

Ed u c a ç ã o  
P r o f i s s i o n a l  
P a u l i s t a

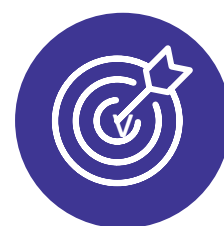
Técnico em  
Ciência de  
Dados

# Jargões e inglês da ciência de dados

## Vocabulário, em inglês, na ciência de dados (parte II)

### Aula 1

Código da aula: [DADOS]ANO1C1B2S10A1



## Objetivos da aula:

- Conhecer os termos técnicos, em inglês, utilizados na área de ciência de dados e inteligência artificial.



## Competências da unidade (técnicas e socioemocionais):

- Aprender a pensar de forma crítica e analítica;
- Trabalhar em equipe em busca de um objetivo;
- Desenvolver *networking*, curiosidade e autonomia;
- Expandir o conhecimento e vocabulário técnico.



## Recursos didáticos:

- Recurso audiovisual para exibição de textos, vídeos e imagens;
- Acesso ao laboratório de informática e/ou internet.



## Duração da aula:

50 minutos.

## Cronograma da aula de hoje

Vocabulário, em inglês, na ciência de dados (parte II)

- ✓ Introdução de vocabulário técnico em inglês
- ✓ Exemplos e seus significados
- ✓ Atividade

# Introdução

A ciência de dados é um campo de estudos multidisciplinar que usa técnicas científicas, métodos, algoritmos e sistemas para extrair conhecimento e *insights* de dados estruturados e não estruturados.

A capacidade de entender e comunicar, eficazmente, essas técnicas e métodos é essencial para qualquer cientista de dados. E uma grande parte disso é o **domínio do vocabulário técnico em inglês**.



## Importante

Em geral, para as carreiras técnicas, o **inglês** é o idioma mais importante. Isso ocorre pela facilidade de escrita, leitura, contato (estamos, diariamente, em contato com o idioma!), pela produção acadêmica e diversos outros aspectos.



# ***Big data (grandes volumes de dados)***

**Big data** é um termo que se refere a grandes volumes de dados (tanto estruturados quanto não estruturados), que são tão extensos que os métodos tradicionais de processamento são inadequados para lidar com eles. Podem incluir imagens, textos, vídeos, logs de transações e muito mais

Esses dados são gerados por várias fontes, como:

- ✓ redes sociais;
- ✓ dispositivos móveis;
- ✓ sensores;
- ✓ máquinas.



## Exposição

# **Big data (grandes volumes de dados)**

Vamos conhecer os termos em inglês associados ao *big data*:

	Significado
Hadoop	Um <i>software</i> de código aberto projetado para armazenamento e processamento de grandes conjuntos de dados distribuídos.
Spark	Uma plataforma de computação em <i>cluster</i> rápida, projetada para processamento de <i>big data</i> .
Data mining (mineração de dados)	Processo de descoberta de padrões em grandes conjuntos de dados.
Data warehouse (depósito de dados)	Sistema de armazenamento de dados em grande escala que atua como um repositório central para integrar dados de uma ou mais fontes distintas.
ETL ( <b>extract, transform, load</b> - extração, transformação, carga)	É o processo que permite que as empresas reformatem e limpem dados, assim como movam-nos de várias fontes e os carreguem em outro banco de dados, data mart ou data warehouse, para análise, ou em outro sistema operacional, para apoiar um processo de negócios.
Data wrangling (preparo de dados)	Processo de transformação de dados similar a ETL, porém com foco maior na limpeza e preparação de dados para análise.

Elaborado especialmente para o curso.

## Exposição

# Machine learning (aprendizado de máquina)

	Significado
NoSQL (not only SQL)	Um tipo de banco de dados capaz de lidar com grandes volumes de dados estruturados e não estruturados.
Data lake (lago de dados)	Um repositório de armazenamento que contém uma grande quantidade de dados brutos em seu formato nativo.
Data analytics (análise de dados)	O processo de examinar conjuntos de dados para tirar conclusões sobre as informações que contêm.
MapReduce	Modelo de programação para processar e gerar grandes conjuntos de dados com um método paralelo e distribuído.
Hive	Uma infraestrutura de <i>data warehouse</i> que fornece consulta e análise de dados armazenados no <i>hadoop</i> .
Pig	Uma plataforma de alto nível para criar programas <i>mapreduce</i> com o <i>hadoop</i> .

Elaborado especialmente para o curso.



## Exposição

# Machine learning (aprendizado de máquina)

	Significado
HBase	Um banco de dados distribuído e não relacional construído no <i>hadoop</i> .
Zookeeper	Serviço de coordenação de alta <i>performance</i> para sistemas distribuídos.
Kafka	Plataforma de <i>streaming</i> de eventos distribuída, capaz de lidar com trilhões de eventos por dia.
Data scientist (cientista de dados)	Profissional responsável por extrair <i>insights</i> e conhecimentos de grandes volumes de dados.

Elaborado especialmente para o curso.

## Exposição

# Machine learning (aprendizado de máquina)

	Significado
Distributed systems (sistemas distribuídos)	Sistemas de <i>software</i> e <i>hardware</i> de rede que funcionam em conjunto para um objetivo comum.
Cluster computing (computação em cluster)	Uso de vários computadores conectados para formar um único e poderoso sistema de computação.
Cloud computing (computação em nuvem)	Entrega de recursos de computação, como servidores, armazenamento, bancos de dados, rede, <i>software</i> , análise e inteligência, pela Internet ("a nuvem").
Batch processing (processamento em lote)	Um método de execução de tarefas de computação ( <i>jobs</i> ), em que os dados são processados em grandes quantidades em um período de tempo específico.

Elaborado especialmente para o curso.

## ***Machine learning* (aprendizado de máquina)**

	Significado
Real-time processing (processamento em tempo real)	Um método de processamento de dados em que as informações recebidas são processadas e transmitidas quase instantaneamente.
Structured data (dados estruturados)	Dados que são organizados de forma pré-definida, ou que seguem um modelo específico, como dados armazenados em um banco de dados relacional.
Unstructured data (dados não estruturados)	Dados que não possuem uma estrutura ou modelo predefinido, como <i>e-mails</i> , <i>posts</i> de redes sociais, vídeos, fotos etc.
Semi-structured data (dados semi-estruturados)	Dados que não são totalmente estruturados, mas contêm marcadores para separar elementos de dados e impor hierarquias de registros e campos.

Elaborado especialmente para o curso.

## Exposição

# ***Machine learning* (aprendizado de máquina)**

	Significado
Data cleansing (limpeza de dados)	Processo de detecção e correção (ou remoção) de erros e inconsistências em um conjunto de dados ou base de dados.
Predictive analytics (análise preditiva)	A prática de extrair informações de conjuntos de dados existentes para prever futuras tendências ou comportamentos.
Data privacy (privacidade de dados)	Envolve o tratamento de dados com respeito à sua relevância e seu propósito, garantindo que os dados estejam protegidos contra acesso ou uso não autorizado.
In-memory computing (computação em memória)	Armazenamento de informações na memória principal de um servidor dedicado, em vez de um disco rígido, para produzir tempos de resposta mais curtos.

Elaborado especialmente para o curso.



## Exposição

# Machine learning (aprendizado de máquina)

	Significado
Scalability (escalabilidade)	A capacidade de um sistema lidar com o aumento da carga de trabalho, adicionando recursos ao sistema.
Data governance (governança de dados)	É o gerenciamento geral da disponibilidade, usabilidade, integridade e segurança dos dados usados em uma empresa. É um conjunto de processos, funções, políticas, normas e métricas que garantem o uso efetivo e eficiente dos dados.
Business intelligence (inteligência de mercado)	Envolve a coleta, integração, análise e apresentação de dados empresariais para ajudar na tomada de decisões. A função de analista de <i>business intelligence</i> é a deriva deste conceito.

Elaborado especialmente para o curso.





Vamos  
fazer uma  
**atividade**

## Leitura de texto em inglês

1. Em duplas, façam a leitura do texto:  
SHAHZAN. *Big data explained in plain and simple english*. Medium, 2 maio 2019. Disponível em: <https://medium.com/swlh/big-data-explained-38656c70d15d>. Acesso em: 26 fev. 2024.
2. Extraia as principais ideias discutidas pelo autor.
3. Faça um resumo do que entendeu do texto e uma lista de tópicos e seus significados em inglês.



Vamos  
fazer um  
**quiz**

# Qual é o foco do processo de *data wrangling*?

Armazenamento  
seguro de dados

Limpeza e preparação  
de dados

Criptografia de dados  
sensíveis

Distribuição de dados  
em rede





Vamos  
fazer um  
**quiz**

## Qual é o foco do processo de *data wrangling*?



**Armazenamento  
seguro de dados**

**Limpeza e preparação  
de dados**



**Criptografia de dados  
sensíveis**

**Distribuição de dados  
em rede**



### RESPOSTA CORRETA!

*Data wrangling* é o processo de transformação de dados com foco maior na limpeza e preparação de dados para análise.



Vamos  
fazer um  
**quiz**

## O que é *hadoop*?

Um banco de dados  
NoSQL

Uma linguagem de  
programação

*Software* de código  
aberto para *big data*

Uma plataforma de  
análise de dados





Vamos  
fazer um  
**quiz**

## O que é *hadoop*?



Um banco de dados  
NoSQL

Uma linguagem de  
programação



*Software* de código  
aberto para *big data*

Uma plataforma de  
análise de dados



### RESPOSTA CORRETA!

*Hadoop* é um *software* de código aberto projetado para armazenamento e processamento de grandes conjuntos de dados distribuídos.



Vamos  
fazer um  
**quiz**

## O que significa ETL?

*Encode, transfer, load*

*Extract, transform, load*

*Execute, translate, link*

*Encrypt, transmit, launch*





Vamos  
fazer um  
**quiz**

## O que significa ETL?



***Encode, transfer, load***

***Extract, transform, load***



***Execute, translate, link***

***Encrypt, transmit, launch***



### RESPOSTA CORRETA!

ETL é o processo que permite às empresas mover dados de várias fontes, reformatá-los e limpá-los, e carregá-los em outro banco de dados, *data mart*, ou *data warehouse* para análise, ou em outro sistema operacional para apoiar um processo de negócios.



O que nós  
**aprendemos?**

© Getty Images

## Hoje, desenvolvemos:

- 1** Introdução ao vocabulário técnico em inglês com foco nos termos técnicos utilizados na área de ciência de dados e inteligência artificial.
- 2** Conhecimento sobre *big data* e *machine learning*, termos mais usados e o que significam.
- 3** Leitura de artigo em inglês e identificação das principais informações sobre *big data*.



# Saiba mais

O *Dicionário do Programador* é o quadro semanal em que você aprende mais sobre termos, tecnologias ou palavras do nosso maravilhoso mundo da programação! Para saber mais sobre *big data*, assista:

CÓDIGO FONTE TV. *Big data* // Dicionário do Programador. Disponível em:  
<https://youtu.be/lpfE8B9H9cl>. Acesso em: 26 fev. 2024.



# Referências da aula

AMARAL, F. *Introdução à ciência de dados: mineração de dados e big data*. Rio de Janeiro: Alta Books, 2018.

CHACON, S.; STRAUB, B. *Pro Git*. USA: Apress, 2022. Disponível em: <https://github.com/progit/progit2-pt-br/releases/download/2.1.46/progit.pdf>. Acesso em: 27 fev. 2024.

JUPYTER. *Jupyter project*, [s.d.]. Disponível em: <https://jupyter.org/>. Acesso em: 27 fev. 2024.

MATPLOTLIB. *Matplotlib documentation*, [s.d.]. Disponível em: <https://matplotlib.org/>. Acesso em: 26 fev. 2024.

NUMPY. *NumPy documentation*, [s.d.]. Disponível em: <https://numpy.org/>. Acesso em: 27 fev. 2024.

PYDATA. *Pandas documentation*, [s.d.]. Disponível em: <https://pandas.pydata.org/>. Acesso em: 27 fev. 2024.

PROVOST, F.; FAWCETT, T. *Data science para negócios: o que você precisa saber sobre mineração de dados e pensamento analítico de dados*. Rio de Janeiro: Alta Books, 2018.

RODRIGO, T. *Mapas mentais: nosso jeito de raciocinar, desenhando*. Medium, 27 jul. 2020. Disponível em: <https://tiagorodrigos.medium.com/mapas-mentais-nosso-jeito-de-raciocinar-desenhando-c6cd4d125272>. Acesso em: 26 fev. 2024.

SHAHZAN. *Big data explained in plain and simple english*. Medium, 2 maio 2019. Disponível em: <https://medium.com/swlh/big-data-explained-38656c70d15d#:~:text=What%20is%20Big%20Data%3F,within%20a%20given%20time%20frame>. Acesso em: 26 fev. 2024.

SIMPLILEARN. *Big data in 5 minutes | What is big data? | Big data analytics | Big data tutorial*. Disponível em: <https://youtu.be/bAyrObI7TYE>. Acesso em: 26 fev. 2024.

Identidade visual: Imagens © Getty Images



Ed u c a ç ã o  
P r o f i s s i o n a l  
P a u l i s t a

Técnico em  
Ciência de  
Dados

**Educação  
Profissional  
Paulista**

Técnico em  
**Ciência de  
Dados**

# **Introdução à Ciência de Dados, ferramentas e jargões da área**

## **Introdução à informática**

### **Aula 2**

**[DADOS]ANO1C1B2S11A2**

# Exposição



## Objetivos da Aula

Introduzir os conceitos de sistemas operacionais.



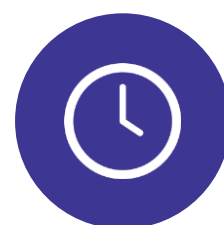
## Competências da Unidade (Técnicas e Socioemocionais)

- Compreender e dominar os principais conceitos e noções de sistemas operacionais na informática;
- Analisar situações em que cada tipo de sistema operacional pode ser utilizado e identificar os principais equipamentos e finalidades de aplicação para seus usos;
- Trabalhar em grupos.



## Recursos Didáticos

- Recurso audiovisual para a exibição de textos, vídeos e imagens;
- Artigos selecionados para leitura.



## Duração da Aula

50 minutos.



# Conceitos fundamentais de sistemas de arquivo

Os sistemas de arquivos são vitais para os sistemas operacionais, gerenciando o armazenamento, a organização e o acesso aos dados em mídias de armazenamento.

Eles determinam a nomenclatura, a alocação e a segurança dos arquivos dentro de uma estrutura de diretórios. **Conheça cada um:**

NTFS ( <i>New Technology File System</i> )	ext4 ( <i>Fourth Extended File System</i> )	FAT32 ( <i>File Allocation Table 32</i> )
Utilizado pelo Windows, é conhecido por suportar grandes volumes e complexas configurações de permissões, tornando-o adequado para ambientes empresariais.	Popular no Linux, é reconhecido por sua robustez e eficiência no gerenciamento de muitos arquivos pequenos, ideal para servidores e sistemas de banco de dados.	Aloca o primeiro bloco de memória grande, oferecendo um bom equilíbrio entre performance e simplicidade.

Elaborado especialmente para o curso.

# Conceitos fundamentais de sistemas de arquivo

A estrutura de um sistema de arquivos inclui metadados que armazenam informações sobre os arquivos, como:

- tamanho;
- data de criação;
- permissões;
- localização;

Esses dados são **essenciais** para o gerenciamento eficiente dos arquivos e a interação do usuário/aplicativos com o sistema de arquivos.

Além disso, sistemas de arquivos contemporâneos incluem funcionalidades como compressão e criptografia de dados, **otimizando** o uso do espaço e resguardando informações sensíveis.

# Exemplos de conceitos fundamentais de sistemas de arquivo

### Exemplo 1:

O sistema de arquivos NTFS no Windows inclui definir permissões complexas de arquivo e tem suporte para **grandes volumes de armazenamento**, adequado para empresas e servidores.

### Exemplo 2:

O sistema de arquivos **FAT32** é muito usado em dispositivos de armazenamento removíveis, como pen drives, devido à sua compatibilidade universal entre sistemas operacionais, mesmo tendo limitações de tamanho de arquivo e volume.

# Exemplos de conceitos fundamentais de sistemas de arquivo

### Exemplo 3:

No Linux, o **sistema de arquivos ext4** é amplamente utilizado por sua eficiência em gerenciar um vasto número de pequenos arquivos.



### Tome nota

No Linux, o ext4 é ideal para servidores de arquivos e sistemas de bancos de dados.



# Implementação de sistemas de arquivos

A implementação dos sistemas de arquivos define como os dados são escritos e lidos nos dispositivos de armazenamento. Existem métodos de alocação de arquivos, como:

- **Alocação contígua:** simples e eficaz para acessos sequenciais, mas pode causar fragmentação.
- **Alocação encadeada:** utilizada no FAT32, permite fácil expansão dos arquivos, mas acessos aleatórios são mais lentos.
- **Alocação indexada:** usada no ext4, conta com **inodes** que direcionam aos blocos de dados, agilizando o acesso direto aos arquivos.



### Tome nota

**Inodes** são estruturas essenciais em vários sistemas de arquivos UNIX, como o ext3 e ext4 no Linux. Eles guardam dados importantes de um arquivo, menos o nome e o conteúdo. "Inode" vem de "index node".

# Implementação de sistemas de arquivos

Os sistemas de arquivos precisam **gerenciar a integridade e a organização dos dados**, especialmente em situações de falhas do sistema ou desligamentos inesperados.

Sistemas atuais usam o ***journaling***, que anota alterações antes de efetivá-las no sistema principal. Isso permite restaurar o sistema de modo rápido e seguro após falhas, mantendo a consistência dos dados.



### Tome nota

O ***journaling*** é uma técnica em sistemas de arquivos que registra as mudanças em um *log* antes de aplicá-las ao sistema principal. Assim, se houver uma falha durante uma gravação, o sistema operacional verifica o *log* (journal) para finalizar as alterações pendentes e prevenir a corrupção de dados.

# Exemplos de implementação de sistemas de arquivos

### Exemplo 1:

O método de alocação contígua é simples e facilita o acesso sequencial, mas pode levar à fragmentação ao longo do tempo.

Isso pode resultar em ineficiência para sistemas antigos ou sistemas embarcados com necessidades de armazenamento previsíveis.

### Exemplo 2:

Sistemas de arquivos que utilizam alocação encadeada, como o FAT32, permitem a expansão fácil de arquivos adicionando-se mais blocos à cadeia.

Contudo, o acesso aleatório é mais lento, já que cada bloco precisa ser lido em sequência.



# Exemplos de implementação de sistemas de arquivos

## Exemplo 3:

O uso de **inodes** no sistema de arquivos **ext4** permite uma alocação indexada, na qual cada arquivo tem um inode que aponta para seus blocos de dados.



### Tome nota

Alocação indexada eficiente facilita o acesso direto e a expansão de arquivos.

# Gerenciamento de espaço em disco

**Objetivo:** a administração eficaz do espaço em disco é crucial para **maximizar a eficiência do armazenamento** e assegurar a integridade dos dados.

Métodos como *bitmap* ou listas encadeadas são vitais para monitorar os blocos utilizados e disponíveis.

Além disso, sistemas de arquivos com **journaling**, a exemplo do ext4 e NTFS, reforçam a confiabilidade por meio de um log de transações que registra as alterações planejadas nos arquivos.

**Permitindo que:**

👍 O sistema se recupere de forma estável após interrupções.

# Gerenciamento de espaço em disco

O **LVM, ou gerenciamento de volumes lógicos**, oferece uma maneira avançada e flexível de administrar o espaço em disco.

Permite ao administrador do sistema alterar o tamanho de partições e volumes sem interrupções, adaptando o espaço disponível conforme as demandas mudam.

- Vantajoso em servidores, nos quais a necessidade por armazenamento pode mudar bastante.

Os sistemas de arquivos precisam gerir a **fragmentação**: acontece quando arquivos são armazenados em partes dispersas no disco, afetando o desempenho.

Para resolver isso, utilizam-se ferramentas de **desfragmentação** para reorganizar os dados e otimizar o acesso aos arquivos.



## Exemplo

No Windows, desfragmentar é um procedimento de manutenção regular para juntar fragmentos de arquivos e melhorar a velocidade de leitura e escrita no disco.



# Exemplos de gerenciamento de espaço em disco

### Exemplo 1:

Sistemas com ***journaling***, como o ext4, podem recuperar-se rapidamente de falhas, registrando mudanças em um *journal* antes de aplicá-las ao sistema de arquivos principal, garantindo a integridade dos dados após um *crash*.

### Exemplo 2:

O uso do **LVM** em servidores Linux permite o redimensionamento dinâmico de volumes sem interromper o acesso aos dados, facilitando a gestão de espaço em ambientes que requerem flexibilidade de armazenamento.

# Exemplos de gerenciamento de espaço em disco

## Exemplo 3:

A desfragmentação periódica em sistemas de arquivos **NTFS no Windows** reorganiza os dados para melhorar a velocidade de acesso e aumentar o espaço contíguo disponível.



### Tome nota

Essa realização otimiza o desempenho do sistema.

Vamos  
fazer uma  
**atividade**

## Brainstorming: seminário sobre sistemas operacionais mais utilizados no mercado

No seminário sobre sistemas operacionais, exploraremos e debateremos os mais usados atualmente, focando em aprofundar o conhecimento técnico, as aplicações práticas, o perfil dos usuários e o impacto das comunidades de desenvolvedores.

20 min

Em grupo

Cada grupo de alunos será responsável por pesquisar, analisar e apresentar um esboço abrangente que cubra os seguintes pontos:

- 1 Introdução aos sistemas operacionais mais utilizados:** apresentem uma visão geral dos sistemas operacionais dominantes no mercado, incluindo uma breve história e suas estatísticas atuais de uso e popularidade.
- 2 Características técnicas:** analisem as características-chave dos sistemas operacionais dominantes, focando em gestão de memória, processamento, arquivos e segurança e como cada um maneja atualizações e suporte.
- 3 Casos de uso e aplicativos:** identifiquem os perfis de usuários típicos e discutam os casos de uso mais comuns, incluindo aplicações industriais, comerciais e pessoais para cada sistema operacional.
- 4 Desenvolvimento e comunidade:** examinem o modelo de desenvolvimento (código aberto vs. código fechado) de cada sistema operacional, o suporte comunitário e corporativo e como isso influencia sua evolução e adoção.
- 5 Tendências e futuro:** reflitam sobre as inovações recentes e as tendências futuras para os sistemas operacionais, considerando o impacto da computação em nuvem, da inteligência artificial e da Internet das Coisas (IoT).





# Seminário sobre sistemas operacionais mais utilizados no mercado

Durante a apresentação, estejam preparados para responder às seguintes perguntas-chave:

- 1** Quais são as principais diferenças nos recursos de gerenciamento de memória e processamento entre os sistemas operacionais mais utilizados?
- 2** Como cada sistema operacional atende a diferentes tipos de usuários e casos de uso, e quais são as aplicações mais significativas para cada um?
- 3** Qual é o papel das comunidades de desenvolvimento e do modelo de licenciamento na trajetória de sucesso de um sistema operacional?

**Entrega para hoje:** Cada grupo precisa enviar um resumo detalhado da apresentação com fontes usadas no AVA.

**Na próxima aula, cada grupo realizará sua apresentação,** expondo suas descobertas e percepções à turma.



Momento  
de **reflexão**

© Getty Images





O que nós  
**aprendemos**  
**hoje?**

© Getty Images

## Hoje desenvolvemos:

- 1** O entendimento aprofundado sobre **NTFS, ext4 e FAT32**, discernindo como cada um serve a propósitos variados, do NTFS robusto ao FAT32 simples e ao ext4 eficiente no Linux.
- 2** Conhecimentos em gestão de sistemas de arquivos, avaliando métodos de alocação, como **contíguo, encadeado e indexado**, e seu impacto na eficiência de armazenamento e acesso a dados.
- 3** A compreensão do **LVM e do journaling** para garantir a consistência e a recuperação de dados, enfatizando a relevância dessas técnicas para a evolução dos sistemas **operacionais e as demandas por eficácia**.



# Saiba mais

“Qual sistema operacional você usa?”

Você pensa logo no Windows, no terminal do Linux ou no ícone do Android? Mas sistemas operacionais vão além das interfaces gráficas e dos *shells*. Surpreso? Não é só sobre clicar em ícones ou digitar comandos!

Conheça mais no curso de introdução ao sistema operacional.

PESSÔA, C. Sistemas operacionais: conceito e estrutura. *Alura*, 2023. Disponível em: <https://www.alura.com.br/artigos/sistemas-operacionais-conceito-estrutura>. Acesso em: 11 mar. 2024.

# Referências da aula

PESSÔA, C. Sistemas operacionais: conceito e estrutura. *Alura*, 2023. Disponível em: <https://www.alura.com.br/artigos/sistemas-operacionais-conceito-estrutura> Acesso em: 11 mar. 2024.

TANENBAUM, A. S.; BOS, H. *Sistemas operacionais modernos*. São Paulo: Pearson Education do Brasil, 2016.

Identidade visual: Imagens © Getty Images



**Educação  
Profissional  
Paulista**

Técnico em  
**Ciência de  
Dados**

**Educação  
Profissional  
Paulista**

Técnico em  
**Ciência de  
Dados**

# **Introdução à Ciência de Dados, ferramentas e jargões da área**

## **Controle de versão com Git e GitHub**

### **Aula 1**

**[DADOS]ANO1C1B2S12A1**

# Exposição



## Objetivos da Aula

Como instalar o Git em todas as três plataformas para as quais ele se encontra disponível.



## Competências da Unidade (Técnicas e Socioemocionais)

- Usar ferramentas de desenvolvimento de software como Git e GitHub;
- Trabalhar em equipe, compartilhando conhecimentos, contribuindo com ideias e colaborando para alcançar objetivos comuns.



## Recursos Didáticos

- Recurso audiovisual para exibição de textos, vídeos e imagens;
- Recurso computacional com internet.



## Duração da Aula

50 minutos.



# Relembrando a importância do Git

- **Controle de versão eficiente:** o Git permite o controle de versão eficiente, rastreando alterações em projetos de software ao longo do tempo. Isso possibilita a colaboração e o gerenciamento de código de forma organizada.
- **Colaboração em equipe:** com o Git, múltiplos desenvolvedores podem trabalhar simultaneamente no mesmo projeto, mesclando suas alterações de forma harmoniosa e evitando conflitos.
- **Histórico de alterações:** cada alteração no código é registrada, permitindo revisão histórica e identificação fácil de quando e por quê certas mudanças foram feitas. Isso é crucial para solucionar problemas e entender a evolução do projeto.

# Relembrando a importância do Git

- **Branching e experimentação:** o Git possibilita a criação de branches (ramificações independentes do código principal). Isso permite que os desenvolvedores experimentem novas funcionalidades ou corrijam bugs sem afetar diretamente o código principal.
- **Reversão de mudanças:** se algo der errado, é possível reverter para versões anteriores do código. Essa capacidade é valiosa para lidar com erros ou para desfazer alterações indesejadas.
- **Distribuição e compartilhamento:** o Git facilita a distribuição e o compartilhamento de código entre diferentes sistemas e desenvolvedores. Plataformas como GitHub e GitLab se tornaram populares para hospedar repositórios Git e promover colaboração em escala global.

# Como vamos lidar com o Git

O Git está disponível para **Windows**, **macOS** e **Linux**. Porém, apesar de todos apresentarem vantagens, vamos focar o Sistema Operacional Windows.

Além disso, estão disponíveis diversos métodos de instalação e uso, mas o nosso foco será aprender a base fundamental, que está na linha de comando.

## Os benefícios de aprender o Git pela linha de comando

**Controle total:** a linha de comando fornece controle total sobre as operações do Git. Comandos como **commit**, **push**, **pull** e **merge** oferecem uma compreensão profunda das ações realizadas.

**Flexibilidade:** a linha de comando permite personalização e automação avançadas por meio de scripts, proporcionando flexibilidade no gerenciamento do fluxo de trabalho.

**Compatibilidade com ambientes:** em muitos casos, servidores remotos e ambientes de desenvolvimento podem ser acessados exclusivamente pela linha de comando, tornando essencial a familiaridade com essa abordagem.

Em resumo, instalar e aprender o Git via linha de comando é crucial para um desenvolvedor, pois proporciona um controle preciso sobre o código, facilita a colaboração eficiente em equipe e oferece habilidades essenciais para gerenciar projetos de software de maneira eficaz.



# Instalando o Git

Antes de começar a utilizar o Git, é necessário tê-lo **disponível** em seu computador. Mesmo se já estiver instalado, é aconselhável considerar a atualização para a versão mais recente. A instalação pode ser feita por meio de um pacote, utilizando outro instalador, ou, ainda, baixando o código-fonte e realizando a compilação por conta própria.

# Instalando o Git no Linux

Se for do seu desejo, é possível fazer a instalação das ferramentas essenciais do Git no **Linux**, por meio de um instalador binário, utilizando a ferramenta de gerenciamento de pacotes inclusa na sua distribuição. Se estiver utilizando o Fedora (ou qualquer distribuição baseada em RPM, como RHEL ou CentOS), você pode empregar o seguinte comando com o "dnf":

### Código:

```
$ sudo dnf install git-all
```

# Instalando o Git no Linux

Se estiver utilizando uma distribuição fundamentada em **Debian**, como o **Ubuntu**, experimente o seguinte comando:

**Código:** \$ sudo apt install git-all

Para outras opções, você pode encontrar instruções detalhadas sobre a instalação em várias distribuições Unix no site oficial do Git. Disponível em: <https://git-scm.com/download/linux>. Acesso em: 14 mar. 2024.

Fonte: GIT, [s.d.].

# Instalando o Git no macOS

Há diversas formas de instalar o Git no macOS, sendo que a mais simples, provavelmente, é a instalação do Xcode Command Line Tools. Para versões do macOS a partir do Mavericks (10.9), você pode realizar a instalação, executando o seguinte comando no Terminal:

**Código:** `$ git --version`

Caso não esteja instalado, será solicitada a instalação. Se preferir uma versão mais atualizada, é possível realizar a instalação por meio de um instalador binário. Um instalador dedicado para macOS é mantido e pode ser baixado diretamente do site oficial do Git. Disponível em: <https://git-scm.com/download/mac>. Acesso em: 14 mar. 2024.

Fonte: GIT, [s.d.]



# Instalando o Git no Windows

Existem várias opções para instalar o Git no **Windows**, sendo que para instalar a compilação mais atual disponível para download no site do Git basta acessar: <https://git-scm.com/download/win> (Acesso em: 14 mar. 2024.) e o download iniciará **automaticamente**.

É importante notar que esta compilação é conhecida como **Git para Windows**, uma implementação específica, **diferente** do Git original. Para mais informações, visite: <https://gitforwindows.org> (Acesso em: 14 mar. 2024).

Se preferir uma instalação automatizada, é possível utilizar o pacote **Git no Chocolatey**. Ressalta-se que o pacote Chocolatey é mantido pela comunidade de tecnologia, proporcionando uma alternativa para facilitar o processo de instalação no ambiente Windows.

Fonte: GIT, [s.d.].

## Instalando o Git no Windows

Normalmente, os computadores atuais são todos produzidos com a capacidade de 64-bit, porém, procure verificar a capacidade do seu computador antes da instalação. Contudo, prossiga, clicando no link dentro deste local destacado.

### Download for Windows

[Click here to download](#) the latest (**2.44.0**) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **about 1 month ago**, on 2024-02-23.

#### Other Git for Windows downloads

##### Standalone Installer

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)

##### Portable ("thumbdrive edition")

[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

#### Using winget tool

Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.44.0**. If you want the newer version, you can build it from [the source code](#).

### Now What?

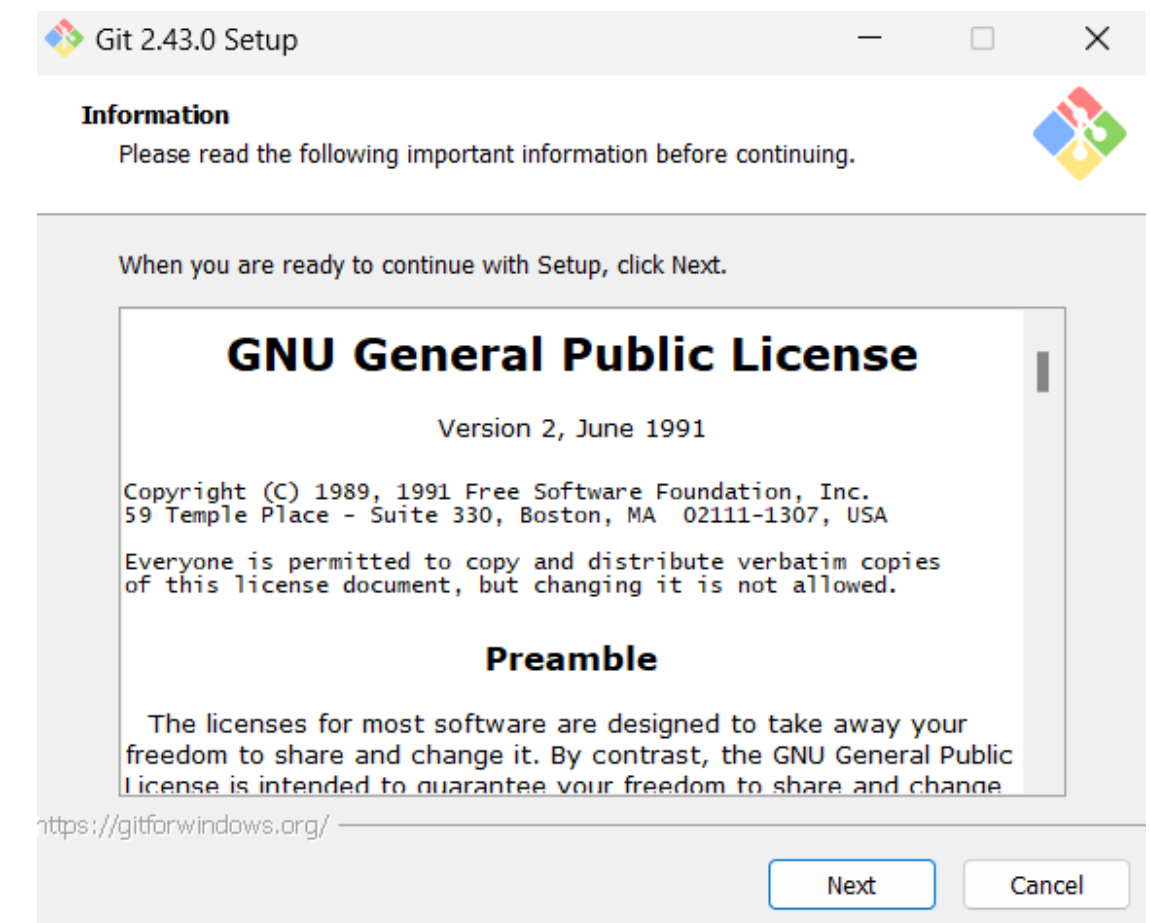
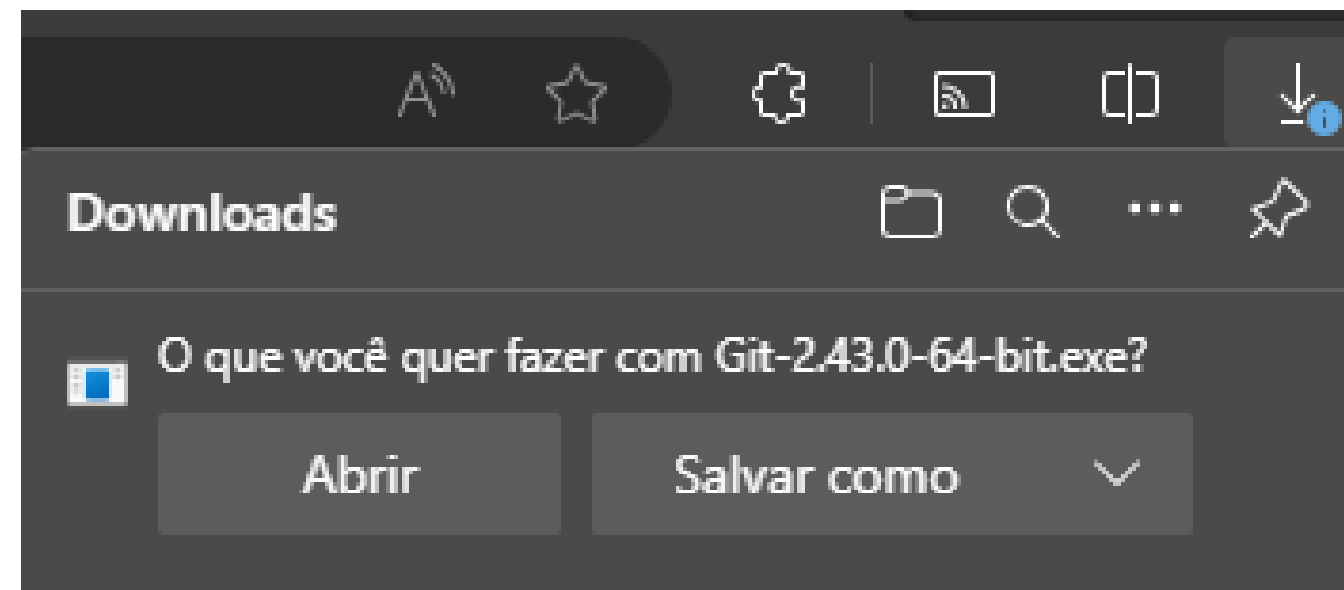
Now that you have downloaded Git, it's time to start using it.

Reprodução - GIT, [s.d.].

## Exposição

# Instalando o Git no Windows

Uma vez que a instalação foi finalizada, conclua em seu computador. Em seguida, aparecerá o instalador do Git. Entretanto, sugerimos que **leia tudo** que for possível e selecione conforme a sua preferência. Basta clicar em “Next” até que seja concluída a instalação.



Reprodução – GIT, [s.d.].





Vamos  
fazer um  
**quiz**

**Em qual sistema operacional o  
Git pode ser instalado?**

Windows

macOS

Linux

Todas as alternativas  
anteriores





Vamos  
fazer um  
**quiz**

## Em qual sistema operacional o Git pode ser instalado?



Windows

macOS



Linux

Todas as alternativas  
anteriores



### FEEDBACK GERAL DA ATIVIDADE

O Git é uma ferramenta de controle de versão distribuído e multiplataforma, projetado para funcionar em uma variedade de ambientes de desenvolvimento. Portanto, ele é compatível com todos os sistemas operacionais mencionados na questão.



Vamos  
fazer um  
**quiz**

## Durante a instalação do Git, recomenda-se:

Apenas clicar em  
“Next” até finalizar.

Ler com atenção e selecionar  
as opções coerentes.

Cancelar a instalação.

Aceitar os termos de uso  
e não ler mais nada.





Vamos  
fazer um  
**quiz**

## Durante a instalação do Git, recomenda-se:



**Apenas clicar em  
"Next" até finalizar.**

**Ler com atenção e selecionar  
as opções coerentes.**



**Cancelar a instalação.**

**Aceitar os termos de uso  
e não ler mais nada.**



### FEEDBACK GERAL DA ATIVIDADE

Durante a instalação do Git, recomenda-se "Ler com atenção e selecionar as opções coerentes". Isso ajuda a garantir que o Git seja instalado corretamente e que quaisquer configurações personalizadas necessárias para o seu ambiente de desenvolvimento sejam selecionadas.



Vamos  
fazer um  
**quiz**

**Selecione a alternativa que corresponde  
a uma das importâncias do Git:**

**Colaboração em  
equipe**

**Controle de versão  
do código**

**Histórico das versões  
do código**

**Todas as alternativas  
anteriores**





Vamos  
fazer um  
**quiz**

**Selecione a alternativa que corresponde a uma das importâncias do Git:**



**Colaboração em  
equipe**

**Controle de versão  
do código**



**Histórico das versões  
do código**

**Todas as alternativas  
anteriores**



### **FEEDBACK GERAL DA ATIVIDADE**

A alternativa que corresponde a uma das importâncias do Git é "Todas as anteriores". O Git é uma ferramenta fundamental para colaboração em equipe, controle de versão do código e manutenção do histórico das versões do código.



© Getty Images

O que nós  
**aprendemos**  
**hoje?**

## Hoje desenvolvemos:

- 1** A compreensão da praticidade na distribuição e no compartilhamento de código com Git;
- 2** A aplicação da colaboração simultânea no Git;
- 3** O conhecimento sobre a instalação de Git para os principais sistemas operacionais.



# Saiba mais

Veja a **documentação oficial** para a instalação do Git:

GIT. 1.5 Primeiros passos – Instalando o Git.  
Disponível em: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>. Acesso em: 14 mar. 2024.



# Referências da aula

CHACON, S.; STRAUB, B. *Pro Git*. EUA: Apress Open, 2014. Disponível em: <https://git-scm.com/book/en/v2>. Acesso em: 14 mar. 2024.

GIT. Baixar para Linux e Unix. Disponível em: <https://git-scm.com/download/Linux>. Acesso em: 14 mar. 2024.

GIT. Baixar para macOS. Disponível em: <https://git-scm.com/download/mac>. Acesso em: 14 mar. 2024.

GIT. Baixar para Windows. Disponível em: <https://git-scm.com/download/win>. Acesso em: 14 mar. 2024.

GIT. Git para Windows. Versão 2.44.00. Disponível em: <https://gitforwindows.org/>. Acesso em: 14 mar. 2024.

REGINO, A. *Git e GitHub: seu código versionado: aprenda de uma vez por todas e sem enrolação* (Programação para Iniciantes). Devs, 2022.

Identidade visual: Imagens © Getty Images



**Educação  
Profissional  
Paulista**

Técnico em  
**Ciência de  
Dados**