

[◀ VOLTAR](#)

# Fundamentos da rede em Python

Apresentar os fundamentos de uma rede cliente-servidor em Python.

## NESTE TÓPICO

> A REDE TCP/IP

> Dê uma olhada nos links abaixo para saber mais sobre a linguagem Python:



Marcar  
tópico



Olá alunos,

Lembrando: a linguagem Python é uma linguagem de programação sequencial, estruturada e orientada a objetos, principalmente Python 3, que tem todo o seu núcleo apoiado na orientação a objetos.

Então a linguagem possui uma parte de programação sequencial e estruturada que permite a criação de funções.

No início, período conhecido como primeiro paradigma da computação (década de 60), a computação era centralizada em um CPD (Centro de Processamento de Dados), com o domínio das linguagens FORTRAN e COBOL.

No segundo paradigma (década de 70), os terminais aparecem nos departamentos da empresa, conhecidos como terminais “burros” (não tinham CPU e nem memória), pois eram conectados diretamente com o Main-Frame (computador central).

Veja na foto abaixo, um CPD dos anos 60:



Main-Frame: O computador Central

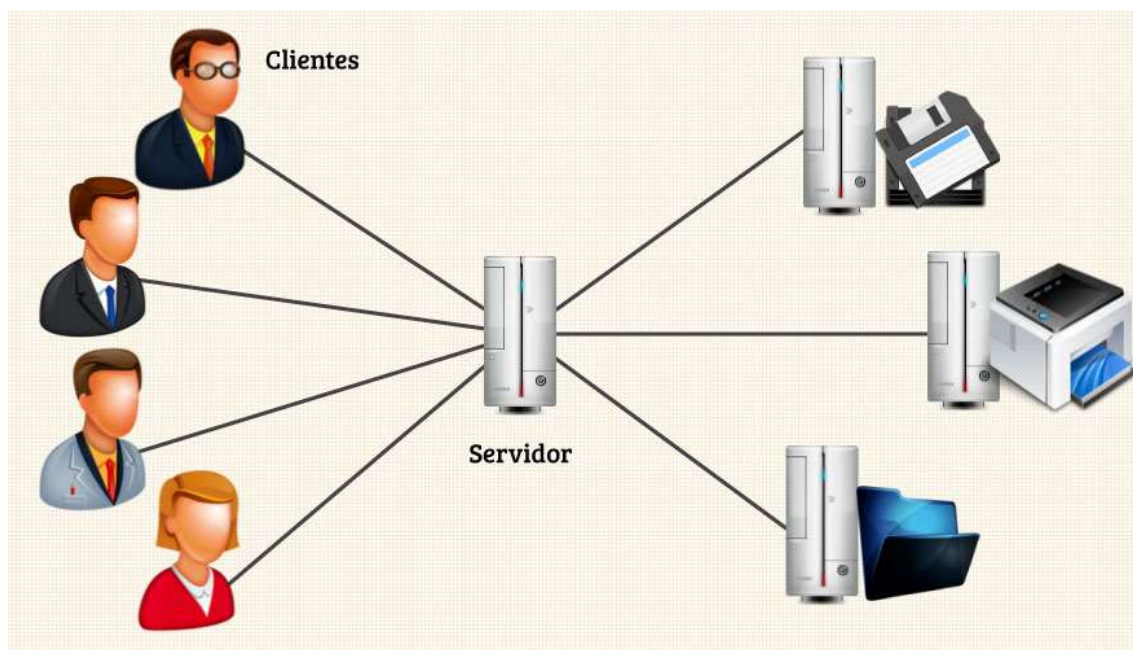
A partir da década de 80, no terceiro paradigma da computação, surge o micro-computador e a comunicação entre computadores é uma realidade frequente.

A partir da década de 90 até os dias atuais, estamos no quarto paradigma da computação: a computação móvel (em qualquer lugar) e o mundo conectado em redes.

Em relação a arquitetura de redes, basicamente hoje, temos dois tipos: a arquitetura de duas e a de três camadas.

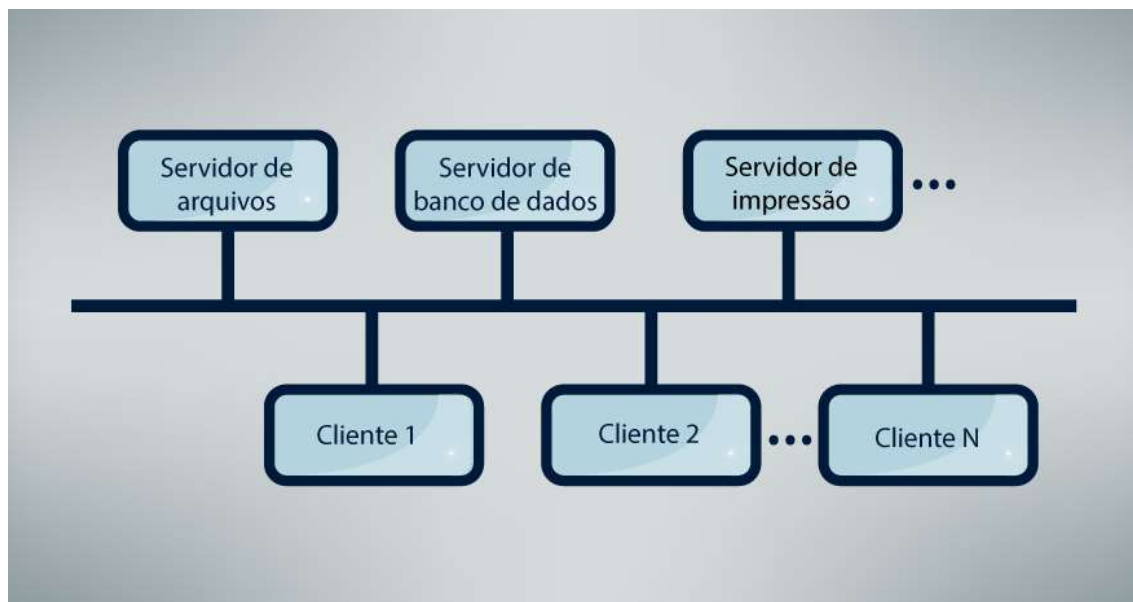
A de duas camadas é conhecida como Client-Server (Cliente-Servidor) e a arquitetura de três camadas: Cliente-WEB (aplicação)-Servidor.

Veja o diagrama abaixo que representa a arquitetura cliente-servidor:



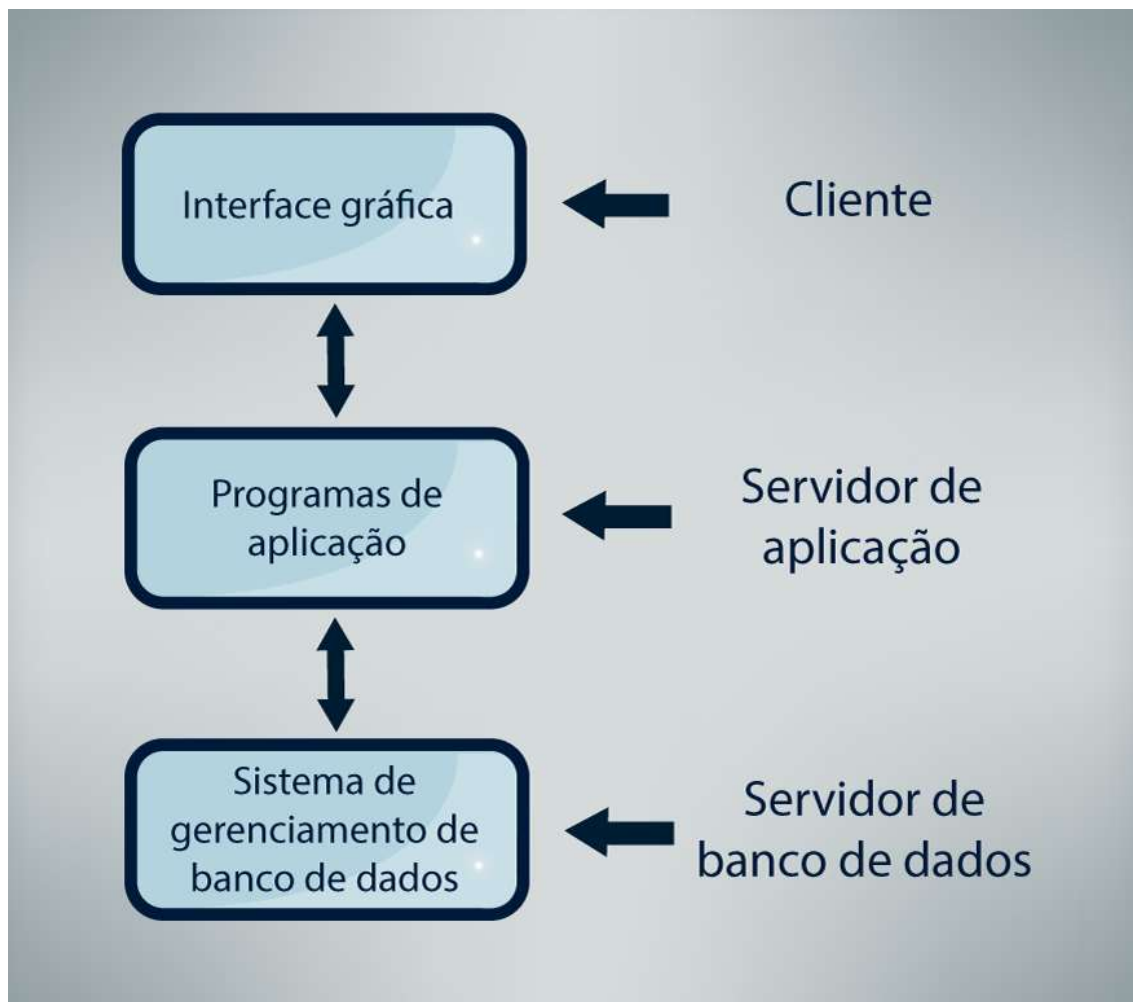
Arquitetura Client-Sever

Veja no diagrama abaixo, a representação da arquitetura cliente-servidor de duas camadas:



Arquitetura Client-Server de duas camadas

Veja no diagrama abaixo, a representação da arquitetura cliente-servidor de três camadas:



Existem aplicações Python tanto para desktop como para servidor. Em aplicações para desktop, Python ainda gatinha, pois as linguagens Java e C# têm um maior emprego, já para servidor, é onde temos maiores aplicações Python.

Desde o primeiro paradigma da computação, a ideia sempre foi colocar os computadores em rede, mas os sistemas operacionais e as aplicações só rodavam naquele hardware, ou melhor eram criados para aquele computador com aquela versão e isto dificultava a comunicação entre os computadores.

Com o sistema operacional UNIX, isto começa a mudar e a ideia de se trabalhar com scripts começa a ser popularizada. Conectar computadores em rede significa a comunicação entre dados e aplicações em hardwares diferentes.

Hoje, você consegue, por exemplo, executar transações na sua conta bancária, utilizando seu tablet, celular ou laptop, através de aplicações com interfaces gráficas, acessando a conta que está em um Main-Frame com aplicações construídas no velho COBOL!.

## A REDE TCP/IP

Toda esta comunicação é estabelecida por protocolos (padrões) e o padrão mais utilizado é o **TCP/IP** (Transmission Control Protocol / Internet Protocol): a informação é empacotada (datagram) e “etiquetada” (endereçada).

Tudo em que há comunicação e troca de dados tem que ser endereçado: os arquivos que estão no seu disco, um pendrive que você “espeta” no seu laptop, uma mensagem que você envia para alguém.

Estes endereços são numerados e estes números obedecem a um padrão, no caso da rede, o padrão é o IP.

No início era um conjunto de 4 blocos com 8 bits cada ( $4 \times 8 = 32$  bits, então  $2$  elevado a  $32 = 4.294.967.296$ ), que hoje é classificado como **IPv4** (versão 4).

Como estamos no quarto paradigma, o mundo ligado em rede e o **IPv4** terá sua quantidade de endereços (mais de 4 bilhões) esgotados devida a demanda por conexões em redes, foi criado e oficializado a partir de 2012, o **IPv6** que passa de 32 para 128bits, o suficiente para que cada pessoa do planeta tenha mais mil endereços IP! ( $2$  elevado a 128 de endereços).

A ideia é que este protocolo padronizado pode ser interpretado pela fonte e pelo destino.

No caso, entre o computador cliente e o servidor. Python tem suporte para as redes de comunicação utilizando a sua classe (módulo) **socket** que suporta tanto os endereços do **IPv4** quanto do **IPv6** com os protocolos **TCP** e também para os protocolos **UDP** (User Datagram Protocol) que são protocolos menos confiáveis (não há garantia que a mensagem que você enviou vai chegar ao destino).

O **socket** divide os dados em pacotes pequenos e desempacota no destino.

Resumindo, o **socket** é uma espécie de porta ou “portal” por onde os dados transitam entre o cliente e o servidor, intermediário entre a camada de aplicação e de transporte (as famosas camadas de rede do modelo OSI que ficam na placa de rede!). Lembrando: a camada de rede de **aplicação** é utilizada pelo HTTP, SMTP, FTP e a camada de **transporte** é utilizada pelo TCP, UDP, NetBEUI.

Veja na figura abaixo, a representação da camada OSI:

Camada OSI

### SAIBA MAIS...

Dê uma olhada nos links abaixo para saber mais sobre a linguagem Python:

<https://www.python.org/doc/> (<https://www.python.org/doc/>)

<https://wiki.python.org/moin/PythonBooks>  
(<https://wiki.python.org/moin/PythonBooks>)

Neste tópico vimos os principais conceitos de redes: as arquiteturas, os principais protocolos, os meios de transmissão e as linguagens utilizadas, como Python.

## Quiz

Exercício Final

Fundamentos da rede em Python

INICIAR ➤

## Referências

SUMMERFIELD, M. *Programação em Python 3*: Uma introdução completa à linguagem Python. Rio de Janeiro Alta Books, 2012. 495 p.

MENEZES, N. N. C. *Introdução à programação com Python*: algoritmos e lógica de programação para iniciantes. 2. ed. São Paulo: Novatec, 2014. 328 p.

SWEIGART, AL. *Automatize tarefas maçantes com Python*: programação prática para verdadeiros iniciantes. São Paulo: Novatec, 2015. 568 p.

PYTHON, doc. Disponível em: <<https://www.python.org/doc/>>. Acesso em: Junho/2018.

PYTHON, books. Disponível em: <<https://wiki.python.org/moin/PythonBooks>>. Acesso em: Junho/2018.



Avalie este tópico



ANTERIOR

Funções de preenchimento

Biblioteca

(<https://www.uninove.br/conhec-a->

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Recebendo uma conexão em um servidor Python

© Todos os direitos reservados

Ajuda?

PRÓXIMO

([https://ava.uninove.br/seu/curso=](https://ava.uninove.br/seu/curso/python))

Python

