ESCOLA DOM JOÃO BECKER

TÉCNICO EM INFORMÁTICA

REDES DE COMPUTADORES

**APOSTILA – FUNDAÇÃO BRADESCO**

Luciano Juliano Dutra Escobar

Porto Alegre

2023

Sumário

[1. MÓDULO 1 - PROGRAMAÇÃO ESTRUTURADA VS ORIENTADA A OBJETOS 4](#_Toc132616499)

[1.1. Programação estruturada 4](#_Toc132616500)

[**1.1.1.** **Sequência** 4](#_Toc132616501)

[**1.1.2.** **Decisão** 4](#_Toc132616502)

[**1.1.3.** **Iteração** 5](#_Toc132616503)

[1.2. Programação orientada a objetos 6](#_Toc132616504)

[2. CAPÍTULO 2 9](#_Toc132616505)

[2.1. Tópico 1: Características e topologia 9](#_Toc132616506)

[**2.1.1.** **O que são *LANs*?** 9](#_Toc132616507)

[**2.1.2.** **Topologia** 9](#_Toc132616508)

[2.1.2.1. Topologia em anel 9](#_Toc132616509)

[2.1.2.2. Topologia em barramento 10](#_Toc132616510)

[2.1.2.3. Topologia em estrela 10](#_Toc132616511)

[2.2. Tópico 2: Elementos de uma Rede 10](#_Toc132616512)

[**2.2.1.** **Placa de rede** 10](#_Toc132616513)

[**2.2.2.** **Equipamentos de Distribuição e Conexão de Rede** 11](#_Toc132616514)

[2.2.2.1. *Hub* 12](#_Toc132616515)

[2.2.2.2. *Switch* 12](#_Toc132616516)

[2.2.2.3. Roteador 12](#_Toc132616517)

[2.2.2.4. Repetidor de sinal 13](#_Toc132616518)

[2.2.2.5. *Gateway* 13](#_Toc132616519)

[2.3. Tópico 3: Cabeamento Estruturado 14](#_Toc132616520)

[***2.3.1.*** ***Data Centers*** 14](#_Toc132616521)

[**2.3.2.** **Cabeamento estruturado** 14](#_Toc132616522)

[3. CAPÍTULO 3: PROTOCOLOS DE REDE 21](#_Toc132616523)

[3.1. Tópico 1: Arquitetura de Rede 21](#_Toc132616524)

[**3.1.1.** **Protocolos de redes** 21](#_Toc132616525)

[**3.1.2.** **Arquitetura de rede** 22](#_Toc132616526)

[**3.1.3.** **Modelo *OSI*** 23](#_Toc132616527)

[3.2. Tópico 2: Protocolo TCP/IP 24](#_Toc132616528)

[**3.2.1.** **Protocolo TCP/IP** 24](#_Toc132616529)

[**3.2.2.** **Endereçamento IP** 25](#_Toc132616530)

[**3.2.3.** **Endereçamentos válidos** 27](#_Toc132616531)

[**3.2.4.** **Endereços reservados** 27](#_Toc132616532)

[4. LEITURA RECOMENDADA 29](#_Toc132616533)

1. MÓDULO 1 - PROGRAMAÇÃO ESTRUTURADA VS ORIENTADA A OBJETOS
   1. Programação estruturada

Neste primeiro módulo, conheceremos um pouco mais sobre a programação estruturada (como funciona e qual a sua estrutura) e sobre programação orientada a objetos (como funciona e alguns conceitos).

Antes de entender as diferenças entre estas duas programações, é necessário entender os conceitos de cada uma.

Imagine que quando falamos dessas duas programações, estamos falando de duas pessoas diferentes. É importante saber que apesar dessas duas pessoas terem o mesmo objetivo de vida, que é programar, cada uma tem seu jeito particular de pensar.

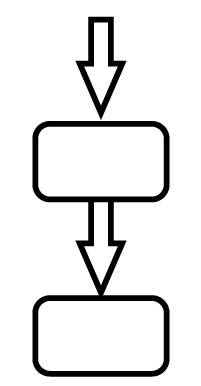
Vamos entender melhor como é a programação estruturada e como ela funciona?

É preciso saber que este tipo de programação segue uma lógica e sequência de pensamentos de uma máquina. Ou seja, a sua lógica é direcionada à linguagem de máquina, já que ela realiza o que foi orientado pelo programador por meio de uma linguagem estruturada.

Para saber mais sobre isso, observe o recurso abaixo. As imagens contidas nele representam fluxogramas (símbolos para representação de regras e procedimentos lógicos que levam a solução de um problema) das interações já mencionadas: sequência, decisão e iteração.

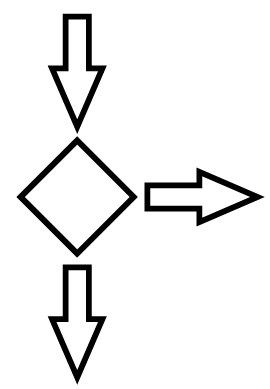
* + 1. **Sequência**

Uma sequência lógica para realizar determinada ação, ou seja, os comandos.



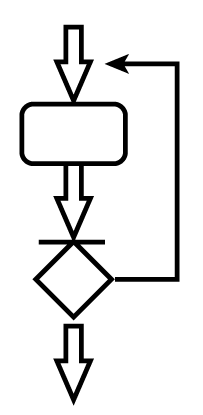
* + 1. **Decisão**

Uma ou mais decisões devem ser testadas pelo programa, por exemplo: If Else, Else If, SWITCH-CASE.



* + 1. **Iteração**

É uma repetição que pode ser interrompida quando a condição for satisfeita, por exemplo: mostre na tela números de 1 a 100, o programa iniciará mostrando o número 1 na telae só finalizará quando chegar ao número 100, que neste caso é a condição a ser satisfeita. As estruturas a serem utilizadas para realizar esta tarefa pode ser a FOR, a WHILE e a DO WHILE.



Diferente da programação orientada a objetos, a estruturada segue processos (sequência contínua de fatos) para conseguir atingir o seu objeto final, que é a solução de um determinado problema. Imagine que existe um programa a ser desenvolvido, onde várias etapas precisam ser cumpridas. Na programação estruturada esse programa é dividido em vários processos pequenos e depois que todos estão resolvidos, os juntamos para formar a resolução final do problema.

Para entender de forma mais clara o conceito apresentado, observe o exemplo a seguir.

Em uma pizzaria, você trabalha como pizzaiolo e o problema que precisa resolver é o de fazer uma pizza de calabresa. Muitos processos devem ser realizados para que você consiga realizar suas tarefas, não é mesmo? Então, para isso você segue um passo a passo, cada uma das atividades, como:

* Preparar a massa;
* Colocar o recheio;
* Ligar o forno, se já não estiver ligado;
* Colocar a pizza para assar; e
* Retirar a pizza do forno depois de 10 minutos.

Pronto, você conseguiu preparar a pizza de calabresa corretamente e é assim que a programação estruturada funciona, ou seja, ela é realizada por sequência.

Neste tópico, você compreendeu acerca do conceito de programação estruturada, a qual é fundamentada em processos destinados à resolução de um determinado problema. Porém, ainda é necessário conhecer a programação orientada a objetos. Desta forma, a seguir, vamos nos aprofundar nesse conceito.

* 1. Programação orientada a objetos

Agora que você compreendeu sobre a lógica por trás da programação estruturada, vamos entender como funciona a programação orientada?

Como você viu, as duas são um modo de programação, mas cada uma possui a sua própria lógica de funcionamento.

Enquanto a programação estruturada obedece a uma lógica de máquina, a programação orientada a objetos funciona de um modo mais similar ao raciocínio humano. Os seus códigos são organizados de acordo com objetos, e não processos, como no caso da programação estruturada.

O modo de programar orientado a objetos surgiu da necessidade de deixar a programação mais fácil, possibilitando ao desenvolvedor entregar um software que satisfaça completamente o cliente. Algo muito difícil e às vezes, quase impossível, com a programação estruturada.

Para que isso ocorra, é necessário que você, como programador, ensine a máquina a pensar como os seres humanos, mostrando como o nosso mundo funciona.

Certo, mas como se faz isso?

Primeiro você precisa compreender alguns conceitos! Vamos lá?

* Objeto: são elementos do mundo real para o mundo da programação. O objeto é criado a partir de uma classe. É algo que se visualiza, se utiliza e assume um papel no domínio do problema. Um exemplo disso é um Fusca 1964.
* Atributos: são as características do objeto. Considerando o exemplo do Fusca 1964, os atributos deles são: ano de fabricação que é 1964, que ele tem um formato arredondado e o motor fica na parte traseira.
* Métodos: são os comportamentos do objeto. Ainda de acordo com o exemplo, o Fusca pode se locomover, transportando pessoas, cargas etc.
* Classes: são os agrupamentos que descrevem todos os objetos de um único tipo. Por exemplo, o fusca pertence à classe dos carros. Não importa a sua marca ou ano. O que importa é que são objetos que se encaixam nesta classe, que serve como molde para a criação dos objetos. É na classe que são moldados os objetos que serão utilizados no programa a ser desenvolvido.

Veja outro exemplo, reunimos alguns atributos que podem ser inseridos em uma classe de um objeto do tipo “pessoa”.

Nas palavras apresentadas, é possível notar alguns atributos relacionados à classe pessoa (bonita, engraçada, esperta, grande, gorda etc.), ou seja, nessa classe, é possível encontrar pessoas (objetos) com tais características (atributos).

Já os métodos são todos os tipos de ação que um objeto pode realizar. No caso de um carro podemos pensar em ações como dar ré, ir para frente, frear etc. No caso de uma pessoa, podemos pensar em ações como comer, correr, falar, andar, entre outras diversas ações que uma pessoa pode realizar.

Até este ponto, você estudou acerca do conceito de método e de instância, os quais, juntos com os de objeto, classe e atributos, são essenciais para a prática da programação orientada a objetos.

Agora que possui uma base teórica mais consistente em relação à programação estruturada e à programação orientada a objetos, a seguir, acompanhe uma reflexão comparativa entre ambas.

* 1. Comparação entre programação estruturada e a POO

Neste tópico, iremos promover uma comparação entre a programação estruturada e a programação orientada a objetos, com o intuito de não só reforçar as características singulares de cada uma, mas também consolidar o que você já sabe em relação a elas.

A primeira coisa que precisamos ter em mente é que os dois tipos de programação possuem suas vantagens e desvantagens de uso pelos desenvolvedores. Além disso, determinar qual a melhor programação a ser utilizada pode depender do tipo de programa que será desenvolvido.

Então, preparado para aprender as diferenças entre elas? Observe a imagem a seguir.



Perceba que na programação estruturada as funções são utilizadas globalmente na aplicação, já em POO, essas funções são aplicadas aos dados de cada objeto.

Agora, observe que a programação estruturada , quando realizada corretamente, tem a probabilidade de ter o desempenho superior ao da orientada a objeto. Isso ocorre, pois ela é procedural[[1]](#footnote-1) e realizada em sequencia, onde cada linha de código é executada logo após a outra, sem desvios. Ao contrário do que ocorre na programação orientada a objetos.

Notou que é possível analisar claramente as diferenças entre esses dois modos de programar quando falamos de dados?

Deste modo, você percebeu que os procedimentos (funções) são os mesmos, o que muda é onde eles são declarados.

Como a programação estruturada possui uma linguagem mais parecida com a da máquina, ela permite que o programador utilize melhor o desempenho do hardware, resultando em um código mais Eficiente.

Até aqui, analisamos as várias vantagens da programação estruturada. Mas você deve estar se perguntando, por que então utilizar a programação orientada a objetos?

Para trazer ainda mais benefícios quando falamos em desenvolver aplicações mais modernas. Pois, hoje em dia o hardware possui uma ótima capacidade de processamento. Logo, o desempenho da aplicação não é mais uma grande preocupação, isso fez com que a POO se tornasse muito utilizada no mundo todo. Outra vantagem apresentada por esta programação é que ela possibilita reutilizar o código, além de ser um código muito próximo ao mundo real, o que fortaleceu o seu crescimento mundialmente.

1. MÓDULO 2 – OS QUATRO PILARES DA POO

No módulo anterior, você estudou sobre a diferença entre programação estruturada e a programação orientada a objetos.

Agora, neste módulo, você aprofundará mais seus conhecimentos, sobretudo, estudando sobre alguns elementos que o ajudará a entender melhor a programação orientada a objeto, explicando seus quatro mais importantes componentes: abstração, encapsulamento, herança e polimorfismo. Vamos começar?

* 1. Tópico 1: Abstração

Na POO, damos o nome de abstração para o processo de aproximar o mundo real do mundo da programação, sendo o seu objetivo, simplificar um problema difícil. Para isso, a abstração leva em conta os aspectos importantes de um determinado ponto de vista e desconsidera os aspectos restantes.

Assim como representado na imagem, a abstração é a criação de uma classe abstrata, que é uma classe incompleta, como se fosse um quebra-cabeça. Assim, esta classe não permite a criação de instâncias e obriga a implementação de todos os métodos da classe assinados como ‘abstract’.

Resumindo, a abstração se concentra apenas nas informações que são importantes para o seu propósito. Dessa forma, ela mantém suas classes o mais simples possível, concentrando-se apenas no que é importante para uma determinada finalidade.

A abstração nada mais é do que você abstrair coisas, ou seja, quando você tem algo muito grande, mas não há necessidade de cadastrar todas as informações.

Vamos entender melhor este conceito a partir do seguinte exemplo:

as, mas as três topologias mais comuns em uma rede *LAN* são: Anel, Barramento e Estrela.

* + 1. **Topologia**

Na tela anterior falamos em topologia. Você sabe o que isso significa? Topologia diz respeito à forma como os equipamentos estão dispostos em uma rede. Clique nas setas abaixo para conhecer as características das três topologias mais comuns em uma rede *LAN*, conforme mencionado.

* + - 1. Topologia em anel

Como o seu nome indica, uma topologia em anel é uma topologia em que os computadores estão ligados sob a forma de um anel ou circunferência. Um único cabo é compartilhado por todos os dispositivos e os dados viajam em uma direção, como uma rodada.

Cada estação está conectada a apenas duas outras estações, quando todas estão ativas. Uma desvantagem é que se, por acaso apenas uma das máquinas falhar, toda a rede pode ser comprometida, já que a informação só trafega em uma direção, que no caso é “circular”.

* + - 1. Topologia em barramento

Imagine uma conversa com diversas pessoas falando ao mesmo tempo. Fica uma confusão, não é mesmo? Para resolver, é preciso um padrão de comunicação: se alguém falar, os outros deverão ficar quietos.

É assim com a topologia barramento. Ela permite que apenas um equipamento envie informações por vez. As outras máquinas ficam "barradas", ou seja, não conseguem enviar qualquer tipo de mensagem durante esse tempo.

* + - 1. Topologia em estrela

Nesse tipo de rede, todas as estações são conectadas por um dispositivo (*HUB* ou *Switch*) principal.

Por meio desse dispositivo todas as estações podem se comunicar entre si, com estações remotas e terminais.

A grande vantagem da sua utilização é que cada estação se comunica diretamente, assim quando ocorre um problema de comunicação com a estação, apenas ela estará parada enquanto todas as outras continuarão trabalhando normalmente.

* 1. Tópico 2: Elementos de uma Rede
     1. **Placa de rede**

Os computadores se comunicam, usando como linguagem os protocolos de rede. Mas você já pensou como ocorre, de fato, essa comunicação?

As pessoas trocam informações pronunciando palavras e, para isso, utilizamos órgãos da fala, certo? Os computadores, por sua vez, utilizam as placas de rede. Elas são o *hardware* que permite a troca de informações entre os computadores. Sua função é controlar todo o envio e recebimento de dados entre um equipamento e outro.

As placas de rede podem ser definidas em função de três características e são elas:

Taxa de transmissão

Quanto à taxa de transmissão, existem basicamente quatro tipos de placas:

* Placa *Ethernet*, que suporta velocidade de transmissão de 10 Mbps, 100 Mbps, 1000 Mbps, 10000Mbps.
* Placa do tipo *Token Ring*, que suporta 4 Mpbs e 16 Mbps até 300Mps dependendo da configuração do cabeamento.
* Placa *Wi-Fi*, que suporta velocidades de transmissão de 2Mbps até um limite ainda não definido, porém atualmente estudos mostram 9,6Gbps.
* Placa de fibra óptica, que suporta velocidades na casa dos Terabits, porém de utilização extremamente restrita, mais comumente nas operadoras brasileiras, girarem em torno de 100mbps a 300mbps.

Cabos de rede suportados

Quanto aos cabos de rede suportados, as placas:

* *Ethernet* são compatíveis com cabos do tipo par trançado e coaxiais.
* *Token Ring* têm conectores apenas para par trançado.
* Fibra óptica.

Barramento

Quanto ao barramento, os tipos de placas são:

* Barramento *ISA (Industry Standard Architecture).*
* Barramento *PCI (Peripheral Component Interconnect).*
* Barramento *PCI-X (Peripheral Component Interconnect Extended).*
* Barramento *AGP (Accelerated Graphics Port).*
* Tecnologia *PCI Express* (mais comum atualmente).
* Barramentos *AMR, CNR* e *ACR*.
* *VESA (VESA Loca Bus).*
* *MCA (Micro Channel Architecture).*
* *EISA (Extended Industry Standard Architecture).*
  + 1. **Equipamentos de Distribuição e Conexão de Rede**

Nas próximas telas, você conhecerá alguns equipamentos que conectam os computadores às redes, transmitindo e recebendo informações. Talvez você já tenha ouvido falar do roteador, que é um dois mais conhecidos, mas você tem ideia, por exemplo, do que seja um *hub*? Ou um *switch*?

Cada vez que você manda ou recebe um e-mail, navega em *sites* ou acessa uma rede social, eles estão lá, cada um cumprindo seu papel, enquanto você simplesmente digita um endereço eletrônico, uma senha ou dá um comando qualquer em um computador.

* + - 1. *Hub*

O *hub*, ou concentrador, é uma espécie de ponto central que interliga diversos computadores. Todos os cabos da rede se concentram no *hub*, cuja principal vantagem é permitir que novos computadores sejam adicionados à rede sem que ela precise ser totalmente desativada.

Em outras palavras: com o *hub*, o sinal se mantém ativo, mesmo que a rede sofra alterações em sua estrutura. Isso acontece porque o *hub* possui um elemento conhecido como repetidor.

Porém, de acordo com Brito (2003), o *hub* tem um ponto fraco.

Ponto fraco do *Hub*

Segundo Brito (2013), o *hub* é um equipamento bem antigo, um dos primeiros usados pelas empresas em redes locais. Basicamente, ele conecta os computadores de uma rede e possibilita a transmissão das informações entre eles.

Porém, é exatamente nesta transmissão que está o seu ponto fraco: ao pegar a informação de um computador para enviar, ele passa esta informação por todos os computadores até encontrar o destinatário final. Isso causa um tráfego enorme na rede, provocando lentidão.

Outro problema é que esse equipamento permite expor os dados a qualquer um que esteja conectado à rede, podendo gerar sérios problemas de segurança. Por esses e outros motivos, o *hub* tem sido substituído pelo *switch*, o qual ainda é usado em menor escala.

* + - 1. *Switch*

Também chamado de comutador, este equipamento possui a mesma função do *hub*. Sua principal diferença está na forma de transmitir os dados: ele recebe a informação e a repassa única e exclusivamente para o destinatário final. Isso é muito importante sob o ponto de vista de segurança, pois evita que os dados transmitidos sejam expostos a outros computadores na mesma rede.

Assim como os *hubs*, os *switches* também possuem portas para conexão dos cabos de rede. A principal diferença está na quantidade das portas.

Alguns modelos possuem 48 portas, por exemplo. Veja na figura um *switch* de 24 portas.

* + - 1. Roteador

Já vimos que o *hub* e o *switch* servem para enviar dados de uma máquina para outra dentro de uma mesma rede local, certo? E quando temos redes metropolitanas, por exemplo, interligando dois prédios em distâncias maiores? Neste caso, cada prédio terá sua própria rede, com seus próprios *switches*, e para conectar essas duas redes será preciso usar um roteador.

Além de permitir a conexão entre duas redes distintas, o roteador é capaz de exercer o mesmo papel do *switch*, mas de forma mais inteligente. Ele estabelece a comunicação entre duas máquinas ou redes e ainda consegue escolher a melhor rota que um dado seguirá no caminho entre elas. Isso faz com que a velocidade dessa transmissão seja maior e o risco de perda de dados diminua consideravelmente.

Saiba Mais!

Para Brito (2013): "O roteador é um equipamento que faz o papel de um intermediador, possibilitando a troca de pacotes entre redes separadas. O uso de aparelhos do gênero é comum em situações em que é necessário interligar redes diferentes, mas que, ao mesmo tempo, precisam ser mantidas isoladas. Na prática, quem está em uma delas não consegue enxergar diretamente a outra, a menos que utilize o dispositivo como caminho para isso".

Já Morimoto (2011) destaca que: "Usando roteadores, é possível interligar um número enorme de redes diferentes, mesmo que situadas em países ou continentes diferentes. Note que cada rede possui seu próprio roteador e os vários roteadores são interligados entre si. É possível interligar inúmeras redes diferentes usando roteadores".

* + - 1. Repetidor de sinal

Sua principal função é receber todos os pacotes emitidos por um computador em uma das redes que o interligam e repeti-los nas demais redes sem realizar qualquer tipo de tratamento sobre eles. Além disso, estes equipamentos amplificam o sinal que recebem de um emissor antes de enviá-lo para o próximo destino. Isto é importante porque permite alcançar grandes distâncias, não importa o quanto um computador esteja longe do outro.

* + - 1. *Gateway*

Quando você aprendeu sobre arquitetura de redes, viu que existe uma padronização, lembra? Cada rede tem apenas um protocolo, mas existem diferentes arquiteturas de rede e protocolos no mundo. Diante disso, você imagina como as diferentes redes se comunicam? A resposta é: usando o *gateway*.

*Gateways* são componentes indispensáveis para alcançar as comunicações entre terminais ligados a redes heterogêneas que usam protocolos diferentes. São equipamentos que podem ser um computador com duas (ou mais) placas de rede, ou um dispositivo dedicado, cujo objetivo é permitir a comunicação entre duas redes com arquiteturas diferentes, como também compartilhar uma conexão com a internet entre várias estações. Permitem traduzir os endereços e os formatos de mensagens presentes em redes diferentes.

* 1. Tópico 3: Cabeamento Estruturado

Grandes empresas, prédios comerciais, universidades ou *data* *centers* exigem que suas redes sejam montadas com a utilização de cabeamento estruturado. Mas você sabe quais são as razões dessa exigência?

Ao instalar um cabeamento estruturado, o principal objetivo é disponibilizar pontos de rede em todos os pontos do prédio em que eles forem necessários. Para isso, os cabos passam pelos corredores e se encontram em um ponto central, onde ficam diversos equipamentos de rede, como *roteadores* e *switches* que farão a distribuição do sinal.

* + 1. ***Data Centers***

*Data Center* é um local que disponibiliza diversos servidores para empresas, universidades, prédios etc. Esses servidores são também chamados de *storages*. Apesar de toda esta facilidade, é importante que o *data center* garanta, entre outras questões, segurança, desempenho e disponibilidade.

Saiba Mais!

Nos prédios, a montagem de uma rede é bem difícil, devido principalmente à arquitetura (altura, salas, andares, etc). Já imaginou se cada sala tivesse sua própria fiação e cabos? Um prédio com fios subindo pelas paredes e saindo pelas janelas para alcançar os postes? E os servidores e todos os outros equipamentos necessários ao funcionamento de telefone e internet?

* + 1. **Cabeamento estruturado**

Para que tudo funcione perfeitamente, é preciso que se instale um sistema organizado, eficiente e seguro. Ou seja, um cabeamento estruturado. Para entender melhor como funciona, acompanhe o infográfico na próxima tela.

1 – Entrada do prédio

Na entrada do prédio está localizada a ligação entre os cabos do prédio e os cabos externos, aqui situa o ponto onde é feito a *interface* (ligação) entre o cabeamento da rua e o cabeamento vertical / horizontal do prédio. Nesta sala estão presentes os cabos, os equipamentos de conexão, os dispositivos de proteção e também outros equipamentos necessários para conectar a instalação externa ao sistema de cabos local, tais como, linhas de telefones, links de internet, cabos ligando o prédio a outros prédios vizinhos e assim por diante.

2 – *Gateway*

Pode ser um computador ou um dispositivo dedicado. Seu objetivo é permitir a comunicação entre duas redes com arquiteturas diferentes, como também compartilhar uma conexão com a internet entre várias estações.

3 – Armário de telecomunicações (AT)

É a área que aloja o equipamento do sistema de cabeamento de telecomunicações. Aqui pode ser considerado um ponto de distribuição, de onde saem os cabos que vão até os pontos individuais, pois seria inviável puxar um cabo separado para cada um dos pontos de rede. Geralmente, usa-se um *rack* contendo todos os equipamentos. Também podem ser instalados em uma sala ou um armário de acesso restrito, contendo terminadores mecânicos, conectores de cruzamento (*crossconnects*), terminadores para o sistema de cabeamento horizontal e vertical (*patch panel*), além de outros dispositivos e equipamentos.

4 – Cabeamento vertical

O cabeamento vertical, também conhecido como *backbone*, trata-se de um acesso que atravessa cada andar do edifício ligando os ATs existentes em cada um dos andares. Os ATs são ligados a um switch ou um roteador, na sala de equipamentos por meios de cabos também verticais. Se a distância permitir, podem ser usados cabos de par trançado, mas é muito comum usar cabos de fibra óptica para esta função.

5 – Cabeamento horizontal

O cabeamento horizontal permite a conexão entre os cabos que ligam o armário de telecomunicações às tomadas onde são conectados os computadores da rede. São cabos permanentes, instalados como parte do cabeamento inicial, os quais continuam sendo usados por muito tempo.

6 – Área de trabalho

São as salas e os ambientes contendo as tomadas, onde ficam os computadores que serão conectados à rede.

7 – Sala de equipamentos

Este é o local para abrigar equipamentos de telecomunicações, de conexão e instalações de aterramentos e de proteção. É considerado distinto do armário de telecomunicações devido à natureza ou complexidade dos equipamentos que contém. Aqui ficam os servidores, switches e os roteadores principais. A ideia é que essa sala seja uma área de acesso restrito, onde os equipamentos fiquem protegidos. Em um prédio, ela fica normalmente no andar térreo.

Na norma ABNT NBR 14565:2012, as tomadas das áreas de trabalho não são chamadas de "pontos de rede", e sim de "pontos de telecomunicações". Isso acontece porque o cabeamento estruturado prevê também o uso de cabos de telefone e de outros tipos de cabos de telecomunicação, não se limitando aos cabos de rede.

A norma EIA/TIA 568A (ANIXTER, 1996) estabelece que os componentes de cabeamento entre a tomada de telecomunicações e as estações de trabalho devem ser simples, baratos e permitam flexibilidade de deslocamento, sem comprometer a conexão física.

O cabo UTP é um cabo sem blindagem que passou a ser o mais utilizado no uso das redes. Cada segmento pode chegar no máximo a 100 metros.

As especificações determinam quantos trançamentos são permitidos por metro de cabo, e o número de trançamentos depende do propósito para qual o cabo será utilizado.

A EIA/TIA (Associação da Indústria Eletrônica e de Telecomunicações) especificou o tipo de cabo UTP a ser usado em várias situações de cabeamento.

Existem cabos de cat1 até cat7. Como os cabos cat5 são suficientes tanto para redes de 100 quanto de 1000 megabits, eles são os mais comuns e mais baratos, mas os cabos cat6 e cat6a estão se popularizando e devem substituir os cabos cat5 nos próximos anos.

* Categoria 1: se refere ao cabo telefônico tradicional, que pode transportar voz, mas não dados. Muitos cabos telefônicos anteriores a 1983, eram cabos dessa categoria.
* Categoria 2: certifica cabos UTP para transmissões de dados de até 4 Mbps. Possui 4 pares de fios.
* Categoria 3: este foi o primeiro padrão de cabos de par trançado desenvolvido especialmente para uso em redes. Certifica cabos UTP para transmissões de dados de até 16 Mbps.
* Categoria 4: certifica cabos UTP para transmissões de dados de até 20 Mbps. Possui 4 pares de fios. Assim como as categorias 1 e 2, a categoria 4 não é mais reconhecida pela TIA e os cabos não são mais fabricados, ao contrário dos cabos de categoria 3, que continuam sendo usados em instalações telefônicas.
* Categoria 5: certifica cabos UTP para transmissões de dados de até 100 Mbps. Possui 4 pares de fios. Atualmente, eles foram substituídos pelos cabos categoria 5e (o "e" vem de *enhanced*/melhorado), uma versão aperfeiçoada do padrão, com normas mais estritas, desenvolvidas de forma a reduzir a interferência entre os cabos e a perda de sinal, o que ajuda em cabos mais longos, perto dos 100 metros permitidos.
* Categoria 5e: certifica cabos UTP para transmissões de dados de até 1 Gbps. Possui 4 pares de fios. Tem a mesma aparência dos cabos de categoria 5, mas possuem uma qualidade melhor.
* Categoria 6: esta categoria de cabos foi originalmente desenvolvida para ser usada no padrão Gigabit Ethernet, mas com o desenvolvimento do padrão para cabos categoria 5 sua adoção acabou sendo retardada, já que, embora os cabos categoria 6 ofereçam uma qualidade superior, o alcance continua sendo de apenas 100 metros, de forma que, embora a melhor qualidade dos cabos cat 6 seja sempre desejável, acaba não existindo muito ganho na prática.
* Categoria 6a: para permitir o uso de cabos de até 100 metros em redes 10G foi criada essa nova categoria de cabos. A categoria 6ª (o "a" vem de *augmented*/ampliado). Utiliza um conjunto de medidas para reduzir a perda de sinal e tornar o cabo mais resistente a interferências. É importante notar que existe também diferença de qualidade entre os conectores RJ-45[[2]](#footnote-2) destinados a cabos categoria 5 e os cabos cat6 e cat6a, de forma que é importante checar as especificações na hora da compra.
* Categoria 7: atualmente muito se fala em categorias superiores de cabeamento metálico, chamadas de Categoria 7 e Categoria 7a. Porém, até o momento, se aconselha o uso de cabos ópticos.

Uma das principais vantagens dos cabos de par trançado é que eles podem ser crimpados rapidamente, no local, usando apenas ferramentas simples, diferente dos cabos de fibra óptica, que precisam de uma preparação mais cuidadosa.

Seja em redes domésticas ou corporativas, crimpar o cabeamento é uma necessidade, pois dependendo da arquitetura do local, precisamos passar os cabos por dentro das tubulações ou pelo teto para evitar acidentes com cabos espalhados pelo chão.

Em redes de 10, 100 e 1000 megabits, o comprimento máximo dos cabos é de até 100 metros, tanto entre o *switch* e o PC quanto entre dois *switches*. Como os *switches* atuam como repetidores, você pode usar um cabo de 100 metros do PC até o *switch* e outro de mais 100 metros do *switch* até o dispositivo seguinte sem comprometer o desempenho da rede.

Crimpar

Ato de plugar o cabo de rede ao conector RJ-45 utilizando alicate específico.

A ferramenta básica para crimpar os cabos é o alicate de crimpagem, representado na figura ao lado. Ele "esmaga" os contatos do conector, fazendo com que as facas-contato perfurem a cobertura plástica e façam contato com os fios do cabo de rede.

Ao crimpar os cabos de rede, o primeiro passo é descascar os cabos, tomando cuidado para não atingir os fios internos, que são bastante finos. Normalmente, o alicate inclui uma saliência no canto da guilhotina, que serve bem para isso.

Existem também descascadores de cabos específicos para cabos de rede, como mostrado na figura.

O cabo de par trançado é formado por 4 pares de cabo, sendo que cada par apresenta uma cor distinta. Então, como há 4 pares de cabo, há 4 cores diferentes: laranja, azul, marrom e verde.

Um dos cabos de cada par tem uma cor sólida e o outro é mais claro ou malhado, misturando a cor e pontos de branco.

Diferenciamos os 8 fios pelas cores.

O segundo passo é destrançar os cabos, deixando-os soltos.

Para simplificar o trabalho, descasque um pedaço grande do cabo, uns 5 ou 6 centímetros, para poder organizar os cabos com mais facilidade e depois corte o excesso, deixando apenas o tamanho suficiente que entrará dentro do conector RJ45.

O próprio alicate de crimpagem inclui uma guilhotina para cortar os cabos. A guilhotina faz um corte reto, deixando os fios prontos para serem inseridos dentro do conector.

Saiba Mais!

Ao crimpar, devemos retirar apenas a capa externa do cabo e não descascar individualmente os fios, pois isso, ao invés de ajudar, serviria apenas para causar mau contato, deixando frouxo o encaixe com os pinos do conector.

A função do alicate é fornecer pressão suficiente para que os pinos do conector RJ-45, que internamente possuem a forma de lâminas, esmaguem os fios do cabo, alcançando o fio de cobre e criando o contato.

Como os fios dos cabos de rede são bastante duros, precisamos aplicar certa força para que o conector fique firme, daí a necessidade de usar um alicate resistente.

Existem dois padrões para a ordem dos fios dentro do conector, o EIA 568A e o EIA 568B. A diferença entre os dois é que a posição dos pares de cabos laranja e verde são invertidos dentro do conector.

Resumindo, temos os esquemas abaixo:

Cabo *straight*

O cabo crimpado com a mesma disposição de fios em ambos os lados do cabo é chamado de cabo "reto", ou *straight*. Este é o tipo "normal" de cabo, usado para ligar os micros ao *switch* ou ao roteador da rede.

Cabo *cross-over*

Existe ainda um outro tipo de cabo, conhecido como *cross-over* (também conhecido como cabo *cross*, ou cabo cruzado), que permite ligar diretamente dois micros, sem precisar do *hub* ou *switch*. Ele é uma opção mais barata quando há apenas dois micros.

Para fazer um cabo *cross-over*, crimpe uma das pontas, seguindo o padrão EIA 568B que vimos anteriormente, e a outra, utilizando o padrão EIA 568A.

É preciso um pouco de atenção ao cortar e encaixar os fios dentro do conector, pois eles precisam ficar perfeitamente retos.

Veja que o que protege os cabos contra as interferências externas são justamente as tranças. A parte destrançada que entra no conector é o ponto fraco do cabo, onde ele é mais vulnerável a todo tipo de interferência.

Por isso, é recomendável reduzir, para o mínimo possível, a parte do cabo sem as tranças.

Ao crimpar os cabos de rede, o primeiro passo é descascar os cabos, tomando cuidado para não atingir os fios internos, que são bastante finos.

Normalmente, o alicate inclui uma saliência no canto da guilhotina, que serve bem para isso.

Os testadores de cabos variam em sofisticação. Eles podem ser simples verificadores de continuidade, que verificam individualmente cada um dos quatro pares, avisando caso algum dos fios esteja partido.

Eles também podem ser certificadores com ferramentas de diagnóstico muito mais sofisticadas, que efetivamente medem a qualidade do cabeamento, atestando que o ponto atende a diferentes padrões.

Na figura, temos um testador simples de continuidade, onde o conjunto de *leds* se acende, mostrando o *status* de cada um dos fios. Se algum fica apagado durante o teste, você sabe que o fio correspondente está partido ou uma das pontas cabeadas está danificada.

1. CAPÍTULO 3: PROTOCOLOS DE REDE
   1. Tópico 1: Arquitetura de Rede
      1. **Protocolos de redes**

Você sabe o que acontece com as informações na comunicação entre os computadores? Primeiramente, elas são convertidas em sinais. Essa conversão é feita pelo protocolo. É ele que controla o envio e o recebimento de pacotes de informações em uma rede, numa linguagem que seja facilmente entendida.

Para você entender melhor, observe o exemplo abaixo.

Como ocorre a comunicação entre duas pessoas? Em geral, elas precisam falar a mesma língua para que se entendam, não é mesmo? Se você quiser falar com alguém que fala outra língua, provavelmente vai precisar de um dicionário ou de um tradutor. O protocolo de rede faz esse papel, só que de forma diferente: ele transforma a linguagem em sinais que viajam pela rede. Ou seja, ele é uma espécie de idioma eletrônico.

Podemos dizer que os cabos, os fios, as antenas e as ondas eletromagnéticas são os veículos que levam/transportam as informações nas redes.

As redes têm uma arquitetura composta de camadas (ou níveis), e é por elas que as informações são transmitidas. Assim, podemos dizer que o protocolo é o responsável pela comunicação e linguagem na rede.

Na internet, os protocolos mais usados para a transmissão são o TCP (Protocolo de Controle de Transmissão) e o IP (Protocolo da Internet). Eles também são conhecidos como TCP/IP, pois trabalham em conjunto. Enquanto o TCP controla a transmissão das informações, o IP especifica o formato.

Na arquitetura de redes, a qualidade precisa seguir os padrões estabelecidos por órgão confiável e reconhecido oficialmente. Por isso, assim como o modelo *OSI*, que padroniza a arquitetura das redes, foram criados padrões para os protocolos TCP e IP na internet.

O responsável por essa padronização é o organismo internacional *IETF (Internet Engineering Task Force).*

Mas por que é tão importante ter padrões de qualidade a serem seguidos? Clique no ícone abaixo para saber.

Saiba Mais!

Imagine o seguinte: uma fábrica no Brasil resolve produzir tomadas com três furos, para se adaptar aos novos plugues de três pinos. Porém, sem se preocupar com os padrões, a tal fábrica produz um lote de tomadas com furos muito próximos um do outro, de maneira que os pinos não se encaixam. Esse lote não poderá ser utilizado, certo?

E se a fábrica tivesse seguido os padrões estabelecidos? Teria tomadas onde os novos plugues se encaixam, certo? Você percebe a importância da padronização? Assim também acontece com as redes de computadores: os padrões servem para garantir que tudo funcione da melhor maneira possível, em qualquer lugar do mundo.

Você se lembra do modelo de transmissão de dados, representado na figura abaixo? Bem, ele também serve para explicarmos a arquitetura de rede. Clique nos números em destaque e acompanhe.

1 – Assim como a construção de uma casa deve seguir um projeto arquitetônico, a estrutura de uma rede também precisa ser projetada. A diferença é que o projeto de uma casa prevê paredes, telhado, iluminação, distribuição de água e assim por diante, enquanto a arquitetura de rede é baseada em um conjunto de camadas.

2 – Nesse modelo, pense em cada camada (ou nível) como uma espécie de software, que se comunica com o software de outra camada por meio de sinais eletrônicos.

3 – Observe as interfaces: elas enviam as informações de uma camada para outra. Veja que cada camada tem um protocolo. Ele é responsável por transformar as informações em sinais, em uma linguagem que todas as camadas entendam.

* + 1. **Arquitetura de rede**

De acordo com uma das maiores referências em rede do nosso país, Luiz Fernando Gomes Soares (1995), a maneira mais eficiente de se estruturar uma rede é por meio de um projeto arquitetural fundamentado em níveis, como vimos anteriormente. Isso porque, uma vez definidas as interfaces, as camadas podem ser alteradas sem causar impacto na estrutura global da rede.

De volta à comparação com a arquitetura de uma casa, isso é o mesmo que ter a oportunidade de reformar um banheiro ou uma cozinha sem causar impacto nos outros cômodos, por exemplo. Assim, podemos concluir que, tanto na casa quanto nas redes, a arquitetura torna mais eficiente o processo de manutenção e evolução.

Padronização

A tecnologia de arquitetura em camadas se desenvolveu até que vários fabricantes de computadores criaram seu próprio modelo. Foi aí que surgiu um problema: cada um tinha sua linguagem. Então, como os equipamentos de diferentes fabricantes conseguiriam se comunicar e trocar informações? Logicamente, era impossível.

Surgiu assim a ideia de criar uma arquitetura única, aberta e pública, que permitisse a conectividade entre todos os equipamentos, não permitindo vantagem de um fabricante sobre o outro. Desse modo, a Organização Internacional para Padronização (em inglês, *OSI*; em português, ISO) definiu um modelo conhecido como Modelo de Referências para Sistemas de Interconexão Abertos.

* + 1. **Modelo *OSI***

O modelo *OSI*, ou ISO, que você já conheceu aqui, é formado por sete camadas (ou níveis), representadas no esquema. Perceba que cada camada deste modelo internacional possui um papel a ser cumprido, e a informação precisa passar por todas elas, uma a uma, de forma que chegue ao outro lado processada e compreendida, sem equívocos.

Saiba Mais!

Você lembra do início do capítulo, quando comparamos o papel do protocolo com a comunicação entre pessoas? O protocolo é o tradutor em forma eletrônica.

Ou seja, neste modelo, os protocolos em cada camada atuam como um tipo de conversor que traduz a informação. O objetivo é garantir que a informação depositada na entrada chegue até o fim do processo de maneira íntegra, sem desentendimentos.

Vamos entender sobre cada um dos níveis e qual o seu papel no caminho que as informações percorrem dentro do modelo *OSI*.

[7 - Aplicação](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_1_08.html#collapse-1)

Este nível representa a interface de comunicação para acesso dos usuários aos serviços de rede.

[6 - Apresentação](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_1_08.html#collapse-2)

Formata os dados a serem apresentados para a camada de aplicativo.

[5 - Sessão](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_1_08.html#collapse-3)

Ajuda o nível de transporte a garantir a entrega das mensagens ao fornecer mecanismos para controlar (estruturar) os caminhos utilizados pela camada de transporte.

[4 - Transporte](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_1_08.html#collapse-4)

Como o nível de rede sozinho não consegue garantir que um *bit* vá chegar ao seu destino, o nível de transporte entra em operação. É quem garante que as mensagens sejam entregues sem erros, na sequência e sem perdas ou duplicações para seu destinatário final.

[3 - Rede](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_1_08.html#collapse-5)

Controla a operação da rede, decidindo qual o caminho físico que os dados devem tomar. Ele toma essa decisão com base em informação de prioridade do serviço e em outros fatores obtidos junto à camada de enlace.

[2 - Enlace](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_1_08.html#collapse-6)

Detecta e corrige erros que por acaso ocorram no nível físico, comunicando esses erros ao nível seguinte (de rede).

[1 - Físico](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_1_08.html#collapse-7)

Fornece, no início, as características mecânicas, elétricas, funcionais e de procedimento para ativar, manter e desativar conexões físicas necessárias à transmissão de dados com o próximo nível (de enlace). Em outras palavras, esse nível detecta os dados, em forma de *bits*, que estão trafegando na rede e os transporta para o nível de enlace.

* 1. Tópico 2: Protocolo TCP/IP
     1. **Protocolo TCP/IP**

Já falamos um pouco do protocolo TCP/IP no início do capítulo, você se lembra? É um conjunto de protocolos que são os principais responsáveis pela transmissão de dados na internet, convertendo as informações em uma linguagem que todos entendam. Podemos afirmar que ele é a língua oficial da internet.

Na verdade, é esse conjunto de protocolos que permite a você mandar e receber e-mails, transferir arquivos, acessar milhões de páginas, ouvir músicas, assistir a vídeos e ter um perfil no *Facebook*, por exemplo.

Saiba Mais!

Comparando com a comunicação humana mais uma vez, é como se existisse um idioma universal que todas as pessoas do mundo conseguissem entender.

No caso da internet, os protocolos TCP/IP são esse idioma, capaz de ser entendido por todos os equipamentos do mundo.

Assim como o modelo ISO, o protocolo TCP/IP também funciona com base em camadas. No entanto, enquanto o modelo ISO possui sete camadas, o TCP/IP tem apenas quatro, cada uma com tarefas a cumprir.

Entenda abaixo e veja quais são as tarefas.

[4 - Aplicação](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_2_02.html#collapse-1)

Toda vez que você usa um computador, celular ou outro aparelho conectado à internet, você acessa esta camada. Ou seja, ela é utilizada para que um equipamento envie informações a outros.

Exemplo: quando você digita um endereço de um *site* na internet, por exemplo, quem recebe sua solicitação é a camada de aplicação. Ao digitar o endereço de uma página com as iniciais *HTTP*, a camada de aplicação entende que é para buscar este endereço em outro computador.

[3 - Transporte](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_2_02.html#collapse-2)

Esta camada recebe as solicitações da camada de aplicação, verifica se são válidas e as "transporta" ao destinatário. Até esse ponto só o TCP trabalha, pois estamos falando sobre a forma de transmissão dos dados.

[2 - Rede](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_2_02.html#collapse-3)

É aqui que o IP entra em ação. Ele possui informações de endereço de origem e de destino, além de outras utilizadas para encaminhar dados entre máquinas na internet. Ou seja, é o IP que "sabe" qual é o equipamento que está enviando a informação e para quem ela deverá ser entregue.

[1 - Acesso à rede](https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3324-IRC_v2/pag/3_2_02.html#collapse-4)

Essa camada especifica os detalhes de como os dados são enviados fisicamente pela internet. Ela identifica que tipo de conexão (par trançado ou fibra óptica, por exemplo) está sendo utilizada pelas máquinas.

* + 1. **Endereçamento IP**

Em uma rede TCP/IP os dados trafegam do *host* de origem ao *host* de destino utilizando o endereço IP. O IP é uma identificação de um dispositivo (computador, servidor, etc) em uma rede local ou pública. Cada computador na internet possui um IP (*Internet Protocol* ou Protocolo de internet) único, que é o meio que as máquinas usam para se comunicar na internet.

Os pacotes são os blocos de informações enviados na internet e são divididos em duas partes: o cabeçalho, que, como um envelope, possui as informações de endereçamento da correspondência, e o dado, que é a mensagem a ser transmitida propriamente dita. Cada pacote possui um endereço de destino e um endereço de origem e, a cada roteador no caminho, o endereço de destino é verificado e o pacote encaminhado para o próximo destino.

Saiba Mais!

*Host* é qualquer máquina ou computador conectado a uma rede, podendo oferecer informações, recursos, serviços e aplicações aos usuários. Os *hosts* variam de computadores pessoais a supercomputadores, dentre outros equipamentos.

Todo *host* na internet precisa obrigatoriamente ter um endereço IP. Os roteadores são elementos interconectados e distribuídos por toda a rede. Eles são como as estações de distribuição de correspondências, distribuindo os pacotes para outros roteadores mais próximos do destino final ou do próprio destino, se for o último elemento do caminho. A descoberta do caminho é realizada automaticamente pelos roteadores, por meio dos protocolos de roteamento. Esses protocolos, de maneira geral, se baseiam no anúncio dos vizinhos de um roteador para os seus adjacentes na rede. Assim, os roteadores descobrem todos os caminhos na internet e para qual vizinho ele deve entregar cada pacote.

O endereço da versão 4 do protocolo IP, é dividido em quatro grupos de 8 *bits*, denominados octetos, ou seja, quatro números de 0 a 255, separados por pontos.

Cada *host* possui endereço único de IP e este endereço é composto por 4 *bytes* ou 32 *bits*.

Assim, para dois computadores se comunicarem utilizando o protocolo TCP/IP, cada um deles precisa ter um endereço de rede diferente, pois é por meio do endereço IP que é possível identificar um determinado computador na rede.

Alguns exemplos de endereço IP seriam:

200.241.16.8

11001000 – 11110001 – 00010000 - 00001000

128.10.2.30

10000000 – 00001010 – 00000010 – 00011110

Saiba Mais!

O exemplo mostra um endereço IP com a sua representação binária e a sua representação decimal. A representação binária é separada em quatro blocos de oito bits, já na sua forma decimal, estes blocos são agrupados e separados por ponto.

Inicialmente, os endereços IP foram divididos em classes para facilitar o roteamento de pacotes.

Nesta divisão, um endereço de classe A, por exemplo, tem o seu primeiro octeto reservado para o endereço de rede e os demais são utilizados para as máquinas.

Já no endereço classe B, os dois primeiros octetos são reservados para a rede e os demais para as máquinas.

No endereço de classe C, os três primeiros octetos são reservados para a rede e somente o último octeto para as máquinas.

Isso significa que os endereços de classe C são usados por pequenas redes, até o limite de 256 computadores (utilizando somente um endereço de classe C).

Já os endereços de classe B são para redes maiores, que suportam até 65.536 computadores na mesma rede e os de classe A suportam até 1.677.216².

Saiba Mais!

Existem ainda os endereços de Classe D, que são utilizados para enviar mensagens *multicasting* (uma mensagem enviada por um único endereço IP para vários destinatários) e o *broadcasting* (uma mensagem enviada através de um único endereço IP para todos os destinatários de uma determinada rede).

* + 1. **Endereçamentos válidos**

Observe que o endereço 127.X.X.X não é mostrado na tabela abaixo, isto porque este endereço foi reservado para *Loopback*. Por convenção, toda máquina rodando TCP/IP possui uma interface de *Loopback*, além das interfaces de rede que possui. Esta interface não conecta a rede alguma. Seu objetivo é permitir teste de comunicação inter-processos dentro da mesma máquina. O endereço de *loopback* utilizado pela quase totalidade das implementações é 127.0.0.1.

* + 1. **Endereços reservados**

Assim como a classe de endereços 127.0.0.0 é reservada para *loopback*, existem outros endereços reservados para utilização em redes internas, não válidos na internet.

Para definirmos se um determinado endereço IP é um endereço de rede ou um endereço de máquina, utilizamos o conceito de máscara. A máscara de uma rede vai nos permitir dizer quais endereços são da rede e quais são de máquinas e dentro de qual rede. O formato de escrita da máscara é o mesmo do número IP.

Para exemplificar, vamos considerar que tenhamos um endereço IP classe C, 200.241.16.X. Quais os endereços de máquina possíveis?

No caso mais simples, temos apenas uma rede e não há sub-redes, logo a máscara será:

E os endereços IPs seriam de: 200.241.16.0 à 200.241.16.255.

Só que devemos nos lembrar sempre de retirar os endereços IP extremos. Portanto, nossos endereços válidos são: 200.241.16.1 à 200.241.16.254.

Nosso endereço de rede seria então: 200.241.16.0.

O IP estático (ou fixo) é um número IP dado permanentemente a um computador, ou seja, seu IP não muda, exceto se tal ação for feita manualmente.

O IP dinâmico, por sua vez, é um número que é dado a um computador quando este se conecta à rede, mas que muda toda vez que há conexão.

O método mais usado para a distribuição de IPs dinâmicos é o protocolo *DHCP (Dynamic Host Configuration Protocol).*

Clique nas áreas destacadas na figura para visualizar as propriedades de Protocolo IP.

Do ponto de vista da internet, as máquinas são muito bem identificadas pelo endereço IP. São esses valores (endereços IP) que realmente viajam no pacote IP, conforme já visto anteriormente. No entanto, do ponto de vista do usuário, esses números são difíceis de serem memorizados. É mais fácil para o usuário guardar o nome [www.microsoft.com.br](http://www.microsoft.com.br) do que o seu respectivo endereço IP 23.96.52.53.

A fim de facilitar a vida do usuário, foi criado o serviço de *DNS (Domain Name System* – Sistema de Nome de Domínio), que associa nomes simbólicos a endereços IP, tornando mais fácil ao usuário final a utilização da internet como um todo.

Uma forma de se estabelecer um sistema de nomes seria criar um banco de dados único que contivesse a relação de todas as máquinas da internet. Esse banco de dados estaria em uma máquina (talvez duas, por questão de segurança) e todas as outras máquinas a consultariam para conseguir traduzir os nomes para endereços IPs.

Qualquer máquina que fosse adicionada a essa internet deveria ser cadastrada nesse servidor central de nomes. Para uma internet pequena, esse sistema funcionaria bem, porém, com a sua ampliação, traria alguns inconvenientes.

Saiba Mais!

Devido às limitações de um sistema único, foi escolhido construir um Banco de Dados de forma distribuída. Hoje, a internet possui, em cada país, um órgão responsável, dentre outras atividades, pela criação de domínios e distribuição de endereços IPs.

No Brasil, esse trabalho é realizado atualmente pelo registro.br.

1. LEITURA RECOMENDADA

Amplie seu conhecimento com algumas leituras.

* ANIXTER. Norma TIA/EIA 568-A: Um Guia de Referência sobre as Normas de Cabeamento de Telecomunicações para Edifícios Comerciais. 1996. Disponível em: [http://celepar7cta.pr.gov.br/portfolio.nsf](http://celepar7cta.pr.gov.br/portfolio.nsf/b239398b4e7d02ec03256d9c003fdcb8/bd4306ad15f6462e03256b98003e2c82/$FILE/EIA-TIA-568-A.pdf). Acesso em: 15 maio 2017. 13h02.
* BRASIL. ABNT. ABNT NBR 14565:2012. Cabeamento estruturado para edifícios comerciais e data centers. ABNT – Associação Brasileira de Normas Técnicas, 133 páginas.
* CUKIER, K. MAYER-SCHÖNBERGER, V. Big Data: Como Extrair Volume, Variedade, Velocidade e Valor da Avalanche de Informação Cotidiana.
* KUROSE, J. F. ROSS, K. W. Redes de Computadores e Internet. 3. ed. São Paulo: Pearson Addison Wesley, 2006.
* MORIMOTO, C. E. Redes: Guia Prático. 2. ed. São Paulo: GDH Press e Sul Editores, 2011.
* PERRENOUD, P. Ciência da Computação: uma visão abrangente. 11. ed. Porto Alegre: Bookman, 2013.
* TANENBAUM, A. S. et al. Redes de Computadores. 5. ed. São Paulo: Pearson Education, 2011.
* TANENBAUM, A. S. Redes de Computadores. Capítulo 5: Arquiteturas de Redes de Computadores. 4. ed. Rio de Janeiro: Campus, 2003.

1. Programação procedural ou programação procedimental é um paradigma de programação que se baseia em procedimentos, que são executados numa sequência. Ela é derivada da programação imperativa, mas acrescentando procedimentos, de forma a organizar o código em blocos que possam ser reutilizados. [↑](#footnote-ref-1)
2. RJ-45: Este conector é mais conhecido por ligar cabeamentos de *Ethernet* tendo cada um 8 condutores. Aproximadamente desde o ano 2.000, é utilizado como conector universal para os cabos que compõem uma rede *Ethernet*, mas possui também outras utilizações. [↑](#footnote-ref-2)