



# Utilização de Arquivos em Ciência de Dados

---

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

# Persistência de Dados em Arquivos

---

- Capacidade de armazenar dados de forma que eles sobrevivam ao término da execução de um programa ou sistema.
- Arquivos são uma das formas mais comuns e fundamentais de alcançar essa persistência.
- **Armazenamento Físico:** Quando você salva dados em um arquivo (CSV, JSON, Excel, etc.), esses dados são gravados em um dispositivo de armazenamento físico, como:
  - **Disco Rígido (HDD):** Um disco magnético que armazena dados em superfícies rotativas.
  - **Unidade de Estado Sólido (SSD):** Uma memória flash não volátil que armazena dados eletronicamente.
  - **Pendrive/Cartão SD:** Dispositivos portáteis de armazenamento.
- **Independência da Memória RAM:** A memória RAM é volátil, o que significa que os dados armazenados nela são perdidos quando a energia é desligada. Arquivos, por outro lado, permanecem no dispositivo de armazenamento mesmo sem energia.

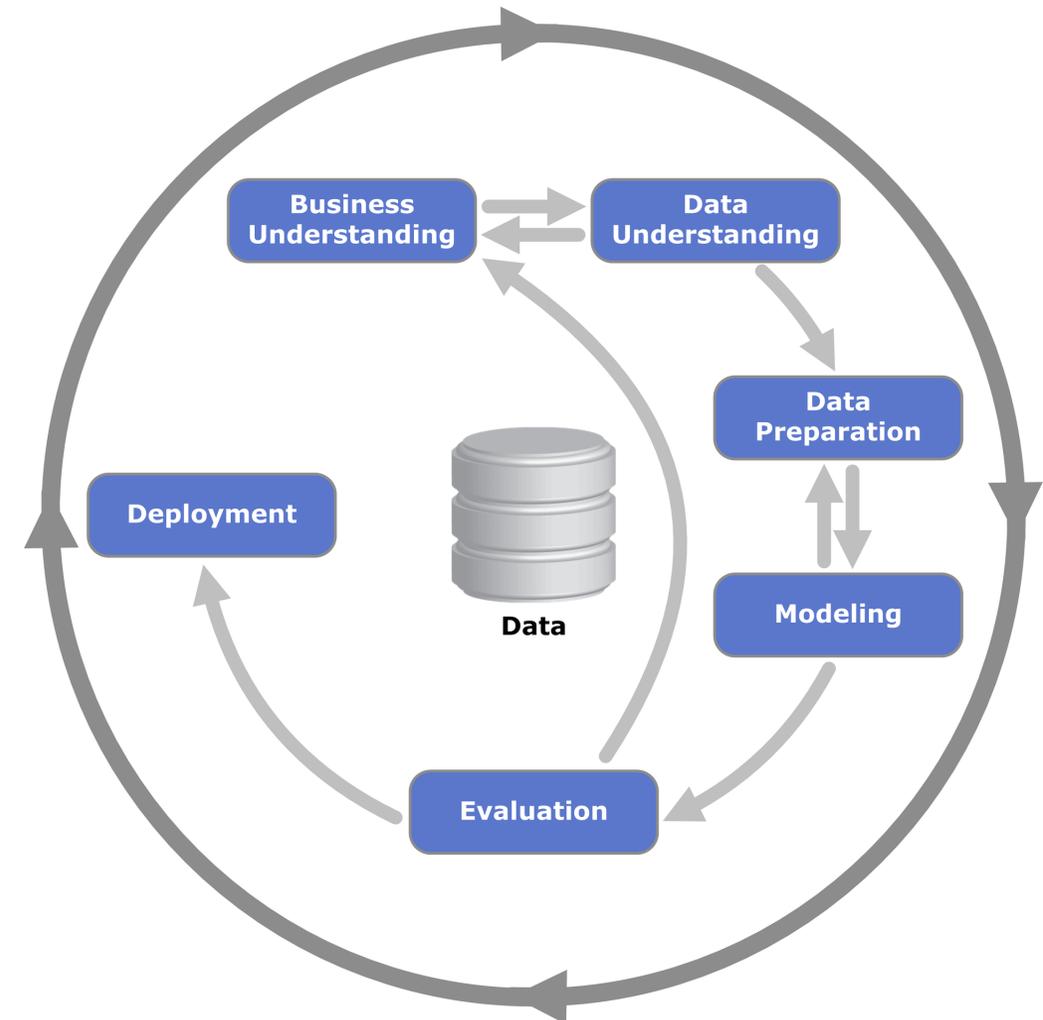
# Contexto histórico - Perspectiva Computacional

---

- **No passado:** Sistemas Gerenciadores de Bancos de Dados (SGBD, ou DBMS do inglês) eram a ferramenta primordial de análise de dados.
  - ~50 anos de pesquisa, ~500.000 transações/segundo, algoritmos muito eficientes.
  - SQL, modelo relacional, data warehouse...
- **No presente:** análise de dados realizada *fora* do SGBD.
  - Arquivos CSV, Python, R, Spark, data lake...
  - O landscape de ferramentas disponíveis é gigantesco.
- **Considerações gerais:**
  - Aplicações de ML são altamente impactadas pela qualidade e quantidade de dados.
  - Acesso aos dados se torna o gargalo do sistema (i.e., mover dados é muito custoso) → Mova a análise e não os dados (e.g., MapReduce, Spark, etc).

# CRISP-DM

- Passo 1. Compreensão do Negócio: Definir o problema e os objetivos do negócio.
- Passo 2. Compreensão dos Dados: Coletar, explorar e avaliar a qualidade dos dados.
- Passo 3. Preparação dos Dados: Limpar, transformar e estruturar os dados para análise.
- Passo 4. Modelagem: Aplicar técnicas estatísticas, de inteligência artificial e/ou machine learning para encontrar padrões.
- Passo 5: Validar modelo e algoritmos, interpretar resultados e verificar se atendem ao objetivo do negócio.
- Passo 6: Implementar o modelo para ser usado no ambiente real e monitorar seu desempenho.



```
39 19 3864 2548 2268 R 0.0 0.0 0:08.60 tmux new
20 678M 5948 4548 S 39.3 0.0 0:07.83 /nix/sto
20 1132M 28660 24508 S 25.4 0.1 0:05.99 foot
39 19 231M 15172 4496 R 17.6 0.0 0:03.16 htop
39 19 230M 13284 4528 S 15.5 0.0 0:01.66 htop
20 16236 7344 6908 S 10.9 0.0 0:02.16 /nix/sto
20 166M 15092 10248 S 9.8 0.0 0:08.79 /run/cur
20 3448M 161M 149M S 8.8 0.5 26:10.48 /nix/sto
20 678M 5948 4548 S 5.2 0.0 0:00.92 /nix/sto
20 678M 5948 4548 R 4.7 0.0 0:00.89 /nix/sto
20 678M 5948 4548 S 4.7 0.0 0:00.89 /nix/sto
20 678M 5948 4548 S 4.7 0.0 0:00.90 /nix/sto
20 678M 5948 4548 S 4.7 0.0 0:00.91 /nix/sto
20 678M 5948 4548 S 4.7 0.0 0:00.92 /nix/sto
20 678M 5948 4548 S 4.1 0.0 0:00.90 /nix/sto
20 678M 5948 4548 S 4.1 0.0 0:00.90 /nix/sto
20 3103M 241M 104M S 3.1 0.8 2:54.24 /nix/sto
20 1132M 28660 24508 S 3.1 0.1 0:00.53 foot
20 1132M 28660 24508 S 3.1 0.1 0:00.57 foot
20 1132M 28660 24508 S 3.1 0.1 0:00.54 foot
20 51.46 373M 54092 S 2.6 1.2 15:14.36 /nix/sto
20 678M 5948 4548 S 2.6 0.0 0:00.52 /nix/sto
```

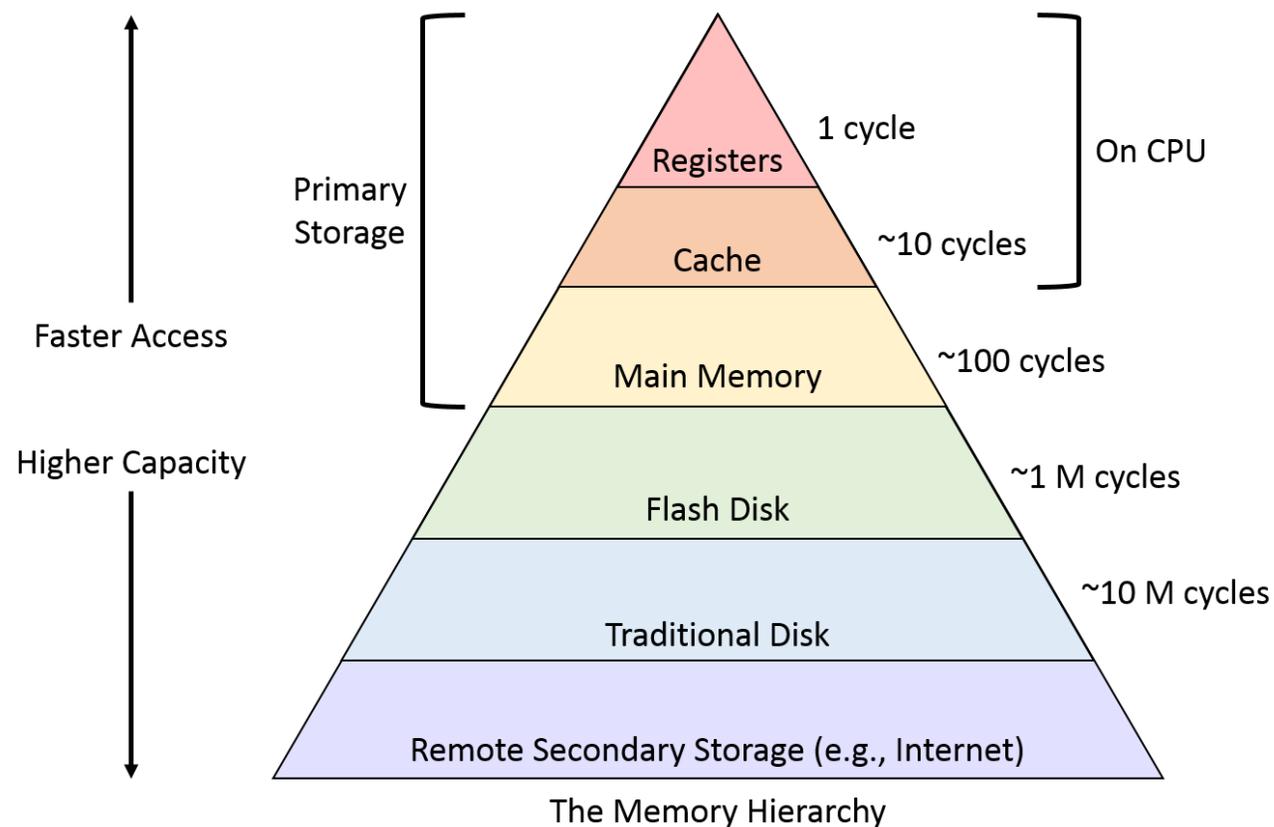
```
F3 Search F4 Filter F5 Free F6 SortBy F7 Nice F8 Nice F9 Kill F10
```

```
0 31% 237% 0.0% 0 100.00x
EPZRSY02:05okenepi2enot synch
ENEBEL304BEmkhangap2ull files
ENFILE:03 Ewchngy 0p22 files
EQWMEED20DT100m2ngop2ediles
EBUVER06ADev30e0onared2odce bi
EBU8Y336NDevic2eabrgament2ob
EDDWT08:49m2r2c2o22rdu2ent no
EUBARRECE49102oN2c2o22rdr2voppnd
ENETRESET:102 Network dropped
E2eREADY:114 Operation alread
ENQREADY151010p2rdevic2e2req2d
EQONB2BOR5EB1002 devic2e2req2d
E60NNABORTED:103 Software cau
G6SHAM:120 Is a named type fi
EES22M:120FI2e2e2n2s2ed type fi
EEXMSEL17:125e0p2xist2ion cance
EC2NCE62DT125r0p2p2222dn cance
EN2MEK627T2m2e2r2o2k2p2is2edilable
ERBUCK:37 Hadlod2se2s2available
ESRDEE:29 B222e2gd22r2o2s2ek
ESPTPEC29ERA2e22a132e22t2e2 not
ESBTREC69ESA2o2En13r22t2e2 not
ESBRIN69ES98o22d2r2e2r2o2n2l2e2ady
ENDDRINU6E:98 22d2d2e2s22v2a2il2e2b2y
END2ATN061PR0R22e293 22v2a2il2e2b2l2e2n
EP2B2DTN47UP2P20RT:93r22r2o2t2o2l n
E13RST:27 l2e2v2l2d3r2e2ct2e2ty
E1S2IR:21 l2e2s2o2d2i2e2c2t2o2r2a2c2il
```

# Critérios

- Recuperar e armazenar o mais rápido possível
- Uso ótimo do espaço de memória
  - Quando os dados não estão sendo usados, a memória é liberada imediatamente
  - Minimizar o tempo em que a memória está reservada, mas não utilizada
  - Os dados devem ocupar o menor espaço necessário

# Hierarquia de memória



Fonte da Imagem: [CS31](#)

Ver também: [Latency by Collin Scott](#)

# Formatos Comuns

---

- **TXT (Text Files):**
  - Arquivos de texto simples, podem conter dados formatados ou não.
  - Requer análise customizada para extrair informações relevantes.
- **CSV (Comma Separated Values):**
  - Simples, amplamente utilizado para dados tabulares.
  - Delimitador (vírgula, ponto e vírgula, etc.) define a separação entre colunas.
  - Pode ter ou não cabeçalho com os nomes das colunas.
- **JSON (JavaScript Object Notation):**
  - Formato flexível para dados semiestruturados.
  - Representa dados como pares chave-valor e listas.
  - Útil para APIs e dados hierárquicos.
  - Problemas comuns: aninhamento complexo, validação de schema.

# Formatos Comuns (cont.)

---

- **JSON (JavaScript Object Notation):**
  - Formato flexível para dados semiestruturados.
  - Representa dados como pares chave-valor e listas.
  - Útil para APIs e dados hierárquicos.
  - Problemas comuns: aninhamento complexo, validação de schema.
- **Parquet**
  - Formatos colunares otimizados para leitura e escrita de grandes datasets.
  - Mais eficientes que formatos linha-orientados (CSV) para análises exploratórias e machine learning.

# Metadados do Arquivo

---

- **Encoding:** A codificação de caracteres utilizada no arquivo (UTF-8, Latin-1, ASCII). Encoding incorreto pode levar a erros de leitura e caracteres estranhos.
- **Delimitadores/Separadores:** O caractere usado para separar os campos em arquivos CSV ou outros formatos delimitados.
- **Cabeçalho:** A presença ou ausência de uma linha de cabeçalho com os nomes das colunas.
- **Tipo de Dados por Coluna:** Entender o tipo de dado (string, inteiro, float, data) que cada coluna representa.

# Permissões de Acesso

---

- Quem tem permissão para fazer o quê?
- O sistema controla o acesso a objetos por sujeitos.
- **Objeto:** qualquer coisa que precise ser protegida: por exemplo, uma região de memória, um arquivo, um serviço.
  - Com operações diferentes dependendo do tipo de objeto.
- **Sujeito:** entidade ativa que utiliza os objetos, ou seja, um processo.
  - Threads dentro de um processo compartilham as mesmas permissões de acesso.
  - O sujeito pode também ser o próprio objeto, por exemplo, terminar uma thread ou um processo.

JULIA EVANS  
@bork

# unix permissions

drawings.jvns.ca

There are 3 things you can do to a file

↓ read ↓ write ↓ execute

ls -l file.txt shows you permissions  
Here's how to interpret the output:

rw- rw- r-- bork staff  
↑ ↑ ↑  
bork (user) staff (group) ANYONE  
can can can  
read & write read & write read

File permissions are 12 bits

setuid setgid  
↓ ↓  
000 110 110 100  
sticky rwx rwx rwx  
user group all

For files:

- r = can read
- w = can write
- x = can execute

For directories it's approximately:

- r = can list files
- w = can create files
- x = can cd into & modify files

110 in binary is 6

So rw- r-- r--  
= 110 100 100  
= 6 4 4

chmod 644 file.txt  
means change the permissions to:

rw- r-- r--

simple!

setuid affects executables

\$ls -l /bin/ping

rwS r-x r-x root root  
↑  
this means ping always runs as root

setgid does 3 different unrelated things for executables, directories, and regular files



Fonte da Imagem: [Julia Evans](#)

# Dúvidas e Discussão

---