixedContent>
  <w:AlwaysShowPlaceholderText>false</w:AlwaysShowPlaceholderText>
  <w:DoNotPromoteQF/>
  <w:LidThemeOther>PT-BR</w:LidThemeOther>
  <w:LidThemeAsian>X-NONE</w:LidThemeAsian>
  <w:LidThemeComplexScript>X-NONE</w:LidThemeComplexScript>
</w:WordDocument>

```

Exemplo de código HTML poluído.

Observe que uma simples linha com o texto “Exemplo de código poluído” é transformada pelo Ms Word nesse gigantesco código HTML. Dessa forma, podemos concluir que o Ms Word não é um bom editor de HTML, visto que o código HTML gerado por ele é um código poluído.

12

4 - SOFTWARE EDITOR HTML

Um bom exemplo de software editor de HTML, que produz um resultado limpo, é o KompoZer. O KompoZer possui duas interfaces, uma WYSIWYG e uma de código HTML.

→ Baixe, instale e abra o KompoZer; passada a tela de apresentação, escreva: exemplo de código limpo.

Observe agora que o programa apresenta três botões na parte central abaixo:



Botões de visualização do KompoZer.

Esses botões permitem você ver o código em formato WYSIWYG ou código HTML.

Clique na opção do meio, “Split”. Veja que a tela é dividida em duas, e na parte de baixo o programa apresenta o mesmo conteúdo de cima, mas no formato HTML:

```

<body>
  exemplo de código limpo

</body>
  
```

Conteúdo no formato HTML limpo.

13

Na parte de cima, vamos inserir os seguintes comandos:

1. No menu, clique em “Insert” e depois em “Table”
2. Na tela que aparece, posicione o mouse na posição de duas linhas e duas colunas, e clique com o mouse.
3. Na tela do programa, escreva o nome de quatro pessoas em cada uma das células da tabela.
4. Posicione o mouse no final da tabela, clique e mantenha pressionado o botão do mouse no quadrado branco entre as duas linhas, arraste para a esquerda, diminuindo assim o tamanho da tabela.

Sua tela deverá parecer com a da imagem abaixo:

exemplo de código limpo	
Pedro	Paulo
João	José

Texto e tabela preenchida.

Na barra de botões, clique agora na opção “Source” e veja como está seu código HTML até o momento.

14

Seu código HTML deverá estar algo parecido com o da imagem a seguir:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
  http-equiv="content-type">
  <title></title>
</head>
<body>
  exemplo de código limpo<br>
  <table style="text-align: left; width: 294px; height: 60px;" border="1"
  cellpadding="2" cellspacing="2">
    <tbody>
      <tr>
        <td style="vertical-align: top;">Pedro<br>
        </td>
        <td style="vertical-align: top;">Paulo<br>
        </td>
      </tr>
      <tr>
        <td style="vertical-align: top;">João<br>
        </td>
        <td style="vertical-align: top;">José<br>
        </td>
      </tr>
    </tbody>
  </table>
  <br>
</body>
</html>

```

Tela “Source” mostrando o código HTML formatado.

Observe na imagem acima algumas instruções HTML, como:

- A primeira linha que indica o padrão de HTML utilizado no documento;
- A marcação `<html>` que indica que o conteúdo abaixo é HTML (e não um script ou programação PHP por exemplo);
- A marcação `<head>` que indica, no cabeçalho, o padrão de caracteres acentuados que estamos utilizando;
- A marcação `<title></title>` que indica o título do documento HTML, que aparece no nosso navegador quando abrimos o documento. Veja que no momento nosso documento não tem título.
- Posicione o mouse entre essas marcações e escreva “este é um exemplo de HTML”. Ao final, a linha deve estar assim: `<title>este é um exemplo de HTML </title>`
- Depois vemos a marcação `<body>`, que indica o corpo do documento, ou seja, o conteúdo em si.
- Dentro de `<body>`, encontramos marcações como `<tbody>`, indicando que ali virá uma tabela.
- Logo a seguir, vemos `<tr>` que indica linhas de tabela e `<td>` que indica colunas.
- Códigos como `style="vertical-align: top;"` que indicam o posicionamento e o alinhamento das informações dentro da tabela.
- E ao final, diversas marcações de encerramento (fechamento do conteúdo) com as marcas de `</>` sinalizando o fechamento.

15

4.1 - Exercício prático

Vamos aprender na prática alguns comandos básicos de edição de texto. Clique no botão “Design”. Observe que o programa possui uma barra de formatação fácil de entender. Na barra, vemos opções

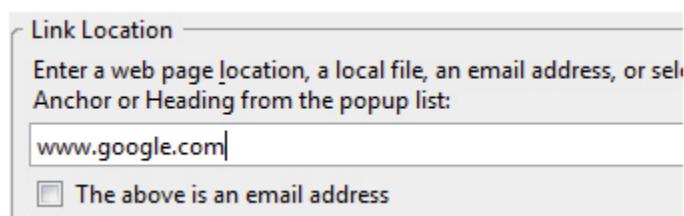
para tamanho de fonte, negrito, itálico, sublinhado, alinhamento de texto, listas numeradas, listas pontuadas, identação de texto, e outros.



Ainda, no menu “Insert”, temos opções para inserir imagens, links e outros. A partir da tela de “design”, execute as seguintes operações:

1. Clique no botão de negrito (letra “B” na barra de botões”, e digite: teste em negrito, pressione enter.
2. Desmarque o botão negrito, clique no botão de itálico (letra “I”), e digite: teste em itálico, pressione enter.
3. Pressione agora negrito, itálico e sublinhado, e digite: teste em negrito com itálico e sublinhado, pressione enter.
4. Clique no botão de lista numerada () e digite o nome de quatro frutas. Clique novamente no botão de lista numerada para cancelar a operação.
5. Clique no botão de lista de pontos () e digite o nome de quatro cores. Clique novamente no botão de lista numerada para cancelar a operação.
6. Digite “clique aqui para acessar o google”, marque o texto com o mouse, clique agora no menu em “Insert” e “link”. Na janela que abriu, digite “www.google.com” e clique em ok. Clique em outro local da tela para desmarcar o texto.

16



Tela “Source” mostrando o código HTML formatado.

7. Digite “este texto possui letras grandes”, marque o texto com o mouse e clique duas vezes no botão de aumentar o tamanho da letra ().
8. Digite “texto em vermelho”, marque o texto, pressione o botão de cor do texto (a caixinha preta) (), escolha a cor vermelha e clique em ok. Desmarque o texto e veja que ele ficou em vermelho.

9. Digite “texto alinhado no centro”, marque o texto, clique no botão de alinhamento central ().

10. Digite “Título nível 1”, marque o texto, e clique na caixa de tipo de texto ( ) e mude para “Heading 1”.

11. Digite “texto usando a fonte verdana”, e na opção de fonte ( ), clique e mude para “verdana”.

17

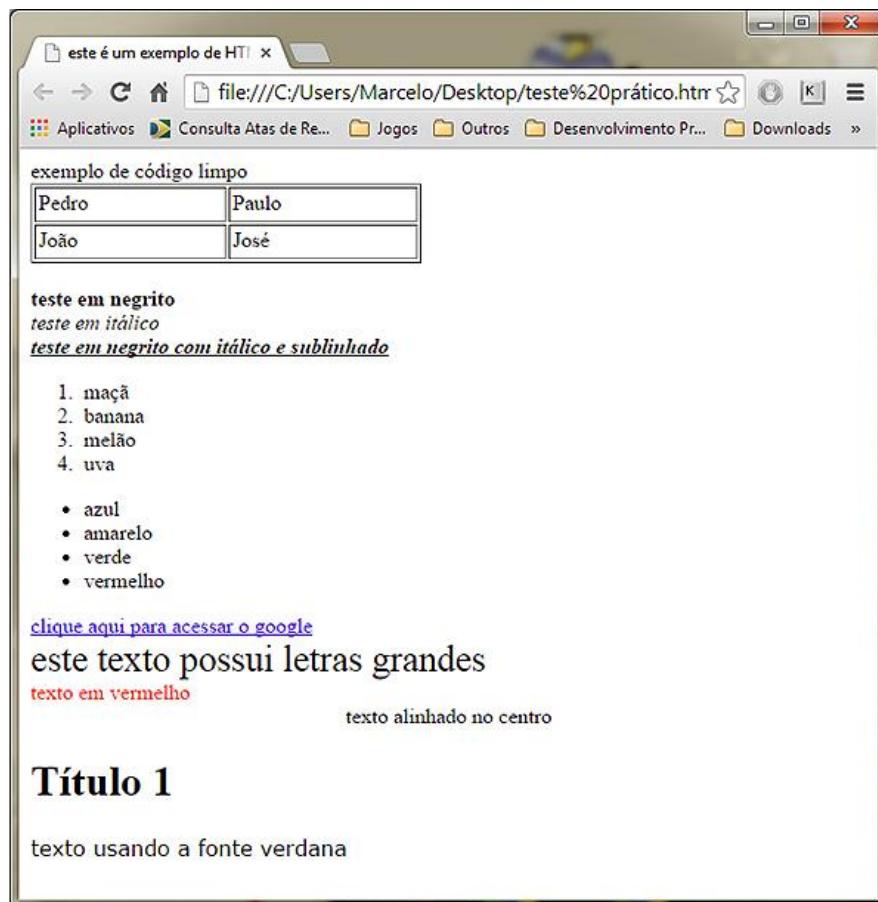
12. Clique no botão de “split”, na parte inferior da tela. Agora, vá marcando com o mouse pequenos pedaços do que você escreveu e veja na parte inferior como é o código HTML para o trecho que você marcou:

 clique aqui para acessar o google
 este texto possui letras grandes

 `clique aqui para acessar o google`



13. Clique no botão de salvar (), salve o arquivo como “teste prático”. Clique no botão HTML (), e o conteúdo será mostrado no seu navegador padrão.



Arquivo que criamos sendo visualizado no navegador.

18

Volte agora ao *software KompoZer*, clique no botão “source” (parte inferior da tela) e veja como ficou todo o código HTML do seu arquivo. Dê uma pequena analisada no conteúdo do arquivo para você ver os comandos de cada trecho do arquivo:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN "http://www.w3.org/TR/html4/strict.dtd">
<body>
exemplo de código limpo<br>
<table style="text-align: left; width: 294px; height: 60px;" border="1"
cellpadding="2" cellspacing="2">
<tbody>
<tr>
<td style="vertical-align: top;">Pedro<br>
</td>
<td style="vertical-align: top;">Paulo<br>
</td>
</tr>
<tr>
<td style="vertical-align: top;">João<br>
</td>
<td style="vertical-align: top;">José<br>
</td>
</tr>
</tbody>
</table>
<br>
<span style="font-weight: bold;">teste em negrito</span><br>
<span style="font-style: italic;">teste em itálico<br>
<span style="text-decoration: underline;"><span
style="font-style: italic;"><span style="font-weight: bold;">teste em
negrito com itálico e sublinhado<br>
</span></span></span></span>
<ol>
<li>maçã</li>
<li>banana</li>
<li>melão</li>
<li>uva</li>
</ol>
<ul>
<li>azul</li>
<li>amarelo</li>
<li>verde</li>
<li>vermelho</li>
</ul>
<a href="www.google.com">clique aqui para acessar o google</a><br>
<big><big><big>este texto possui letras grandes</big></big></big><br>
<span style="color: red;">texto em vermelho</span><br>
<div style="text-align: center;">texto alinhado no centro<br>
</div>
<h1>Título 1</h1>
<span style="font-family: Verdana;">texto usando a fonte verdana</span><br>
<span style="font-weight: bold;"><br>
</span>
</body>
</html>

```

Conteúdo do arquivo no formato HTML

Concluímos que ao utilizar um bom editor de HTML, nem precisamos saber a fundo o que significa cada comando HTML, pois produzimos o conteúdo de forma fácil, assim como produzidos documentos comum utilizando ferramentas como o Ms Word.

19

5. ESTRUTURA DO HTML

Um documento HTML válido precisa seguir obrigatoriamente a estrutura composta pela instrução `<!DOCTYPE>` e pelas tags `<html>`, `<head>` e `<body>`.

5.1 - A instrução DOCTYPE

A instrução DOCTYPE não é uma tag HTML, mas uma instrução especial que deve ficar na primeira linha do arquivo HTML. Ela indica para o navegador qual versão do HTML deve ser utilizada para apresentar o conteúdo da página. O padrão mais simples é utilizar <!DOCTYPE html>, que indica para o navegador a utilização da versão mais recente do HTML - a versão 5, atualmente.

Há muitos comandos complicados nessa parte de DOCTYPE que eram usados em versões anteriores do HTML. Hoje em dia, nada disso é mais necessário. O recomendado é sempre usar a última versão do HTML, usando a declaração de DOCTYPE simples:

```
<!DOCTYPE html>
```

Entretanto, ao utilizar *softwares* de edição de HTML, pode ser que você veja expressões de apontem para um formato ou padrão específico. No caso do KompoZer, que por padrão utiliza a versão HTML 4, a linha doctype que ele cria é assim:

Para fins deste módulo, não se preocupe em alterar esse valor. Quando você for criar um programa Web, seu *software* de programação irá configurar essa linha para você.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Conteúdo do arquivo no formato HTML

20

5.2 - A tag <html>

Na estrutura do nosso documento, antes de tudo, inserimos uma tag <html>. Dentro dessa tag, é necessário declarar outras duas tags: <head> e <body>.

Essas duas tags são "irmãs", pois estão no **mesmo nível hierárquico** em relação à sua tag "pai", que é <html>.

```
<html>
    <head></head>
    <body></body>
</html>
```

21

5.3 - A tag <head>

A tag <head> contém informações sobre nosso documento que são de interesse somente do navegador, e não das pessoas que leem o conteúdo da página HTML. São informações que não serão exibidas na área do documento no navegador.

A especificação obriga a presença da tag de conteúdo `<title>` dentro do nosso `<head>`, permitindo especificar o título do nosso documento, que normalmente será exibido na barra de título da janela do navegador ou na aba do documento.

Outra configuração muito utilizada, principalmente em documentos cujo conteúdo é escrito em um idioma como o português (que tem caracteres acentuados e cedilha), é a configuração da codificação de caracteres, chamado de encoding ou charset.

Podemos configurar qual codificação queremos utilizar em nosso documento por meio da configuração de charset na tag `<meta>`. Um dos valores mais comuns usados hoje em dia é o UTF-8, também chamado de Unicode. Há outras possibilidades, como o latin1, muito usado antigamente.

O UTF-8 é a recomendação atual para encoding na Web por ser amplamente suportada em navegadores e editores de código, além de ser compatível com praticamente todos os idiomas do mundo.

```
<html>
  <head>
    <title>Mirror Fashion</title>
    <meta charset="utf-8">
  </head>
  <body>
  </body>
</html>
```

22

5.4 - A tag `<body>`

A tag `<body>` contém o corpo do nosso documento, que é exibido pelo navegador em sua janela. É necessário que o `<body>` tenha ao menos um elemento "filho", ou seja, uma ou mais tags HTML dentro dele.

```
<html>
  <head>
    <title>Este é o título da nossa página</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Esta frase tem a formatação título 1</h1>
    <p>Esta frase tem a formatação simples de um parágrafo</p>
  </body>
</html>
```

Nesse exemplo, usamos a tag `<h1>`, que indica um título, a tag `<p>`, que indica um parágrafo.

5.5 - A tag `
`

A tag `
` indica uma quebra de linha. É utilizada quando queremos fazer com que o texto a seguir fique abaixo do texto atual. Veja o exemplo abaixo:

```
<body>
  <p>Esta frase está numa linha<br>
  Já esta frase está em outra linha<br></p>
</body>
```

23

5.6 - As tags <h1><h2>...<h6>

Textos geralmente contêm capítulos, cada capítulo pode ter vários subníveis. Da mesma forma, uma página HTML pode conter títulos e subtítulos. Os níveis de cada marcação representa uma profundidade hierárquica, ou seja, o nível mais alto é o h1 (representa, por exemplo, o capítulo 1). O próximo nível é o h2 (representa, por exemplo, o capítulo 1.1). O próximo nível é o h3 (representa, por exemplo, 1.1.1). Em HTML é possível até 6 níveis de profundidade hierárquica. Para ficar mais claro veja o exemplo hipotético abaixo:

O uso do nível correto, além de influenciar no tamanho padrão de exibição do texto, também tem impacto nas ferramentas que processam HTML. As ferramentas de indexação de conteúdo para buscas, como o Google, Bing ou Yahoo! levam em consideração essa ordem e relevância. Os navegadores especiais para acessibilidade (deficientes visuais) também interpretam o conteúdo dessas tags de maneira a diferenciar seu conteúdo e facilitar a navegação do usuário pelo documento.

Nível Hierárquico	Correspondente HTML
Cap 1	h1
Cap 1.1	h2
Cap 1.2	...h2
Cap 1.2.1	h3
.....Cap 1.2.2	h3
Cap 2	h1
Cap 2.1	h2

24

5.7 - A tag <p>

A tag <p> indica um parágrafo de texto. É a marcação indicada para colocarmos um texto comum. Cara parágrafo pode ser separado de duas formas: a) ou com inserção de uma quebra de linhas (tag
), que não é recomendado; ou b) com a criação de várias sequências <p> e </p>. Veja o exemplo abaixo:

```
<p>Um parágrafo de texto.</p>
<p>Outro parágrafo de texto.</p>
```

5.8. Tags de ênfase

Quando queremos dar uma ênfase diferente a um trecho de texto, podemos utilizar as marcações de ênfase. Podemos deixar um texto "mais forte" com a tag `` ou deixar o texto com uma "ênfase acentuada" com a tag ``. Também há a tag `<small>`, que diminui o tamanho do texto.

Por padrão, os navegadores ajustarão o texto dentro da tag `` em negrito e o texto dentro da tag `` em itálico. Existem ainda as tags `` e `<i>`, que atingem o mesmo resultado visualmente, mas as tags `` e `` são mais indicadas por definirem nossa intenção de significado ao conteúdo, mais do que uma simples indicação visual. Veja o exemplo abaixo que produz o mesmo resultado:

```
<p>Olá, o meu nome é <strong>Marcelo</strong>.</p>
<p>Olá, o meu nome é <b>Marcelo</b>.</p>
```

Resultado do uso das tags `` e ``

Olá, o meu nome é **Marcelo**.
Olá, o meu nome é **Marcelo**.

25

5.9 - As tags `` e `<figure>`

A tag `` define uma imagem em uma página HTML e necessita de dois atributos preenchidos: "src" e "alt". O primeiro aponta para o local da imagem e o segundo, um texto alternativo para a imagem caso essa não possa ser carregada ou visualizada.

O HTML 5 introduziu duas novas tags específicas para imagem: `<figure>` e `<figcaption>`. A tag `<figure>` define uma imagem com a conhecida tag ``. Além disso, permite adicionar uma legenda para a imagem por meio da tag `<figcaption>`.

```
<figure>
  
  <figcaption>'O universo é pequeno'</figcaption>
</figure>
```

5.10 - Tag de comentário `<!-- -->`

Quando iniciamos um projeto, utilizamos poucas tags HTML. Como desenvolvimento das páginas, conteúdo e outros arquivos, nosso projeto atinge um tamanho razoável de elementos que pode gerar uma certa confusão. Para manter o código mais legível, é recomendada a adição de comentários antes da abertura e após o fechamento de tags estruturais (que conterão outras tags). Dessa maneira, nós podemos identificar claramente quando um elemento está dentro dessa estrutura ou depois da mesma.

```
<!-- início do cabecalho -->
<div id="header">
  <p>Esse parágrafo está "dentro" da área principal.</p>
</div>
<!-- fim do cabecalho -->

<p>Esse parágrafo está "depois" da área principal.</p>
```

5.11 - A estrutura dos arquivos de um projeto

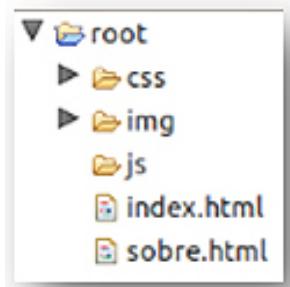
Como todo tipo de projeto de *software*, existem algumas recomendações quanto à organização dos arquivos de um site. Não há nenhum rigor técnico quanto a essa organização e, na maioria das vezes, você vai adaptar as recomendações da maneira que for melhor para o seu projeto.

Como um site é um conjunto de páginas Web sobre um assunto, empresa, produto ou qualquer outra coisa, é comum todos os arquivos de um site estarem dentro de uma só pasta e, assim como um livro, é recomendado que exista uma "capa", uma página inicial que possa indicar para o visitante quais são as outras páginas que fazem parte desse projeto e como ele pode acessá-las, como se fosse o índice do site.

Esse arquivo de índice é um padrão adotado pelos servidores de páginas Web. Se desejamos que uma determinada pasta seja servida como um site e dentro dessa pasta existe um arquivo chamado `index.html`, esse arquivo será a página inicial a menos que alguma configuração determine outra página para esse fim. Exemplo

Dentro da pasta do site, no mesmo nível que o `index.html`, é recomendado que sejam criadas mais algumas pastas para manter separados os arquivos de imagens, as folhas de estilo CSS e os scripts. Para iniciar um projeto, teríamos uma estrutura de pastas como a demonstrada na imagem.

Muitas vezes, um site é servido por meio de uma aplicação Web e, nesses casos, a estrutura dos arquivos depende de como a aplicação necessita dos recursos para funcionar corretamente. Porém, no geral, as aplicações também seguem um padrão bem parecido com o que estamos adotando para o nosso projeto.



Exemplo de estrutura de arquivos de um site

Exemplo

Quando digitamos <http://www.google.com.br>, na verdade, dentro do servidor, estamos abrindo o seguinte endereço e arquivo: <http://www.google.com.br/index.html>.

26

6 - CRIANDO UM PROGRAMA PARA LER E INTERPRETAR CÓDIGO HTML

É pouco usual, mas há programas que leem conteúdo de páginas HTML e de arquivos HTML. Esses programas têm por objetivo analisar a estrutura do arquivo ou identificar (capturar) informações contidas no HTML. A forma mais tradicional de ler um arquivo HTML é semelhante à forma com que lemos arquivos XML: por um componente interpretador de HTML (um exemplo seria o WebBrowser). Esses componentes conseguem separar estrutura de conteúdo, e assim permitir diversos tipos de análise desejados por um programa.

Devido à raridade desse tipo de necessidade, bem como da grande semelhança com o tratamento de arquivos XML (já citados anteriormente), não vamos aqui tratar de códigos de programação sobre esse tema. Caso seja do seu interesse, pesquise na Web “como interpretar código HTML com visual basic”. Se desejável, substitua a expressão Visual Basic por outras linguagens de programação do seu interesse.

Para saber mais sobre html

Há na Web centenas e centenas de apostilas, cursos e livros sobre HTML, gratuitos e pagos. Consulte no Google por “apostila grátis html”, por exemplo, para ver muito material sobre o assunto.

27

RESUMO

Neste módulo, aprendemos:

- a. Que o HTML é o modo de estruturar páginas Web. Todas as páginas Web são formatadas em HTML.
- b. Que o HTTP é o protocolo padrão de transporte de páginas HTML.
- c. Que o HTML utiliza tags de marcação para identificar comandos e formatos.
- d. Que a primeira linha do arquivo HTML é a DOCTYPE que identifica o tipo de arquivo HTML.
- e. Que há editores HTML que geram código sujo, poluído, e por isso devem ser evitados.
- f. Há diversas marcações em HTML, é importante saber o significado delas, mas um bom editor de texto HTML irá inserir as marcações para você automaticamente.

UNIDADE 3 – ARQUIVOS DE INTERNET E MULTIMÍDIA

MÓDULO 3 – ARQUIVOS DE ÁUDIO

01

1 - TIPOS DE ÁUDIO

Olá, seja bem-vindo a mais uma etapa do nosso estudo. Neste módulo, iremos aprender características sobre os arquivos de áudio.

O primeiro som digital que um computador emitiu foi um bip. Naquela época, os computadores levavam muito tempo para ligar e carregar o sistema em memória. Esses computadores também não possuíam tela (monitor). Luzes e bips informavam aos operadores que o computador estava pronto para uso ou que o processamento havia sido concluído.

De lá pra cá muita coisa mudou, hoje os computadores são capazes de sintetizar a voz humana com perfeição, criar músicas imitando qualquer tipo de instrumento musical, gravar e modelar sons e voz, enfim, temos no computador um completo estúdio de som.

Há três tipos de formato de áudio relacionados com o computador:

Áudio analógico	Áudio digitalizado (ou encoded)	Áudio 100% digital
É aquele que está fora do ambiente do computador ou de um equipamento digital (como um CD). Saiba+	É o áudio analógico que foi “gravado” e “convertido” em áudio digital. Saiba+	Foi criado dentro do computador ou por meio de equipamentos digitais, como um teclado de música (sintetizador) ou mesmo um softwares intetizador. Saiba+

O áudio analógico é armazenado em fitas magnéticas (como as antigas fitas K7 e fitas rolo) e LPs. O áudio digital/digitalizado é armazenado em arquivos de computador em formato binário ou em equipamentos digitais como CDs, DVDs e outros.

Áudio Analógico

Quando ouvimos uma TV aberta analógica, rádio FM e AM, também são sinais de áudio analógico. O áudio analógico pode ser criado por equipamentos analógicos, como instrumentos musicais não eletrônicos, voz humana, ruídos da natureza, etc. Pode também ser armazenado em equipamentos analógicos, como gravador em LPs, fitas K7, gravadores portáteis e outros. O áudio analógico é sujeito a várias interferências, que geram ruídos, zumbidos, chiados, distorções e falhas. Para esse áudio ser enviado para o computador, ele precisa ser digitalizado (processo esse também chamado de codificação ou encode).

Áudio Digitalizado

Um CD de música é um exemplo de áudio digitalizado. Quando o cantor grava sua música em um estúdio de música, cada componente musical é gravado individualmente por equipamentos que digitalizam e convertem o som da voz do cantor e dos instrumentos musicais em som digital. Posteriormente, esses sons são misturados para formar a música e então criar o CD de música. O processo de conversão do som analógico para digital acarreta uma perda, e é sabido que o som digital nunca poderá representar o som analógico de maneira plena. No entanto, a evolução tecnológica dos processos de conversão atingiu um grau elevado de precisão ao ponto de não deixar transparecer nenhuma distinção perceptível ao ouvido humano entre o som analógico e sua representação digital. Há equipamentos que não exigem áudio digitalizado de alta qualidade; exemplo: quando utilizamos um celular, nossa voz é digitalizada com baixa qualidade e transmitida pelas linhas telefônicas até a outra pessoa, onde o som é reproduzido.

Áudio 100% digital

Esses tipos de som são livres de interferência (zumbido, distorção etc.).

02

1.1 - Tipo de contêiner versus tipo de codificação

Em se tratando de áudio, dois parâmetros importantes devem ser levados conta:

O primeiro deles é o **tipo de contêiner**.

O **contêiner** é simplesmente o tipo de arquivo (extensão) que guarda as informações do áudio.

Os contêineres mais famosos são o WAV, o MP3 e o MID. Mas existem dezenas de tipos de contêineres, são alguns deles: AAC, CDA, FLAC, RMI, OGG, AIF, AU, VOC, WMA, ASF, M3U, e outros tantos. Cada contêiner tem uma determinada característica, utilidade e qualidade sonora.

O segundo parâmetro é o **tipo de codificação** (codec).

A **codificação** é o método pelo qual o som foi digitalizado ou digitalmente construído.

Há codificadores que compactam o conteúdo (e geralmente há uma leve perda de qualidade nessa compactação, muitas vezes imperceptível aos nossos ouvidos), outros codificadores não compactam o conteúdo, e assim geram arquivos bastante grandes.

03

2 - TIPOS DE ARQUIVOS (CONTÊINERES)

2.1 - Arquivo WAV (ou Wave)

O padrão WAV foi um dos primeiros formatos de áudio que existiu. Ele foi criado pela Microsoft e pela IBM com o objetivo de armazenar informações de áudio em computadores. O WAV é um contêiner que permite alguns tipos de codificação, sendo o PCM o mais comum.

Como o arquivo WAV utiliza principalmente áudio não comprimido (sem perdas) esse é o formato preferido para gravar sons profissionais de alta qualidade / alta fidelidade (HI-FI). O áudio em formato WAV pode ser editado e manipulado com relativa facilidade usando softwares. Por ser um formato sem compressão, o WAV ocupa um espaço muito grande de armazenamento.

O formato WAV é limitado a arquivos menores de 4 GB, devido ao uso de inteiros de 32 bits para gravar o campo de tamanho no cabeçalho de arquivo. Apesar disto, um arquivo WAV pode conter aproximadamente 6.6 horas de áudio em qualidade de CD (44.1 kHz, 16-bit estéreo). Neste padrão, o arquivo WAV consome 10,1 Mbytes por minuto de gravação.

Informações técnicas detalhadas sobre o formato WAV podem ser encontradas aqui:
<http://pt.wikipedia.org/wiki/WAV>.

04

2.2 - Arquivo MP3

O MP3 basicamente é um **formato de compressão de áudio**. Ficou popular por ser um formato de compressão com perdas de qualidade quase imperceptíveis ao ouvido humano.

O método de compressão com perdas empregado na compressão do MP3 consiste em retirar do áudio tudo aquilo que o ouvido humano normalmente não conseguiria perceber, devido a fenômenos de mascaramento de sons e de limitações da audição humana (embora pessoas com ouvido absoluto possam perceber tais perdas). O ouvido humano consegue captar frequências de 20 Hz até 20 kHz.

O formato padrão de arquivos MP3 é de 128 kbps, que representa cerca de 90% de redução do arquivo de som (quando comparado com o WAV/PCM), ou seja, cada minuto de áudio em formato MP3 ocupa apenas 1 MByte de espaço em disco.

O máximo de qualidade de um arquivo MP3 é 320 kbps (cerca de 2,3 MBytes por minuto de áudio). Ainda, temos os formatos intermediários de 192 kbps e 256 kbps, cuja escolha depende da relação custo-benefício desejada, onde o tamanho do arquivo pode ser reduzido em detrimento da qualidade/fidelidade do som. Um CD-R comum pode armazenar mais de 12 horas de áudio em MP3 de 128 kbps.

05

2.3 - Arquivo MID (ou MIDI)

MIDI (Musical Instrument Digital Interface) na verdade não é um tipo de arquivo de áudio. É um protocolo de comunicação entre um instrumento eletrônico e um módulo ou um *hardware*. Ele geralmente é utilizado para definir componentes do sistema, como “controlador MIDI”, “interface MIDI”, “conector MIDI” etc.

O arquivo MIDI não contém dados de áudio, e sim instruções de como produzi-lo. Instrumentos, notas, ritmos, intensidades etc., como se fosse uma partitura musical. Todas essas instruções serão lidas por um sintetizador (*software* ou *hardware*) para que sejam executadas.

O Windows possui um pacote de sons MIDI. É por este motivo que ele pode ser executado no Windows Media Player, como se fosse um arquivo de áudio. Porém, se você utilizar este mesmo arquivo MIDI em um sintetizador, os sons gerados serão os desses equipamentos. Ou seja, a música pode ser a mesma, mas o som será diferente. O formato MIDI também não comporta voz humana, somente instrumentos eletrônicos. Muitas pessoas produzem música por meio de softwares sintetizadores MIDI.

Um exemplo de utilização do MIDI é um teclado de piano digital ligado ao PC, que por sua vez executa algum software de sintetização, transformando os toques nesse teclado (que não possui som nenhum, geralmente) em sons de Pianos e outros instrumentos.

06

3 - TIPOS DE CODIFICADORES

Há três tipos de codificadores (codecs):

1. Sem compressão

Nestes codecs as informações são armazenadas exatamente como são recebidas, então os arquivos gerados são bem grandes, pois não há compressão de informações. Algumas vantagens de codecs que não usam compressão de dados é que eles exigem pouco poder de processamento e são ótimos para edição. A principal desvantagem é que os arquivos gerados por estes codecs ocupam muito espaço. Exemplo: PCM e AIFF.

2. Compressão sem perdas

Com estes codecs é diferente. As informações são comprimidas e os arquivos gerados são bem menores do que os originais. Uma grande vantagem deles é que nenhuma informação é perdida, então quando você quiser usar outro codec a qualidade será preservada, pois a informação processada pelo decodificador é a mesma recebida pelo codificador. A média de compactação desses codecs é de 40 a 60%. Exemplo: APE, M4A, FLAC, WMA e Squish. Uma tabela comparativa de codecs sem perda pode ser vista aqui: <http://wiki.hydrogenaud.io/index.php?title=Lossless>.

3. Compressão com perdas:

Neste tipo de codec parte das informações recebidas pelo codificador são descartadas para diminuir o tamanho do arquivo. Bons codificadores descartam informações que geralmente não são percebidas pelos sentidos humanos. Estes são os codecs mais utilizados hoje, mas não são muito usados para uso profissional. Os arquivos gerados por estes codecs são geralmente os menores e o poder de processamento requerido é quase sempre maior que os requeridos pelos codecs sem-compressão. Quando o codec é muito complexo, chega a necessitar mais poder de processamento que os codecs onde há compressão sem perdas. A média de compactação desses codecs é de 25 a 90% Exemplo: MP3, AAC, OGG e Vorbis.

07

3.1 - Codificador PCM

Modulação por Código de Pulso ou em inglês **Pulse-Code Modulation (PCM)** é um método usado para codificar sinais analógicos, representando as ondas de som em formato de sinal digital.

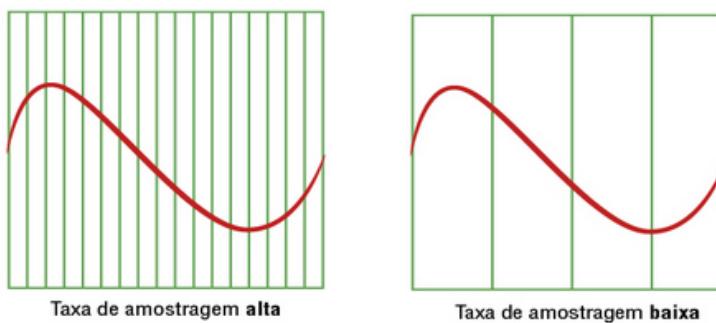
É a forma padrão para áudio digital em computadores e em vários formatos de Blu-ray, DVD e Discos Compactos (Compact Disc - CD), e também outros usos, tal como os sistemas digitais de telefones.

Um fluxo de PCM é uma representação digital de um sinal analógico. Há no PCM duas propriedades básicas que determinam a sua fidelidade (qualidade) ao sinal analógico original:

- a **taxa de amostragem**, que é o número de vezes por segundo que amostras são tomadas;

- e a **profundidade de bit**, que determina o número de possíveis valores digitais que cada amostra pode tomar.

Entender como o PCM funciona não é difícil, mas exige de você um pouco de conhecimento sobre ondas sonoras. Sabemos que o som é uma onda e, mesmo misturando vários tipos de sons, ainda será uma onda. O trabalho de digitalizar um som analógico depende de sabermos como gravar as informações dessa onda. Veja no gráfico a onda em vermelho. Ela representa uma onda sonora sendo que o eixo X representa o tempo e o eixo Y representa a amplitude da onda.



Para podermos digitalizar essa onda, primeiro precisamos fatiar o tempo. Quanto mais fatias de tempo tivermos, maior será a qualidade.

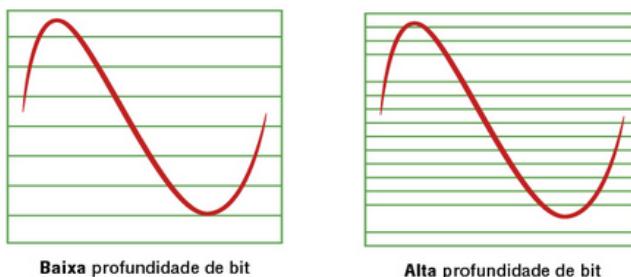
A “velocidade” ou o “período de tempo” em milissegundos em que fatiamos a onda é a taxa de amostragem, ela está representada no eixo X do gráfico pelas linhas verticais azuis.

Taxa de amostragem

Uma taxa de amostragem de 10 Hz significa que faremos 10 medidas em um segundo. Uma taxa de amostragem de 1.000 Hz representa 1.000 medidas em um segundo. Quanto maior a taxa de amostragem (maior frequência), mais valores serão identificados para calcular a onda digitalizada, consequentemente, mais qualidade o som terá. Por outro lado, como iremos computar mais informação, também teremos um arquivo maior.

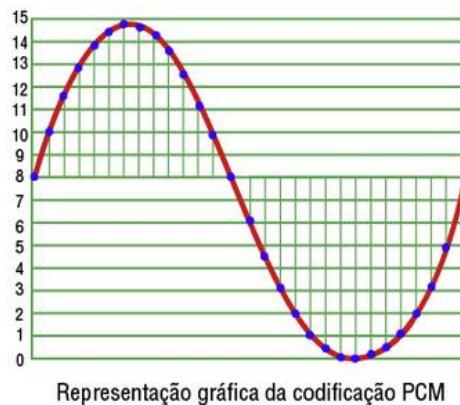
08

Já a **profundidade de bits** representa o tanto de amostras possíveis de um conjunto de valores possíveis. Quanto mais números tiver esse conjunto, mais preciso e próximo do valor analógico será o valor digital escolhido.



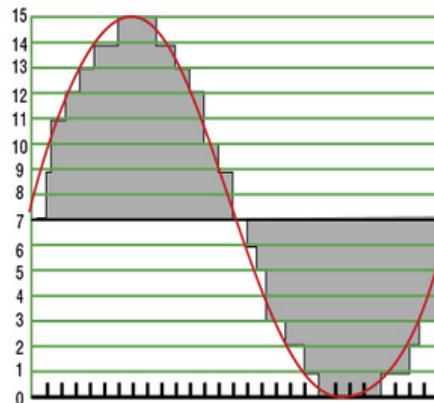
Quando juntamos a taxa de amostra e a profundidade de bits temos uma matriz de possibilidades de medição. Quanto mais pontos tiver nossa matriz, mais próximo do real será nossa medida. Portanto, a qualidade do áudio digital dependerá de quantos pontos temos para escolher no relacionamento do sinal analógico com o digital.

A figura abaixo ilustra os pontos azuis que representam as medidas digitais do áudio analógico (em vermelho). Esse processo chama-se **modulação**.



09

A imagem abaixo ilustra como o som analógico (em vermelho) fica quando convertido em digital (em cinza). A diferença entre um e outro representa a diferença da qualidade sonora do som digital (ou digitalizado).



Sobreposição entre o áudio analógico e o áudio digitalizado.

CDs de áudio são codificados em PCM (mas com o contêiner RBA), que é o mesmo utilizado em arquivos WAV. O padrão RBA foi definido pela Sony e Philips nos anos 80, tornando-se padrão mundial logo em seguida. O RBA, a título de curiosidade, define os padrões de um CD de música, entre eles:

- o tempo máximo de áudio em um CD é de 79 minutos e 50 segundos, incluindo pausas;
- a duração mínima de uma música (uma trilha) é de 4 segundos;
- a quantidade máxima de músicas em um CD é de 99.

Para um CD de áudio, a taxa de amostragem é de 44,1 kHz (44.100 medidas por segundo). Há 16 bits de profundidade, o que representa medidas que vão do -32.768 a +32.767. Com isso, cada minuto de áudio de um CD consome 10,1 Mbytes de informação.

10

3.2 - Taxa de amostragem versus qualidade

Já aprendemos que **quanto maior a taxa de amostragem, maior será a qualidade da nossa gravação**. Mas, de que adiantaria uma taxa de amostragem muito alta se o som original já é de baixa qualidade? Nada!

Então, para prevenir perda de espaço de arquivo, desnecessário por conta de uma baixa qualidade do som de origem, podemos utilizar taxas de amostragem baixas em áudios de baixa qualidade. Uma sugestão de taxa de amostragem compatível com a qualidade do áudio é:

- **8 kHz a 11 kHz** → Para áudio de muito baixa qualidade como rádio AM, discos 78 rotações, áudio de telefone.
- **22 kHz a 32 kHz** → Para áudio de baixa qualidade como fitas K7, gravadores de áudio portáteis.
- **44,1 kHz** → Para áudios de boa qualidade como CDs, rádio FM, LPs, áudio para TV digital e TV a cabo.
- **48 kHz** → Para áudios de alta qualidade DVDs, edição de som ao vivo, áudio para Cinema.
- **96 kHz** → Para áudios de altíssima qualidade como gravações de estúdios de som.

11

3.3 - Profundidade de Bits versus qualidade

Já aprendemos que **quanto maior a taxa de bits, maior será a qualidade da nossa gravação**. Mas, de que adiantaria uma taxa de bits muito alta se o som original já é de baixa qualidade? Nada!

Então, para prevenir perda de espaço de arquivo, desnecessário por conta de uma baixa qualidade do som de origem, podemos utilizar taxas de bit baixas em áudios de baixa qualidade. Uma sugestão de taxa de bit compatível com a qualidade do áudio é:

- **8 bit** → Para áudio de muito baixa e baixa qualidade como rádio AM, fitas K7, discos 78 rotações, áudio de telefone, gravadores de áudio portáteis.
- **16 bit** → Para áudios de boa qualidade como CDs, rádio FM, LPs, áudio para TV digital e TV a cabo.
- **24 bit** → Para áudios de alta qualidade como CDs e DVDs, edição de som ao vivo, áudio para Cinema.
- **32 bit** → Para áudios de altíssima qualidade como gravações de estúdios de som.

12

4 - SUGESTÕES DE CONTÊINERES E CODIFICAÇÕES

Diante de tantas opções, é muito útil conhecer alguns padrões de contêineres e codificações a serem utilizados no dia a dia com o objetivo de ter o melhor custo x benefício na escolha:

Necessidade	Sugestão / recomendação
Preciso gravar aulas, áudio de reuniões, gravações telefônicas ou a voz humana. Meu objetivo é	1. Utilize WAV/FLAC, gravação em mono, 8bits, 8kHz, gerará o arquivo de excelente

conseguir reproduzir e entender o conteúdo, sem me preocupar com a melhor qualidade possível.	qualidade e menor tamanho possível. 2. Outra alternativa seria MP3 com até 22 kHz. O arquivo será um pouquinho maior que o WAV e com qualidade quase igual.
Preciso converter um CD ou um DVD, ou mesmo um vídeo do Youtube em formato de áudio	1. Converta utilizando MP3, 44,1 kHz e 16 bits. 2. Se o CD ou DVD for de altíssima qualidade, opte por MP3, 44,1 kHz e 24 bits.
Sou um audiófilo e quero converter e ouvir músicas no melhor padrão possível.	1. Utilize FLAC. 2. Use MP3, 44,1 kHz, 24 bits.
Sou engenheiro de som, produtor musical ou outro profissional que preciso guardar sons, músicas e instrumentos na melhor qualidade possível para posteriormente produzir CDs, DVDs ou trilhas sonoras.	1. Utilize WAV/PCM, gravação em pistas separadas, 32bits, 96kHz.
Sou DJ e preciso ter uma grande quantidade de músicas disponíveis para tocar em shows, boates e festas.	1. Use MP3, 44,1 kHz, 24 bits. 2. Use MP3, 44,1 kHz, 16 bits.
Gostaria de colocar uma música disponível para baixar em um site.	1. Use MP3, 44,1 kHz, 16 bits. 2. Use MP3, 44,1 kHz, 24 bits.

13

5 - COMO FAZER

5.1 - Como fazer para extrair o áudio de um vídeo no YouTube?

Há várias formas de se extrair o áudio de um vídeo no YouTube, basicamente, você pode optar por instalar um *software* no seu computador ou acessar um site que faz isso por você. Para localizar esse tipo de solução, nada melhor que o nosso Google, não é mesmo? Abra o Google e escreva “como extrair áudio de um vídeo no YouTube?”. Veja o que o Google traz como resposta:



como extrair áudio de um vídeo no youtube?

Web Vídeos Shopping Notícias Imagens Mais ▾ Ferramentas de pesquisa

Aproximadamente 1.780.000 resultados (0,33 segundos)

Extrair áudio de vídeos do youtube sem programas Muito ...
www.youtube.com/watch?v=A6xumj9mTgo ▾
 23/09/2013 - Vídeo enviado por planetgamersBR
 Like , Favorite no Vídeo Para Dar Aquela Força ! Ganhe Games Steam , Origin , Desura Totalmente ...

COMO EXTRAIR O AUDIO DE UM VIDEO - YouTube
www.youtube.com/watch?v=uJ83sKWIJE ▾
 18/06/2012 - Vídeo enviado por jorebri
 Neste tutorial, utilize o programa grátis A TUBE CATCHER para extrair o audio de um vídeo, sem ...

COMO EXTRAIR AUDIO DO YOUTUBE SEM PROGRAMAS ...
www.youtube.com/watch?v=Z1eDRgrtPaw ▾
 02/05/2013 - Vídeo enviado por jussimar almeida
 This video is unavailable. You need Adobe Flash Player to watch this video. ... COMO EXTRAIR AUDIO DO YOUTUBE SEM PROGRAMAS.

Extrair Audio Youtube - Busca Baixaki
www.baixaki.com.br/busca/?q=extrair+audio+youtube ▾
 Resultados da busca para extrair audio youtube no Baixaki. ... Baixe vídeos em alta definição e extraia o áudio de filmes do YouTube com uma ferramenta leve ... ListenToYouTube YoutubeFisher VidToMP3 HearYoutube

YouTube to MP3 Converter - Youtube vídeo download ...
www.video2mp3.net/pt/ ▾
 Use facilmente nosso conversor youtube para mudar de vídeo em um ... qualquer vídeo do YouTube ou para um arquivo de áudio de alta qualidade MP3.
 Obter a extensão Go Premium Ajudar Login

14

Note que o Google lhe oferece uma série de opções, mas aqui seguem duas sugestões para você testar, ambas gratuitas:

- www.youtube-mp3.org/pt - por meio desse site, basta colar a url do vídeo que você quer converter em MP3, clicar em “Converter Video”, aguardar alguns instantes para processar o vídeo, e pronto, você poderá baixar o arquivo MP3 clicando no link “download”.



YouTube mp3

Video convertido com sucesso para mp3
 Título: AC/DC - Back In Black
 Duração: 4 minutos
[Download](#)

<https://www.youtube.com/watch?v=pAgnJDJN4VA>

Converter Video

Interface do Youtube mp3

- <http://www.atube.me/> - permite baixar áudio ou o vídeo com áudio em vários formatos. Após instalar, basta copiar a url no campo correspondente e clicar em download. Permite ainda baixar vários áudios e vídeos ao mesmo tempo. Ainda, permite tocar a música enquanto é baixada. Por fim, aceita arrastar a url para a janela dele e então efetuar o download. Diferente do método anterior, este possui uma variedade enorme de funcionalidades, codecs etc.

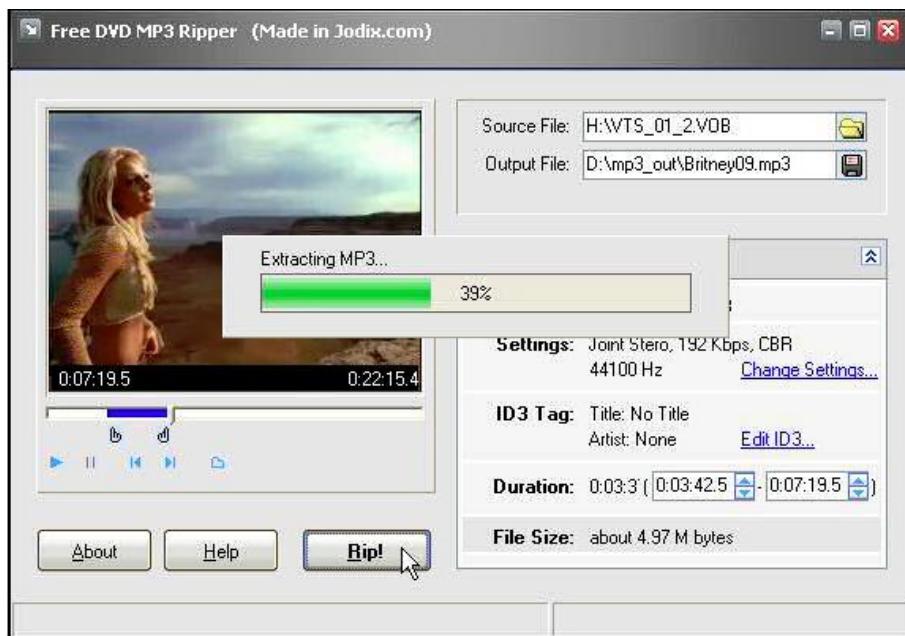


15

5.2 - Como extrair áudio MP3 de um DVD?

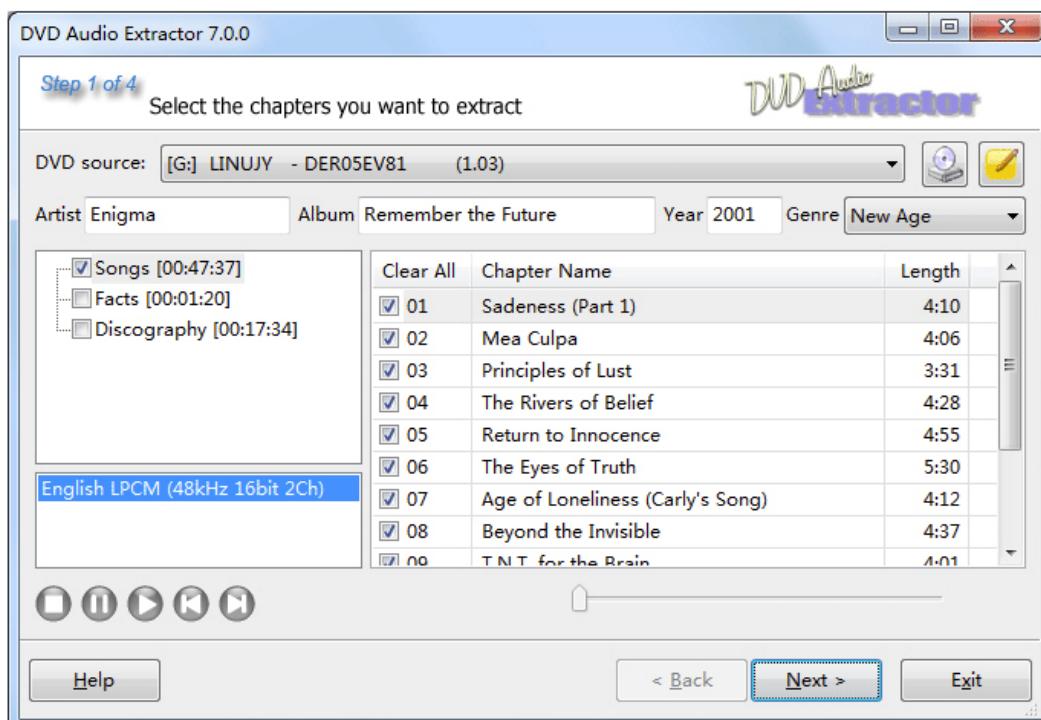
Usando a mesma técnica de pesquisa no Google, encontramos uma série de programas gratuitos para extrair MP3 de um DVD, entre eles, seguem duas recomendações:

- <http://www.dvd-mp3.org/> (Free DVD MP3 Ripper). Programa fácil de usar, basta colocar o DVD no leitor, selecionar a pasta onde as músicas serão gravadas e clicar no botão “Rip!”. Também é possível extrair apenas pequenos trechos do DVD.



Interface do Free DVD MP3 Ripper

- <http://www.dvdæ.com/> (DVD Audio Extractor). Bastante similar ao anterior. Realiza as mesmas tarefas. Tem a vantagem de possuir versões para Windows, Linux e Mac.



Interface do DVD Audio Extractor

5.3 - Como converter formatos de áudio diferentes?

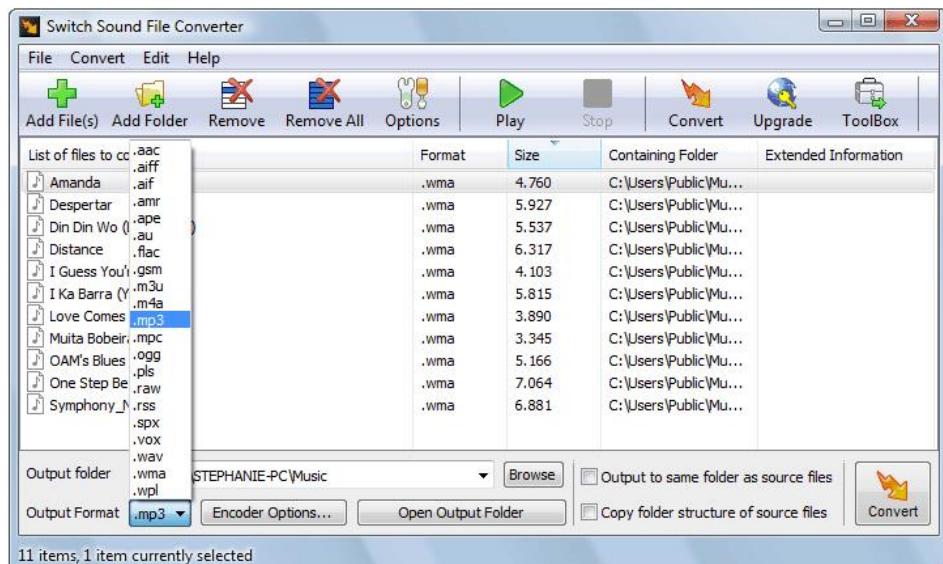
Novamente, recorrendo ao Google, ele nos sugere alguns programas gratuitos para converter formatos diferentes de áudio, seguem duas recomendações:

- <http://online-audio-converter.com/pt/> (Online Audio Converter), programa Web, permite conversão entre os seguintes tipos de arquivo de áudio: MP3, WAV, iPhone ringtone, M4A, Flac, OGG, MP2 e AMR. Muito fácil de usar, tem a desvantagem de precisar enviar e fazer download de arquivos, o que o torna lento. Boa opção para converter apenas um ou outro arquivo.



Interface do Online Audio Converter

- <http://www.nch.com.au/switch/> (Switch Audio File Converter Software), compatível com PC e Mac, permite conversão entre os seguintes tipos de arquivo de áudio: WAV, MP3, AU, AIF/AIFF, GSM, VOX, RAW, OGG, FLAC, AAC, M4A, MP2, WMA, RM/RA/RAM, DVF, MSV, DSS, CDA, MOV e AMR.

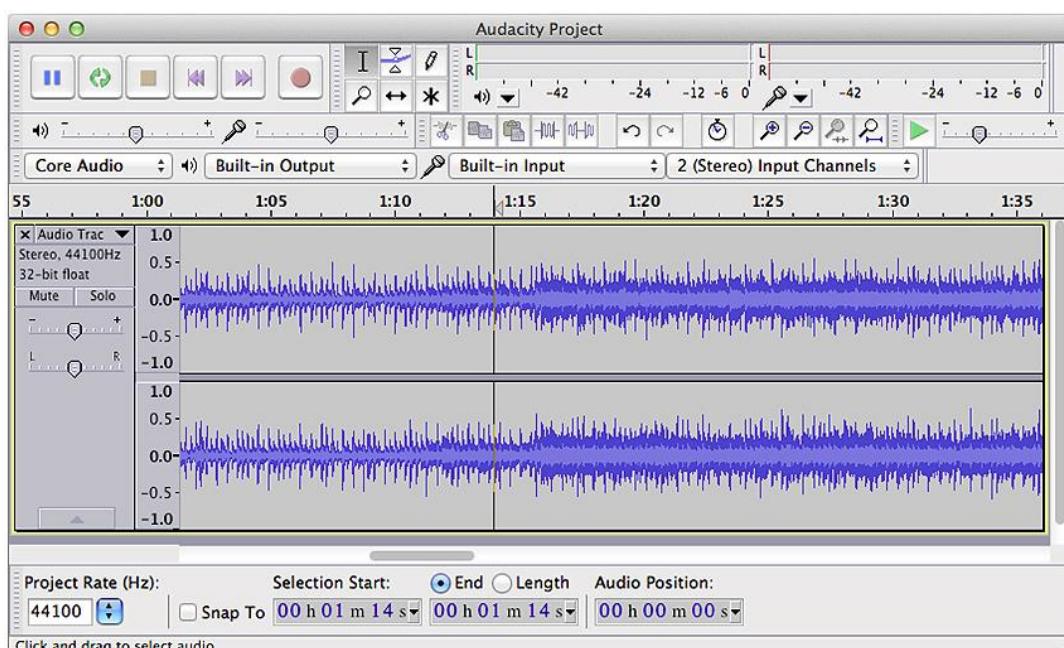


Interface do Switch Audio File Converter Software.

17

5.4 - Como editar um arquivo de áudio

Há várias formas e programas que permitem a edição de arquivos de áudio. Aqui estamos falando de manipulações completas como criação de efeitos, normalização de som, cortar um pedaço do áudio, equalizar, aplicar ajustes etc. Nesse aspecto encontramos uma infinidade de softwares gratuitos e pagos. Na área dos gratuitos, um dos mais poderosos e famosos é sem dúvida o “Audacity”, disponível neste endereço: <http://audacity.sourceforge.net/?lang=pt-BR>.

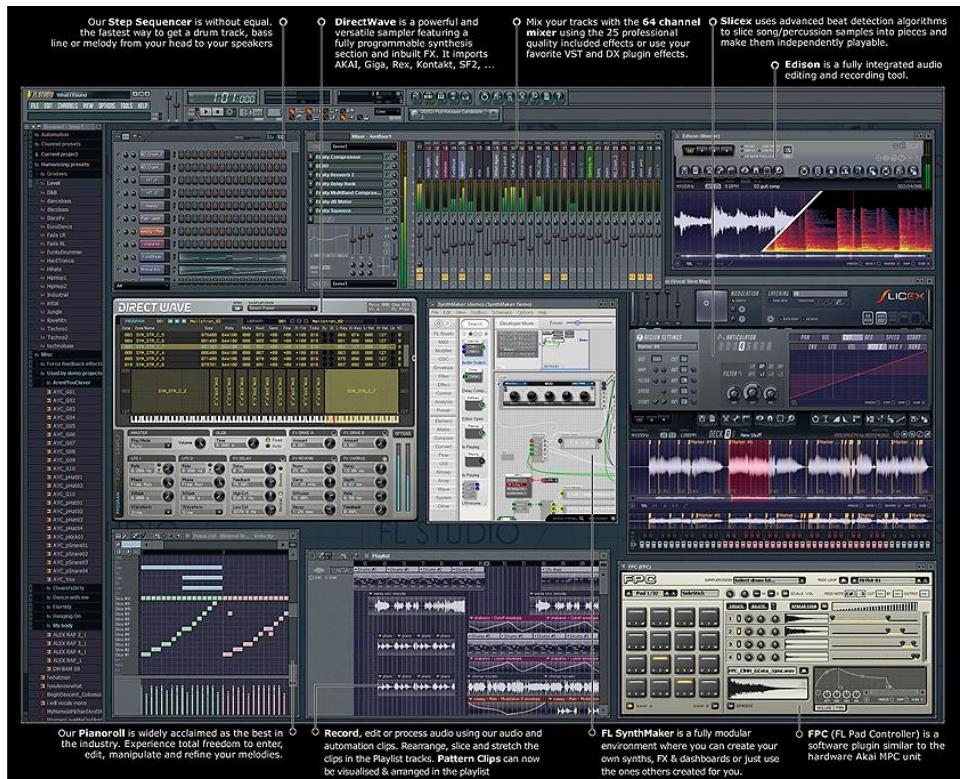


Interface do Audacity.

5.5 - Como criar áudio digital

Há muitos softwares que geram sons digitais, mas quando o assunto é música eletrônica o “Fruity Loops” é certamente um dos mais famosos. Há bandas que utilizam apenas o Fruity Loops como gerador de som: bateria, guitarra, teclados e até mesmo a voz humana pode ser sintetizada com softwares como esse. Infelizmente o Fruity Loops é um *software* pago, não existe ainda similar a ele que seja gratuito.

Se você quiser ver o poder do Fruity Loops, dê uma olhada no YouTube, clicando aqui: https://www.youtube.com/results?search_query=fruity+loops, olhe, por exemplo, esta música aqui: <https://www.youtube.com/watch?v=gTFpYscK48Q>.



Interface do Fruity Loops

Muitos exemplos de músicas feitas no Fruity Loops você encontra aqui: <https://soundcloud.com/explore>.

RESUMO

Neste módulo, aprendemos:

- a. Que para o mundo da informática, existem três tipos de formato de áudio: analógico, digitalizado e digital.
- b. Que o áudio analógico é aquele que está fora do ambiente do computador.
- c. Que o áudio analógico precisa ser convertido (codificado) para se tornar um áudio digital.
- d. Que áudio digital é aquele que foi criado e armazenado em um equipamento digital.
- e. Que há vários contêineres e vários tipos de codificação de arquivos de áudio digital.
- f. Que codificação é o método pelo qual o som foi digitalizado ou digitalmente construído.
- g. Que o padrão WAV/PCM é um padrão sem perdas e que gera áudio de altíssima qualidade.
- h. Que o padrão MP3 tem baixíssima perda, quase imperceptível aos ouvidos humanos.
- i. Que arquivos MIDI são como partituras musicais na qual o equipamento reproduz o som dos instrumentos de acordo com as informações do arquivo MIDI.
- j. Os tipos de codificadores de áudio são: sem compressão, compressão sem perdas e compressão com perdas.
- k. Os codificadores sem compressão possuem máxima qualidade, mas geram arquivos grandes.
- l. Os codificadores com compressão e sem perdas mantém a máxima qualidade, e geram arquivos um pouco menores.
- m. Os codificadores com compressão e com perdas geram arquivos bem menores, mas podem sofrer perda de qualidade.
- n. O PCM é o mais famoso dos codificadores sem perdas.
- o. Há muitos softwares gratuitos que permitem qualquer tipo de manipulação de áudio que você possa precisar.

UNIDADE 3 – ARQUIVOS DE INTERNET E MULTIMÍDIA

MÓDULO 4 – ARQUIVOS DE VÍDEO

01

1 - TIPOS DE VÍDEO

Seja bem-vindo mais um módulo de estudo. Neste módulo, iremos aprender características sobre os arquivos de vídeo.

Até o meio da década de 1970, os computadores não possuíam tela. Seus resultados eram apresentados em impressões. Operadores interagiam com os computadores por meio de ouvir bips de comandos. No final da década de 70 até o início da década de 90 os computadores possuíam as famosas telas em verde ou em âmbar, que só representavam texto ou gráficos de baixíssima qualidade.



Monitor CGA monocromático



Monitor CGA com 4 cores

Mas foi somente após o ano 2.000 que o potencial de vídeos coloridos realmente explodiu. De lá pra cá os computadores ganharam um potencial de processamento gráfico muito expressivo. Hoje, computadores podem processar vídeos de forma igual ou superior aos equipamentos gráficos específicos.

02

Editoração de vídeos, vídeos em 3D, vídeos de CAD, simuladores, jogos, filmagem, cinegrafia, vídeo-aulas, reuniões com vídeo conferência, são alguns dos milhares de usos para o novo potencial dos vídeos do século 21.

Da mesma forma que tratamos de áudio analógico e digital, os sistemas e arquivos de vídeo possuem o mesmo tipo de qualificação:

Vídeo analógico	Vídeo digitalizado (ou encoded)	Vídeo 100% digital
É aquele que está fora do ambiente do computador ou de um equipamento digital (como um CD). Saiba+	É o vídeo analógico que foi “gravado” e “convertido” em vídeo digital. Para o vídeo analógico ser enviado para o computador, ele precisa ser digitalizado (processo esse também chamado de codificação ou <i>encode</i>). Saiba+	É aquele que foi criado dentro do computador (por meio de softwares de editoração e criação de vídeo) ou por meio de equipamentos digitais, como uma filmadora digital ou um celular moderno.

Geralmente, 99,9999% dos arquivos de vídeo digital contém também áudio digital. É raríssimo, nos dias de hoje, vermos arquivos de vídeo que não contêm áudio. Portanto, além das características da codificação do vídeo, teremos associado também, a codificação do áudio.

Saiba+ vídeo analógico

Quando assistimos uma TV aberta analógica, ou utilizamos filmadoras antigas de Fita (VHS, Betamax etc.), estamos falando de sinais de vídeo analógico.

Saiba+ vídeo digitalizado (ou encoded)

Gravar aquelas velhas fitas VHS em DVD é um exemplo de conversão para vídeo digitalizado.

03

1.1 - Tipo de contêiner *versus* tipo de codificação

Em se tratando de vídeo, dois parâmetros importantes devem ser levados conta:

- O primeiro deles é o tipo de **contêiner**.

O contêiner é simplesmente o tipo de arquivo (extensão) que guarda as informações do vídeo (e do áudio associado).

Os contêineres mais famosos são o AVI, MP4, MKV, FLV, OGV, MOV, RM, 3GP e WMV. Cada contêiner tem uma determinada característica, utilidade e qualidade de imagem e som.

- O segundo parâmetro é o tipo de **codificação** (codec).

A codificação é o método pelo qual o vídeo e o áudio foram digitalizados ou digitalmente construídos.

Há codificadores que compactam o conteúdo (e geralmente há uma leve perda de qualidade nessa compactação, muitas vezes imperceptível aos nossos ouvidos), outros codificadores não compactam o conteúdo, e assim geram arquivos bastante grandes. Aqui estamos falando de dois codificadores independentes, um para a imagem e outro para o áudio. Os codecs mais famosos são Xvid, DivX, H264 e Mpeg.

04

1.2 - Streaming

Streaming, sem tradução direta para o português, mas algo como “transmissão por fluxo”, refere-se à capacidade de transmitir a informação em tempo real. Esse recurso é muito importante, pois permite que você possa assistir ao vídeo enquanto ele é “baixado” para o seu computador.

Um exemplo clássico são os vídeos do YouTube, iTunes e Netflix, todos os vídeos que lá residem estão em um formato que permite streaming. Para ver um comparativo completo dos sistemas que suportam streaming acesse

aqui: http://en.wikipedia.org/wiki/Comparison_of_streaming_media_systems.

Vídeos que não possuem o recurso de *streaming* precisam ser “baixados” completamente para só depois ser possível de serem tocados.

Para vídeos que são disponibilizados na Internet, o ideal é que sempre haja suporte ao **streaming**. Sistemas de vídeo conferência, por exemplo, como o MSN, o Skype, e outros tantos, todos eles precisam do recurso de **streaming** para funcionarem.

Você verá logo adiante que há tipos de arquivos de vídeo que suportam *streaming* e outros que não suportam. Contêineres mais antigos são compatíveis com mais players (equipamentos, como o DVD, e softwares, como o Windows Media Player, para mostrar os vídeos), entretanto, a maioria não

suporta *streaming*. Já os contêineres mais modernos, que suportam *streaming*, não são compatíveis com todos os players existentes no mercado.

Atualmente, o padrão de codificação vídeo mais moderno que existe é o H264. Veremos mais abaixo as características desse codificador.

05

1.3 - Proporção de tela

Proporção de tela refere-se à razão de proporção entre a largura e a altura da imagem.

No início das transmissões de TV, por volta dos anos 50, a imagem era levemente arredondada. Já nos anos 60 adotou-se o padrão de imagem retangular. A proporção da imagem de TV analógica, definido naquela época, era de 4:3, ou seja, a altura da imagem deveria corresponder a 75% ($3/4$) da largura da imagem. Esse padrão foi utilizado até o final dos anos 90, onde as TVs de tela plana trouxeram um novo padrão de imagem, o 16:9. Atualmente, a transmissão digital e quase todas as TVs de tela plana adoram o padrão de 16:9.



Proporção 4:3



Proporção 16:9

06

Já o cinema utiliza outro padrão de proporção, o 21:9, nesse formato, a imagem é muito mais retangular. Essa proporção, mais larga do que alta, foi criada pelo próprio formato dos cinemas.

Se uma tela de cinema fosse na proporção 16:9, por exemplo, ela precisaria ter uma tela de projeção muito alta, ou uma redução na largura da imagem. Exemplo: um cinema com 21 metros de largura de tela, precisa de 9 metros de altura de tela. Se a proporção do cinema fosse 16:9, a tela precisaria ter 12 metros de altura. Se a proporção fosse 4:3, a tela precisaria ter 16 metros de altura!



Proporção 21:9

Quando você criar vídeos é muito importante que saiba onde ele será apresentado, pois assim você poderá escolher a proporção de tela mais adequada ao seu público-alvo.

07

As telas antigas de computador adotavam o padrão 4:3, já as telas mais modernas utilizam o padrão 16:9. Alguns raros modelos utilizam o padrão 21:9.

Veja essa imagem comparativa:



08

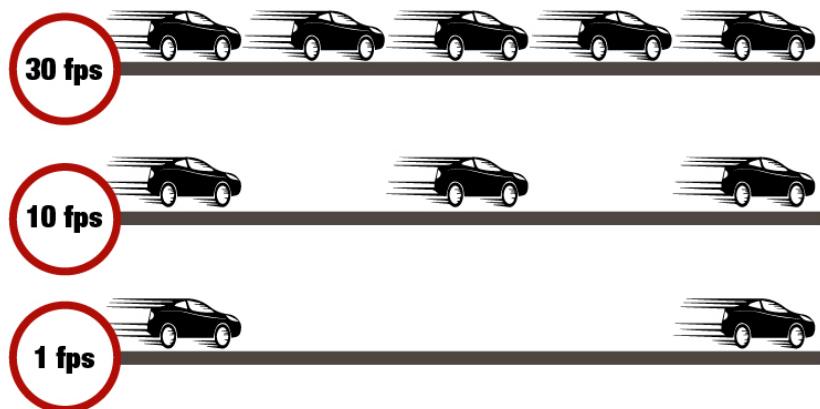
1.4 - Imagens por Segundo (*Frames Per Second – FPS*).

Um vídeo é uma série de imagens estáticas sobrepostas.

Desde o cinema mudo, fazemos vídeos a partir de várias e várias fotos colocadas em sequência. Cada formato de vídeo possui uma taxa de quantidade de imagens (fotos) em um segundo de vídeo. Atualmente há três padrões de imagem:

- **24 imagens por segundo:** padrão utilizado para cinema;

- **30 imagens por segundo:** padrão utilizado para TV digital;
- **60 imagens por segundo:** padrão para TV digital de alta qualidade;



Há ainda outros padrões, como os utilizamos para criar imagens em câmera lenta, que vão de 100 a até 10.000 imagens por segundo.

09

1.5 - Taxa de bits (*bit rate*)

O *bit rate* controla a quantidade de informação que um arquivo de vídeo irá conter.

Essa taxa é medida em bits por segundo (bps).

Diretamente falando, quando maior o *bit rate*, maior será a quantidade de informação e, portanto, maior será a qualidade da imagem e o tamanho do arquivo gerado.

Para os formatos de vídeo padrão, temos os seguintes intervalos de *bit rate*:

- **TV comum (SD)** – de 2.000 a 5.000 kbit/s;
- **TV HD (720p)** – de 5.000 a 10.000 kbits/s;
- **TV Full HD (1080p)** – de 10.000 a 20.000 kbits/s;

10

1.6 – Resolução

Resolução refere-se ao tamanho da imagem.

Para os formatos de vídeo padrão, temos as seguintes resoluções:

- **TV comum (SD) em formato 4:3** – 640 x 480 pixels;

- **TV comum (SD) em formato 16:9 – 640 x 360 pixels;**
- **TV HD (720p) em formato 16:9 – 1280 x 720 pixels;**
- **TV Full HD (1080p) em formato 16:9 – 1920 x 1080 pixels;**

A maioria dos *softwares* de vídeo conferência utiliza o padrão de 640 x 480 pixels para comunicação.

11

2 - TIPOS DE ARQUIVOS (CONTÊINERES)

2.1 - Arquivo AVI

AVI vem do termo em inglês *Audio Video Interleaved*, algo como **áudio e vídeo intercalado**. O AVI é um padrão criado pela Microsoft em 1992. O padrão AVI mantém as informações de áudio e vídeo integradas, mantendo assim a sincronização entre ambos. Assim como vemos em um DVD, um arquivo AVI pode conter vários arquivos de áudio associados ao mesmo vídeo, permitindo assim a possibilidade de termos um filme com vários idiomas de áudio.

Por ser um padrão antigo, os arquivos AVI possuem algumas **limitações**, entre elas:

- Não possui informação sobre a proporção entre a largura e a altura do vídeo, gerando distorções em algumas reproduções. Exemplo de erro de proporção causada por configuração errada de arquivo AVI:



Imagen 16:9 em formato 4:3



Imagen 16:9 em formato 16:9

- Não suporta codecs de compressão que são baseados em leitura de quadros futuros (que é um dos métodos mais modernos usados atualmente, como o MPEG-4 ou o H264).
- Não suporta codificação com bitrate variável, que é uma técnica mais moderna para maior compressão e qualidade de imagens (o áudio MP3, por exemplo, suporta bitrate variável).
- Não suporta *Streaming*, ou seja, não pode ser utilizado em Internet para visualização em tempo real.

Modelos mais recentes de contêineres, como o Matroska, Ogg e MP4, resolveram todos os problemas citados.

12

2.1.1 - Padrão DV AVI

O padrão DV AVI é um tipo especial de AVI para vídeo comprimido com o padrão DV, muito utilizado por filmadoras digitais.

A maioria dos *softwares* de edição de vídeos (de filmadores) utiliza o padrão DV AVI como o mais comum padrão de edição.

Ferramentas líderes de mercado, como Adobe Premiere, tem como principal formato de edição o padrão DV AVI.

As filmadoras mais modernas suportam outros codecs como o DivX, o Mpeg-4 ou o H264.

13

2.2 - Arquivo MP4

O padrão MP4, também conhecido por MPEG-4 parte 14, é um formato altamente comum, baseado no padrão QuickTime, para armazenar áudio de vídeo. Foi criado em 2003 e suporta *streaming*.

Há uma formatação especial do MP4 para *smartphones*, denominada 3GP, que foi feita com o intuito de permitir vídeo *streaming* pela rede 3G dos celulares.

A Sony, por exemplo, é uma das empresas que produz material exclusivamente em formato MP4. Todos os vídeo games desta marca utilizam este formato nos seus equipamentos (exemplo: Play Station e PSP).

14

2.3 - Arquivo MKV

O formato Matroska Multimedia Container, MKV, é um padrão em código aberto extremamente flexível. Permite armazenar vários arquivos de vídeo ao mesmo tempo (permitindo, por exemplo, enxergar uma mesma cena filmada por câmeras diferentes) e de áudio também (permitindo, por exemplo, filmes com idiomas diferentes e qualidades de áudio diferentes).

O MKV permite também arquivos de alta taxa de qualidade (imagens HD, Full HD e mais atuais). O padrão MLV é similar aos padrão AVI, MP4 ou ASF, entretanto, é um padrão totalmente gratuito e aberto, permitindo assim, a criação de players de melhor qualidade (e mais leves). Arquivos MKV suportam, ainda, imagens em 3D, legendas, menus interativos, ajustes de proporcionalidade de imagem, *streaming* e tudo o de mais moderno que existe atualmente.

Atualmente, uma das pouquíssimas desvantagens desse formato é a existência de muitos equipamentos (como algumas smart TVs, tablets, filmadoras e *smartphones*) que ainda não são compatíveis com esse tipo de contêiner.

15

2.4 - Arquivo FLV

O formato FLV refere-se a Flash Video, que é um formato criado pela Adobe. Arquivos FLV são nativamente suportados pelo software Adobe Flash Player. O formato FLV tornou-se o padrão mais bem aceito na Internet para streaming de vídeo, tanto é que a ISO certificou em 2007 o padrão FLV como padrão para Internet.

Os vídeos que você assiste no YouTube, Yahoo Video, Metacafe, Reuters.com e muitos outros sites são todos eles codificados no formato FLV.

Atualmente, o FLV já suporta o codec H.264, tornando-se assim, o melhor formato para codificação de vídeo quando comparamos tamanho de arquivo criado versus qualidade da imagem. Todos os sistemas operacionais e todos os navegadores suportam vídeo no formato FLV. Apenas o sistema operacional Apple iOS não suporta FLV.

2.5 - Arquivo RM ou RMVB

O padrão RealMedia (ou RealMedia Variable Bitrate) é um formato criado pela empresa RealNetworks. Esse padrão é tipicamente utilizado para armazenagem local de arquivos de vídeo. O RM utiliza codificador similar ao x264.

Esse padrão é largamente utilizado para conteúdo asiático, especialmente na China (tanto em TV quanto em filmes). Por essa razão ele se tornou notavelmente presente (muito embora seja pouquíssimo popular no resto do mundo).

16

2.6 - Arquivo QuickTime (MOV e QT)

O padrão QuickTime foi criado pela Apple em 2001. É o padrão de codificação dos vídeos desta empresa. Suporta uma grande variedade de recursos, como múltiplas trilhas de áudio e vídeo, legendas, capítulos e outros.

Ele tem a capacidade de separar o áudio do vídeo em arquivos separados, o que é ótimo quando estamos falando de editoração de vídeo. Por esse motivo é tão famoso por empresas de pequeno e médio porte na criação de conteúdo digital.

O QuickTime usa os mesmos padrões do MP4, portanto, os players de QuickTime também conseguem reproduzir arquivos MP4 (e vice-versa), bastando, em muitos casos de incompatibilidade, apenas renomear o arquivo .MOV para .MP4.

2.7 - Arquivo ASF e WMV

O formato ASF também foi criado pela Microsoft. O ASF nasceu em evolução ao AVI por suportar *streaming* de vídeo. Desde o Windows XP, a Microsoft oferece, gratuitamente, o Windows Movie Maker, que é um software de criação e edição de arquivos WMV e WMA. Arquivos ASF podem residir em servidores de mídia digital ou sites HTTP (incluindo redes locais), sendo em ambos possíveis de assisti-los por meio de *streaming*.

O padrão ASF também suporta metadados, como o nome do filme, do artista, ano de criação, duração do vídeo e outros.

17

2.8 - Arquivos VOB

O padrão VOB é o padrão utilizado em DVD's. O padrão VOB permite vídeo, áudio, legendas e menu. Pode ainda ter seu conteúdo criptografado. Dentro de um DVD, os arquivos de imagem estão dentro da pasta “VIDEO-TS”. O VOB utiliza o codificador MPEG-1 ou MPEG-2 (veja mais abaixo).

Em um DVD, apesar de a imagem de um filme ser contínua, os arquivos internos são divididos em tamanhos de, no máximo, 1 GByte para tornarem-se compatíveis com todos os sistemas de arquivos dos sistemas operacionais, incluindo formatos antigos, como o FAT.

Um DVD, para funcionar com recursos de menus, legendas, navegação de conteúdo e mudança de capítulos, precisam além do arquivo VOB de arquivos auxiliares contendo essas informações, em formato IFO e BUP.

Praticamente todos os DVDs comerciais produzem arquivos com proteção e restrição de cópia. Para tal, os arquivos VOB são criptografados utilizando uma técnica denominada CSS (Content Scramble System). Muitos softwares de ripagem de DVD são capazes de quebrar essa segurança.

Para mais informações sobre a estrutura de um DVD, acesse: http://en.wikipedia.org/wiki/DVD-Video#Directory_and_file_structure

2.9 - Outros formatos e quadro comparativo

Outros formatos e um quadro comparativo entre os contêineres apresentados podem ser vistos aqui: http://en.wikipedia.org/wiki/Comparison_of_container_formats.

18

3 - TIPOS DE CODIFICADORES

Assim como quando falamos de áudio, há também três tipos de codificadores (codecs) para vídeo:

- **Sem compressão**

Nestes codecs as informações são armazenadas exatamente como são recebidas, então os arquivos gerados são bem grandes, pois não há compressão de informações. Algumas vantagens de codecs que não usam compressão de dados é que eles exigem pouco poder de processamento e são ótimos para edição. A principal desvantagem é que os arquivos gerados por estes codecs ocupam muito espaço. Exemplo: LPCM, PDM e PAM.

- **Compressão sem perdas**

Com estes codecs é diferente. As informações são comprimidas e os arquivos gerados são bem menores do que os originais. Uma grande vantagem deles é que nenhuma informação é perdida, então quando você quiser usar outro codec a qualidade será preservada, pois a informação processada pelo decodificador é a mesma recebida pelo codificador. A média de compactação desses codecs é de 40 a 60%. Exemplo: H.264 lossless, VP9 (Google) – para vídeo; ALAC (FFmpeg e QuickTime), DST (MPEG-4), Dolby TrueHD, DTS-HD, FLAC, WMA – para o áudio.

- **Compressão com perdas**

Neste tipo de codec parte das informações recebidas pelo codificador são descartadas para diminuir o tamanho do arquivo. Bons codificadores descartam informações que geralmente não são percebidas pelos sentidos humanos. Estes são os codecs mais utilizados hoje, mas não são muito usados para uso profissional. Os arquivos gerados por estes codecs são geralmente os menores e o poder de processamento requerido é quase sempre maior que os requeridos pelos codecs sem-compressão. A média de compactação desses codecs é de 25 a 90% Exemplos: AIC, AVS, MPEG-1, MPEG-2, MPEG-4, H.264, RealVídeo, VP9 (Google), WMV – para vídeo; Dolby Digital, DTS, MPEG-1, MPEG-2, LAME, AAC, MPEG-4, Vorbis – para o áudio.

19

3.1 - Codificador MPEG-2

O MPEG-2 (também chamado de H.262) é o codec padrão para vídeos em DVD e Blue Ray. Os arquivos em formato .VOB e .TS (presentes em DVDs e discos Blue Ray) são codificados utilizando o MPEG-2. TV a cabo e por satélite também utilizam o padrão MPEG-2 para codificação e transmissão.

3.2 - Codificador MPEG-4

O MPEG-4 é um codec com compressão e com perdas criado em 1998. Tornou-se no mesmo ano um padrão pela ISO para uso na Internet (*streaming*), DVDs, sistemas de vídeo conferência e transmissão de TV Digital.

3.3 - Codificador H.264/MPEG-4 AVC

O H.264, também chamado de MPEG-4 Parte 10 ou MPEG-4 AVC é atualmente o padrão de compressão de vídeo mais moderno que existe. Todas as grandes sites que disponibilizam vídeos (como YouTube, Vimeo, loja iTunes e outros) possuem seus arquivos codificados neste padrão (e salvos no contêiner FLV). Todas as grandes operadores de TV digital, TV a cabo e TV por satélite também utilizam esse codec. Atualmente, todos os novos filmes produzidos em formato Blue-ray utilizam este codificador.

O H.264 é, de longe, o melhor padrão quando comparamos qualidade de imagem (alta) e tamanho de arquivo produzido (o menor de todos). Por isso é tão utilizado atualmente.

Para saber mais sobre H.264, acesse aqui: http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC.

20

3.4 - Outros codecs

Há uma dezena de outros codecs que utilizam variações desses codecs já apresentados. Pesquise na Internet sobre DivX, Xdiv e FFmpeg.

3.5 - Instalando Codecs no computador

Para você realizar as atividades citadas neste módulo, além dos softwares citados, você precisa ter instalado no seu computador os Codecs que você irá utilizar. Para isso, um dos melhores pacotes de codecs gratuitos que encontramos na Internet é o “K-lite Codec Pack”. Esse pacote pode ser encontrado e baixado deste endereço: http://www.codecguide.com/download_kl.htm. Opte pela opção “Full”, é a quem a melhor configuração de Codecs para você.

Diferentemente dos diversos padrões que vimos quando tratamos de áudio, para vídeo podemos, sim, dar uma sugestão direta: 99,9999% do material de vídeo digital hoje é melhor codificável com H.264 e guardado em formato FLV (ou Mp4).

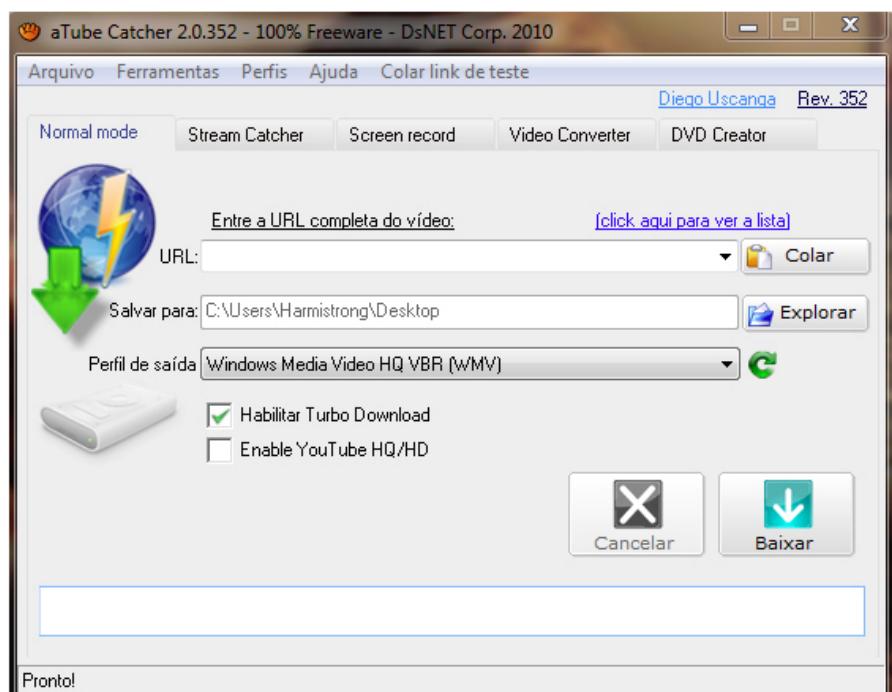
Praticamente todos os equipamentos atuais como computadores, smart TV, celulares smartphones, tablets, sistema de vídeo conferência e sites de vídeo suportam esse formato.

4 - COMO FAZER

4.1 - Como fazer para baixar um vídeo no YouTube?

Há várias formas de se baixar um vídeo no YouTube, basicamente, você pode optar por instalar um software no seu computador ou acessar um site que baixe o arquivo para você. Para localizar esse tipo de solução, nada melhor que o nosso Google, não é mesmo? Abra o Google e escreva “como baixar um vídeo do YouTube?”. Veja que o Google lhe oferece uma série de opções, mas aqui seguem sugestões para você testar, todos gratuitos:

- <http://www.atube.me/> - permite baixar áudio ou o vídeo com áudio em vários formatos. Após instalar, basta copiar a url no campo correspondente e clicar em download. Permite ainda baixar vários áudios e vídeos ao mesmo tempo. Possui uma variedade enorme de funcionalidades, codecs etc.



Interface do aTube Catcher.

- <http://www.videograbber.net/pt/> e <http://www.clipconverter.cc/pt/> - permite baixar vídeos sem precisar de instalar software e em vários formatos, como MP4, 3GP, AVI e MOV.



Interface do ClipConverter

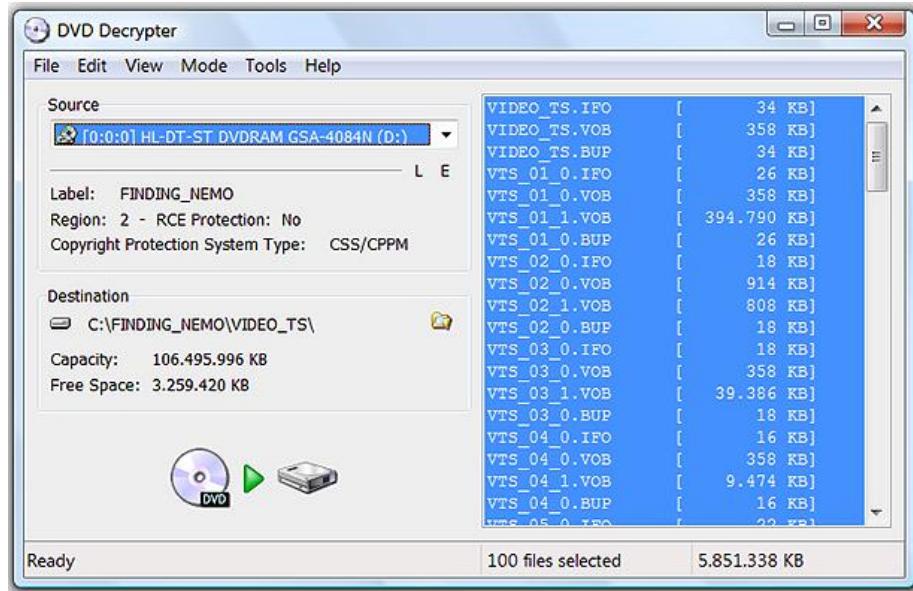


Interface do Video Grabber

23

4.2 - Como converter um DVD em um arquivo no seu computador?

A conversão de um DVD em um arquivo no computador é denominado de “ripagem”. Como normalmente o conteúdo de um DVD é protegido quando a cópia, os softwares que fazem “ripagem” são considerados “ilegais” e, portanto, podem ser proibidos em alguns países (quando utilizados para fins não pessoais). Entretanto, é extremamente fácil encontrar softwares que fazem isso, como o libdvdcss, DeCSS, DVD Decrypter, AnyDVD e o DVD Shrink. Use o Google para encontrar esses softwares.



Interface do DVD Decrypter.

24

4.3 - Como converter formatos de vídeo diferentes?

Há centenas de softwares que fazem conversão de formatos de vídeo. O Wikipedia apresenta um comparativo interessante de ferramentas:

http://en.wikipedia.org/wiki/Comparison_of_video_converters. Há vários softwares gratuitos capazes de converter formatos de vídeos diferentes. Quase todos eles suportam conversão entre os tipos mais comuns de arquivos de vídeo: 3GP, AVI, vídeo Blu-ray, vídeo DVD, FLV, Matroska, MP4, MPEG-PS, Ogg, QuickTime, SVCD, TS, TOD, VCD e WMV. Os softwares gratuitos mais famosos são:

- <http://www.any-video-converter.com/> (Any Video Converter);



25

- <http://www.erightsoft.com/SUPER.html> (Super);



- http://www.freemake.com/free_video_converter/ (Freemake Video Converter)



Interface do Switch Audio File Converter Software

4.4 - Como editar um arquivo de vídeo

Há várias formas e programas que permitem a edição de arquivos de áudio. Aqui estamos falando de manipulações completas como criação de efeitos, normalização de som, cortar um pedaço do vídeo, criar vinhetas, aplicar ajustes, etc. Nesse aspecto encontramos uma infinidade de softwares gratuitos e pagos. Na área dos gratuitos, algumas sugestões seriam:

- Windows Movie Maker;
- VirtualDub;
- Avidemux;
- Ffmpeg;
- Blender;
- VSDC.

Quanto aos pagos, os mais reconhecidos são:

- Adobe Premiere;
- Final cut (Apple);
- iMovie (Apple);
- Sony Vegas.

Uma tabela comparativa pode ser encontrada aqui:

http://en.wikipedia.org/wiki/Comparison_of_video_editing_software.

27

RESUMO

Neste módulo, aprendemos:

- a. Que para o mundo da informática, existem três tipos de formato de vídeo (os mesmos tipos de formatos para áudio): analógico, digitalizado e digital.
- b. Que o vídeo analógico é aquele que está fora do ambiente do computador.
- c. Que o vídeo analógico precisa ser convertido (codificado) para se tornar um vídeo digital.
- d. Que vídeo digital é aquele que foi criado e armazenado em um equipamento digital.
- e. Que há vários contêineres e vários tipos de codificação de arquivos de vídeo digital.
- f. Que codificação é o método pelo qual o vídeo foi digitalizado ou digitalmente construído.
- g. Que o contêiner é o tipo de arquivo onde o vídeo fica armazenado.
- h. Que streaming é a capacidade de um vídeo ser transmitido e assistido ao mesmo tempo.
- i. Que os vídeos são gravados em diferentes formatos de tela.
- j. Que arquivos de vídeo podem conter várias trilhas de vídeo, de áudio, diversas legendas e outras informações.
- k. Que o padrão FLV / H264 é atualmente o melhor método para criação de vídeos digitais.
- l. Que a resolução do vídeo determina o tamanho da imagem e sua qualidade.
- m. Que cada padrão é mais adequado a uma determinada situação.

- n. Os tipos de codificadores de vídeo são: sem compressão, compressão sem perdas e compressão com perdas.
- o. Os codificadores sem compressão possuem máxima qualidade, mas geram arquivos grandes.
- p. Os codificadores com compressão e sem perdas mantém a máxima qualidade, e geram arquivos um pouco menores.
- q. Os codificadores com compressão e com perdas geram arquivos bem menores, mas podem sofrer perda de qualidade.
- r. Que o padrão matroska é gratuito e aberto, e permite várias trilhas de áudio e vídeo no mesmo arquivo.
- s. Que codecs são necessários para assistir os vídeos e podem ser gratuitamente baixados da internet.