

# **SERVIÇOS DE REDES DE COMPUTADORES**

## **UNIDADE 3 - SERVIDOR FTP E PROTOCOLOS DE CORREIO ELETRÔNICO: COMO FUNCIONAM?**

Aline Izida

# Introdução

Transferir arquivos pela internet se tornou uma tarefa trivial para nós, como usuários e como desenvolvedores de páginas e aplicações *web*. Mas você sabe quais protocolos trabalham para fornecer esse serviço em redes de computadores? O protocolo FTP é um desses protocolos, muito útil para desenvolvedores *web* e para quem fornece serviço de hospedagem.

Já o correio eletrônico, também conhecido como *e-mail*, é um dos serviços mais utilizados e importantes que atuam na internet, pois possibilita que as pessoas se comuniquem de forma assíncrona. Tudo bem que você pode mandar uma mensagem no WhatsApp da pessoa e então ela responder quando puder. No entanto, esse tipo de mensagem síncrona é um tanto informal, não permite que você organize as informações em diferentes categorias. Você conhece todas as funções que seu *e-mail* proporciona? A partir do estudo desta unidade, você passará a explorar melhor o que sua caixa de correio eletrônico tem a oferecer.

Empresas utilizam o *e-mail* para manter relacionamentos entre clientes, colaboradores e fornecedores. Você está precisando organizar seus compromissos, separar as mensagens e arquivos que recebe de diferentes trabalhos das mensagens pessoais de compras, por exemplo? Ao estudar esta unidade, veja como o *e-mail* funciona e como pode ser útil.

Ótimos estudos!

## 3.1 FTP: conceitos

O *File Transfer Protocol* (FTP), em português Protocolo de Transferência de Arquivos, é utilizado para transferir arquivos na internet entre dois computadores ou dispositivos, geralmente um sendo cliente e o outro um servidor. Duas características importantes do FTP é que ele proporciona o controle da transferência e a segurança do arquivo. Assim, o FTP se preocupa com a segurança do arquivo e se ele vai chegar íntegro ao destino, considerando seu transporte pela internet (BARRETT, 2010).

## VOCÊ SABIA?



Existe o Protocolo de Transferência de Arquivos Trivial (o TFTP – do inglês *Trivial File Transfer Protocol*). Ele é uma forma simples do FTP, que usa UDP e, portanto, não oferece controle de acesso e não é seguro. É utilizado para transferir arquivos pequenos em redes locais e para configurações especiais em equipamentos eletrônicos.

Antes de prosseguir, vamos deixar claro o significado de arquivo nesse contexto. Consideramos que um arquivo armazena algo, que é considerado um objeto arbitrário, ou seja, um documento qualquer (planilha, de texto etc), um programa de computador, uma imagem, um vídeo, uma música.

Um fator fundamental faz com que essa transferência de arquivos pela internet seja algo complexo: a heterogeneidade das tecnologias de computadores, pois são representações de arquivo diferentes, informações de tipo, nomeação e mecanismos de acesso ao arquivo. Por exemplo, em alguns sistemas, a extensão de um arquivo de imagem é chamada de .jpg, em outros, de .jpeg. Uns sistemas utilizam uma barra (/) para separar nomes de arquivos, enquanto outros utilizam a barra invertida (\), ou seja, sistema de diretórios diferentes. Ademais, alguns sistemas definem que um arquivo de texto é encerrado por um único caractere, enquanto

outros sistemas exigem mais de um, o que significa que a representação dos arquivos pode ser diferente em cada sistema. Além disso, usuários com permissões de acesso diferentes também influenciam nesse contexto, pois um pode acessar determinado arquivo e outro não (COMER, 2016).

Desse modo, com a conexão FTP é possível enviar qualquer arquivo de um computador para o outro ou em um servidor FTP que armazenará os arquivos deixando-os sempre disponíveis para acesso. Na década de 1970, a ideia do FTP surgiu como proposta para transferência segura de arquivos entre computadores e servidores do modelo precursor da internet, a ARPANET. O FTP foi sendo aprimorado conforme o surgimento da internet e da evolução dela, em especial o grande uso dos computadores em residências e em organizações.

De acordo com Comer (2016), os seguintes aspectos são possibilitados através do FTP. Clique nas abas para ver.

- **Conteúdos arbitrários de arquivo**

É possível transferir qualquer tipo de dados, tais como documentos, imagens, músicas ou vídeos.

- **Transferência bidirecional**

Utilizada para baixar arquivos através da transferência do servidor para o cliente ou carregar (upload) arquivos, por meio da transferência do cliente para o servidor.

- **Suporte para autenticação e propriedade**

Cada arquivo pode ser proprietário de alguém com direito à restrição de acesso.

- **Habilidade para navegar em pastas**

O cliente pode navegar entre os conteúdos de um diretório.

- **Mensagens de controle textual**

As mensagens de controle trocadas entre um cliente e um servidor FTP são enviadas como texto ASCII.

- **Acomodação de heterogeneidade**

Detalhes dos sistemas operacionais dos computadores individuais são escondidos e é possível transferir uma cópia de um arquivo entre dois computadores quaisquer.

O problema de heterogeneidade citado anteriormente pode ser resolvido com a definição de três atributos de comunicação, segundo Forouzan (2013):

- estrutura de dados: pode ser estrutura de arquivos (sem estrutura), se trata de um fluxo de *bytes*; estrutura de registro, em que o arquivo é dividido em registros, suportando apenas arquivos de texto ou estrutura de página, em que o arquivo é dividido em páginas, cada uma com um número de um cabeçalho de página, que podem ser armazenadas e acessadas de forma aleatória ou sequencial;
- tipo de arquivo: a transferência de arquivos ASCII, EBCDIC ou de imagem;
- modo de transmissão: pode ser por modo de fluxo, o padrão, entregues do FTP para o TCP como um fluxo contínuo de *bytes*; de bloco, em que os dados podem ser entregues do FTP para o TCP em blocos precedidos de um cabeçalho de 3 *bytes* ou modo comprimido (compactado).

O protocolo pode parecer invisível, mas ele é chamado automaticamente pelo navegador quando o usuário solicita uma transferência de arquivo.

Uma questão importante é que, assim como o HTTP trabalha com o SSL para prover segurança e autenticação (HTTPS), o FTP também precisou ser aprimorado com a adição da camada de *sockets* segura SSL entre a camada de aplicação FTP e a camada TCP. Assim, o FTP será conhecido como FTPS, passando a trabalhar com a aplicação de criptografia para transferência de arquivos.

A seguir, você verá como o FTP trabalha com suas conexões.

### 3.1.1 Funcionamento

Um dos aspectos que mais chama atenção no funcionamento do FTP é o modo como o cliente interage com o servidor. Grosso modo, o cliente estabelece uma conexão com o servidor FTP e faz diversas solicitações para o servidor. Contudo, existe a conexão de controle, que é criada pelo cliente, e nela constam os comandos. Já o servidor abre uma nova conexão cada vez que precisa baixar ou carregar um arquivo. As conexões para transferir arquivos são chamadas de conexões de dados. Logo abaixo, na figura, é demonstrada uma interação simples entre cliente e servidor, considerando as conexões de dados e as de controle.

## VOCÊ QUER LER?



A lista completa dos comandos FTP e dos códigos de retorno definidos para o protocolo foram originalmente definidos pelo IETF na RFC 959. Posteriormente, considerando as questões de segurança das redes modernas, o protocolo foi atualizado pelas RFCs 2228 (HOROWITZ; LUNT, 1997), 2640 (CURTIN, 1999) e 2773 (HOUSLEY; YEE; NACE, 2000), que, então, incluíram especificações que abordam, por exemplo, o processo de encriptação de mensagem FTP e de dados de usuário transferidos na forma de arquivos. Acesse as RFCs em:

RFC 2228: <https://tools.ietf.org/html/rfc2228>;

RFC 2640: <https://tools.ietf.org/html/rfc2640>;

RFC 2773: <https://tools.ietf.org/html/rfc2773>.

As duas conexões no FTP têm tempos de vida diferentes, de modo que a conexão de controle permanece ativa durante toda a interação da sessão FTP, já a conexão de dados é aberta e em seguida fechada para cada atividade de transferência de arquivos. Ela é então aberta toda vez que algum comando que envolve a transferência de arquivos é utilizado, e a conexão é fechada quando o arquivo termina de ser transferido. Assim, quando um usuário inicia uma sessão FTP, a conexão de controle é aberta. Enquanto essa conexão de controle estiver aberta, a conexão de dados pode ser aberta e fechada diversas vezes se vários arquivos precisarem ser transferidos. O FTP usa, dessa forma, duas portas TCP, a porta 21 para conexão de controle e a porta 20 para conexão de dados (FOROUZAN, 2013).

Portanto, de acordo com Comer (2016), o FTP inverte o relacionamento cliente-servidor no momento da conexão de dados. Quando abre uma conexão de dados, o cliente age como se fosse um servidor (espera pela conexão de dados), enquanto o servidor age como se fosse um cliente (inicia a conexão de dados). Depois que a conexão foi utilizada para uma transferência, ela é fechada. A figura abaixo ilustra essa interação de forma simples e direta.

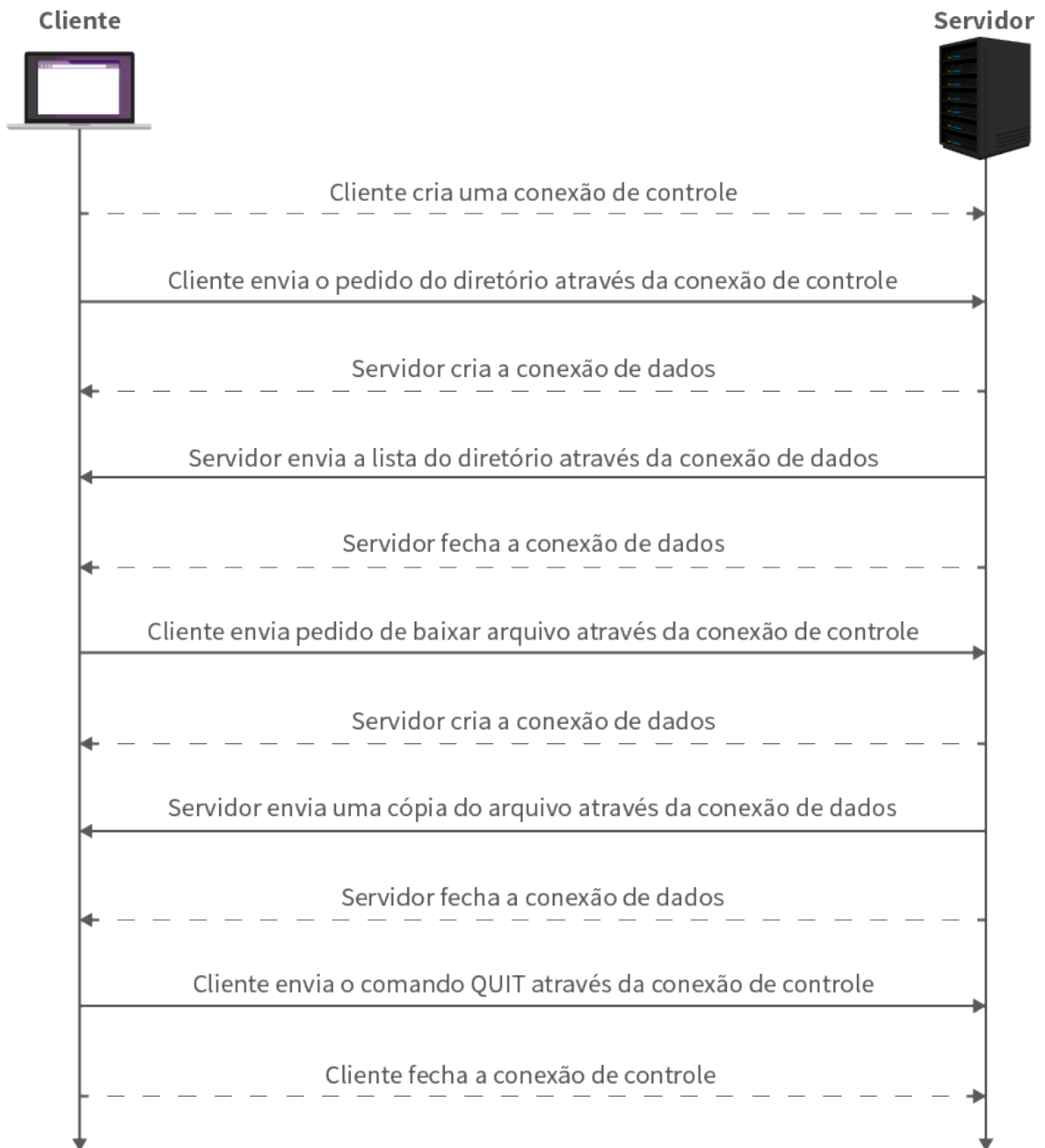


Figura 1 - Ilustração da conexão FTP durante uma sessão típica.

Fonte: COMER, 2016, p. 56.

Contudo, a figura não mostra que depois de criar a conexão de controle, um cliente deve se conectar ao servidor por meio do envio de um *login* e de uma senha; um *login* anônimo com senha de convidado é usado para obter arquivos que são públicos. E um servidor envia um *status* numérico através da conexão de controle como uma resposta a cada pedido, inclusive o de *login*; a resposta permite que o cliente saiba se o pedido era válido.

Desse modo, temos que, para realizar uma conexão com o servidor FTP, são necessários os dados de usuário, senha, endereço do servidor (*hostname*) e porta (no caso, para conexão com o servidor é por padrão a porta 21). No caso do usuário anônimo, não é necessário informar senha. Quando a conexão se estabelece com o servidor FTP, o cliente apresenta várias opções possíveis para manipular os arquivos, tais como listar arquivos, copiar e definir permissões.

## VOCÊ QUER VER?



Esta videoaula (2013) é ministrada de uma forma simples e didática, deixando claro o objetivo do FTP e enfatizando que nem toda transferência de arquivo utiliza FTP. Assista em: <https://www.youtube.com/watch?v=JKwk8WFG1zc>

A figura abaixo mostra o modelo básico do FTP, apresentando o cliente com seus três componentes: interface com o usuário, processo de controle cliente e processo de transferência de dados cliente. Já o servidor tem dois componentes: processo de controle servidor e processo de transferência de dados servidor. A conexão de controle é realizada entre os processos de controle e a conexão de dados entre os processos de transferência de dados (FOROUZAN, 2013).

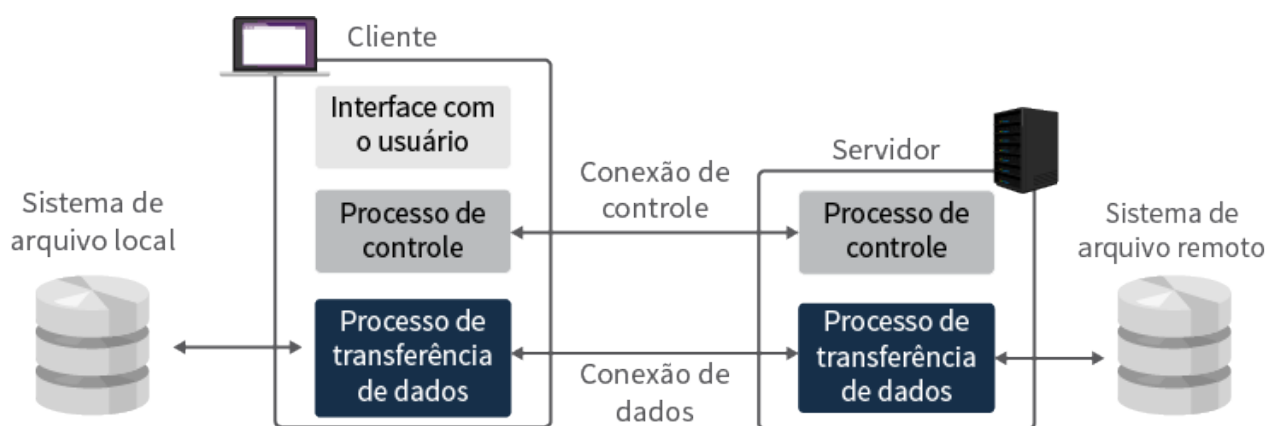


Figura 2 - Modelo básico de funcionamento FTP com os respectivos componentes do cliente e do servidor.

Fonte: FOROUZAN, 2013, p. 59.

Essa separação entre conexão de controle e de dados é responsável pela eficiência do funcionamento do protocolo FTP. A conexão de controle trabalha com regras de comunicação simples, ao passo que uma linha de comando ou uma linha de resposta é transferida de cada vez. A conexão de dados é mais complexa, pois trata da transferência dos arquivos de fato, e esses arquivos têm diversas variações de composição.

### Comandos FTP

Segundo Forouzan (2013), durante a conexão de controle, os comandos são enviados do cliente para o servidor e as respostas são enviadas do servidor para o cliente. Os comandos são enviados a partir dos processos clientes FTP de controle seguindo o formato ASCII com letras maiúsculas, sendo ou não seguidas por argumento. A seguir, veja alguns dos principais comandos, segundo Kurose e Ross (2013). Clique nos itens para conhecer.

<b>USER</b> username	Utilizado para enviar identificação do usuário para o servidor.
<b>PASS</b> password	Utilizado para enviar a senha do usuário para o servidor.

<b>LIST</b>	Utilizado para solicitar ao servidor uma lista com todos arquivos existentes no atual diretório remoto.
<b>R E T R filename</b>	Utilizado para obter um arquivo do diretório atual do <i>host</i> remoto. Ativa o <i>host</i> remoto para que abra uma conexão de dados e envia o arquivo requisitado por essa conexão.
<b>S T O R filename</b>	Utilizado para inserir um arquivo no diretório atual do <i>host</i> remoto.
<b>STRU</b>	Define organização dos dados (F para arquivo, R para registro e P para página).
<b>PORT</b>	É o identificador de porta, o cliente escolhe uma porta.
<b>TYPE</b>	É o tipo padrão de arquivo (A para ASCII, E para EBCDIC e I para imagem).
<b>QUIT</b>	Sai do sistema.

Para cada comando que o usuário gera existe uma resposta da conexão de controle. Essa resposta é enviada do servidor para o cliente, composta de um número de três dígitos e uma mensagem opcional depois desse número. Alguns códigos de resposta do servidor FTP são listados a seguir, conforme Kurose e Ross (2013):

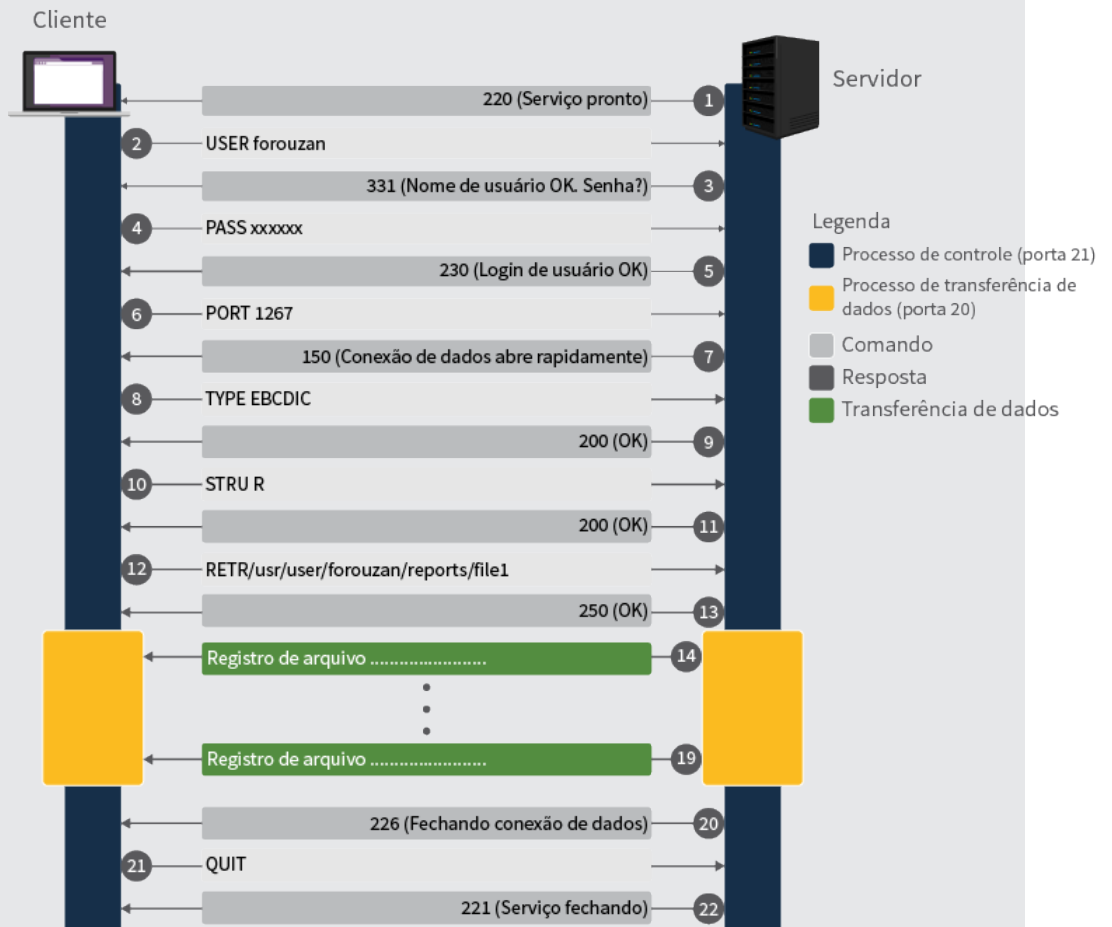
- 331 Nome de usuário OK, senha requisitada;
- 125 Conexão de dados já aberta, iniciando transferência;
- 425 Não é possível abrir a conexão de dados;
- 452 Erro ao escrever o arquivo;
- 226 Fechando conexão de dados;
- 221 Serviço fechando;
- 220 Serviço pronto;
- 200 Comando OK;
- 250 Solicitação de ação em arquivo OK.

Portanto, a transferência de arquivos acontece através da conexão de dados sob o controle dos comandos enviados pela conexão de controle.

## CASO



Veja este exemplo que traz o esquema do funcionamento da transferência de um arquivo FTP (FOROUZAN, 2013, p. 62).



Perceba que a conexão de controle permanece aberta durante todo o tempo, enquanto a conexão de dados é aberta e fechada repetidamente. Consideramos que o arquivo é transferido em seis seções. Logo após a transferência de todos os registros, o processo de controle no servidor anuncia que a transferência do arquivo foi finalizada. Como o processo de controle não tem mais arquivos para recuperar, é emitido o comando QUIT e a conexão é fechada.

O acesso ao servidor do FTP se dá por meio do que chamamos de cliente FTP. Os sistemas operacionais permitem acessar servidores FTP através de linhas de comando. O Windows, por exemplo, tem um FTP embutido no Explorer, o qual possibilita você navegar e manipular arquivos. E existem alguns *softwares* que permitem uma manipulação mais robusta, funcionando como cliente FTP, como é o caso do FileZilla, Cyberduck e WinSCP. Além disso, caso você deseje entrar no FTP sem utilizar um *software*, basta digitar ftp://número do ip em seu navegador para iniciar o acesso.



## VAMOS PRATICAR?



O FileZilla é um dos *softwares* FTP mais utilizados. Ele é gratuito e muito útil para quem gerencia contas de hospedagem de *sites*. Com ele é possível navegar, criar, excluir e editar todos os arquivos que estão armazenados em uma conta de hospedagem, agilizando as tarefas de quem precisa gerenciar seus arquivos, inclusive no que se refere a baixar e subir os arquivos de forma rápida, fácil e segura.

O FileZilla funciona em diversos sistemas operacionais, tais como Windows, Linux, Mac OS, entre outros. Tem como vantagem o fato de ter uma interface intuitiva, além de proporcionar a edição de códigos-fonte para correção, qualquer alteração será salva. É ágil ao transferir seus arquivos pela internet; proporciona um ambiente seguro para guardar seus arquivos, contanto que você também trabalhe com antivírus em sua máquina e estabeleça uma senha segura; é compatível com sistemas de gerenciamento de conteúdo, tal como o WordPress; através dele é possível instalar temas e *plugins* sem necessidade de acessar o painel de controle do seu sistema de gerenciamento de conteúdo (*site*).

Vamos aprender a utilizar o FileZilla, entendendo como o FTP funciona na prática? Baixe e instale o FileZilla e pratique o *upload*, *download* e gerenciamento de versões de arquivos.

Para instalar o FileZilla, basta acessar <https://filezilla-project.org/> e realizar o *download* da versão FileZilla Client, de acordo com o sistema operacional que você utiliza. Na tela que pede para selecionar entre FileZilla Pro e FileZilla, escolha a segunda opção, pois a versão Pro é paga. Siga as instruções de instalação e finalize-a. Neste vídeo tutorial, você pode acompanhar o *download* e instalação do FileZilla: <https://www.youtube.com/watch?v=Fy3BPhW-aEs>.

Você pode aprender a fazer uma conexão FTP com FileZilla com este vídeo tutorial: <https://www.youtube.com/watch?v=j9Ke6UmheLY>.

Agora que você já sabe o que é e como funciona o FTP, vamos descrever, no próximo tópico, os conceitos do correio eletrônico. Assim, você poderá analisar qual a relação entre esses dois conteúdos.

## 3.2 Correio eletrônico: princípios e componentes

O correio eletrônico (*e-mail*) é uma das primeiras aplicações a utilizar a internet. Primeiro, um *software* foi desenvolvido para enviar e receber mensagens, e com o tempo foram aprimorando os serviços oferecidos, tal como poder anexar arquivos de todos os tipos.

## VOCÊ O CONHECE?



Ray Tomlinson é considerado o pai do *e-mail*. Em 1972, o programador americano implementou um sistema de correio eletrônico na ARPANET (primeira rede operacional de computadores e precursora da internet) e tornou possível a troca de mensagens entre máquinas situadas em diferentes localidades. Foi dele a ideia de usar o símbolo @ para endereços de *e-mail* (BARRETT, 2010).

De modo geral, o correio eletrônico utiliza um método de transmissão do tipo armazenar e encaminhar. As mensagens são baixadas para a estação de trabalho. De modo semelhante, o correio eletrônico de saída é armazenado até que ele possa ser encaminhado na rede. Esse é o processo já muito utilizado nos *softwares* de correio eletrônico, tais como Outlook (BARRETT, 2010). No entanto, é muito mais comum, atualmente, a utilização do *webmail*, tais como Gmail e Hotmail, utilizados diretamente no navegador *web*.

Há diversas funções que permitem administrar *e-mails* enviados e recebidos. É possível criar diretórios para organizar *e-mails* recebidos, filtrar mensagens indesejadas (*spam*) ou diretamente para os diretórios criados para determinado tema de mensagens, e outras funções configuráveis de acordo com sua necessidade. Além disso, outros aplicativos interagem com o aplicativo de *e-mail*, como agendas e documentos *on-line*. Com a agenda, é possível criar reuniões e qualquer outro item de tarefa e compartilhar com todos os interessados.

## VAMOS PRATICAR?



Você conhece todas as funções para organização e gerenciamento dos seus *e-mails* na sua caixa de entrada? Por exemplo, você tem uma conta de *e-mail* pessoal, em que recebe mensagens dos *sites* de compras no quais é cadastrado, da sua faculdade, que, por sua vez, podem ser de diversas disciplinas, de pessoas da família, de amigos ou mesmo mensagens de propagandas indesejadas. Com seu *webmail*, pratique a criação de marcadores ou diretórios para filtragem de mensagens. Você pode criar um marcador, que funciona como uma pasta, e dar o nome de FACULDADE, e dentro dela criar pastas para cada disciplina. Quando as mensagens chegarem em sua caixa de entrada, você pode enviá-las para as respectivas pastas. Você pode fazer esse processo para qualquer tipo de mensagem, organizando-as em categorias. Além disso, você pode marcar uma mensagem indesejada como *spam*, assim, quando outra mensagem do mesmo remetente for enviada, ela será enviada automaticamente para a pasta de *spam*. Outra função interessante é salvar um arquivo grande na nuvem, tal como Google Drive, caso você utilize o Gmail e o OneDrive, caso utilize o Hotmail. Quando precisar enviar esse arquivo por *e-mail*, ele é anexado como um *link*, assim, não ocupa espaço no *e-mail*, pois ele está armazenado na nuvem. Pratique essas e as diversas funções da sua conta de *e-mail* e torne suas mensagens mais organizadas, o que facilita o gerenciamento das informações importantes.

Com a aplicação de documentos *on-line*, é possível anexar o documento no *e-mail* sem ter que utilizar o espaço de armazenamento do *e-mail*, já que a aplicação de documentos funciona *on-line*, é como se criássemos um *link* para o documento.

## Funcionamento

Um usuário chama um aplicativo de mensagens diretamente. A interface desse aplicativo fornece mecanismos que permitem que um usuário componha e edite mensagens de saída, bem como leia e processe mensagens recebidas. Um aplicativo de mensagens não age como um cliente ou um servidor e não transfere mensagens para outros usuários. Ele lê mensagens da caixa postal do usuário (ou seja, um arquivo no computador do usuário) e passa mensagens adiante para um aplicativo de transferência de mensagens. Esse aplicativo age como um cliente para enviar cada mensagem de *e-mail* para o seu destino. Além disso, ele também atua como um servidor para aceitar as mensagens recebidas e armazenar cada uma delas na caixa de correio do usuário apropriado (COMER, 2016).

De modo geral, o correio eletrônico funciona no paradigma cliente-servidor, no entanto, não exatamente como estudamos no FTP, por exemplo. Imagine o seguinte caso: quando Josué envia um *e-mail* para Maria, ele pode até esperar uma resposta, mas por algum motivo, Maria pode simplesmente não responder. Isso acontece muito em mensagens de propagandas. No caso de Maria responder Josué, isso poderá ser feito em um tempo que não é logo em seguida ao envio da mensagem. Desse modo, dizemos que o tanto o envio como a resposta são transações de mão única. Assim, Maria pode desligar seu computador e pode responder quando considerar necessário. Embora o servidor esteja ligado, ele não fica esperando a resposta com uma conexão aberta. Para explicar esse cenário da arquitetura de *e-mail*, veja a figura abaixo.

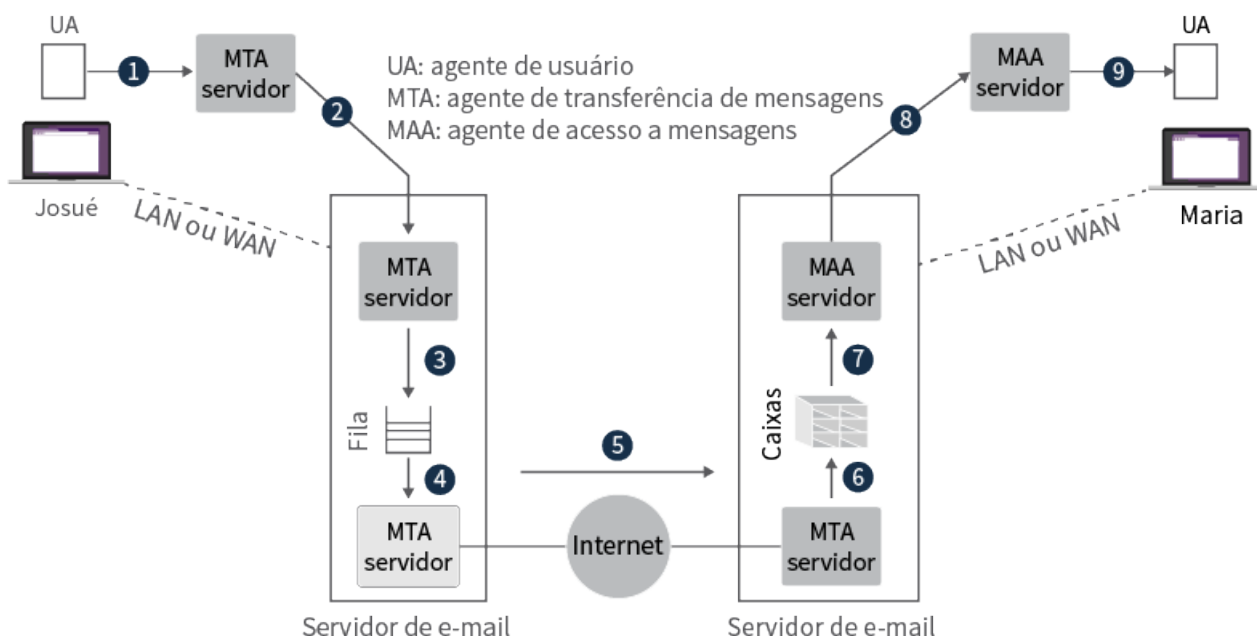


Figura 3 - Cenário comum do funcionamento de um correio eletrônico.

Fonte: Adaptada de FOROUZAN, 2013, p. 64.

O remetente é Josué e o destinatário é Maria. Eles são conectados por meio de uma LAN ou WAN a dois servidores de correio eletrônico. O administrador criou uma caixa de correio para cada usuário e, nelas, as mensagens recebidas são armazenadas. Uma caixa de correio faz parte de uma unidade de disco rígido do servidor, um arquivo especial com restrições de acesso. Somente o dono da caixa de correio tem acesso a ela. O administrador também criou uma fila para armazenar mensagens e serem enviadas (FOROUZAN, 2013).

Na figura, quando Josué precisa enviar uma mensagem para Maria, ele executa um programa AU para preparar a mensagem e enviá-la para seu servidor de *e-mail*. O servidor de *e-mail* do lado de Josué usa uma fila para armazenar as mensagens a serem enviadas. A mensagem, no entanto, precisa ser enviada via internet, do lado de Josué para o lado de Maria, usando um MTA. Então, dois agentes de transferência de mensagens são necessários: um cliente e um servidor. Como a maioria dos programas cliente-servidor na internet, o servidor precisa estar

sendo executado o tempo todo, porque ele não sabe quando um cliente vai pedir uma conexão. O cliente, por outro lado, pode ser ativado pelo sistema quando houver uma mensagem na fila de envio. O AU, no lado de Maria, permite que ele leia a mensagem recebida. Posteriormente, Maria usa um cliente MAA para recuperar a mensagem de um servidor MAA em execução no segundo servidor.

Segundo Comer (2016), as normas de protocolo utilizadas para mensagens na internet podem ser divididas em três tipos amplos. Clique nos itens para conhecê-los.

### **Transferência**

Um protocolo utilizado para mover a cópia de uma mensagem de um computador para outro.

### **Acesso**

Um protocolo que permite ao usuário ter acesso à sua caixa de correio eletrônico para ler ou enviar mensagens.

### **Representação**

Um protocolo que especifica o formato da mensagem quando armazenada no disco.

Desse modo, o mesmo *software* original de *e-mail* pode ser dividido conceitualmente em duas partes separadas: uma aplicação de interface de mensagem e uma aplicação de transferência de mensagem.

## **3.2.1 SMTP: protocolo de transferência de mensagens**

O *Simple Mail Transfer Protocol* (SMTP) é o protocolo padrão que o programa de transferência de mensagens usa para transferir uma mensagem para um servidor por meio da internet. De acordo com Comer (2013, p. 59), o SMTP pode ser caracterizado por:

- Segue o paradigma stream
- Usa mensagem de controle textual
- Transfere somente mensagem de texto
- Permite a um remetente especificar os nomes dos destinatários e checar cada um deles
- Envia uma cópia de uma dada mensagem

A primeira característica importante é que o SMTP transfere apenas texto. A segunda é a capacidade de enviar uma única mensagem para vários destinatários em um determinado computador.

Já Barrett (2010) afirma que o SMTP é um protocolo assimétrico de solicitação-resposta para envio de correio eletrônico de um servidor para outro, que não entrega mensagens para a caixa de um usuário e que não permite que usuários remotos colem correio do servidor. Essas duas últimas características se dão pelo fato de o responsável por elas são os protocolos POP3 e IMAP4. O SMTP trabalha acima da camada TCP-IP e recentemente as portas foram alteradas para trabalharem com segurança SSL característica que anteriormente as portas dos protocolos POP na porta 110, IMAP na porta 143 e SMTP na porta 587 não continham. A alteração teve que ser realizada no início de 2018, ficando os protocolos POP3, IMAP e SMTP nas portas 995, 993 e 465, respectivamente, permitindo apenas conexões com autenticação SSL. Desse modo, o SMTP trabalha na camada de Aplicação que define um conjunto de regras para endereçamento, envio e recebimento de correio eletrônico entre os servidores.

Para transferir mensagens, o SMTP utiliza comandos e respostas. Um comando vai de um cliente para um servidor e a resposta vai de um servidor para um cliente, por exemplo. Cada comando ou resposta termina com um indicador de fim de linha formado por dois caracteres. O comando consiste em uma palavra-chave seguida de zero ou mais argumentos. Veja os comandos na tabela a seguir.

Palavra-chave	Argumento(s)	Descrição
HELLO	Name do <i>host</i> do remetente	Identifica a si próprio
MAIL FROM	Remetente da mensagem	Identifica o remetente da mensagem
RCPT TO	Destinatário-alvo	Identifica o destinatário da mensagem
DATA	Corpo do e-mail	Envia a mensagem de fato
QUIT		Encerra a mensagem
RSET		
VRFY	Nome do destinatário	Verifica o endereço do destinatário
NOOP		
TURN		
EXPN	Lista de discussão	Pede ao destinatário para expandir a lista de discussão
HELP	Nome do comando	Pede ao destinatário que envie informações sobre o comando enviado como argumento
SEND FROM	Destinatário-alvo	Especifica que o e-mail seja entregue apenas para o terminal do destinatário, e não para a caixa de correio
SMOL FROM	Destinatário-alvo	Especifica que o e-mail seja entregue ao terminal ou à caixa de correio do destinatário
SMAL FROM	Destinatário-alvo	Especifica que o e-mail seja entregue ao terminal e à caixa de correio do destinatário

Tabela 1 - Comandos SMTP com suas palavras-chave, argumentos e descrição.

Fonte: FOROUZAN, 2013, p. 68.

Já as respostas são enviadas como um código de três dígitos que pode ser seguido por informação textual. A tabela a seguir mostra algumas respostas.

Código	Descrição
Resposta de conclusão positiva	
211	Estado do sistema ou resposta de ajuda
214	Mensagem de ajuda
220	Serviço pronto
221	Serviço fechando canal de transmissão
250	Comando solicitado completado
251	Usuário não local; a mensagem será encaminhada
Resposta intermediária positiva	
354	Iniciar entrada do e-mail
Resposta de conclusão negativa temporária	
421	Serviço disponível
450	Caixa de correio não disponível
451	Comando cancelado; erro local
452	Comando cancelado; espaço do armazenamento insuficiente
Resposta de conclusão negativa permanente	
500	Erro de sintaxe; comando não reconhecido
501	Erro de sintaxe em parâmetros ou argumentos
502	Comando não implementado
503	Sequência incorreta de comando
504	Comando temporariamente não implementado
550	Comando não executado; caixa de correio implementado
551	Usuário não local
552	Ação solicitada cancelada; espaço de armazenamento excedido
553	Ação solicitada não efetuada; nome da caixa de correio não permitida
554	Falha na transação

Tabela 2 - Respostas aos comandos SMTP.

Fonte: FOROUZAN, 2013, p. 68-69.

Agora que sabemos os comandos e as respostas, vejamos como ocorre o processo de transferência de uma mensagem de *e-mail*. O processo consiste em três fases. Clique nas abas e conheça-as.

- **1. Estabelecimento de conexão**

Após um cliente criar uma conexão TCP para a porta 465, o servidor SMTP inicia a fase de conexão, que envolve:

a. Enviar o código 220 para dizer ao cliente que está pronto para receber mensagens. Se o servidor não estiver pronto, ele envia código 421.

b. O cliente envia a mensagem HELO para se identificar, com seu endereço de nome e de domínio. Isso informa ao servidor o nome de domínio do cliente.

c. O servidor responde com o código 250 ou algum outro, dependendo da situação.

- **2. Transferência de mensagens**

Depois que a conexão for estabelecida entre cliente e servidor SMTP, é possível trocar uma única mensagem entre um remetente e um ou mais destinatários.

a. O cliente envia a mensagem MAIL FROM para especificar o remetente da mensagem. Ele inclui o endereço de e-mail do remetente (caixa de correio e nome de domínio). Esse passo é necessário para fornecer ao servidor o endereço de e-mail de resposta para retornar erros e notificações.

b. O servidor responde com código 250 ou algum outro que seja adequado.

c. O cliente envia mensagem RCPT TO (destinatário), a qual inclui o endereço de e-mail do destino.

d. O servidor responde com o código 250 ou algum outro adequado.

e. O cliente envia a mensagem DATA para iniciar a transferência.

f. O servidor responde com o código 354 (iniciar entrada do e-mail) ou alguma outra mensagem adequada.

g. O cliente envia o conteúdo da mensagem em linhas consecutivas. Cada linha termina com um indicador de fim de linha formado por dois caracteres. A mensagem termina com uma linha que contém apenas um ponto.

h. O servidor responde com o código 250 ou algum outro adequado.

- **3. Encerramento de conexão**

Depois que a mensagem for transferida satisfatoriamente, o cliente finaliza a conexão, o que envolve:

a. O cliente envia o comando QUIT

b. O servidor responde com o código 221 ou algum outro adequado.

A figura a seguir demonstra todos os passos descritos anteriormente que descreveram o processo de transferência de mensagens SMTP em três fases.

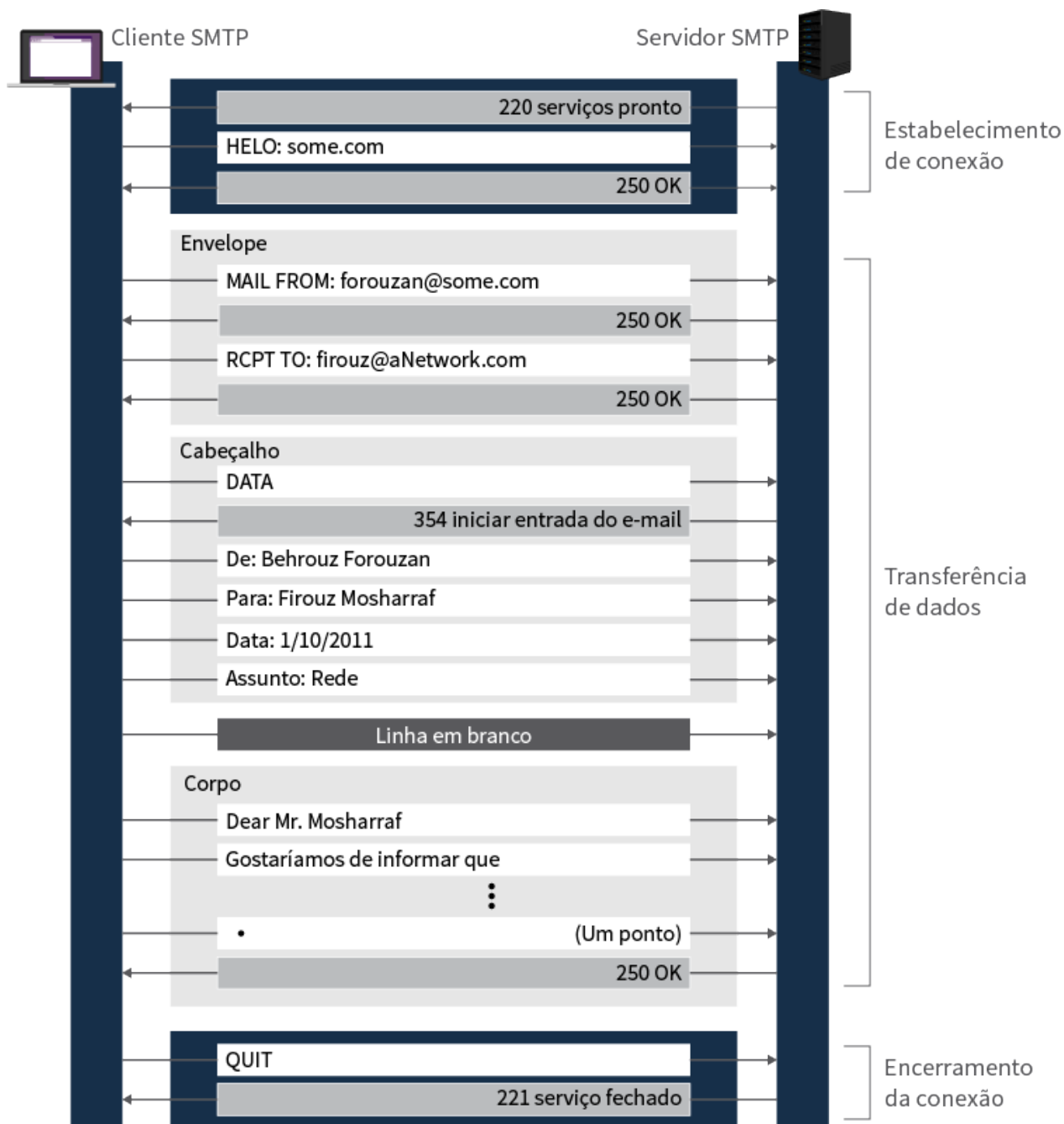


Figura 4 - Esquema que representa o funcionamento da transferência de mensagens com o SMTP.

Fonte: FOROUZAN, 2013, p. 71.

O próximo tópico traz a descrição do funcionamento de outros dois protocolos que complementam o SMTP para que o correio eletrônico funcione corretamente.

### 3.2.2 Protocolos de acesso ao e-mail: POP e IMAP

Em um último estágio, a troca de mensagens de *e-mail* precisa de um agente de acesso às mensagens, isto é, precisa de um protocolo tipo *pull*, que irá promover a recuperação das mensagens do servidor para que o usuário possa acessar através do seu dispositivo de preferência. São protocolos de acesso a mensagens atuais: Protocolo de Agência de Correio, versão 3, do inglês *Post Office Protocol* (POP3) e o Protocolo de Acesso a Correio da Internet, versão 4 (IMAP4 – *Internet Mail Access Protocol*).



Os protocolos de acesso ao *e-mail* possuem, segundo Comer (2016), as características elencadas a seguir. Clique nos itens para ler.

Fornecem acesso a caixa de e-mails do usuário.

Permitem que o usuário veja cabeçalhos, baixe, exclua ou envie mensagens individuais.

Funcionam o cliente no dispositivo do usuário.

Funcionam o servidor no dispositivo onde a caixa de e-mail está operando.

Esses protocolos permitem que a lista de mensagens seja vista pelos usuários sem a necessidade de baixá-las. Isso é útil considerando que uma conexão entre usuário e servidor de *e-mail* pode ficar lenta, o que possibilita o usuário olhar os cabeçalhos e deletar *spams* sem esperar para baixar os conteúdos das mensagens, considerando o acesso por um *smartphone* com acesso precário a internet.

Por ser funcionalmente limitado, o POP3 é simples. Um *software* cliente POP3 é instalado no computador do destinatário e um *software* servidor POP3 é instalado no servidor de *e-mail*. É um protocolo utilizado pelo cliente para buscar as mensagens no servidor do seu domínio. Em configuração padrão, realiza-se a leitura das mensagens disponíveis e logo depois há a remoção do servidor dessa mensagem.

Conforme Forouzan (2013), o acesso ao *e-mail* inicia com o cliente, quando o usuário precisa baixar de seu *e-mail* a caixa de correio que está no servidor. O cliente inicia uma conexão com o servidor na porta TCP 110. Logo após, envia seu nome de usuário e senha para acessar a caixa de correio. O usuário pode listar e recuperar as mensagens de correio, uma por uma.

## VAMOS PRATICAR?



Com certeza, você tem ao menos uma conta de *e-mail*, certo? Em muitos casos, temos duas contas, uma pessoal e outra específica para o trabalho. Gerenciar *e-mails* é uma tarefa fundamental para organizar nossa vida *on-line*, desde mensagens pessoais até cadastros e compras. Ter mais de uma conta torna um cliente de *e-mail* muito útil no dia a dia, pois ele permite enviar, receber, organizar e gerenciar *e-mails*, não só de uma como de diversas contas. São esses *softwares*, chamados de clientes de *e-mail*, que trabalham com os protocolos IMAP e POP. Para conhecer o funcionamento, você pode baixar algum *software* cliente de *e-mail*, como o Outlook, para Windows e macOS, ou Mozilla Thunderbird, para Linux ou Apple Mail nativo do MAC, dentre outros. Pesquise e analise as funções dos clientes de *e-mail* disponíveis para o sistema operacional que você utiliza, baixe e instale em seu computador ou mesmo *smartphone* e após alguns dias utilizando, analise as vantagens e desvantagens em relação à utilização do *webmail*.

O IMAP4 é mais completo e complexo em comparação ao POP3, que apresenta algumas fraquezas. Desse modo, o IMAP4 possui as seguintes funções que o fazem melhor que o POP3, de acordo com Forouzan (2013):

- um usuário pode verificar o cabeçalho do *e-mail* antes de recuperá-lo do servidor;
- permite que o usuário procure por uma sequência específica de caracteres no conteúdo dos *e-mails* antes de recuperá-los;
- possibilita a obtenção do *e-mail* de forma parcial, quando a largura de banda é limitada e o *e-mail* inclui conteúdo multimídia que exija um elevado consumo da capacidade de sua internet;
- possibilita a criação, remoção e renomeação nas caixas de correio no servidor;

- um usuário pode criar uma hierarquia e organização de caixas de correio em uma pasta para armazenamento de *e-mails*.

Assim, o POP3 não possibilita que o usuário organize seus *e-mails* no servidor, porque o usuário não pode ter diferentes pastas. O POP3 também não permite que o usuário verifique parcialmente o conteúdo do *e-mail* antes de obtê-lo.

### 3.2.3 MIME

Através do protocolo adicional MIME (Extensões Multifunção para Mensagens de Internet – do inglês *Multipurpose Internet Extensions*) é possível enviar dados que não sejam apenas ASCII. O MIME transforma dados não ASCII do lado do remetente em dados NVT ASCII e entrega o resultado ao cliente para ser enviado pela internet. Quando chega ao destinatário, a mensagem é transformada novamente para dados originais. A figura a seguir demonstra esse funcionamento.

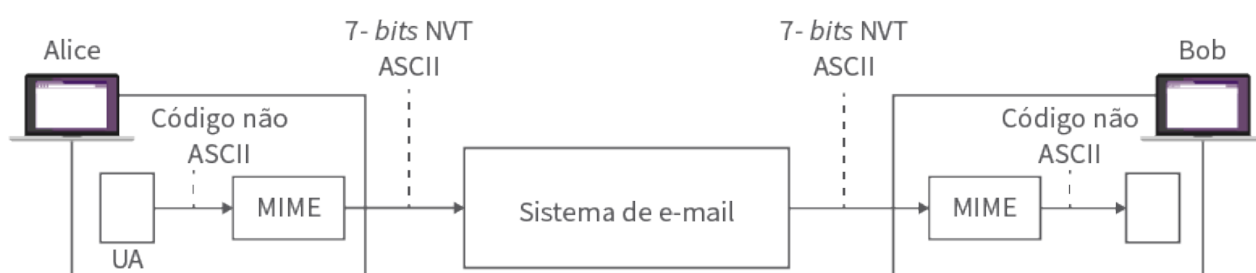


Figura 5 - Funcionamento MIME.  
Fonte: FOROUZAN, 2013, p. 73.

Comer (2016) afirma que o MIME é compatível com os sistemas de *e-mail* que não entendem o padrão MIME ou a codificação. Esses sistemas tratam o corpo da mensagem como um único bloco de texto. O MIME adiciona duas linhas no cabeçalho, uma para declarar que o MIME foi utilizado para criar a mensagem e outra para especificar como a informação MIME está incluída no corpo da mensagem. Por exemplo:

```
MIME-Version: 1.1
```

```
Content-Type: multipart/mixed; boundary=Mime_separador
```

Conforme Forouzan, (2013), depois do cabeçalho, o corpo da mensagem do *e-mail* é exibido. Outros itens de cabeçalhos podem ser inseridos, tais como:

Content-Transfer-Encoding: tipo de codificação

Content-ID: ID da mensagem

Content-Description: explicação textual dos conteúdos não textuais

Desse modo, podemos concluir que o padrão MIME insere linhas extras de cabeçalho para permitir que anexos não textuais possam ser enviados dentro de uma mensagem de *e-mail*. Para isso, esse anexo é codificado como letras imprimíveis e uma linha separadora aparece antes de cada anexo. Segundo Forouzan (2013), como limitação, o MIME pode ser utilizado apenas para a língua inglesa e não pode ser utilizado para enviar arquivos binários de vídeo ou áudio.

## Síntese

Nesta unidade, você aprendeu que o FTP é um protocolo de transferência de arquivos na internet entre dois computadores, geralmente um cliente e outro servidor. Para isso, aprendemos que existem especificidades a

serem tratadas. Também aprendemos como o correio eletrônico, isto é, o *e-mail*, funciona. Enfatizamos que nem toda transferência de arquivo é tratada pelo FTP e uma delas é a de correio eletrônico, pois o protocolo SMTP trata da transferência de mensagens juntamente com a extensão MIME que acrescenta a possibilidade de transferência de arquivos, e os protocolos POP3 e IMAP4 que tratam do acesso ao *e-mail* pelos usuários em seus dispositivos conectados à internet.

Nesta unidade, você teve a oportunidade de:

- compreender o funcionamento do protocolo FTP para transferência de arquivos na internet;
- analisar como os protocolos SMTP, POP3, IMAP4 e MIME trabalham para que mensagens de *e-mail* sejam trocadas na internet;

## Bibliografia

AULA – Prof. Léo Matos – Informática – Serviço FTP. 2013. 1 vídeo (23 min 01 s). Publicado pelo canal Estúdio Aulas Concurso. Disponível em: <https://www.youtube.com/watch?v=JKwk8WFG1zc>. Acesso em: 5 ago. 2019.

BARRETT, D. **Redes de computadores**. Rio de Janeiro: LTC. 2010.

COMER, D. E. **Redes de computadores e internet**. 6. ed. Porto Alegre: Bookman, 2016.

COMO fazer uma conexão FTP. 2017. 1 vídeo (4 min 48 s). Publicado no canal WebLink. Disponível em: <https://www.youtube.com/watch?v=j9Ke6UmheIY>. Acesso em: 5 ago. 2019.

COMO instalar o FileZilla – WebLink. 2017. 1 vídeo (4 min 15 s). Publicado no canal WebLink. Disponível em: <https://www.youtube.com/watch?v=Fy3BPhW-aEs>. Acesso em: 5 ago. 2019.

CURTIN, B. **RFC 2640**: Internationalization of the File Transfer Protocol. Defense Information Systems Agency, 1999. Disponível em: <https://tools.ietf.org/pdf/rfc2640.pdf>. Acesso em: 5 ago. 2019.

FILEZILLA. **FileZilla**. 2019. Disponível em: <https://filezilla-project.org/>. Acesso em: 5 ago. 2019.

FOROUZAN, B. A. **Redes de computadores**: uma abordagem top-down. Porto Alegre: AMGH, 2013.

HOROWITZ, M.; LUNT, S. **RFC 2228**: FTP Security Extensions. Cygnus Solutions, Bellcore, 1997. Disponível em: <https://tools.ietf.org/pdf/rfc2228.pdf>. Acesso em: 5 ago. 2019.

HOUSLEY, R.; YEE, P.; NACE, W. **RFC 2773**: Encryption using KEA and SKIPJACK. SPYRUS, NSA, 2000. Disponível em: <https://tools.ietf.org/pdf/rfc2773.pdf>. Acesso em: 5 ago. 2019.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet**: uma abordagem top-down 6. ed. São Paulo: Pearson Education do Brasil, 2013.

SCHMITT, M. A. R. **Rede de computadores**: nível de aplicação e instalação de serviços. Porto Alegre: Bookman, 2013.