



# SERVIÇOS DA CAMADA DE APLICAÇÃO

## Camada de Aplicação

### Protocolos da Camada de Aplicação

Conforme estudamos, é na camada de aplicação que são executados os processos dos usuários. Nos processos em que eles interagem, realiza-se o que seus usuários esperam. Porém, para que uma aplicação possa trocar dados com outra, é necessário **definir um protocolo de aplicação**.

Mas o que é um **protocolo da camada de aplicação**?

### Resposta

Um protocolo de camada de aplicação define como processos de uma aplicação, que funcionam em sistemas finais diferentes, passam mensagens entre si. Em particular, um protocolo de camada de aplicação define (KUROSE; ROSS, 2013):

- Os tipos de mensagens trocadas, por exemplo, de requisição e de resposta;
- A sintaxe dos vários tipos de mensagens, tais como os campos da mensagem e como os campos são delineados;
- A semântica dos campos, isto é, o significado da informação nos campos;
- Regras para determinar quando e como um processo envia e responde mensagens.

Enquanto o algoritmo da camada de aplicação determina seu funcionamento no ambiente local, o protocolo dela estipula tudo que é necessário para que aplicações em diferentes hospedeiros possam trocar mensagens de maneira estruturada.

Os protocolos públicos da internet são especificados por RFCs. Desse modo, qualquer pessoa é capaz de acessar as especificações de tais protocolos e implementar os próprios softwares.

Para que possamos compreender melhor o funcionamento das camadas de aplicação, analisaremos aquela aplicada na Internet, afinal, trata-se de uma rede de abrangência mundial presente no dia a dia de milhões de pessoas.

## Camadas de Aplicação na Internet

### Serviços da Camada de Aplicação

Já sabemos como funciona a arquitetura do serviço eletrônico. Assista agora a este vídeo para compreender o trabalho dos **protocolos de envio SMTP, POP3 e IMAP**.

Descreveremos a seguir o funcionamento de **três importantes aplicações** das camadas de aplicação na Internet:

#### 1. Serviço Web (Protocolo HTTP)

Implementado pelo protocolo HTTP, que muita gente confunde com a própria Internet.

## 2. Serviço de Correio (Protocolos SMTP, IMAP e POP)

Serviço do correio eletrônico.

## 3. Serviço de Nomes (DNS)

Sistema de resolução de nomes DNS.

# Serviço WEB

## Protocolo HTTP

Definido pelas RFCs **1945** e **2616**, o HTTP (*Hypertext Transfer Protocol*) é o protocolo padrão para transferência de páginas web na internet.

Em 1991, a web foi idealizada no CERN como uma forma de fazer com que grupos de cientistas de diferentes nacionalidades pudessem colaborar por meio da troca de informações baseadas em hipertextos. Em dezembro daquele ano, foi realizada uma demonstração pública na conferência Hypertext 91.

Esse protocolo é constituído de **duas etapas**:

## Etapa 1

Uma página web típica é um documento em formato HTML que pode conter imagens e outros tipos de objetos, como vídeos, texto, som etc.

Para exibir determinada página web, o usuário digita no browser o endereço no qual ela se encontra (ou clica em um hiperlink para essa página), indicando o local em que deve ser buscada. Para que uma página seja transferida do servidor até o browser, um padrão deve ser seguido pelos softwares (cliente e servidor). Ele especifica como o cliente solicita a página, e o servidor a transfere para o cliente.

## Etapa 2

Esse **padrão** é o **protocolo HTTP**. A mensagem HTTP, por sua vez, é carregada pelo outro protocolo: TCP.

Uma interação entre cliente e servidor se inicia quando o cliente envia uma requisição a um servidor. A solicitação mais comum consiste em:

1. Enviar um texto em formato ASCII.
2. Iniciar com a palavra GET.
3. Inserir página solicitada, protocolo utilizado na transferência e servidor a ser contatado.

## Serviço de correio eletrônico

### Correio Eletrônico (e-mail)

Trouxemos um exemplo para esclarecer essa questão:

Os primeiros sistemas de correio eletrônico foram concebidos como um simples sistema voltado para a troca de arquivos. O destinatário da mensagem era especificado na primeira linha do texto.

Bastava então que o sistema procurasse ali para quem a mensagem deveria ser entregue. Porém, com o passar do tempo, surgiram novas necessidades que dificilmente eram atendidas por ele.



A imagem mostra uma janela de aplicativo com o título 'Nova mensagem'. A janela possui uma barra de título cinza com um ícone de envelope à esquerda e botões de minimizar, maximizar e fechar à direita. O corpo da janela é dividido em seções: a primeira seção, rotulada 'Para:', está vazia; a segunda seção, rotulada 'Cc:', também está vazia; e a terceira seção, que ocupa a maior parte da área, contém o texto 'Prezado, envio este e-mail...'.

Em 1982, ainda na era da ARPANET, foram publicadas as **RFCs 821 e 822**, definindo, respectivamente, o protocolo de transmissão a ser utilizado e o formato da mensagem. Entretanto, apesar de ambas resolverem o problema inicial a que se propunham, elas

especificavam que todo o texto deveria ser composto pelo código ASCII.

Tal restrição precisava ser resolvida para ser possível o **envio de mensagens**:

1. Em alfabetos não latinos.
2. Em idiomas sem alfabetos.
3. Que não contêm textos multimídia, como, por exemplo, áudio e vídeo.
4. Com caracteres acentuados.

Para resolver esses novos problemas, foi criada uma **solução** denominada ***Multipurpose Internet Mail Extensions* (MIME)**. O MIME continua utilizando o formato da RFC 822, mas passou a incluir uma estrutura para o corpo da mensagem e **definir regras** para as **mensagens especiais**.

Essa estratégia fez com que tais mensagens pudessem ser enviadas graças à utilização de protocolos e programas de correio eletrônico existentes, havendo somente a necessidade de alterar os programas de envio e recebimento. Atualmente, o protocolo de transmissão *simple mail transfer protocol* (SMTP) é definido pela **RFC 5321**, enquanto o formato da mensagem o é pela **RFC 5322**.

Como é construída a arquitetura do correio eletrônico?

A arquitetura do sistema de correio eletrônico é construída com base em **dois agentes**:

### **Do usuário**

#### **De transferência de mensagens**

O **agente do usuário** é o **programa** que faz a interface do usuário com o sistema de correio eletrônico. É por meio dele que o **usuário**:

1. Faz o envio e o download de mensagens e anexos
2. Lê as mensagens
3. Realiza a pesquisa, o arquivamento e o descarte de mensagens
4. Escreve suas mensagens
5. Anexa arquivos



**Mozilla Thunderbird**



**Microsoft Outlook**



**Eudora**

Já os **agentes de transferência de mensagens** são os responsáveis por fazer com que elas cheguem até o destino. Eles são mais conhecidos como **servidores de correio eletrônico**.



**Postfix**

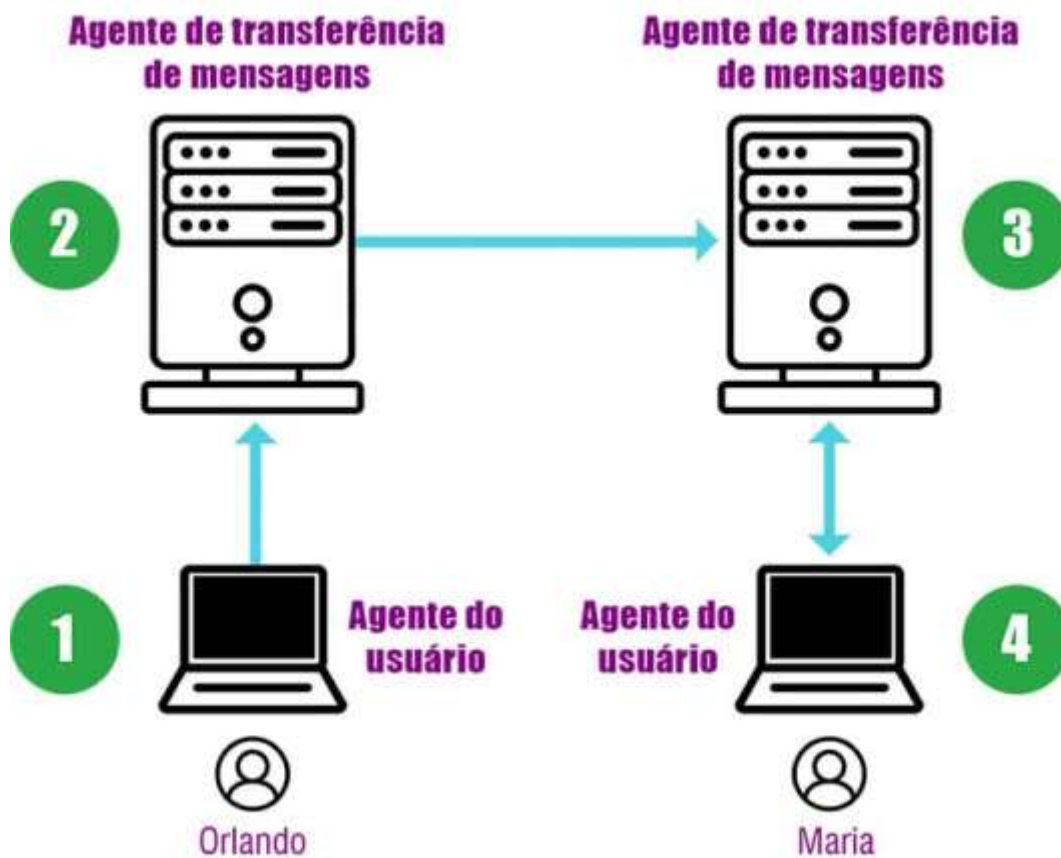


**Zimbra**



**Exchange**

Para entendermos melhor o assunto, analisaremos a seguir a comunicação entre Orlando e Maria. Esse caso explicita uma arquitetura do sistema de correio eletrônico:



A partir da numeração presente na imagem anterior, continuaremos com o exemplo:

1. Orlando deseja enviar uma mensagem para Maria. Após a compor em seu **agente do usuário**, ele solicita seu envio para ela.
2. A mensagem é enviada do agente do usuário de Orlando até seu **agente de transferência de mensagens**, que a recebe, analisa e, em seguida, encaminha-a ao agente de Maria.
3. No destino, tal agente armazena as mensagens que chegam em um local conhecido como **caixa de mensagens** (mailbox), onde cada usuário do sistema possui uma caixa própria.
4. Quando Maria deseja ler suas mensagens, o agente do usuário dela se liga a seu agente de transferência de mensagens e verifica quais estão armazenadas em sua caixa de mensagens.

# Funcionamento do correio eletrônico

Para concluirmos esse estudo, analisaremos **importantes características** dos protocolos apresentados:

## SMTP

O protocolo responsável pela transferência da mensagem até seu destino é o SMTP. Definido pela RFC 5321, ele utiliza o **protocolo de transporte TCP**, obtendo, assim, a **garantia** de que ela será entregue no destino **sem erros**.

O servidor SMTP aguarda por conexões de seus clientes. Quando uma conexão é estabelecida, o servidor inicia a conversação enviando uma linha de texto na qual se identifica e informa se está pronto (ou não) para receber mensagens. Se ele não estiver, o cliente deverá encerrar a conexão e tentar novamente mais tarde.

Caso o servidor esteja acessível, o cliente precisa informar aos usuários a origem e o destino da mensagem. Se o servidor considerar que se trata de uma transferência válida, sinalizará para que ele a envie. Após o envio, o servidor confirma sua recepção e a conexão é encerrada.

## Exemplo

Retomando o caso da comunicação anterior, podemos ver, **na sequência** apresentada adiante, a conversação entre cliente e servidor para estabelecer a transferência da mensagem de orlando@origem.net para maria@destino.net:

**220 Protegido SMTP server**

**hello rayra.origem.net**

**250 Hello rayra.origem.net, pleased to meet you**

**mail from:**

**250 ... Sender ok**

**rcpt to: < maria@destino.net >**

**250 < maria@destino.net >... Recipient ok**

**data**

**354 Please start mail input.**

**subject: Teste de email**

**Primeira linha da mensagem de teste.**



**Segunda linha.**

**Quarta linha.**

.

**250 Mail queued for delivery.**

De acordo com o exemplo de conversação entre cliente e servidor representado acima, é importante observar que na “linha 1” o termo “**Protegido**” refere-se à identificação do servidor que recebe a mensagem, enquanto na “linha 2” o endereço “**rayra.origem.net**” é o nome do hospedeiro que a envia.

## Entrega final

Quando uma mensagem chega ao servidor do destinatário, ela deve ser armazenada em algum local para que possa ser acessada mais tarde (assim que o destinatário estiver on-line). Esse local é a caixa de mensagens.

Como o SMTP é responsável somente pela entrega da mensagem no servidor destino, isso requer a utilização de outro protocolo de modo que o cliente possa buscar suas mensagens no mailbox.

## POP3

A RFC 1939 estipula que o POP3 (*Post Office Protocol version 3*) tem a finalidade de fazer o download das mensagens que se encontram no mailbox do usuário para o sistema local. Caso estejam neste sistema, ele pode utilizá-las em qualquer momento, mesmo sem ter conexão com a internet.

O POP3 é implementado na maioria dos agentes de usuário. Basta configurar os parâmetros de conta e senha do usuário para que o agente faça o download das mensagens. Ele permite o download seletivo delas, assim como apagar as selecionadas no servidor.

## IMAP

Assim como o POP3, o IMAP (*Internet Message Access Protocol*) permite que um usuário tenha acesso às mensagens armazenadas em sua caixa. Porém, enquanto o POP3 é baseado na transferência delas para o sistema local a fim de serem lidas, o IMAP consegue permitir sua leitura diretamente no servidor, dispensando, portanto, a transferência para o sistema local.

Isso será particularmente útil para usuários que não utilizarem sempre o mesmo computador, pois permite que suas mensagens sejam acessadas a partir de qualquer

sistema. Definido pela RFC 3501, o IMAP também fornece mecanismos para criar, excluir e manipular várias caixas de correio no servidor.

## Atenção!

Um webmail não é um protocolo, mas uma **forma oferecida** por alguns sites da web a fim de que os **usuários** possam **ler suas mensagens** de **correio eletrônico**.

Para usar o sistema, o usuário abre uma página web, na qual entra com uma identificação e uma senha. A partir desse momento, ele tem acesso imediato às suas mensagens (de forma parecida com a de um cliente IMAP).

# Serviço de Nomes

## DNS

A comunicação entre hospedeiros na internet ocorre por meio de **endereços binários de rede**. Afinal, para se comunicar com um destino, o hospedeiro precisa conhecer seu endereço.

Entretanto, é bem mais fácil trabalhar com **nomes de hospedeiros** do que com seus endereços de rede. Além de ser muito difícil conhecer todos os endereços dos hospedeiros com os quais precisamos trabalhar, precisaríamos ser notificados toda vez que algum deles mudasse de endereço.

Para resolver esse problema, foi desenvolvido o *Domain Name System* (DNS). Sua finalidade é a criação de um **sistema de nomes** de **forma hierárquica** e baseada em **domínios**. Para acessar um hospedeiro, portanto, basta conhecer seu nome de domínio e fazer uma consulta ao servidor DNS, que é responsável por descobrir seu endereço.

Quais **são os serviços oferecidos** por ele?

## Resposta

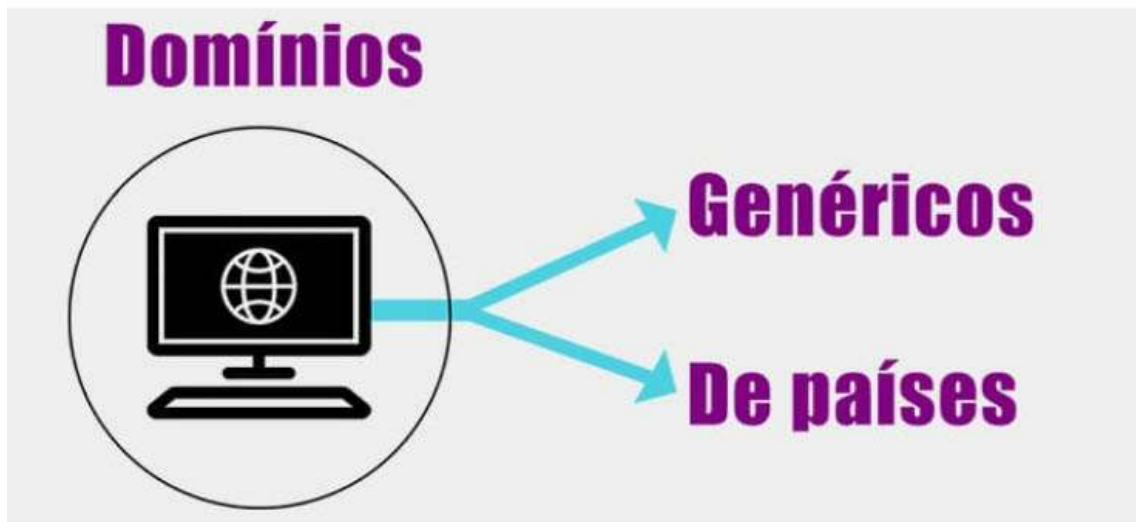
Além do mapeamento de nomes de hospedeiros em endereços IP, o DNS ainda provê:

- Identificação de servidores de correios eletrônicos;
- Apelidos para hospedeiros;
- Distribuição de carga;
- Descoberta de nomes de hospedeiros (mapeamento reverso).

Destacaremos nos tópicos a seguir dois **importantes aspectos** do **DNS**.

# Espaços de Nomes

O espaço de nomes do DNS é dividido em domínios **estruturados em níveis**. Confira a organização do **primeiro nível**:

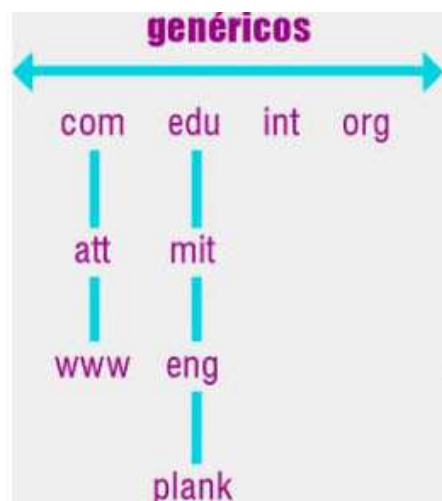


Veja a seguir a diferença entre os domínios genéricos e de países, bem como alguns breves exemplos desses domínios.

#### **Domínios genéricos**

Informam o tipo de organização ao qual o domínio está vinculado. Alguns exemplos são:

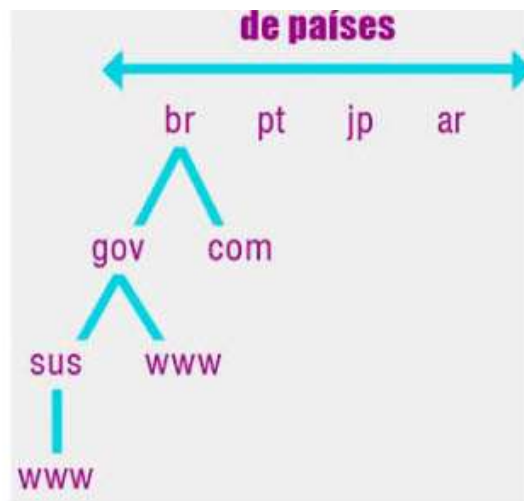
- .com = comercial;
- .edu = instituições educacionais;
- .int = algumas organizações internacionais;
- .org = organizações sem fins lucrativos.



## Domínios de países

Possuem uma entrada para cada país. Alguns exemplos são:

- .br = Brasil;
- .pt = Portugal;
- .jp = Japão;
- .ar = Argentina.



Cada domínio **controla** como **são criados** seus **subdomínios**. Para a criação de um novo domínio, é necessária apenas a **permissão** daquele no qual será incluído.

Não há qualquer restrição sobre a quantidade de subdomínios que podem ser criados dentro de um domínio. Os nomes de domínio não fazem distinção entre letras maiúsculas e minúsculas.

**EDU** e **edu**, por exemplo, são o mesmo.

Os nomes de componentes podem ter **até 63 caracteres**, enquanto os de caminhos completos **não podem ultrapassar os 255**.

O DNS é implementado sobre o protocolo UDP (*User Datagram Protocol*). Trata-se de um protocolo do nível de transporte que não garante a entrega dos dados no destino. Dessa forma, cabe ao software DNS garantir uma comunicação confiável.

## Resolução de Nomes

O espaço de nomes do DNS é dividido em zonas. Independentes, elas possuem um servidor de **nomes principal** e pelo menos um de **nomes secundário**:

## 1. Servidor de nomes principal

Configurado com as informações das zonas sob sua responsabilidade, ele faz o repasse delas para os servidores de nomes secundários.

## 2. Servidor de nomes secundário

Responde pelas zonas caso haja uma falha do servidor de nomes principal.

As **zonas do DNS** definem o que um servidor deve resolver. Se ele for o responsável pela zona pesquisada (servidor autoritativo), deverá fazer a resolução solicitada.

Três principais componentes do DNS

- Registros de recursos armazenados em um banco de dados distribuído;
- Servidores de nomes DNS responsáveis pela manutenção de zonas específicas;
- Solucionadores DNS em execução nos clientes.

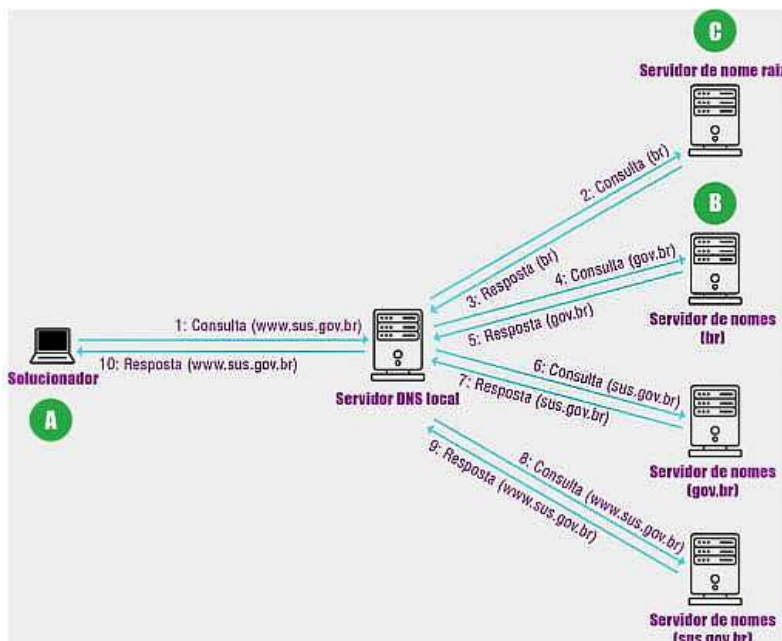
Solucionador X servidor DNS

Quando um solucionador solicita a resolução de um nome para o servidor DNS, pode acontecer o seguinte:

- O servidor DNS é o responsável pela zona: O servidor resolve o nome solicitado e o devolve ao solucionador;
- O servidor DNS não é o responsável pela zona, mas possui a resolução em cache: O servidor envia a resolução ao solucionador;
- O servidor DNS não é o responsável pela zona nem possui a resolução em cache: O servidor precisa realizar uma busca para resolver o nome.

Vamos entender como é feita a busca para a resolução do nome [www.sus.gov.br](http://www.sus.gov.br):

## Funcionamento do DNS



A partir da numeração presente na imagem anterior, continuaremos com o exemplo:

1. Quando a aplicação do cliente solicita a resolução do nome `www.sus.gov.br`, o **solucionador** envia a requisição para o servidor de nomes local, que é o responsável por tratá-la até obter a resposta completa. Desse modo, ele não retorna respostas parciais para o solucionador. A esse tipo de consulta damos o nome de consulta recursiva.
2. Para obter a resposta completa, o servidor de nomes precisa realizar uma série de iterações com outros servidores. Caso nenhuma informação parcial esteja em seu cache, o servidor local primeiramente precisa descobrir quem é o servidor responsável por resolver o **domínio br**.
3. Para isso, ele consulta um servidor de nomes raiz, que indica onde o servidor DNS de “br” pode ser encontrado. O servidor local continua realizando consultas para resolver cada domínio parcial até que haja uma resolução completa. Esse tipo de consulta é conhecido como **consulta iterativa**.

O **excesso de consultas** em um servidor DNS pode levar à **sobrecarga**.

Como evitar esse tipo de problema?

## Resposta

Os servidores devem evitar responder consultas recursivas de clientes não autorizados. Para isso, os administradores de servidores DNS precisam configurar no servidor aqueles autorizados a realizar consultas recursivas. Dessa forma, se houver a consulta de um que não esteja, ela automaticamente será negada.