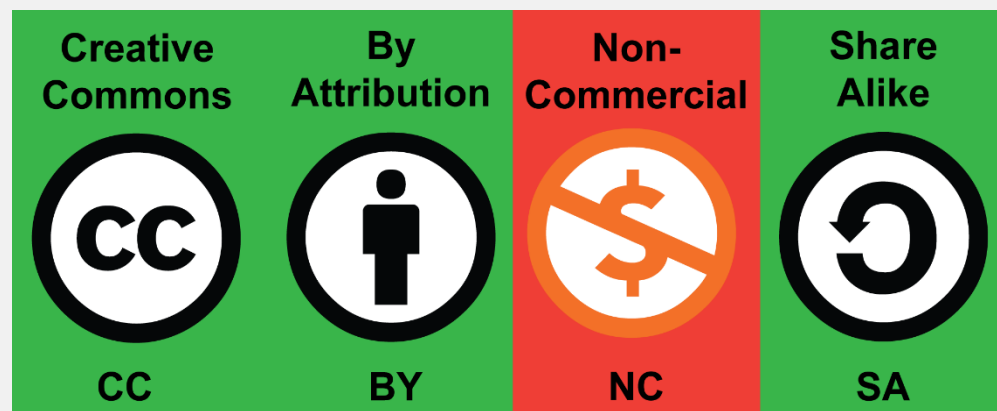


Meios de Armazenamento, Sistemas de Arquivos, Registros e Métodos de Acesso

Prof. MSc. Jackson Antonio do Prado Lima
jacksonpradolima at gmail.com

Departamento de Sistemas de Informação – DSI

Licença



Este trabalho é licenciado sob os termos da Licença Internacional Creative Commons Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional (**CC BY-NC-SA 4.0**)

Para ver uma cópia desta licença, visite
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Histórico de Modificação

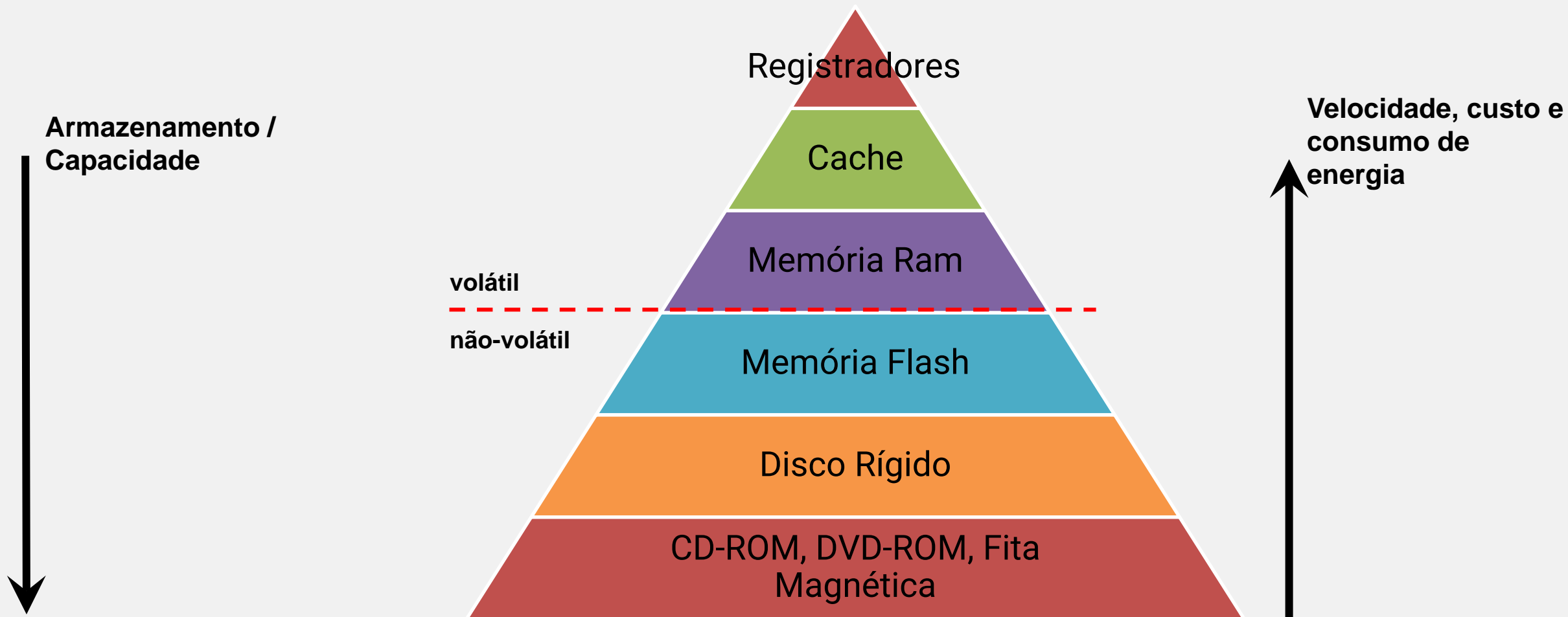
- Esta apresentação possui contribuição dos seguintes professores:
 - Fernando José Muchalski
 - Alex Luiz de Souza
 - Anderson Fabiano Dums
 - Jackson Antonio do Prado Lima

Agenda

- **Meios de armazenamento**
 - Classificação (hierarquia de memória), estrutura dos discos, operações e tempos de acesso
- **Sistemas de arquivos**
 - Características, vantagens e desvantagens, interface
- **Conceitos sobre registro, campo e chave**
- **Métodos de acesso**
 - Acesso sequencial, sequencial indexado, indexado e exemplos

MEIOS DE ARMAZENAMENTO

Hierarquia de Memória



Hierarquia de Memória

- Armazenamento **Primário**:
 - Memória cache e memória principal (RAM)
 - Dados acessados frequentemente (permite acesso rápido e a manipulação de dados – funciona como uma mesa de trabalho)
 - Todos programa precisa ser carregado em memória para ser executado
 - Armazenamento temporário (volátil)

Hierarquia de Memória

- Armazenamento **Secundário**:
 - Discos Magnéticos (HDs)
 - Dados acessados com certa frequência (mais lento que a memória)
 - Armazenamento de arquivos, documentos, programas
 - Armazenamento permanente (não-volátil)

Hierarquia de Memória

- Armazenamento **Terciário**:
 - Fitas magnéticas (*DDS - Digital Data Storage e DAT – Digital Audio Tape*)
 - Memórias Flash
 - Dados que não são acessados com frequência (backup)
 - Armazenamento permanente (não-volátil)

Hierarquia de Memória

Vantagens e desvantagens

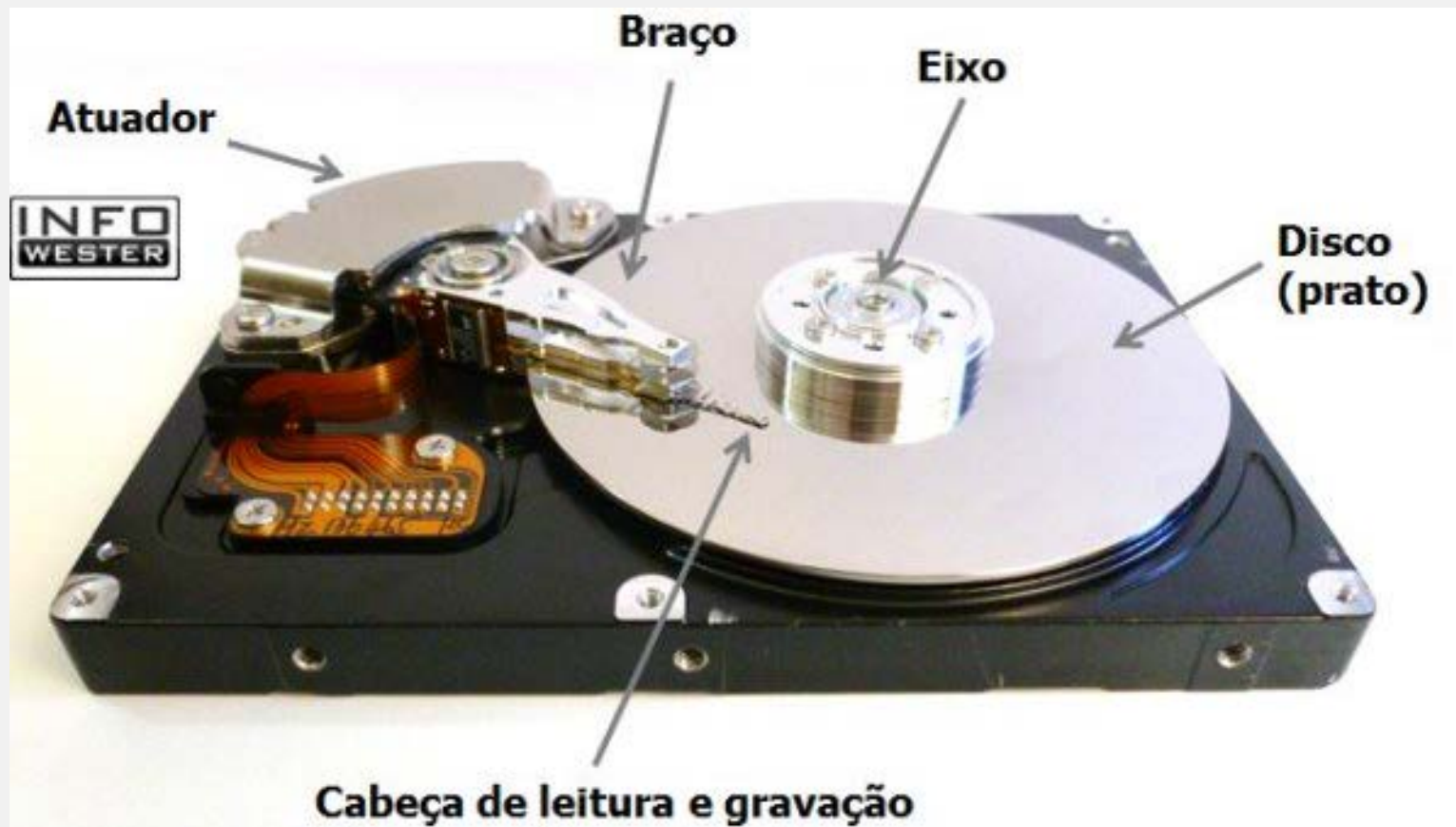
- **Dispositivos** secundários e terciários em **relação** aos primários são:
 - Dispositivos mais lentos
 - Mais baratos (valor \$/byte)
 - Maior **capacidade** de armazenamento
 - São não voláteis
- Disco x Fita:
 - Disco: **armazenamento** aleatório; Fita: sequencial
 - Na **fita** o acesso é mais lento do que o **disco**

Tipos de meios de armazenamento

- **Meios eletrônicos:** cada bit é definido por circuitos eletrônicos formados por capacitores e/ou transistores (portas lógicas). Ex: RAM, cache, flash, SSD.
- **Meios ópticos:** cada bit é definido pela reflexão da luz, detectado em um sensor ótico (existência do feixe refletido). Ex: CD, DVD, BlueRay
- **Meios magnéticos:** cada bit é definido pelo campo magnético detectado em um cabeçote segundo a orientação do campo (polaridade). Ex: HD

Dispositivos Magnéticos

- São os mais antigos e mais utilizados.
- Vantagens:
 - Alta densidade: grande quantidade de espaço em pequeno espaço físico
 - Preço: o custo por Mb é menor que outras mídias
- Desvantagens
 - Mais sujeito a falhas: dispositivo mecânico, partes móveis
 - Mídias sensíveis

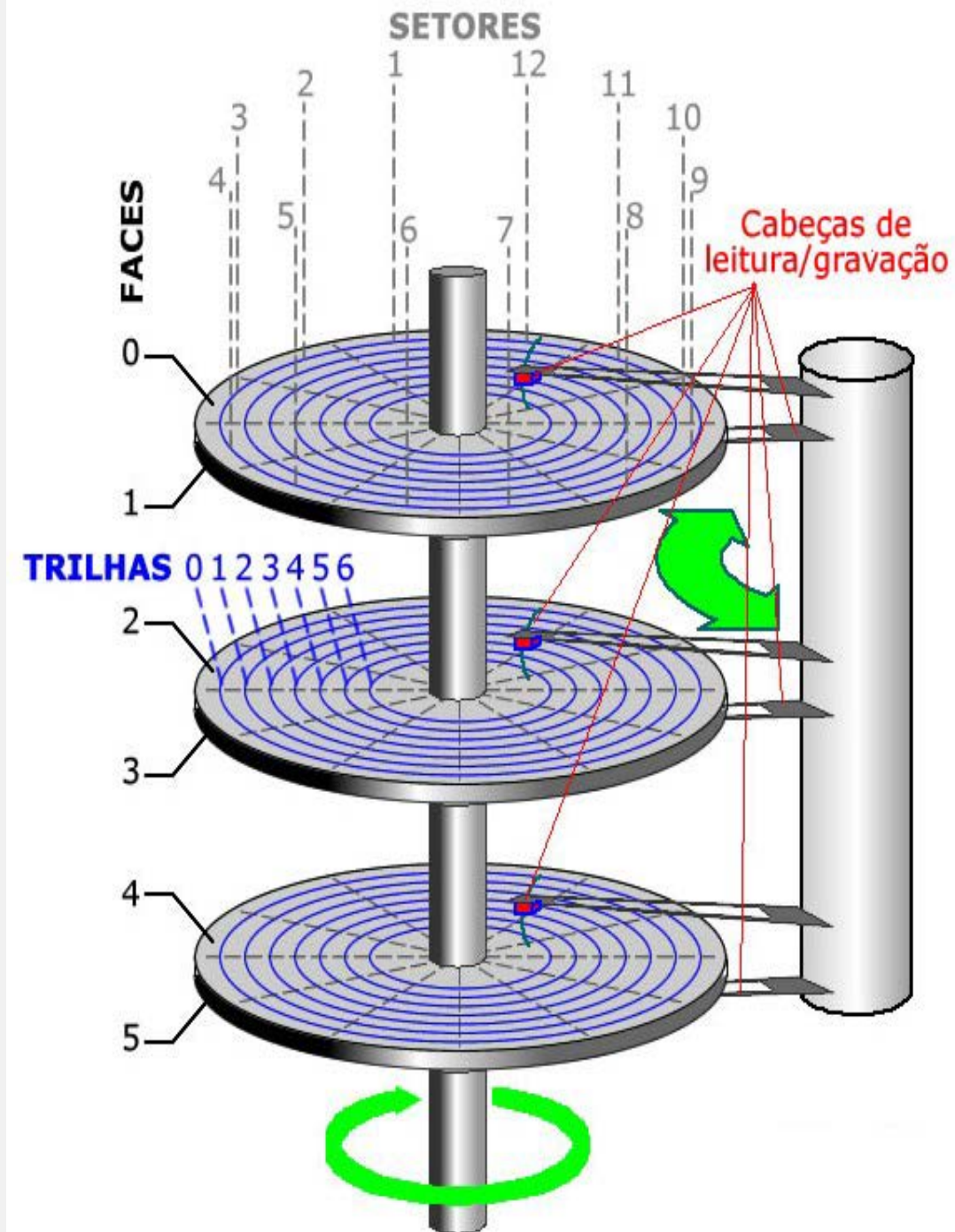


Estrutura do disco

Interior de um HD

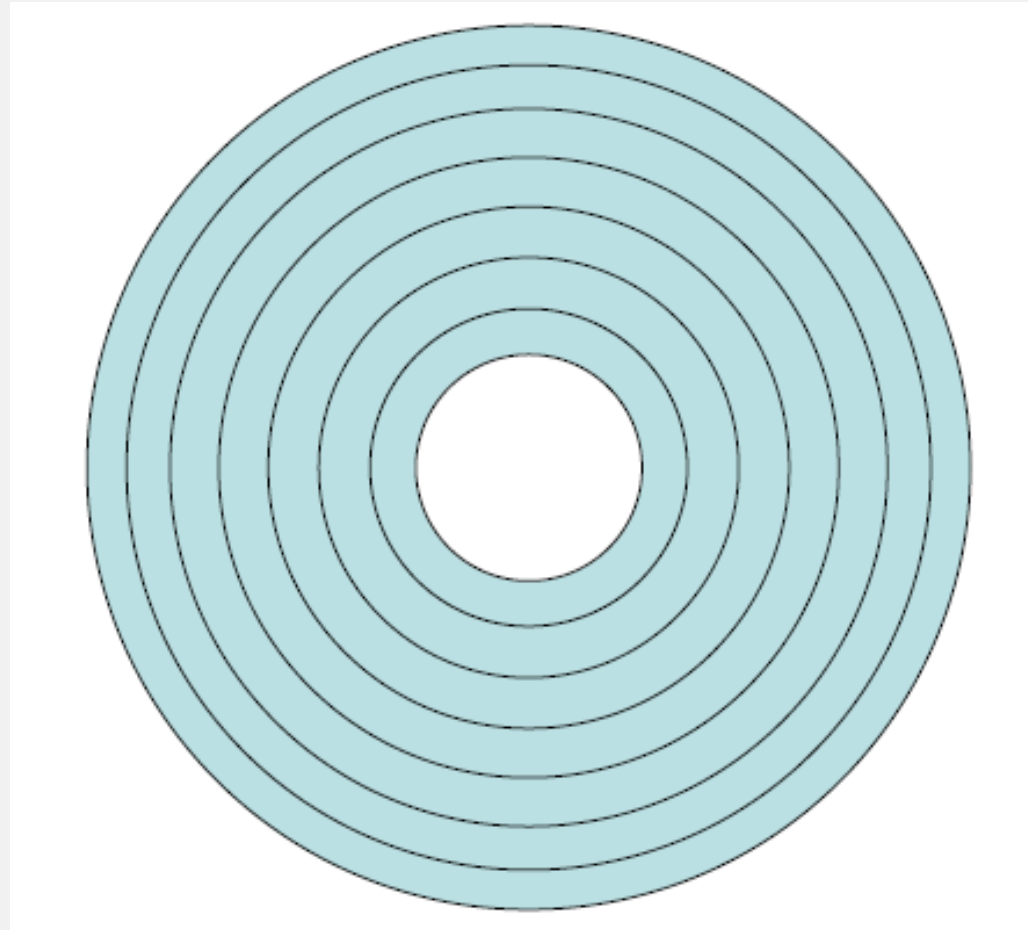
Estrutura do disco

- Os discos rígidos são **divididos** em:
- Setores (menor porção endereçável do disco)
 - Trilhas
 - Cilindros
- O **conjunto** destas informações é denominado **geometria do disco**



Estrutura de disco

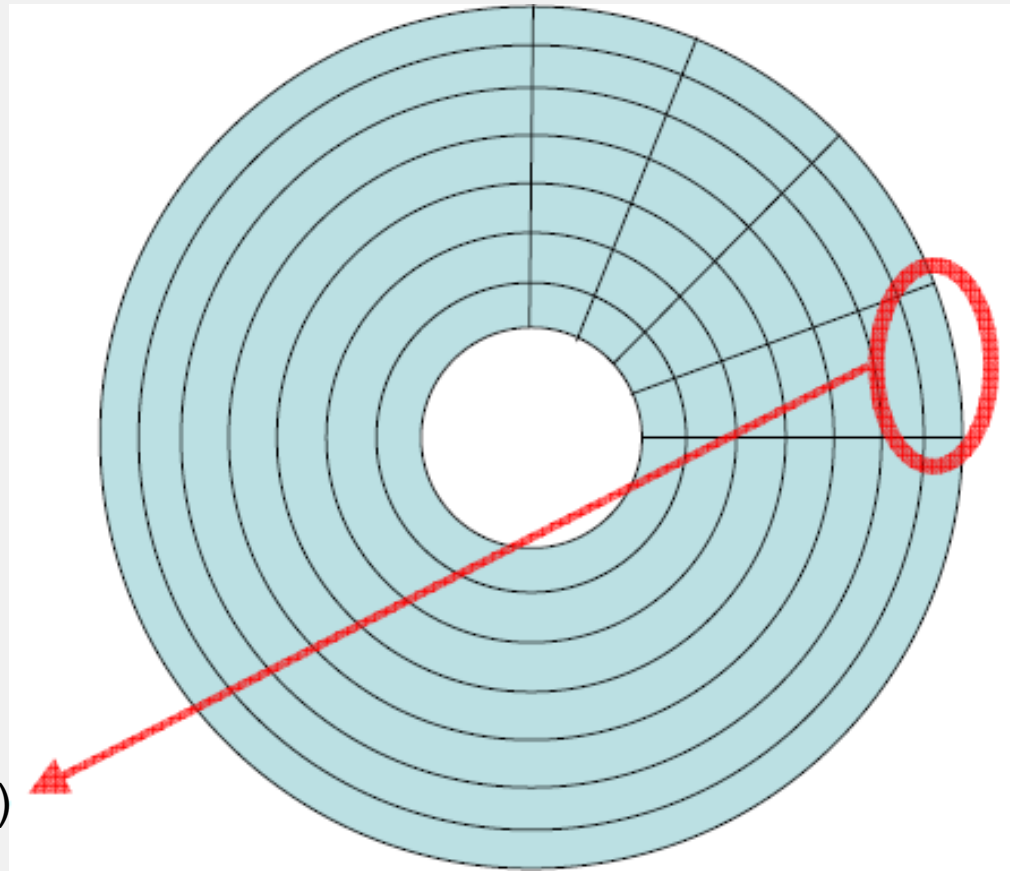
- O HD possui uma **cabeça de leitura** para **cada face** de **disco**, mas **todas** as **cabeças** estão **presas** na mesma **peça de metal**, o **braço de leitura**, por isso **não possuem movimento independente**: para onde uma vai, todas vão.
- O termo **cilindro**, corresponde à **todas** as **trilhas** de **mesmo número**. Ex.: o cilindro 0 é formado por todas as trilhas número 0 (zero), em todas as faces de disco.



Estrutura do disco

Trilhas: círculos concêntricos

Um setor (512 bytes)



Estrutura do disco

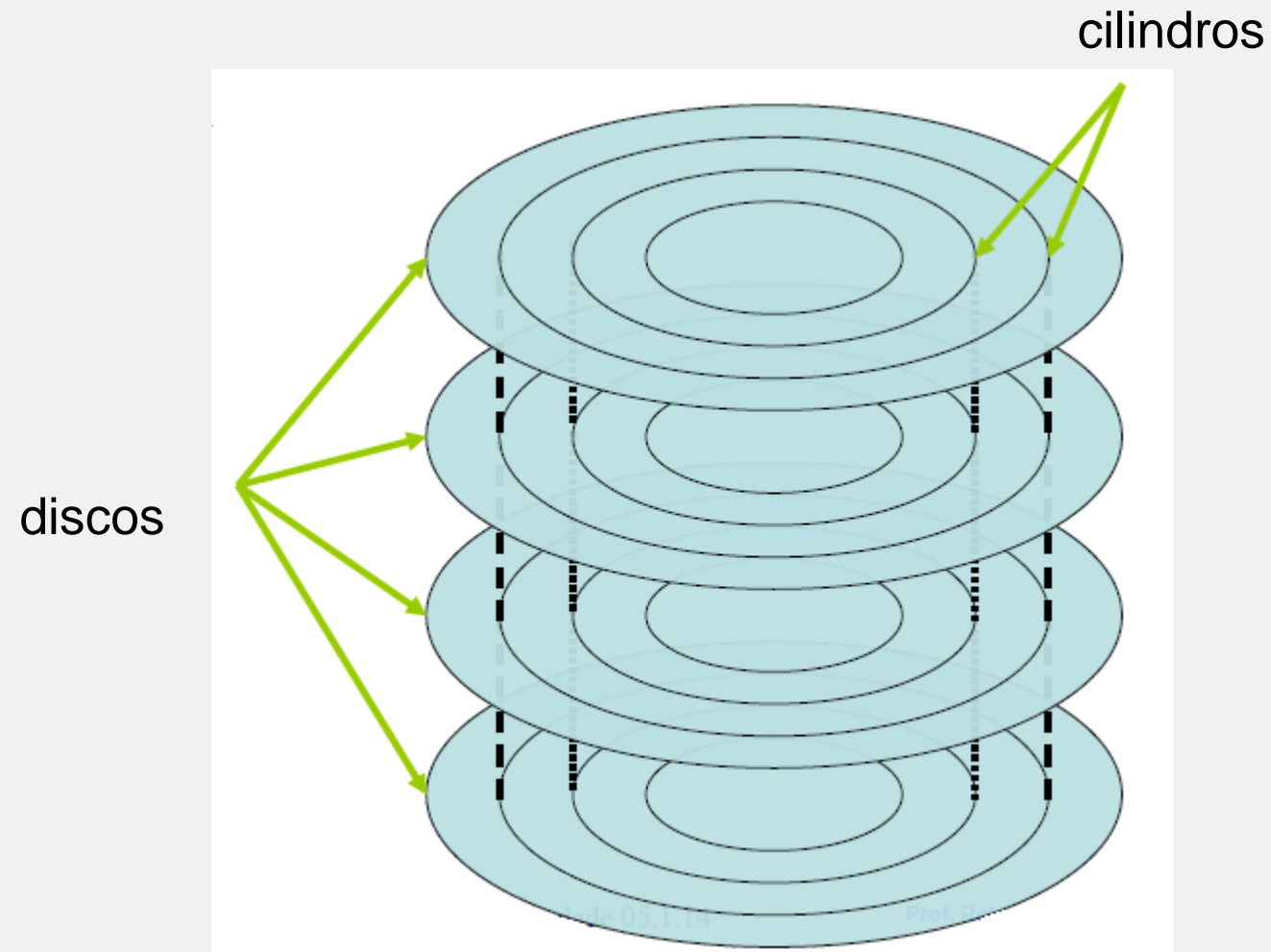
Setores: porção de cada trilha

Estrutura do disco

- HD com N discos internos possui $2N$ faces e uma cabeça para cada face.
- As cabeças não se movimentam independentemente, estão todas presas a mesma peça.
- Quando a cabeça estiver em uma trilha de uma face, todas as outras faces também estarão na mesma trilha.
- Esse é o conceito de **cilindro**.

Estrutura do disco

Cilindro



Operações do Disco para a Memória Principal

- **Divisão física**
 - Corresponde a **geometria do disco** (trilhas, cilindros e setores) - não há como mudar
- **Divisão lógica**
 - Gerado pelo processo de **formatação**
 - O disco é dividido em **blocos** lógicos, também conhecidos como **cluster**.
 - Blocos (cluster) são a unidade de transferência entre o disco e a memória principal.

Operações do Disco para a Memória Principal

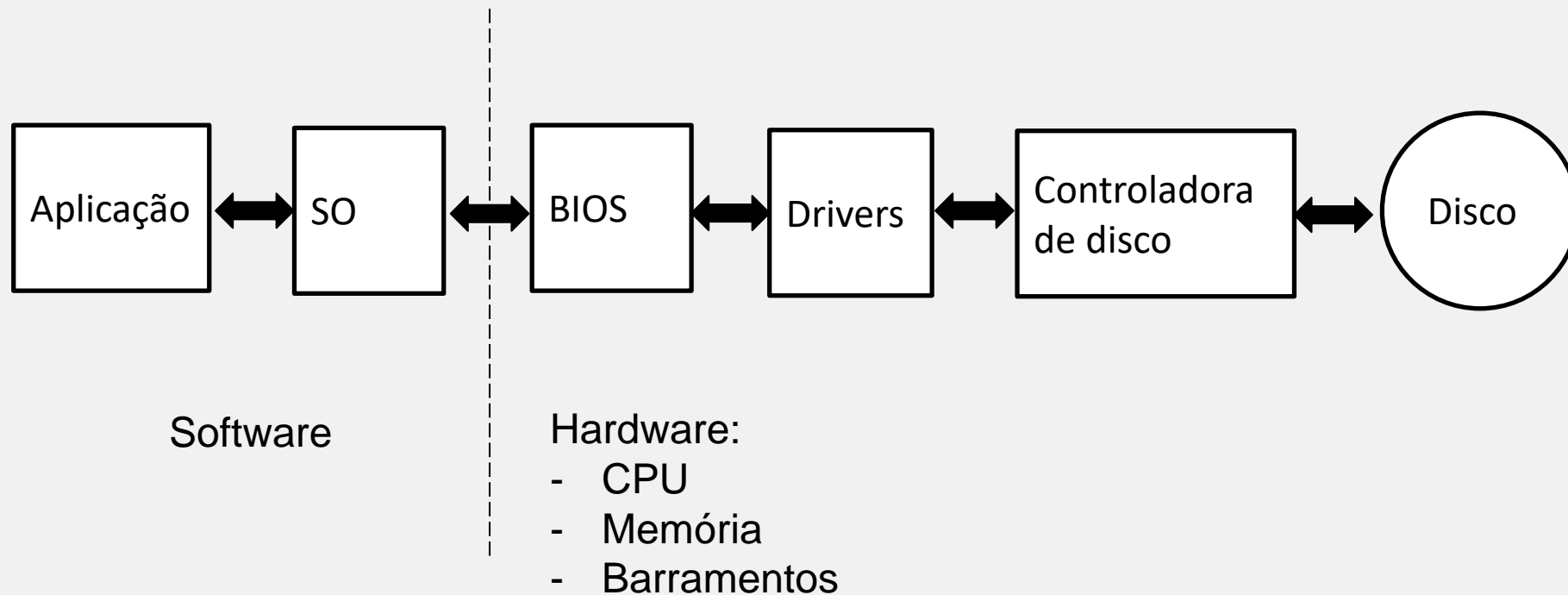
- Os dados **precisam estar** na memória principal para serem **manipulados**
- Bloco (ou *cluster*) é o **nome** da unidade de transferência **entre o disco e a memória** principal
- Se um **único item** no bloco é **necessário**, todo o bloco é transferido
 - Ler ou escrever um bloco = operação de I/O
 - Tempo de acesso = procura + rotação + transferência (latência)

Tempo de Acesso

- É gasto com **procura**
 - Tempo para cabeça **atingir** o bloco necessário
- Com **rotação**
 - Tempo de **espera** para que o **bloco** comece a rodar sob a cabeça do disco
- Com **transferência**
 - Tempo para ler ou escrever no **bloco** = tempo de **rotação** do **bloco** sob a **cabeça** do disco

Tempo de Acesso

Memória Principal – Operação de I/O



Tempo de Acesso

Memória Principal

- Um bloco é a menor unidade do HD que pode ser acessada pelo Sistema Operacional.
- Cada bloco recebe um endereço único.
- Um arquivo grande é dividido em vários blocos para armazenamento.
- Um bloco não pode conter mais de um arquivo, por menor que seja este sempre ocupará ao menos um bloco.

Tempo de Acesso

Memória Principal

- O tamanho do bloco varia de acordo com o **sistema de arquivos** escolhido na formação, exemplos:
 - **FAT16**: de 2Kb até 32Kb
 - **FAT32**: 4 Kb
 - **NTFS**: de 512b até 4Kb
- Observação: quanto menores os blocos, menor a quantidade de espaço em disco desperdiçada.

SISTEMAS DE ARQUIVOS

Sistema de Arquivos

- É a **parte** do Sistema Operacional que **organiza** de forma lógica os **dados** armazenados em **disco**
- Organização padrão:
 - Arquivos: Unidade básica de **dados** enxergada pelo usuário
 - Diretórios: Forma de **classificação** dos arquivos em pastas

Características dos Sistemas de Arquivos

- **Fornecer** uma visão lógica (**abstrata**) do **disco**
- **Proporcionar** o uso eficiente dos **discos**
 - Escalonamento (capacidade de ser colocado em níveis – subpastas)
- **Possibilitar** a organização dos dados
 - Compartilhamento
 - Procura / Classificação
 - Nomeação

Vantagens dos Sistemas de Arquivos

- Do ponto de vista do sistema operacional:
 - Possibilidade de **gerenciamento** do espaço físico
 - **Gerenciar** blocos livres / ocupados
- Do ponto de vista do usuário:
 - Possibilidade de nomear um arquivo
 - Criar, apagar, ler, mover e escrever: manipular arquivos
 - **Controlar** seus direitos de acesso
 - Possibilidade de organização dos arquivos
 - Efetuar backups

Sistema de Arquivos: Interface do Usuário

- A interface **permite** chamadas de sistema:
 - **Manipulação** de arquivos (c/ simples click ou via shell):
 - Open, close, read, write...
- O **sistema operacional** efetua o **vínculo** entre nome simbólico e **lugar** de armazenamento no disco (espaço físico)
- Para o usuário o arquivo é caracterizado:
 - Por um **nome** (visualizado com ou sem extensão)
 - Por **atributos** (somente leitura, leitura e escrita, arquivo de sistema, etc)
 - Por uma **organização lógica** (a estrutura de diretórios, uma extensão)

Arquivo

- Os **arquivos** possuem um **espaço** de endereçamento lógico **contíguo** (ex: fat32 x arq. 100k)
- São **formados** por um conjunto de registros **relacionados**
 - Exemplo: conteúdo do arquivo com dados de clientes
- Conjunto de dados armazenado em um dispositivo físico **não-volátil**.
- Formado por um espaço de endereçamento lógico **contíguo**
- Versáteis em **conteúdo** e **capacidade**
 - Podem conter desde um texto ASCII com alguns bytes até sequências de vídeo com dezenas de gigabytes, ou mais.

Arquivo

- Em sua forma mais simples, um arquivo contém basicamente uma **sequência de bytes**, que pode estar **estruturada** de diversas formas para representar **diferentes tipos de informação**.
- Cabe às **aplicações** interpretá-las.

Arquivo

- Uma forma de organização de arquivo é a **baseada em registros**:
 - Exemplo: arquivo com dados de vários clientes
- Usado por banco de dados para armazenar as coleções de dados em alguma mídia de armazenamento.
- Permite recuperar, atualizar e processar os dados esses dados conforme necessário.

Arquivo

- Na organização baseada em registros um arquivo é visto como uma sequencia de registros.

REGISTRO, CAMPO E CHAVE

Registro

- Unidade lógica **manipulada** por um **programa**, constituída por uma **sequência de itens** chamados de campos ou atributos
 - Por exemplo: <código, nome, data_nascimento>

Formato dos registros

- Tamanho fixo:
 - O **número** de **campos** é fixo
 - Cada **campo** tem o mesmo comprimento em **todos** os **registros**
- Tamanho variável:
 - A quantidade de campos pode variar
 - **Campos** podem ter tamanhos variáveis em **diferentes registros**

Campo

- **Espaço** de um ou mais caracteres **onde** são **armazenados os dados** de mesma natureza
 - Possuem: nome, tipo, tamanho
- **Exemplo:**

Nome do Campo	Tipo	Tamanho
codigo	Int	4
Nome	String	40
data_nascimento	string	10

- Campo: corresponde a cada uma das informações que se deseja modelar a respeito da entidade ou objeto considerado.

Chave

- É um **valor** de um ou mais campos que permite a recuperação de um ou mais registros:
 - Ex: nome (João)

codigo	nome	data_nascimento
10123	João	01-03-1990
10321	José	30-03-1985
10333	Maria	11-11-1991
10432	João	31-12-1980

Tipos de Chave

- **Chave primária:**
 - Possibilita a identificação única dos **registros**
 - Apresenta obrigatoriamente um valor diferente para **cada registro** (Exemplo: código, CPF)
 - Uma chave primária deve ser mínima, isto é, não deve conter campos supérfluos.
- **Chave secundária:**
 - **Permite** a recuperação de registros, mas **não** necessariamente é **única**
 - Exemplo: nome, data_nascimento, salário

Tipos de Chave

- **Chave** de acesso (índice):
 - Chave utilizada para **acessar** (identificar) o **registro desejado**
- **Chave** de ordenação:
 - Chave utilizada para **estabelecer** a ordem dos registros

MÉTODOS DE ACESSO

Métodos de Acesso

Acesso Sequencial

- O acesso sequencial refere-se ao **armazenamento e recuperação** de registros na **ordem** de "um após do outro"
- Os registros geralmente **são armazenados** numa **ordem ascendente ou descendente**, por um código de registro

codigo	veiculo	ano
00001	Gol	2009
00002	Palio	2007
00003	Fox	2011
00004	Celta	2008
...

Métodos de Acesso

Acesso Sequencial

- É **usado** principalmente nos **meios de processamento** por lotes (onde, geralmente, ocorre uma modificação em todos os registros de um arquivo ou na grande maioria deles).
 - Exemplo, a **atualização** de **determinado percentual** no campo “salário”
- Desvantagem
 - É **ineficiente** para aplicações nas quais **apenas** uma **pequena proporção** dos registros **são afetadas** por um dado lote de transações
 - Pois, neste caso, **pode ser necessário** ter que **passar** todo o arquivo para **atualizar** uns **poucos** registros

Métodos de Acesso

Acesso Sequencial

- A **leitura é feita comparando-se** o argumento de pesquisa com cada registro lido, de **forma sequencial**
- Na **inclusão deve ser gerado** um novo arquivo **a partir** do atual, para **intercalar** o novo registro (qualquer modificação no arquivo gera um novo arquivo)
- Na **exclusão**, assim como na **inclusão**, deve ser gerado um novo arquivo a partir do atual, eliminando o **registro desejado**

Métodos de Acesso

Acesso Sequencial

- Outra forma de exclusão, para **arquivos em disco**, é **indicar** em um **campo adicional** um **estado de registro excluído** (um campo para “marcar” a exclusão lógica do registro, sem excluí-lo fisicamente/de verdade do arquivo)

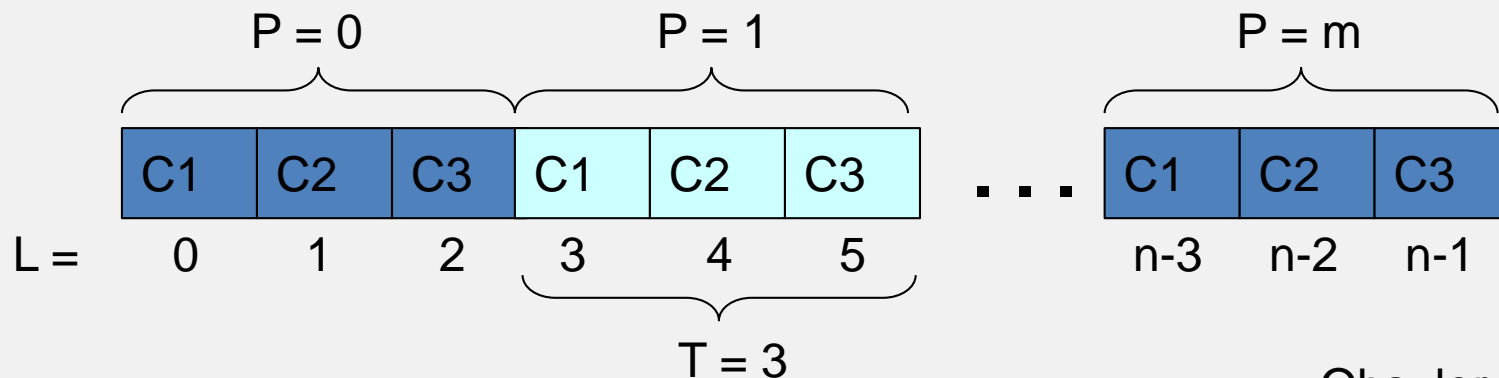
codigo	veiculo	ano	status
00001	Gol	2009	1
00002	Palio	2007	0
00003	Fox	2011	0
00004	Celta	2008	1
...

Métodos de Acesso

Acesso Sequencial

- Um **exemplo de acesso** a registros de tamanho fixo é **dado** pela **fórmula**: $L = (T * P)$
 - **L = Localização inicial**: posição do início do registr
 - **T = Tamanho do registro**: tamanho do registro em bytes
 - **P = Posição Registro**: a posição do registro desejado

Ex: $P = 1$ trará a localização inicial do segundo registro



Métodos de Acesso

Acesso Sequencial Indexado

- **Formado** por um arquivo sequencial (arquivo de registros) e por um índice (arquivo de índices)
- Utilizam **chaves** que servem de índice para a **localização** do endereço do registro (p. ex.: a linha do arquivo sequencial)
 - Um **índice** é sempre específico (único) para uma **chave** de acesso
- A **finalidade** de um índice é **permitir** a rápida determinação do endereço de um registro, dado um **argumento** de **pesquisa** (p. ex.: um código ou uma palavra)

Métodos de Acesso

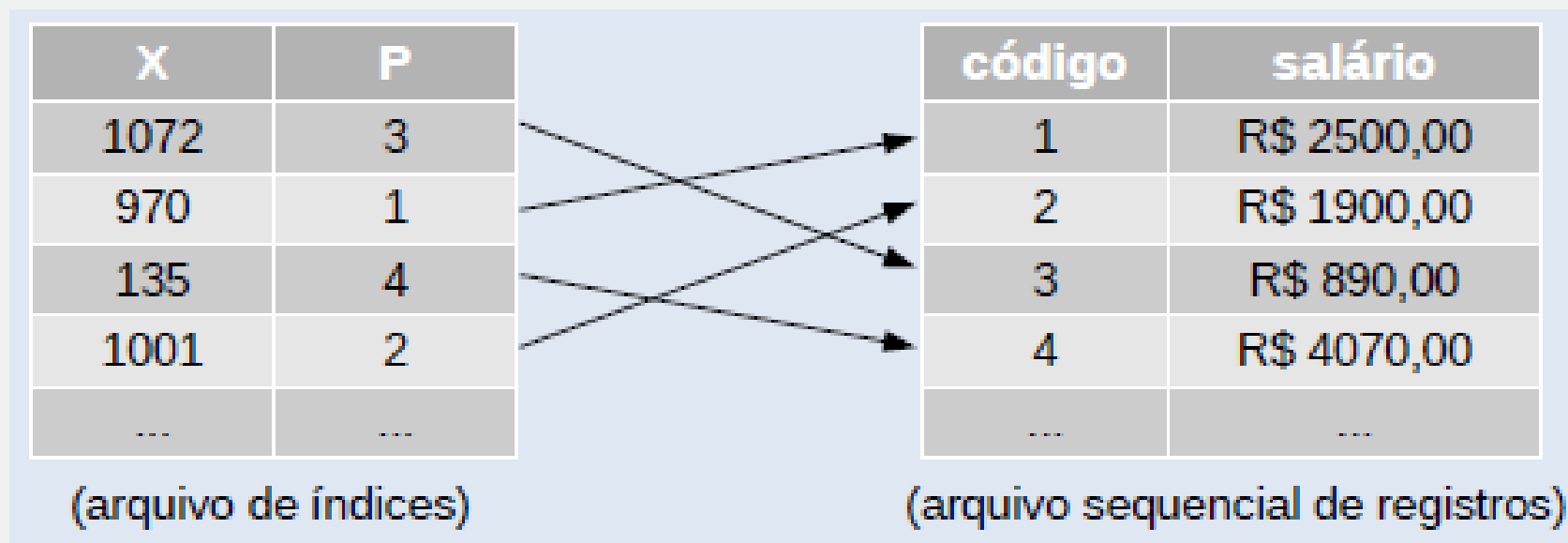
Acesso Sequencial Indexado

- Utiliza o **princípio** da pesquisa sequencial, onde **cada** registro é **lido** sequencialmente até encontrar uma **chave de acesso** igual ou maior que chave de pesquisa (ex.: maior = interromper busca; igual = registro encontrado; menor = ir próximo registro)
- **Providências** necessárias para **aumentar** a eficiência deste **método de acesso**:
 - O arquivo sequencial **deve** ser **mantido** ordenado por um **campo** chave de acesso (ou chave de registro)
 - Um arquivo de índices contendo **pares de valores** $\langle X; P \rangle$ **deve** ser **criado**, onde X **representa** uma chave de acesso e P o endereço do registro **apontado** pela **chave** X

Métodos de Acesso

Acesso Sequencial Indexado

- Exemplo: **estrutura** dos **arquivos** para acesso sequencial indexado de um **conjunto** de **15** registros (4 mostrados)



- X = chave de acesso, P = índice único (código do funcionário)

Métodos de Acesso

Acesso Indexado

- Os registros são **acessados** sempre através de uma ou mais organização de **índices**
- Cada organização de índices tem uma **chave diferente** para acessar os registros de dados
- Não há qualquer compromisso com a **ordem física** de intercalação dos registros

EXERCÍCIOS

Exercícios

1. Dado um arquivo sequencial, com registros de tamanho fixo, contendo números de **CPF** (11 bytes) seguidos de **datas de nascimento** (8 bytes). Identifique:
- Quantos registros existem no arquivo?
 - Qual a posição do 3º registro? (Crie uma formula)
 - Liste um registro por linha e separando os campos por “,”

1	2	3	4
12345678901	2345678901	2345678901	2345678901
-----	-----	-----	-----
0214783452314021984081894723280904197901			
6443766302110199609178983440180519861677			
3422110231220010017325545008091968040901			
8769025111981007890899900909199917531351			
51210101987			

Exercícios

2. Utilizando o mesmo arquivo (String) do exercício 1, faça:

- Um **método** que mostre na tela os registros de acordo com a imagem ao lado;
- Salve os registros em um arquivo csv.
 - Lembrando que em um arquivo os campos são separados por “,”
 - Não esqueça de colocar a descrição do campo no csv, ou seja, inicie o arquivo com: cpf,data de nascimento
 - Verifique o resultado abrindo o arquivo csv no Excel ou no Calc.

021.478.345-23	14/02/1984
081.894.723-28	09/04/1979
016.443.766-30	21/10/1996
091.789.834-40	18/05/1986
167.734.221-10	23/12/2001
001.732.554-50	08/09/1968
040.901.876-90	25/11/1981
007.890.899-90	09/09/1999
175.313.515-12	10/10/1987

DICA: para obter os campos do registro você pode utilizar

- `substring()` ou
- `regex`

Exercício Bônus

1. Utilizando o programa desenvolvido para os exercícios anteriores, ajuste-o para:
 - Aceitar arquivos que o usuário informar (você pode definir o tipo: .txt, .dat)
 - Utilizar um *parser* para os parâmetros que o usuários pode utilizar (Ver exemplo do professor)

Leitura Complementar

- Características e funcionamento dos HDs
<http://www.infowester.com/hd.php>
- O que é SSD (Solid-State Drive)
<http://www.infowester.com/ssd.php>
- Capítulo 5 – **Sistemas de Arquivos** do livro **Sistemas Operacionais: Projeto e Implementação**, de Andrew Tanenbaum

Obrigado

*jacksonpradolima.github.io
github.com/ceplan*