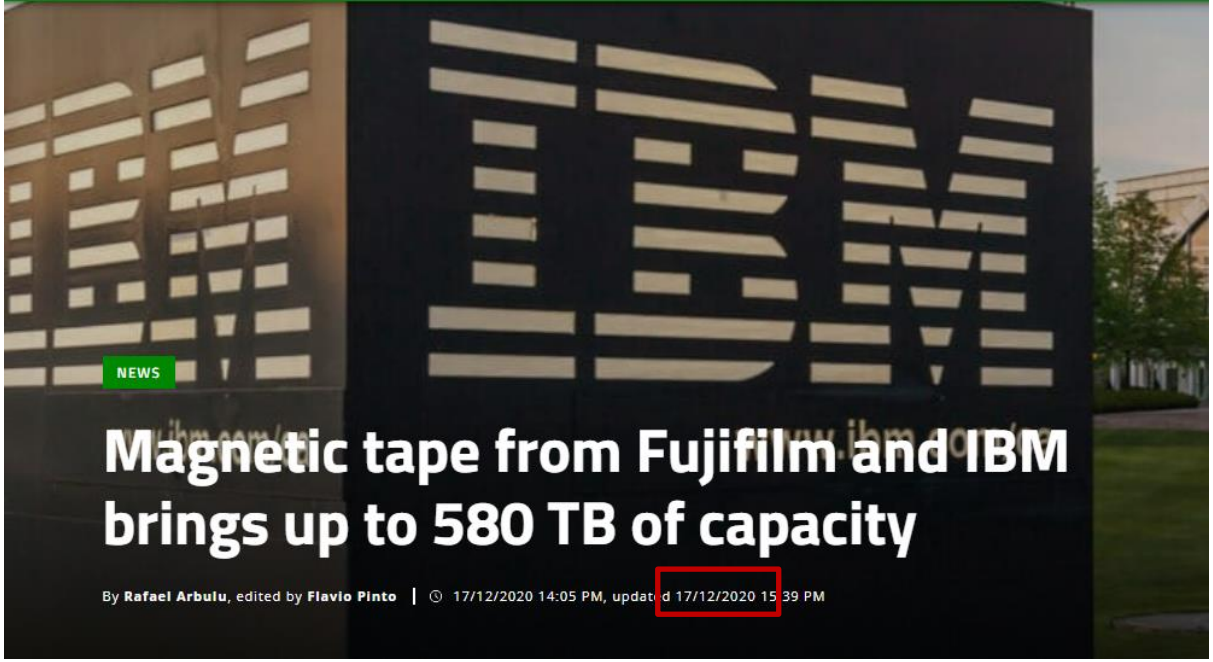
# Memória Secundária

Organização e Recuperação de Dados

Profa. Valéria

UEM – CTC – DIN

Slides preparados com base no Cap. 3 do livro FOLK, M.J. & ZOELLICK, B. *File Structures*. 2nd Edition, Addison-Wesley Publishing Company, 1992.




Fonte: https://olhardigital.com.br/en/2020/12/17/noticias/fita-magnetica-da-fujifilm-e-ibm-traz-ate-580-tb-de-capacidade/

- Durabilidade → ~30 anos

- Fonte: **https://www.ibm.com/blogs/research/2020/12/tape-density-record/**

## IBM's Tale of the Tape

Nearly 70 years of tape innovation: reliable, secure & energy efficient for Hybrid Clouds

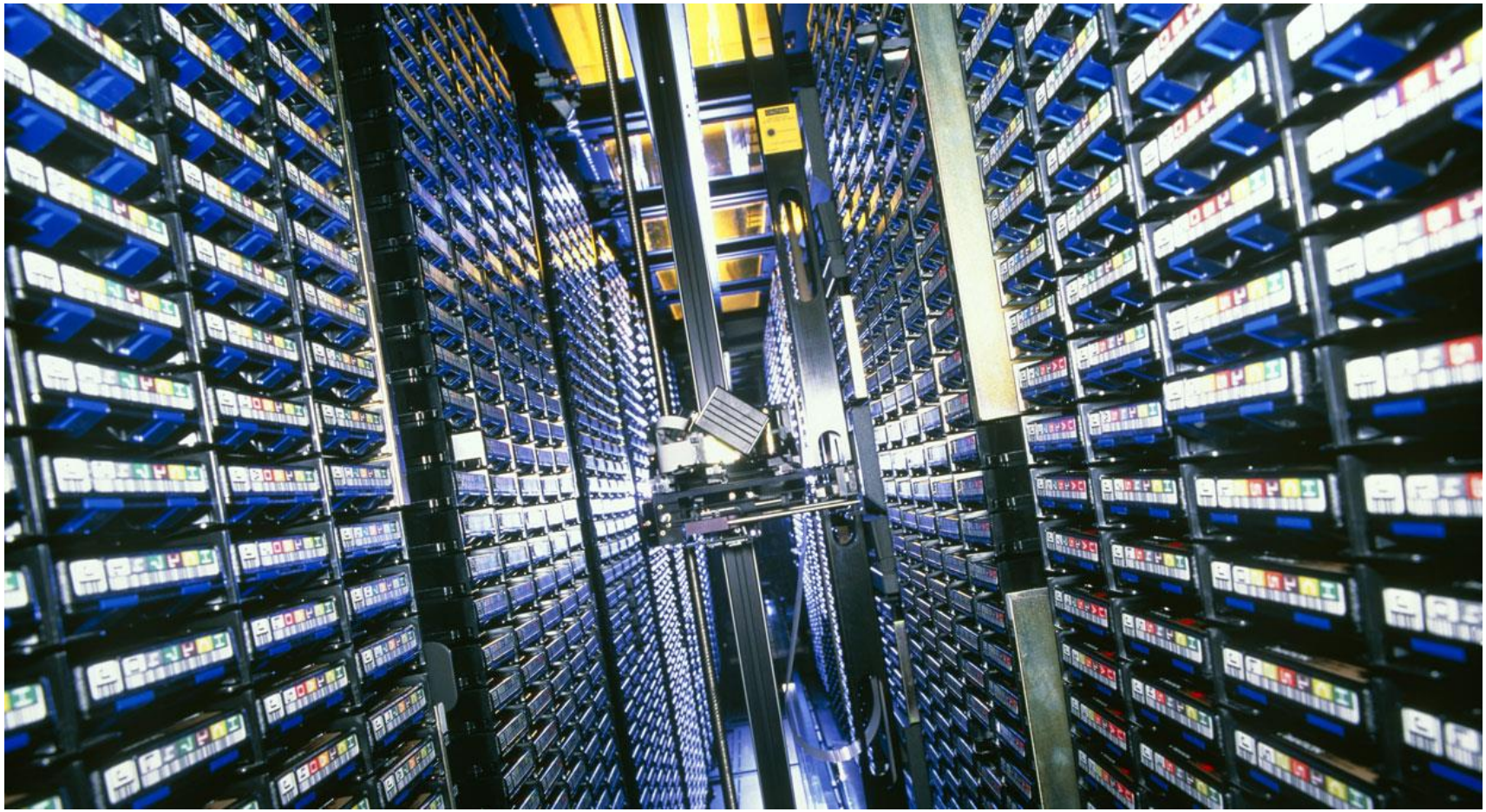
| | 2006 | 2010 | 2014 | 2015 | 2017 | 2020 |
|---|---|---|---|---|---|---|
| Areal Density (bits per sq inch) | 6.67 Billion | 29.5 Billion | 85.9 Billion | 123 Billion | 201 Billion | 317 Billion |
| Cartridge Capacity (Terabytes) | 8 | 35 | 154 | 220 | 330 | 580 |
| # of Books Stored* | 8 Million | 35 Million | 154 Million | 220 Million | 330 Million | 580 Million |
| Track Width | 1.5 µm | 0.45 µm | 0.177 µm | 0.140 µm | 103 nm | 56.2 nm |
| Linear Density (bits per inch) | 400'000 | 518'000 | 600'000 | 680'000 | 818'000 | 702'000 |
| Tape Material | Barium Ferrite | Barium Ferrite | Barium Ferrite | Barium Ferrite | Sputtered Media | Strontium Ferrite |
| Tape Thickness (micrometers - µm) | 6.1 | 5.9 | 4.3 | 4.3 | 4.7 | 4.3 |
| Tape Length (meters) | 890 | 917 | 1255 | 1255 | 1098 | 1255 |



* assumes 1MB of text data per book

IBM

# Bibliotecas de fitas podem armazenar centenas de petabytes



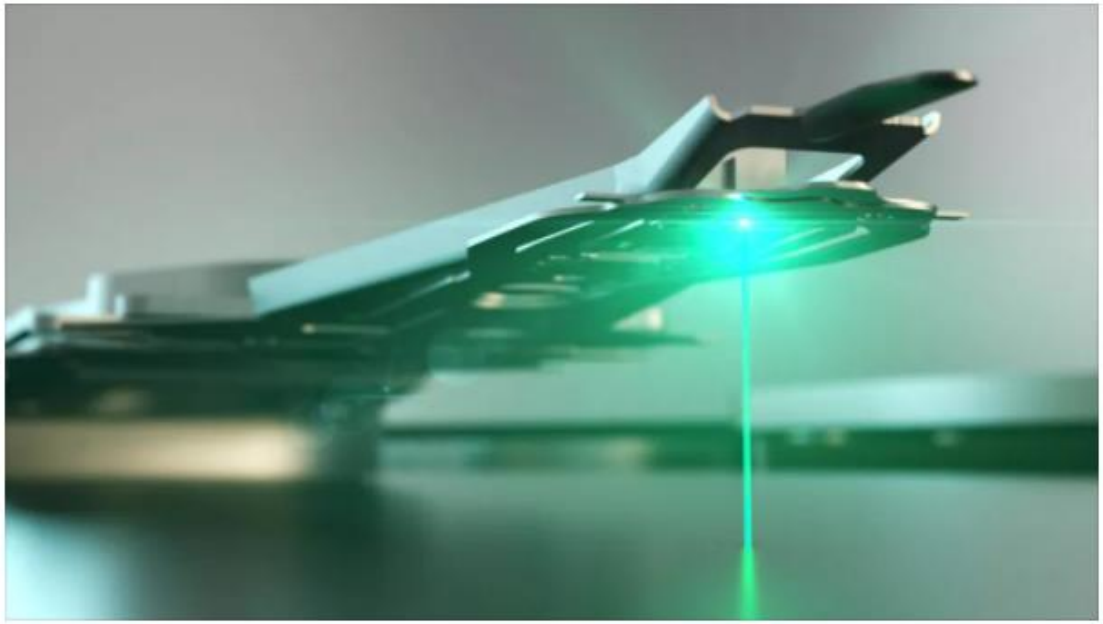
**Fonte**: https://spectrum.ieee.org/computing/hardware/why-the-future-of-data-storage-is-still-magnetic-tape

# Seagate Ships 20TB HAMR HDDs Commercially, Increases Shipments of Mach.2 Drives

By Anton Shilov January 23, 2021

Seagate continues to advance HDD technologies forward.

Comments (26)



(Image credit: Seagate)

- Fonte: https://www.tomshardware.com/news/seagate-ships-hamr-hdds-increases-dual-actuator-shipments

Atualmente, a Seagate tem um HD de 24TB e a Western Digital de 26TB
Ambos para uso em servidor
Custo = ~$480,00


Share

| Who | What |
|---|---|
| SEAGATE TECHNOLOGY | 60 TERABYTE(S) |
| **Where** | **When** |
| UNITED STATES () | 2016 |

In 2016, the US data-storage company Seagate Technology revealed they had created a solid-state drive (SSD) with a capacity of 60 Terabytes. A Terabyte corresponds to 1,000,000,000,000 bytes – in terms of data storage, the equivalent of about 210 DVDs or 1,423 CDs full of data.

Fonte:
https://www.guinnessworldrecords.com/world-records/73071-greatest-disk-storage-capacity

Para uso em servidor
Custo = ~$40.000,00

Fonte:
https://nimbusdata.com/press/nimbus-data-launches-worlds-largest-solid-state-drive-100-terabytes-power-data-driven-innovation/

## Nimbus Data Launches the World's Largest Solid State Drive – 100 Terabytes – to Power Data-driven Innovation

ExaDrive DC series raises the bar in SSD power efficiency, density, and write endurance

Irvine, CA March 19, 2018 – Nimbus Data, a pioneer in flash memory solutions, today announced the ExaDrive® DC100, the largest capacity (100 terabytes) solid state drive (SSD) ever produced. Featuring more than 3x the capacity of the closest competitor, the ExaDrive DC100 also draws 85% less power per terabyte (TB). These innovations reduce total cost of ownership per terabyte by 42% compared to competing enterprise SSDs, helping accelerate flash memory adoption in both cloud infrastructure and edge computing.

# Memória secundária

- Por que os discos e fitas são lentos?
  - Porque eles são **dispositivos eletromagnéticos** que envolvem partes mecânicas, enquanto outro tipos de memória são eletrônicas
- SSDs não possuem partes mecânicas, por isso são dispositivos secundários mais rápidos
  - Mas ainda são milhares de vezes mais lentos do que a memória RAM
- Fitas são dispositivos seriais, enquanto HDs e SSDs são dispositivos de acesso aleatório

# Discos magnéticos

- 1956 → Primeiro disco rígido comercial – IBM 350
- 50 discos de aproximadamente 60 cm de diâmetro
- Capacidade de 3,75 MB
- ~1,50m de altura e 1,80m de comprimento
- Pesava mais de 1 tonelada
- Alugado por $750 ao mês (~ R$3850,00)

- Em 1973, IBM lançou o Winchester, que é considerado o pai dos HDs modernos

Fonte: https://www.computerhistory.org

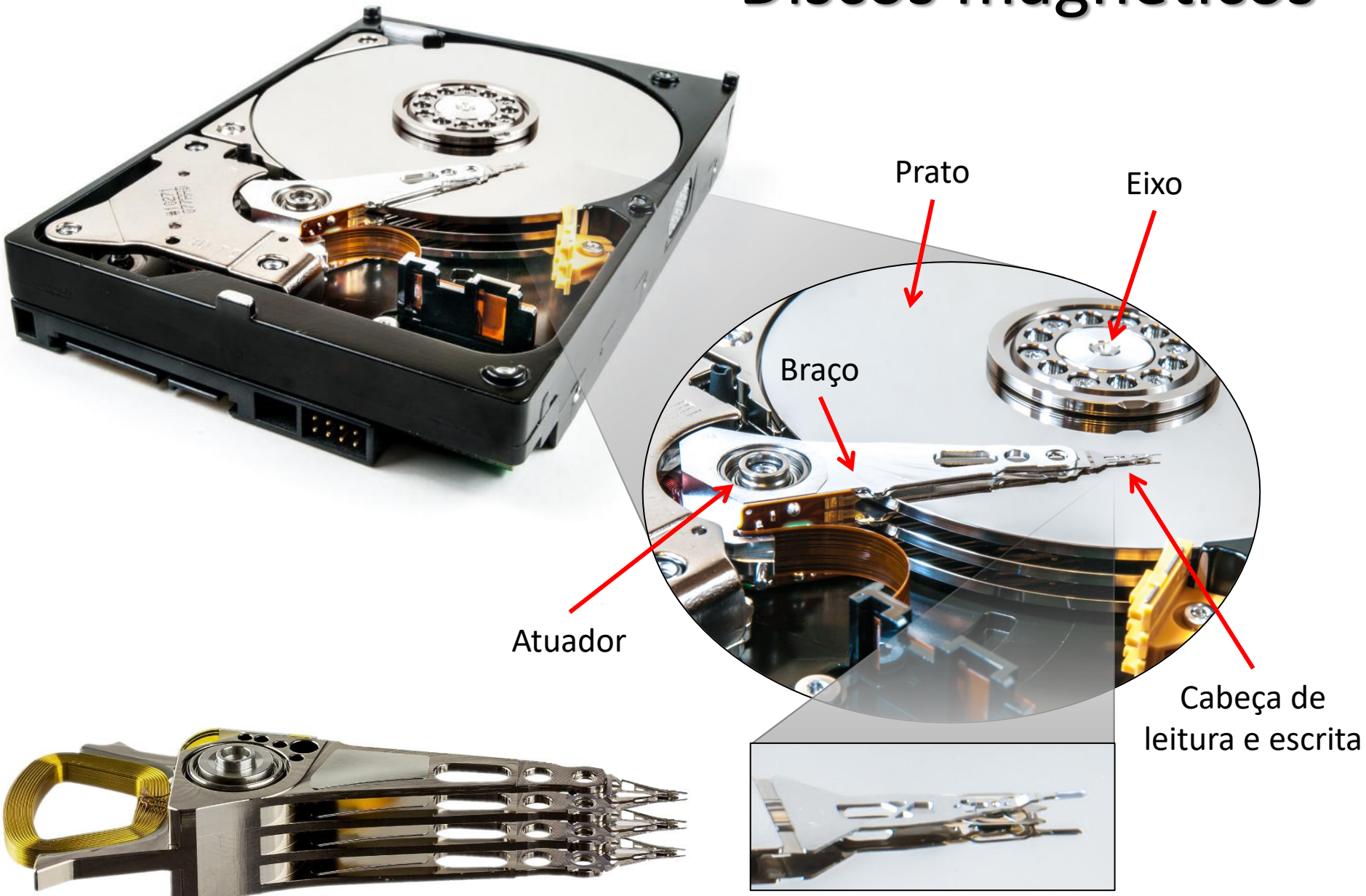
# Discos magnéticos

- **Exemplos**
  - **Hard disks** (discos rígidos)
    - Alta capacidade
    - Baixo custo por bit
    - Lento em relação à RAM
  - **Floppy disks** (disquete)
    - Baixa capacidade
    - Baixo custo
    - Muito mais lento que o HD
  - **ZIP disks**
    - Capacidade de até 750 MB
    - Velocidade: Floppy < ZIP < HD

# Discos magnéticos

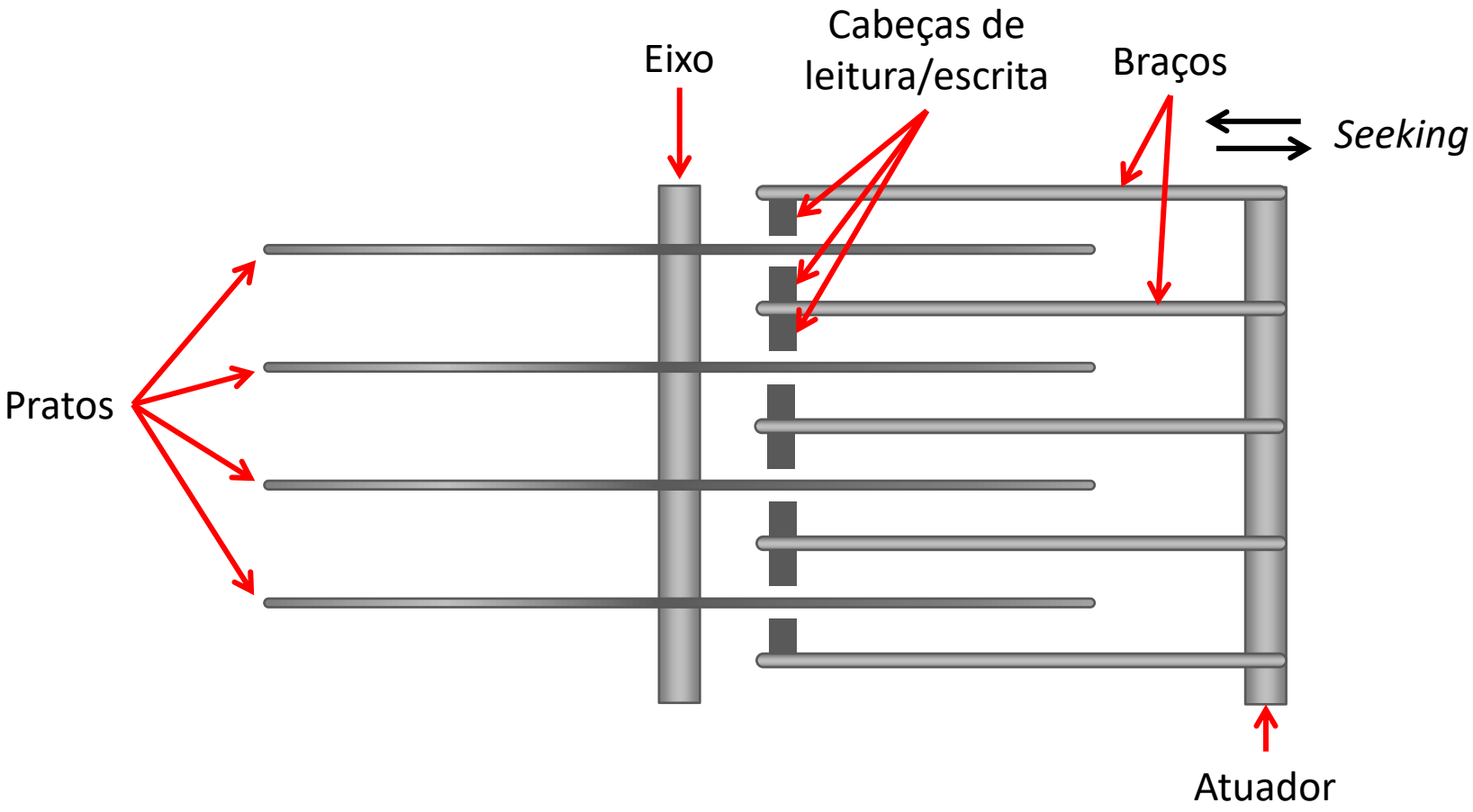
- **Composição**
  - Um ou mais pratos circulares sobrepostos atravessados por um eixo que os rotaciona
    - Floppy Disks e Zip Disks possuem apenas um prato flexível, enquanto que Hard Disks possuem geralmente entre um e quatro pratos rígidos
  - Os pratos são construídos com materiais não magnéticos e revestidos por uma cobertura magnética extremamente fina
    - Floppy Disks e Zip Disks utilizam material plástico enquanto Hard Disks utilizam alumínio ou vidro. Geralmente as duas superfícies do prato são magnetizáveis
  - Cabeças de leitura/escrita posicionadas sobre as superfícies do prato são responsáveis pela leitura e gravação dos dados
    - Nos discos rígidos a distância entre a cabeça de leitura/escrita e um prato é extremamente pequena (≈3 nm). Ela é mantida pela pressão aerodinâmica produzida pela rotação do prato sob a cabeça
  - As cabeças de leitura/escrita são sustentadas por um braço e um atuador é responsável por posicioná-las na superfície do prato
    - Esse posicionamento é chamado de ***seeking***

# Discos magnéticos



# Discos magnéticos

- **Esquema dos componentes de um disco**



# Discos magnéticos

- **Funcionamento**
  - Durante uma leitura/gravação a cabeça permanece parada enquanto o prato gira sob ela
    - Esta disposição dá origem à organização dos dados no prato como um conjunto de anéis concêntricos chamados de **trilhas**
  - Pulsos elétricos enviados a uma bobina na cabeça de L/E produzem campos magnéticos que alteram a polaridade de pequenas regiões de uma trilha do prato
    - A diferença no sentido da polaridade entre cada região codifica a informação binária (0s e 1s) da trilha
  - Durante a leitura, quando as regiões magnetizadas passam sob a cabeça, mudanças de polaridade induzem pulsos elétricos na bobina da cabeça que são convertidos novamente para a informação binária equivalente

# Discos magnéticos

- **Organização dos discos**
  - Os dados são armazenados em trilhas sucessivas na superfície de um prato
  - Cada trilha é dividida em setores
    - Um setor é a menor parte endereçável do disco (512B/4KB)
  - Quando é realizada uma chamada READ() por um byte específico
    - O disco localiza a superfície, a trilha e o setor corretos
    - Os dados de um setor inteiro são copiados para um *buffer*
    - A partir do *buffer*, se encontra o byte específico que foi requisitado
  - Normalmente os discos vêm setorizados de fábrica (formatação física)

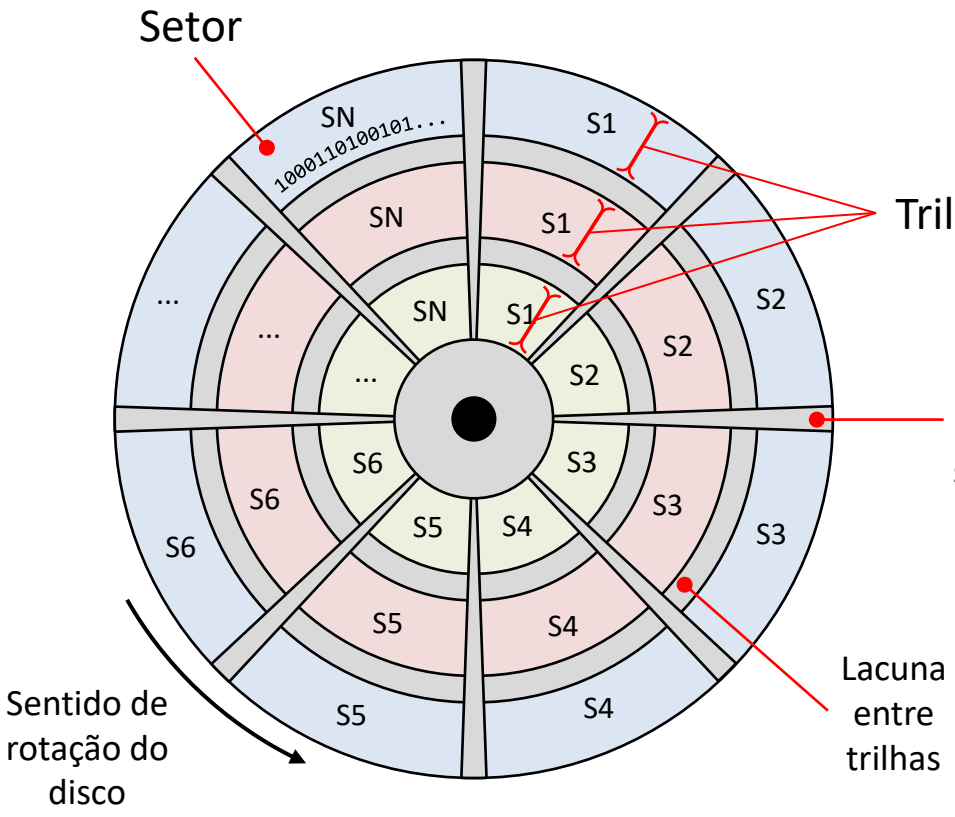
# Discos magnéticos

- **Organização dos discos**
  - Os dados são armazenados em trilhas sucessivas na superfície de um prato
  - Cada trilha é dividida em setores
    - Um setor é a menor parte endereçável do disco (512B/4KB)
  - Quando é realizada uma chamada READ() por um byte específico
    - O disco localiza a superfície, a trilha e o setor corretos
    - Os dados de um setor inteiro são copiados para um *buffer*
    - A partir do *buffer*, se encontra o byte específico que foi requisitado
  - Normalmente os disc

Importante notar que, logicamente, o disco é visto como um vetor contínuo de setores, i.e., a organização física é transparente para o sistema.

# Discos magnéticos

- **Organização dos discos**

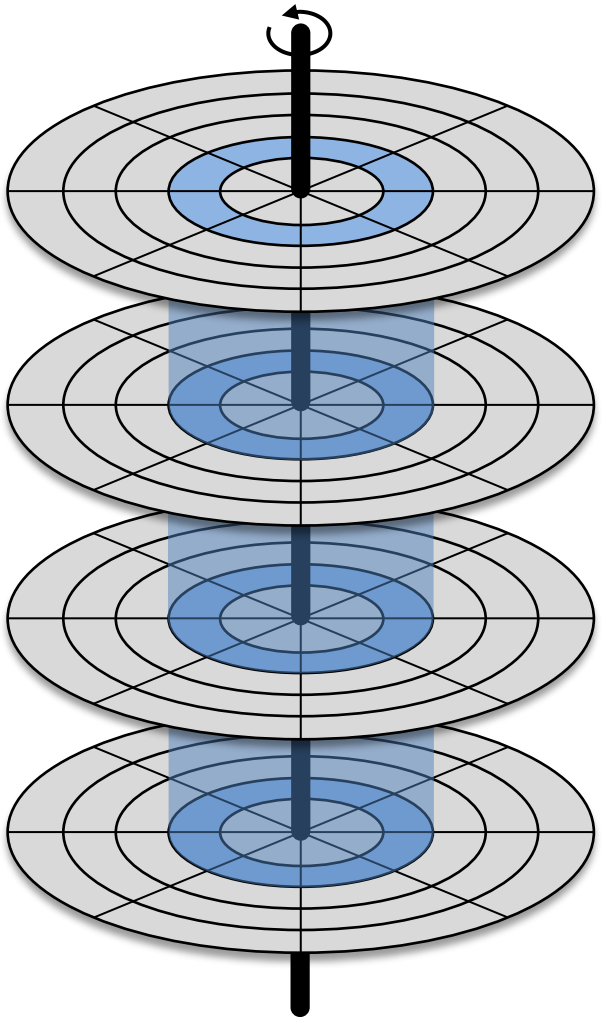



As lacunas entre os setores evitam que seja imposta uma precisão absurda para a cabeça de L/E

As lacunas entre trilhas evitam, ou pelo menos minimizam, erros devido ao desalinhamento da cabeça ou simplesmente à interferência de campos magnéticos.

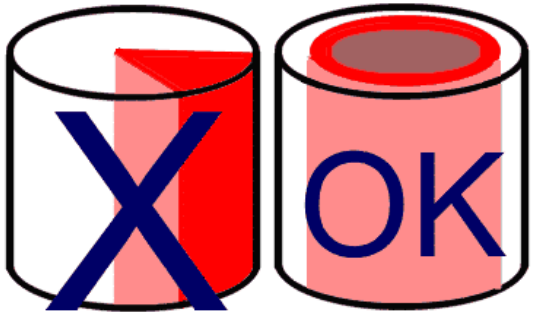
# Discos magnéticos

- **Organização dos discos**
  - Em um disco formado por diversos pratos, o conjunto de trilhas na mesma direção (sobrepostas em pratos diferentes) forma um cilindro



# Discos magnéticos

- **Organização dos discos**
  - Toda a informação em um cilindro pode ser acessada sem *seeking* adicional
    - O *seeking* normalmente é a parte mais lenta em uma operação de leitura/escrita
  - As cabeças se posicionam sobre o cilindro desejado e alternam sucessivamente entre si a leitura de cada trilha
    - Mesmo possuindo várias cabeças de leitura, apenas uma delas pode ser usada de cada vez, de forma que a controladora precisa constantemente chavear entre elas durante a leitura

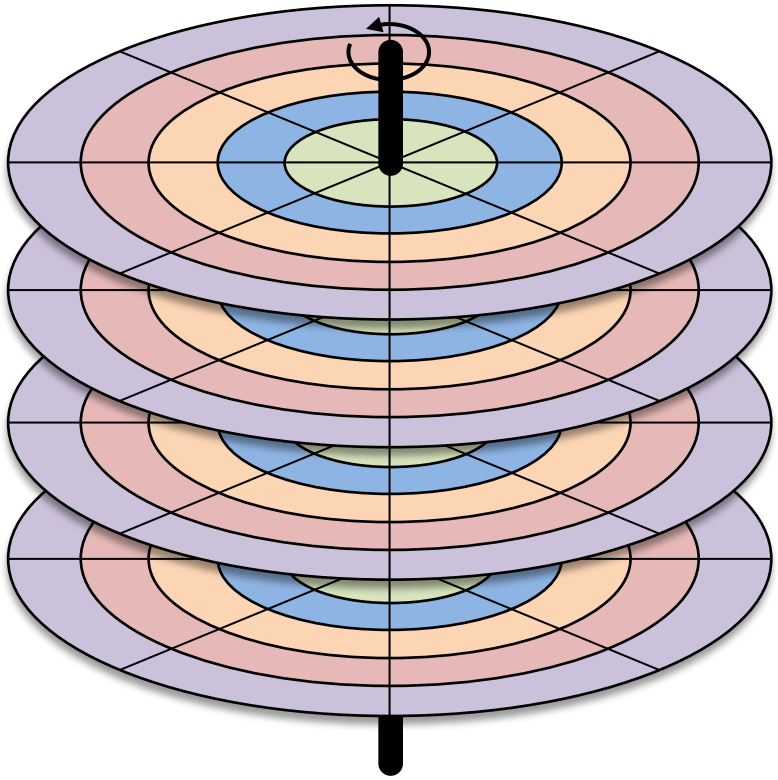


# Discos magnéticos

- **Exemplo (hipotético)**

Discos = 4
Superfícies = 8
Trilhas/superfície = 5
Cilindros = 5
Trilhas = 40
Setores/trilha = 8
Setores = 320

**iguais**



# Discos magnéticos

- **Estimativa de capacidade**
  - A quantidade de dados que pode ser armazenada em uma trilha depende do quão densamente os bits podem ser armazenados
  - A densidade depende da qualidade da mídia e da precisão das cabeças de leitura/escrita
    - **Exemplo:** Disco rígido 1TB Seagate 7200RPM (± R$300,00)
      - 16.383 trilhas/superfície e 63 setores (de 4k) por trilha (256 KB/trilha)

# Discos magnéticos

- **Estimativa de capacidade**
- Sendo que:
  - Número de Cilindros = Número de Trilhas/Superfície
  - Número de Trilhas do Cilindro = 2 × Número de Pratos

- Temos que:
  - Capacidade da **Trilha** =

    Número de bytes por Setor × Número de Setores por Trilha
  - Capacidade do **Cilindro** =

    Capacidade da Trilha × Número de Trilhas do Cilindro
  - Capacidade do **Disco** =

    Capacidade do Cilindro × Número de Cilindros

# Discos magnéticos

- **Estimativa de capacidade e espaço necessários**
- Exemplo
  - Propriedades do disco
    - 512 bytes/setor
    - 63 setores/trilha
    - 16 trilhas/cilindro
    - 4080 cilindros
  - Quantos **cilindros** são necessários para armazenar um arquivo de 50.000 registros de 256 bytes cada?

# Discos magnéticos

- **Estimativa de capacidade e espaço necessários**
- Exemplo
  - Propriedades do disco
    - 512 bytes/setor
    - 63 setores/trilha
    - 16 trilhas/cilindro
    - 4080 cilindros

Capacidades:
1 trilha = 63 setores de 512 bytes = 32.256 bytes
1 cilindro = 16 trilhas = 1.008 setores = 516.096 bytes

  - Quantos **cilindros** são necessários para armazenar um arquivo de 50.000 registros de 256 bytes cada?
    - Como cada setor tem 512 bytes, é possível armazenar 2 registros por setor, sendo necessário um total de 25.000 setores
    - Em 1 cilindro temos 1.008 setores (63 setores x 16 trilhas)
    - Então, para armazenar 25.000 setores são necessários 24,8 cilindros (25.000 setores/1.008 setores)

# Discos magnéticos

- ***Clusters***
  - Organização lógica que visa aumentar o desempenho e mantida pelo **Gerenciador de Arquivos** (*File Manager),* presente no S.O.
    - Quando um arquivo é acessado por uma aplicação, é o gerenciador de arquivos quem associa o arquivo lógico às suas posições físicas
    - O arquivo é visto como uma série de *clusters*
  - Para fazer esse mapeamento, o gerenciador de arquivos utiliza uma tabela de alocação de arquivos (chamada de *File Allocation Table* (FAT) em alguns S.Os.)

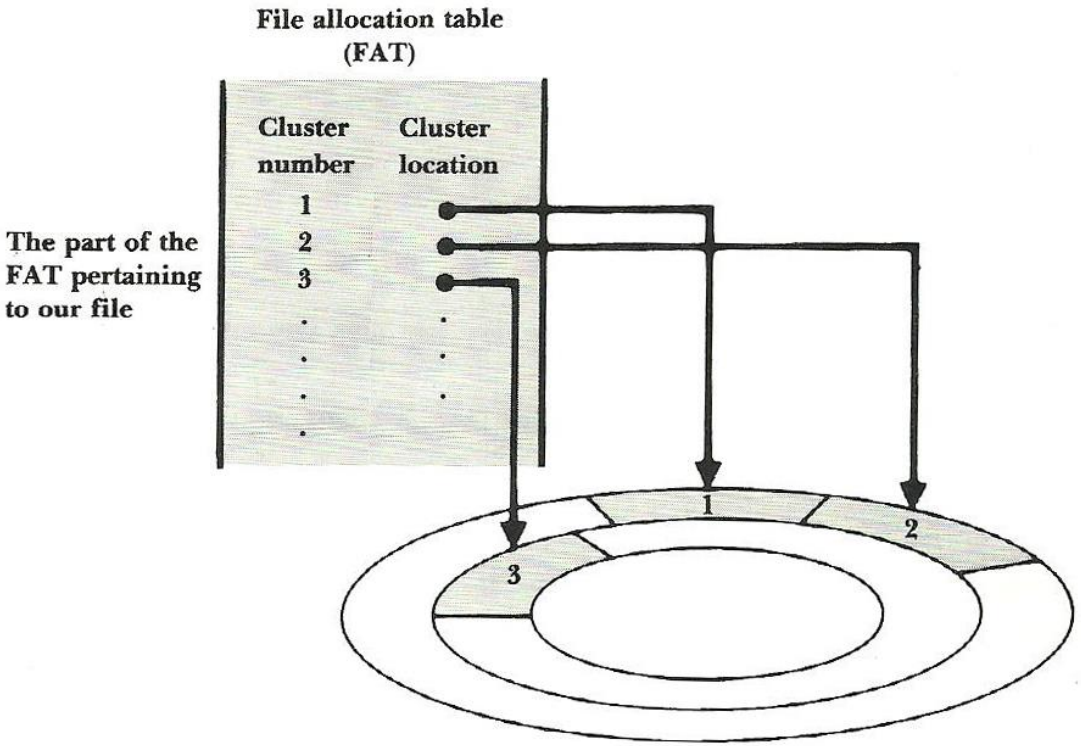
# Discos magnéticos

- ***Clusters***
  - Em vez da tabela de alocação endereçar setores, endereça *clusters*
    - Um *cluster* é um conjunto de um ou mais setores contíguos do disco
      - Todos os setores de um *cluster* podem ser lidos sem *seeks* adicionais e sem a necessidade de consultas adicionais à tabela de alocação
    - A tabela de alocação contém uma lista de todos os *clusters* de um arquivo, ordenados de acordo com a ordem lógica dos setores que eles contém

- O tamanho do **setor** é uma característica do **disco**
- O tamanho do **cluster** é uma característica do **S.O.**

# Discos magnéticos

- ***Clusters***
  - Cada *cluster* do disco é usado para um único arquivo, ou seja, em um mesmo *cluster* não haverá informações sobre mais de um arquivo



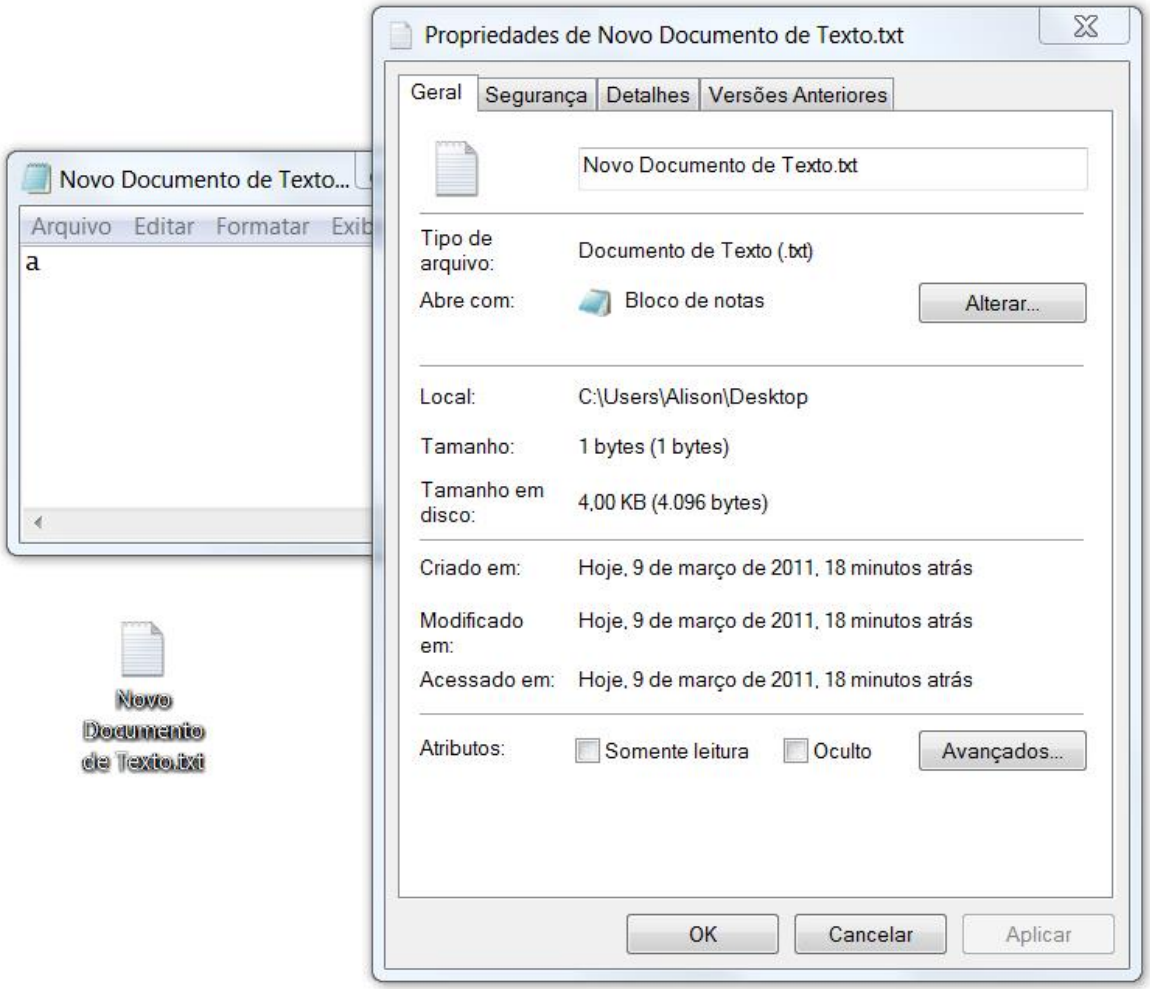
# Discos magnéticos

- **Comparação do tamanho de *cluster* entre partições**

| Tamanho do volume | Tamanho do cluster de FAT16 | Tamanho do cluster de FAT32 | Tamanho do cluster de NTFS |
|---|---|---|---|
| 7 MB–16 MB | 2 KB | Não compatível | 512 bytes |
| 17 MB–32 MB | 512 bytes | Não compatível | 512 bytes |
| 33 MB–64 MB | 1 KB | 512 bytes | 512 bytes |
| 65 MB–128 MB | 2 KB | 1 KB | 512 bytes |
| 129 MB–256 MB | 4 KB | 2 KB | 512 bytes |
| 257 MB–512 MB | 8 KB | 4 KB | 512 bytes |
| 513 MB–1,024 MB | 16 KB | 4 KB | 1 KB |
| 1,025 MB–2 GB | 32 KB | 4 KB | 2 KB |
| 2 GB–4 GB | 64 KB | 4 KB | 4 KB |
| 4 GB–8 GB | Não Compatível | 4 KB | 4 KB |
| 8 GB–16 GB | Não compatível | 8 KB | 4 KB |
| 16 GB–32 GB | Não compatível | 16 KB | 4 KB |
| 32 GB–2 TB | Não compatível | Não compatível | 4 KB |

# Discos magnéticos

- ***Clusters***




**Exemplo:**
Arquivo de 1 byte ocupando 4 KB (tamanho do *cluster*)

# Discos magnéticos

- **Custos de acesso a disco**
  - Em termos de tempo, o custo de um acesso ao disco é a soma de três tempos:
    - **Posicionamento** (*seek time*)
      - Tempo necessário para mover a cabeça leitura/escrita até a trilha correta – depende da distância a ser percorrida
    - **Latência** (*rotational delay* ou *latency*)
      - Tempo necessário para a cabeça de leitura/escrita se posicionar no setor correto
    - **Transferência** (*transfer time*)
      - Tempo necessário para um byte ser lido na superfície do disco e transferido para o *buffer* interno da controladora

# Custos de acesso ao disco

- **Posicionamento (*Seek time*)**
  - É impossível saber exatamente quantas trilhas serão percorridas durante as buscas
  - O que se faz é estimar um **tempo médio**, assumindo que as posições iniciais e finais para cada acesso são aleatórias
  - Por meio de estudos empíricos, estimou-se que uma busca percorre, em média, 1/3 do total de trilhas, sendo que o tempo gasto para percorrer esse número de trilhas tem sido usado pelos fabricantes como indicador do **tempo médio de busca (*seek time*)**
    - O tempo médio divulgado pelos fabricantes está entre 5ms à 10ms

# Custos de acesso ao disco

- **Posicionamento (*Seek time*)**
  - O que ocorre quando um arquivo está armazenado em cilindros consecutivos e é acessado sequencialmente?
    - O *seek time* é reduzido! (melhor situação)
  - O que ocorre quando dois arquivos, localizados em extremos opostos do disco (um no cilindro mais externo e outro no mais interno), são acessados alternadamente?
    - O *seek time* é alto! (pior situação)

- O tempo de *seek* tende a ser mais caro em sistemas multiusuários em que vários processos concorrem pelo uso do disco

# Custos de acesso ao disco

- **Latência (*Rotational delay*)**
  - Tempo gasto para a cabeça de leitura/escrita encontre o setor desejado
    - Estima-se que o tempo médio seja a metade do tempo de uma rotação (na prática, esse tempo costuma ser menor)
    - Também chamado de ***Average Latency***
    - No início dos anos 90, com discos de 3.600 rpm, a metade do tempo de uma rotação era de 8,3ms
    - Atualmente os discos de 7.200 rpm têm latência média de 4,16ms
  - Em uma leitura/escrita sequencial envolvendo trilhas de um mesmo cilindro, não há latência na alternância entre as trilhas
    - Existe apenas um tempo (muito pequeno) de comutação entre as cabeças de leitura/escrita

# Custos de acesso ao disco

- **Tempo de transferência (*Transfer time*)**
  - Uma vez que a cabeça de leitura/escrita está sob o setor desejado, ele pode ser transferido
  - O tempo de transferência é dado por:
    - O tempo para transferir uma trilha inteira normalmente é o tempo de uma rotação
    - **(n° bytes transferidos/n° bytes na trilha) x tempo de rotação**
    - Exemplo:
      - Tempo de transferência de 1 KB em um disco com 32 setores de 512 bytes por trilha (16.384b) e tempo de rotação de 8,2ms:

        (1.024/16.384) x 8,2 = 0,51ms
      - Neste exemplo o tempo está expresso em bytes/ms. Os fabricantes costumam utilizar MB/s

# Custo de acesso ao disco

- O modo de acesso ao arquivo pode afetar drasticamente os custos de tempo de acesso
  - Dois modos de acesso:
    - Sequencial: o máximo do arquivo é processado em cada acesso
    - Aleatório: apenas um registro é processado por vez

- **Exemplo**
  - Determinar o tempo necessário para ler um arquivo com 40.000 registros de 256 bytes cada um
    - Vamos calcular o tempo para leitura do arquivo com acesso sequencial e aleatório e comparar os resultados

# Exemplo

- Vamos considerar as seguintes especificações do disco:

| | |
|---|---|
| **Tempo médio de seek** | 13 ms |
| **Tempo de latência** | 8,3 ms |
| **Tempo de transferência** | 16,7 ms/trilha ou 1.229 bytes/ms |
| **Bytes por setor** | 512 |
| **Setores por trilha** | 100 |
| **Trilhas por cilindro** | 12 |
| **Trilhas por superfície** | 1.748 |
| **Tamanho do cluster** | 10 setores (5.120 bytes) |

# Exemplo

- Tamanho do arquivo ⇨ 40.000 registros de 256 bytes
  - Se cada *cluster* tem 5.120 bytes (10 setores), podemos armazenar 20 registros por *cluster* (5.120/256)
  - Será necessário um total de 2.000 *clusters* para armazenar todos os registros (40.000/20)
  - Como cada trilha possui 10 *clusters* (100 setores), serão necessárias 200 trilhas
- Vamos assumir o pior caso, no qual as 200 trilhas que armazenam o arquivo estão espalhadas **aleatoriamente** no disco
  - Situação extrema, mas que pode ocorrer em discos no limite da capacidade, especialmente com muitos arquivos pequenos

# Exemplo

- Custo com acesso sequencial
  - Para cada trilha, em que são lidos setores consecutivos, o processo de leitura envolve os seguintes custos:
    - Tempo médio de *seek* = 13 ms
    - Tempo de latência = 8,3 ms
    - Tempo de transferência de uma trilha = 16,7 ms

**Para 200 trilhas** =

(13 + 8,3 + 16,7) x 200 = 7.600 ms = **7,6 segundos**

# Exemplo

- Custo com acesso aleatório
  - Para cada registro, a operação de leitura envolve os seguintes custos:
    - Tempo médio de *seek* = 13 ms
    - Tempo de latência = 8,3 ms
    - Tempo de transferência de um *cluster* = 1,67 ms

      (Tempo de 16,7 ms/trilha, como cada trilha = 10 clusters, então é gasto 16,7/10 ms por cluster)

**Para 40.000 registros** =

(13 + 8,3 + 1,67) x 40.000 = 918,8 s = **15,3 minutos**

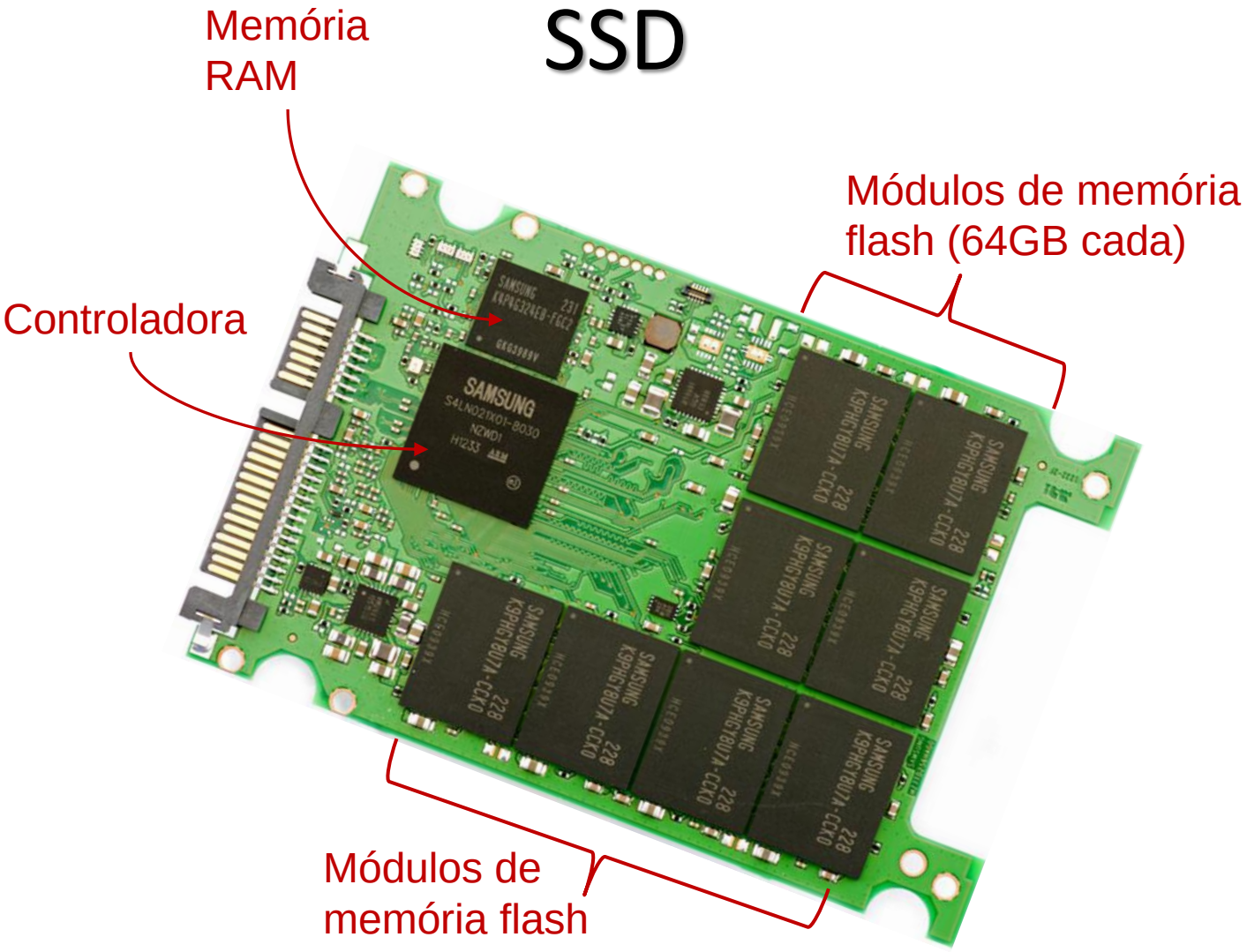
# Custo de acesso

- A diferença entre o acesso sequencial e acesso aleatório é muito grande!
  - No exemplo anterior: 7,6 seg vs. 15,3 min
  - Essa diferença se deve à quantidade de *seeks* – 200 no 1º caso contra 40.000 do 2º caso
- Por isso é aconselhável que o máximo de informação necessária seja lida em cada acesso
  - Evitando o frequente posicionamento da cabeça de leitura/escrita para cada registro

# SSD

- ***Solid-state drives* (SSD)** são compostos por componentes eletrônicos
  - O armazenamento geralmente é feito em módulos de memória *flash*
- Devido a essa característica, o custo de acesso a qualquer posição do SSD será o mesmo
  - Não há tempo de *seek*, nem latência rotacional
  - Mas existe uma latência (tempo para se iniciar uma operação) e um tempo de transferência (*throughput*)
    - Esses tempos dependem do tipo de memória usada, da controladora e da interface do SSD

# SSD




Exemplo: **Samsung SSD 840 Pro (512 GB)**

Fonte: http://codecapsule.com/2014/02/12/coding-for-ssds-part-2-architecture-of-an-ssd-and-benchmarking/

# SSD

- A menor unidade endereçável de um SSD é uma página
  - O tamanho de uma página muda de um SSD para outro, variando entre 2KB e 16KB
  - Assim como acontece com os setores de um disco, a página é a menor unidade de leitura e escrita
- As páginas são agrupadas em blocos
  - O tamanho dos blocos também varia de um SSD para outro, entre 256KB e 4MB
    - P.e., o Samsung SSD 840 EVO tem blocos de 2.048 KB (256 páginas de 8 KB cada)

# SSD

- Diferentemente dos setores de um HD, as páginas de um SSD não podem ser sobrescritas
  - Quando uma alteração é necessária, a página é copiada para um *buffer*, alterada e escrita em uma nova página livre
  - A página antiga é marcada como "velha" e ficará assim até o seu bloco ser apagado e a página fique livre novamente
- Essa operação faz com que as escritas sejam mais lentas do que as leituras
- Além disso, apenas blocos podem ser apagados
  - Para que um bloco com alguma página "velha" seja apagado, o coletor de lixo copiará as páginas ativas para um novo bloco, deixando o bloco pronto para ser apagado por completo
  - Uma vez apagado, as páginas do bloco ficam livres novamente

# SSD

- Além de não poderem ser sobrescritas, memórias *flash* também têm um número máximo de escritas possíveis
  - Elas desgastam conforme são escritas e apagadas
  - Por conta disso, a controladora distribuirá o uso dos blocos de memória para que o desgaste não seja maior em uma região do que em outra
- Todos esses fatores fazem com que as controladoras sejam complexas e adicionem custo nas operações em termos de tempo, especialmente nas escritas
- Ainda assim, os SSDs são centenas de vezes mais rápidos do que os HDs, especialmente quando se trata de acesso aleatório

# Memória secundária

- **Independentemente do tipo de dispositivo usado como memória secundária, ele será o gargalo do sistema**
  - Gargalo → quando um dispositivo mais lento afeta o desempenho de outros mais rápidos, se tornando um fator limitante do sistema
- Por isso é importante que acessos à memória secundária sejam feitos da forma eficiente