

Marcelo Teixeira de Azevedo

# Sistemas distribuídos

**Dados Internacionais de Catalogação na Publicação (CIP)**  
**(Jeane Passos de Souza - CRB 8ª/6189)**

---

Azevedo, Marcelo Teixeira de  
Sistemas distribuídos / Marcelo Teixeira de Azevedo. – São Paulo:  
Editora Senac São Paulo, 2021. (Série Universitária)

Bibliografia.  
e-ISBN 978-65-5536-633-4 (ePub/2021)  
e-ISBN 978-65-5536-634-1 (PDF/2021)

1. Redes de computadores I. Título. II. Série

21-1270t

CDD – 004.6  
BISAC COM043000

---

**Índice para catálogo sistemático**  
**1. Rede de computadores 004.6**

# SISTEMAS DISTRIBUÍDOS

Marcelo Teixeira de Azevedo





## Administração Regional do Senac no Estado de São Paulo

### **Presidente do Conselho Regional**

Abram Szajman

### **Diretor do Departamento Regional**

Luiz Francisco de A. Salgado

### **Superintendente Universitário e de Desenvolvimento**

Luiz Carlos Dourado

## Editora Senac São Paulo

### **Conselho Editorial**

Luiz Francisco de A. Salgado

Luiz Carlos Dourado

Darcio Sayad Maia

Lucila Mara Sbrana Sciotti

Jeane Passos de Souza

### **Gerente/Publisher**

Jeane Passos de Souza (jpassos@sp.senac.br)

### **Coordenação Editorial/Prospecção**

Luís Américo Tousi Botelho (luis.tbotelho@sp.senac.br)

Dolores Crisci Manzano (dolores.cmanzano@sp.senac.br)

### **Administrativo**

grupoedsadministrativo@sp.senac.br

### **Comercial**

comercial@editorasenacsp.com.br

### **Acompanhamento Pedagógico**

Monica Rodrigues dos Santos

### **Designer Educacional**

Diogo Maxwell Santos Felizardo

### **Revisão Técnica**

Rodrigo Assirati Dias

### **Preparação e Revisão de Texto**

Camila Lins

### **Projeto Gráfico**

Alexandre Lemes da Silva

Emília Corrêa Abreu

### **Capa**

Antonio Carlos De Angelis

### **Editoração Eletrônica**

Sidney Foot Gomes

### **Ilustrações**

Sidney Foot Gomes

### **Imagens**

Adobe Stock Photos

### **E-pub**

Ricardo Diana

Proibida a reprodução sem autorização expressa.  
Todos os direitos desta edição reservados à

Editora Senac São Paulo  
Rua 24 de Maio, 208 – 3º andar  
Centro – CEP 01041-000 – São Paulo – SP  
Caixa Postal 1120 – CEP 01032-970 – São Paulo – SP  
Tel. (11) 2187-4450 – Fax (11) 2187-4486  
E-mail: editora@sp.senac.br  
Home page: <http://www.livrariasenac.com.br>

© Editora Senac São Paulo, 2021

# Sumário

## **Capítulo 1** **Introdução aos sistemas distribuídos, 7**

- 1** Conceitos básicos de sistemas distribuídos, 8
  - 2** Modelos de sistemas distribuídos, 11
  - 3** Fundamentos da linguagem de programação Python, 15
- Considerações finais, 17
- Referências, 18

## **Capítulo 2** **Sistemas concorrentes, 19**

- 1** Os sistemas concorrentes e a memória compartilhada, 20
  - 2** Uso de processos e threads, 22
  - 3** Implementação de mutex e semáforo, 24
- Considerações finais, 27
- Referências, 28

## **Capítulo 3** **Comunicação em sistemas distribuídos, 29**

- 1** Exemplos de comunicação em sistemas distribuídos, 30
  - 2** Protocolos de aplicação, 32
  - 3** Conceito de middleware e implementação da RPC, 34
- Considerações finais, 35
- Referências, 35

## **Capítulo 4** **Objetos distribuídos, 37**

- 1** Conceito de objetos distribuídos e arquitetura de objetos, 38
  - 2** Operações com objetos remotos, 42
- Considerações finais, 43
- Referências, 44

## **Capítulo 5** **Sincronização e transação, 45**

- 1** Sincronização e transação em sistemas distribuídos, 46
  - 2** Bancos de dados transacionais, 47
- Considerações finais, 51
- Referências, 52

## **Capítulo 6** **Sistemas distribuídos na web, 53**

- 1** Modelos de sistemas distribuídos na web, 54
  - 2** Implementação de serviços web, 56
- Considerações finais, 59
- Referências, 59

## **Capítulo 7** **Sistemas em nuvem, 61**

- 1** Visão geral sobre os sistemas em nuvem, 62
  - 2** Tipos de serviços em computação em nuvem, 63
- Considerações finais, 67
- Referências, 67

## **Capítulo 8**

### **Aplicação de sistemas em nuvem em serviços de TI, 69**

**1** Soluções de TI *versus* sistemas distribuídos, 70

**2** Exemplos de uso de sistemas distribuídos em sistemas de informação, 73

Considerações finais, 76

Referências, 76

### **Sobre o autor, 79**

# Introdução aos sistemas distribuídos

A computação, em seu estágio inicial, era focada na centralização de seu processamento, e os principais atores desse modelo foram o main-frame e os terminais de acesso. Com o passar do tempo, foi proposta uma mudança dessa arquitetura centralizada e rígida, através da descentralização do processamento, quando surgiram, então, os primeiros sistemas distribuídos.

Desde então, os sistemas distribuídos têm evoluído, sendo uma área da computação que se relaciona com o nosso dia a dia, por meio, por exemplo, de serviços e aplicações web, computação peer-to-peer e redes de sensores.

Este capítulo tem como objetivo contextualizar e apresentar os conceitos essenciais de sistemas distribuídos e, adicionalmente, fornecer uma visão geral sobre a linguagem de programação Python. Para atingir esse objetivo, o capítulo está dividido em três partes:

1. Conceitos básicos de sistemas distribuídos.
2. Modelos de sistemas distribuídos.
3. Fundamentos da linguagem de programação Python.



### PARA PENSAR

Você já parou para refletir como os sistemas distribuídos afetam diretamente a nossa vida? E como estão intimamente conectados com a forma que interagimos com as empresas, o comércio eletrônico e a sociedade?

## 1 Conceitos básicos de sistemas distribuídos

É importante mencionar que existem várias definições de sistemas distribuídos presentes na literatura; dois elementos são comuns a todas elas: computadores e redes. De uma maneira geral, todas as definições concluem que sistemas distribuídos são computadores interligados em rede que se comunicam e coordenam suas ações através de mensagens. Neste capítulo, utilizaremos a definição de Tanenbaum e Van Steen (2008, p. 1): “Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”.

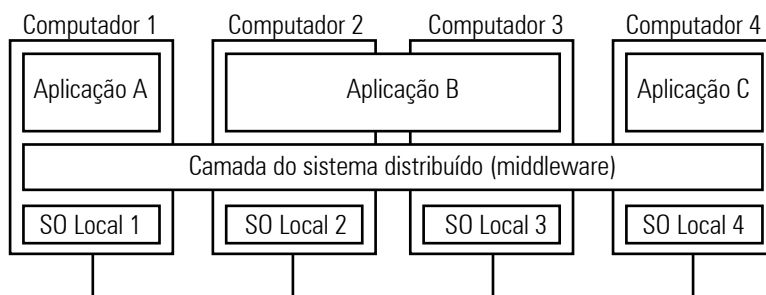
Portanto, o compartilhamento de recursos é um dos principais objetivos da existência dos sistemas distribuídos, e a distância não é um impeditivo para a interligação dos componentes, de modo que eles podem estar separados por qualquer distância: podem estar em diferentes



continentes, regiões e prédios ou mesmo na mesma sala (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Em vista disso, para que seja possível a comunicação entre os computadores e redes heterogêneas, proporcionando uma visão de sistema único, é necessária uma camada de abstração chamada de middleware. Na figura 1, é mostrado o sistema distribuído organizado como middleware, sendo possível visualizar quatro computadores interligados em rede e três aplicações (A, B e C). A aplicação B está sendo executada de maneira distribuída nos computadores 2 e 3. A camada de software (middleware) se estende por todas as máquinas, proporcionando a mesma interface para cada aplicação.

**Figura 1 – Sistema distribuído organizado como middleware**



Fonte: Tanenbaum e Van Steen (2008, p. 2).

Para que fique clara a contextualização de um sistema distribuído, é importante mencionar exemplos reais. Alguns exemplos clássicos são: a internet, as intranets, as tecnologias baseadas em dispositivos móveis e as aplicações e os serviços baseados em computação em nuvem e internet das coisas.

Segundo Coulouris, Dollimore e Kindberg (2007), a internet é um conjunto de redes de computadores de diferentes tipos que são interligadas, permitindo que os usuários, onde quer que estejam, façam uso de diversos serviços, tais como: world wide web (www), transferência de

arquivos, entre outros. Já a intranet faz parte da internet, porém é administrada de forma separada e com limites para que somente usuários internos e com permissão a acessem.



## PARA SABER MAIS

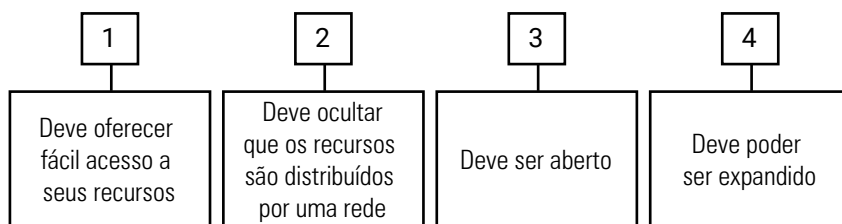
O Nic.Br é uma entidade sem fins lucrativos com funções administrativas e operacionais relativas ao domínio de internet “br”. Em seu canal oficial no YouTube, há uma série de vídeos explicando como funciona a internet, e destacamos estes dois:

- “Como funciona a internet? Parte 1: o protocolo IP”.
- “Como funciona a internet? Parte 2: sistemas autônomos, BGP, PTTs”.

Adicionalmente, recomendamos o vídeo “A internet das coisas”, também disponível no canal do Nic.Br, com um bom conteúdo sobre o tema.

Sistemas distribuídos procuram atender às necessidades dos usuários, tais como desempenho, interoperabilidade, disponibilidade, entre outras. E para atender a essas necessidades de forma que a experiência seja positiva para o usuário, eles devem respeitar alguns requisitos básicos. Segundo Tanenbaum e Van Steen (2008), existem quatro metas importantes que devem ser cumpridas para que o esforço da implementação desses sistemas seja positivo. São elas:

**Figura 2 – Metas para implementação de sistemas distribuídos**



Fonte: adaptado de Tanenbaum e Van Steen (2008, p. 3).

Portanto, a utilização de sistemas distribuídos é diferente das soluções tradicionais de aplicações, devido ao fato de os componentes estarem separados por uma rede, evitando-se, assim, problemas de confiabilidade, segurança, heterogeneidade, latência, largura de banda e administrações.



### IMPORTANTE

Atender aos requisitos dos usuários e de aplicações é um fator fundamental para o sucesso de um sistema distribuído; portanto, é muito importante conhecer os desafios e mecanismos de melhorias. A Amazon Web Services é uma empresa líder no segmento de aplicações que utiliza sistemas distribuídos e conhece muito bem os desafios para uma correta implementação. Em seu site oficial, disponibilizou o artigo “Desafios com sistemas distribuídos”, de Jacob Gabrielson, para leitura.

## 2 Modelos de sistemas distribuídos

Como já abordado anteriormente, os sistemas distribuídos são computadores independentes conectados por uma rede de comunicação, sendo que para o usuário final trata-se de um único dispositivo proveniente do serviço. Em relação ao objetivo desses sistemas, Tanenbaum e Van Steen (2008) dividem os sistemas distribuídos em três tipos:

- **Sistemas de computação distribuídos:** relacionados às atividades de computação de alto desempenho.
- **Sistemas de informação distribuídos:** utilizados em empresas com grandes quantidades de aplicações que precisam ser integradas (interoperabilidade).
- **Sistemas distribuídos pervasivos:** utilizados em cenários que requerem mobilidade.



## PARA SABER MAIS

O site [top500.org](http://top500.org) mantém uma lista com os 500 computadores mais rápidos do mundo, na qual o Brasil aparece com o supercomputador Santos Dumont (SDumont), que possui um total de 18.144 núcleos de CPU, distribuídos em 756 nós computacionais com 64 GB de memória cada. Mais informações sobre o SDumont podem ser obtidas a partir de uma pesquisa simples em sites de busca, usando as palavras-chave “sdumont” e “machine”. Acesse o site [top500.org](http://top500.org), clique em “TOP500 LIST” e fique por dentro de tudo que acontece no universo dos supercomputadores.

## 2.1 Sistemas de computação distribuídos

Os sistemas de computação distribuídos têm como principal objetivo a execução de tarefas de computação de alto desempenho, divididas em dois subgrupos: computação de cluster e computação em grade.

A computação de cluster é constituída de hardwares, geralmente do mesmo tipo, denominados mestre e nó de comunicação, conectados por meio de uma rede local de alta velocidade, na qual um programa é executado em paralelo em várias máquinas.

Já na computação em grade os dispositivos são diferentes em relação ao hardware, ao software e à tecnologia empregada, com recursos de distintas organizações agrupados para permitir a colaboração de um grupo de pessoas.



## NA PRÁTICA

O projeto [Beowulf.org](http://Beowulf.org) é um exemplo prático da computação de cluster. Trata-se de clusters de alta performance que utilizam hardware comum, rede e software aberto, com o objetivo de baratear custos em relação às soluções de mercado. Fazendo uma pesquisa simples em sites de busca com as palavras-chave “beowulf” e “cluster”, pode-se obter mais informações sobre o tema.

Já o projeto SETI@Home é um programa de computação em grade da Universidade da Califórnia que visa obter evidências de vida inteligente extraterrestre utilizando o processamento de mais de 330 mil computadores. Mais detalhes sobre esse projeto podem ser obtidos com uma pesquisa simples em sites de busca pela expressão “seti@home”.

---

## 2.2 Sistemas de informação distribuídos

A arquitetura cliente/servidor proporcionou a utilização de programas que eram executados diretamente no servidor, e, posteriormente, esse mesmo servidor enviava uma resposta aos seus clientes remotos.

Com o avanço da computação e a evolução do setor, esse modelo passou a não atender de forma satisfatória aos requisitos de novas demandas dos clientes, requerendo que as aplicações soubessem se comunicar entre si. Portanto, os dados envolvidos nas transações deveriam ser distribuídos entre vários servidores.

Assim, foi requerida a presença de um monitor de transação (TP monitor), responsável pela coordenação e execução da transação, com a função de permitir o acesso a vários servidores e bancos de dados.

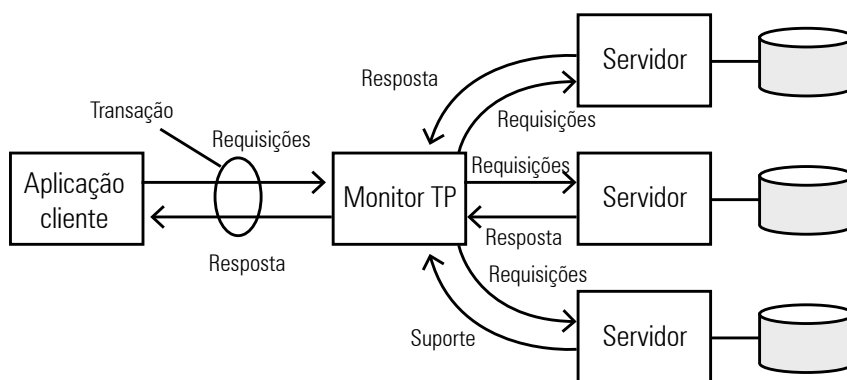
Na figura 3, é descrita uma aplicação cliente que envia uma requisição, a qual chega ao monitor TP, que direciona para um dos três servidores disponíveis, retornando em seguida uma resposta para a aplicação cliente.

A necessidade de diálogo entre aplicações originou modelos diferentes de comunicação, destacando-se (TANENBAUM; VAN STEEN, 2008):

- **Chamadas de procedimento remoto (remote procedure calls – RPC):** trata-se do envio de uma requisição a um outro componente, executando uma chamada de procedimento local.

- **Invocações de método remoto (remote method invocations – RMI):** trabalha da mesma forma que as RPC, porém funciona com objeto em vez de com aplicações.
- **Middleware orientado à mensagem (message-oriented middleware – MOM):** trabalha com o envio de mensagens (publicar/subscrever) para pontos lógicos de contato.

Figura 3 – Monitor TP em sistemas distribuídos

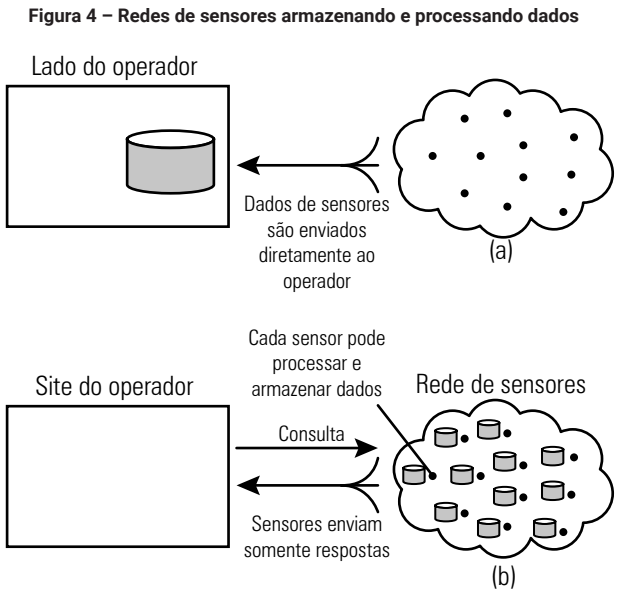


Fonte: Tanenbaum e Van Steen (2008, p. 14).

## 2.3 Sistemas distribuídos pervasivos

Os sistemas distribuídos pervasivos relacionam-se a sensores, que possuem características próprias, tais como: tamanho pequeno, alimentação por bateria, mobilidade, ausência de controle administrativo humano e conexão sem fio. Tais características são diferentes das que vimos até agora, que eram focadas em nós fixos, conexão permanente e alta qualidade da rede. Redes de sensores são um exemplo de sistema distribuído pervasivo, que pode monitorar condições ambientais, tráfego, aplicações médicas, entre outras aplicações.

A figura 4 apresenta uma rede de sensores em duas partes. A parte (a) da figura exibe uma rede de sensores na qual os dados gerados pelos nós sensores são enviados diretamente para um banco de dados. Na parte (b), é possível visualizar os nós sensores armazenando as informações localmente, sendo que os sensores somente enviam respostas às consultas que são realizadas diretamente a eles.



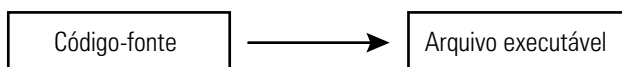
Fonte: Tanenbaum e Van Steen (2008, p. 18).

### 3 Fundamentos da linguagem de programação Python

Uma linguagem de programação é uma forma de escrita padronizada, com regras bem definidas, que tem como objetivo principal enviar instruções a computadores (BARBOSA, 1996). O seu estágio inicial é a elaboração de um código-fonte, que, após compilado, é transformado em um programa, conforme ilustrado na figura 5, na qual uma seta no meio de dois retângulos representa o processo de compilação. O

primeiro quadrado é o do código-fonte, que, após ser submetido ao processo de compilação, gera um arquivo executável. Atualmente, existem inúmeras linguagens de programação; entre as existentes, nosso foco será a Python.

**Figura 5 – Processo de compilação**



Segundo Banin (2018), Python é uma linguagem de programação conhecida por ser objetiva e intuitiva, sendo desenvolvida, ampliada e utilizada desde a década de 1990. Durante a sua trajetória de desenvolvimento, reuniu as seguintes características:

- portabilidade;
- código livre;
- simplicidade e robustez;
- fácil de aprender;
- grande aplicabilidade.

Em 2020, a linguagem Python encontrava-se nas versões 2.7.x e 3.8.x. As versões possuem diferenças fundamentais e incompatibilidade entre si, sendo que a Python 2.7.x já é considerada uma versão legada, e a Python 3.8.x, a nova versão oficial.



### **IMPORTANTE**

No site oficial da Python, no item “glossário”, encontra-se a descrição sobre a incompatibilidade de versões e toda a parte de documentação da versão 3.8.x. Lá, também encontra-se uma ferramenta de conversão do código-fonte da versão Python 2.7.x para Python 3.8.x.



A versão da Python para diversas plataformas pode ser baixada no site da organização que gerencia seu desenvolvimento, através do link “download”, disponibilizado na página principal. Também é possível utilizar IDEs (integrated development environment) para incorporar recursos e funcionalidades no desenvolvimento de programas.



## IMPORTANTE

No site da Real Python, pesquisando a palavra-chave “install”, é possível verificar um roteiro simples de instalação da Python para diversas plataformas (Windows, macOS, Linux, iOS e Android).

Caso não deseje instalar a Python em seu computador, você pode utilizar a versão web, disponibilizada no site oficial da empresa Repl.it, no item “languages”.

As IDEs disponíveis para Python podem ser verificadas na wiki oficial da Python, através de uma busca pelo termo “IDE”.

Na figura 6, é possível visualizar o código-fonte e a execução resultante da compilação de um programa simples, que mostra a mensagem “Bem-vindo ao ambiente Python!”, utilizando a IDE repl.it.

Figura 6 – Programa simples



## Considerações finais

Neste capítulo, foram apresentados os principais conceitos relacionados a sistemas distribuídos: sua definição, caracterização e tipos. Além disso, vimos como os sistemas distribuídos estão intimamente conectados com o nosso dia a dia e inseridos definitivamente na vida moderna, com exemplos e reflexões de como eles afetam nossa vida.

Também foi explorada genericamente a linguagem de programação Python e sua importância no contexto da computação e do desenvolvimento de softwares.

## Referências

BANIN, Sérgio Luiz. **Python 3**: conceitos e aplicações – uma abordagem didática. São Paulo: Érica, 2018. 263 p.

BARBOSA, Valmir. **An introduction to distributed algorithms**. Boston: MIT Press, 1996.

BASHAM, Bryan; SIERRA, Kathy; BATES, Bert. **Use a cabeça!**: servlets & JSP. Rio de Janeiro: Alta Books, 2005.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas distribuídos**: conceitos e projeto. 4. ed. Porto Alegre: Bookman, 2007.

LYNCH, Nancy. **Distributed algorithms**. Nova York: Morgan Kaufmann Publishers, 1996.

TANENBAUM, Andrew S.; VAN STEEN, Maarten. **Sistemas distribuídos**: princípios e paradigmas. São Paulo: Prentice Hall, 2008.