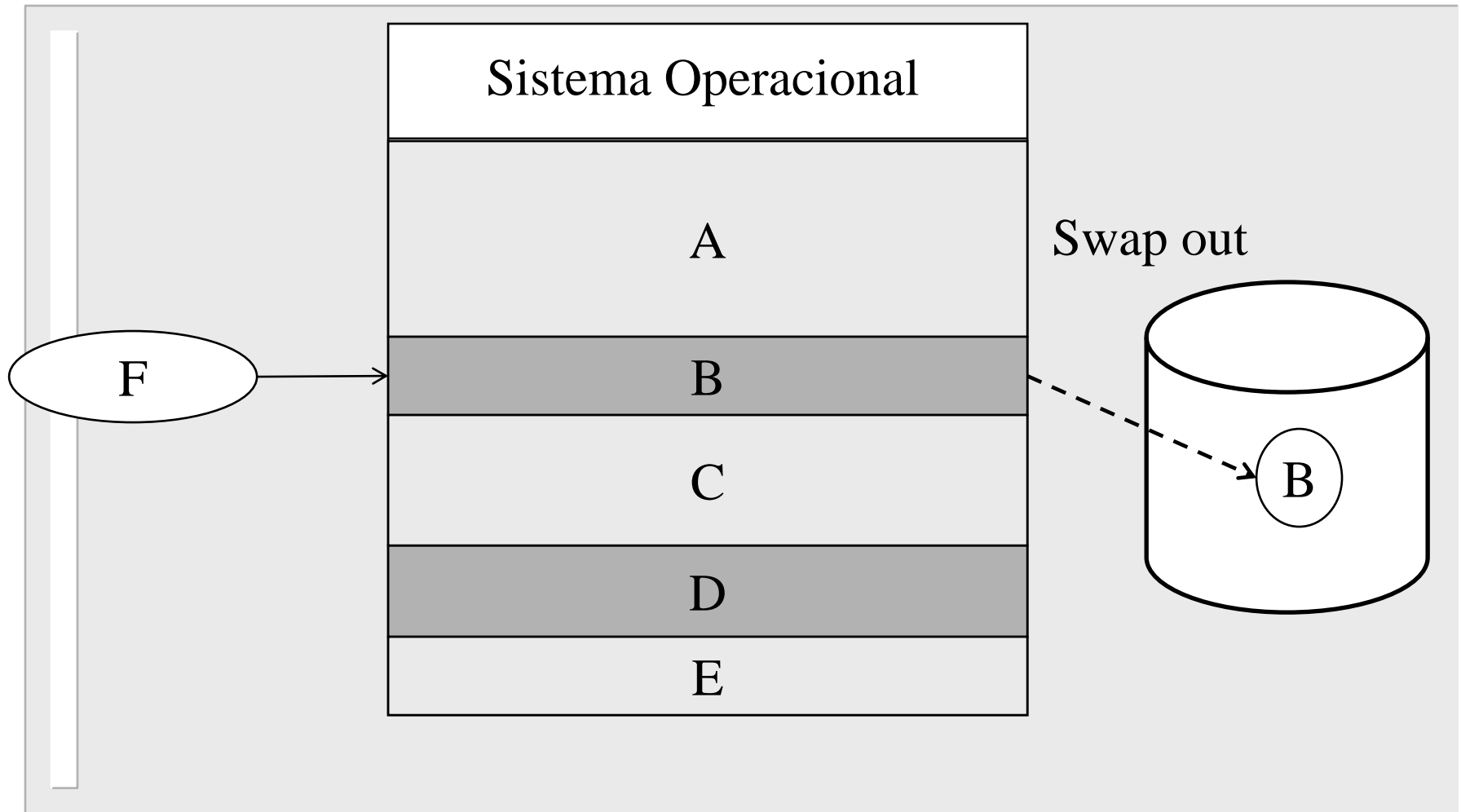


Danton Cavalcanti Franco Junior  
falecom@dantonjr.com.br

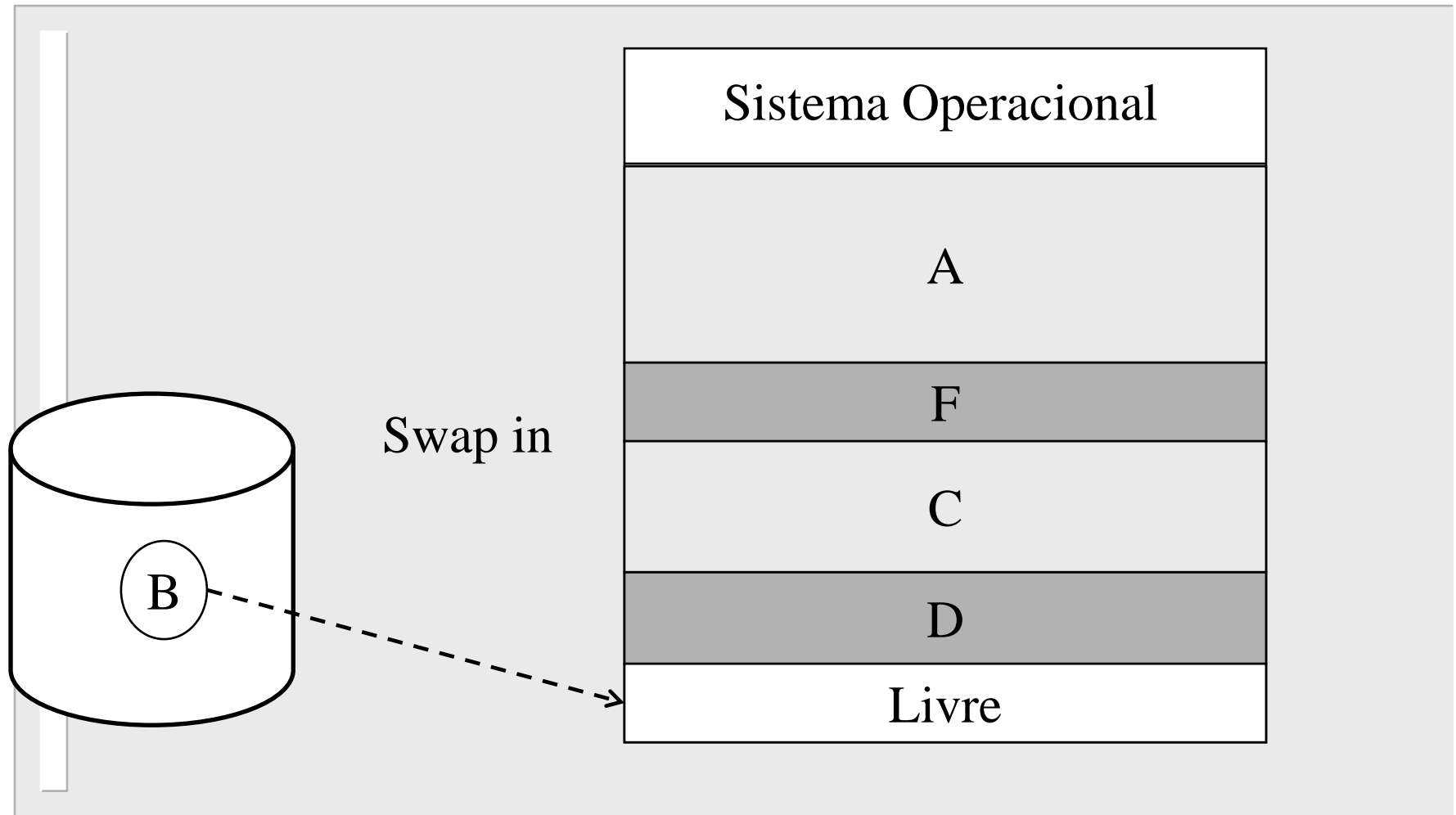
# *Memória Principal*

- **Swapping:** Resolver o problema de insuficiência de memória.
  - Nos mecanismos anteriores, o programa sempre permanecia na memória.
  - O sistema escolhe um programa residente que é levado para o disco (swap out) retornando para a memória principal (swap in) como se nada tivesse ocorrido.
  - Tempo gasto para o carregamento (relocação dinâmica)

# *Memória Principal*



# *Memória Principal*



# ***MMU***

- ❑ **Memory Management Unit** - Unidade de gerência de Memória. É o componente do hardware responsável por prover os mecanismos usados pelo Sistema Operacional para gerenciar a memória.
- ❑ Vai mapear os endereços lógicos gerados pelos processos nos correspondentes endereços físicos que serão enviados para a memória.

# *MMU*

- **Mecanismos de proteção de memória**
  - **Registrador de limite:** Testa o intervalo. Se não estiver no intervalo é um endereço ilegal.
  - **Registrador de Base e Limite:** Testa o limite inicial para o endereço lógico e soma ao registrador de base para obter o endereço físico.

# *Memória Virtual*

- ❑ Técnica sofisticada e poderosa de gerência de memória, onde memória principal e secundária são combinadas (ilusão de existir mais memória que a principal).
- ❑ Permite a execução de programas que não foram carregados para a memória física, pois desvincula seu tamanho do tamanho da memória física.
- ❑ Minimiza o problema da fragmentação.

# *Memória Virtual*

- ❑ **Espaço de Endereçamento virtual**
- ❑ Muito próximo da idéia de um vetor, referência da posição em que um dado se encontra.
- ❑ Os programas não fazem referência aos endereços físicos (endereços reais), mas sim aos endereços virtuais.



# *Memória Virtual*

- No momento da execução, o endereço virtual é traduzido para o endereço físico, pois o processador apenas acessa posições da memória principal.
- Espaço de endereçamento virtual é o nome que se dá ao conjunto de endereços virtuais que os processos podem endereçar.

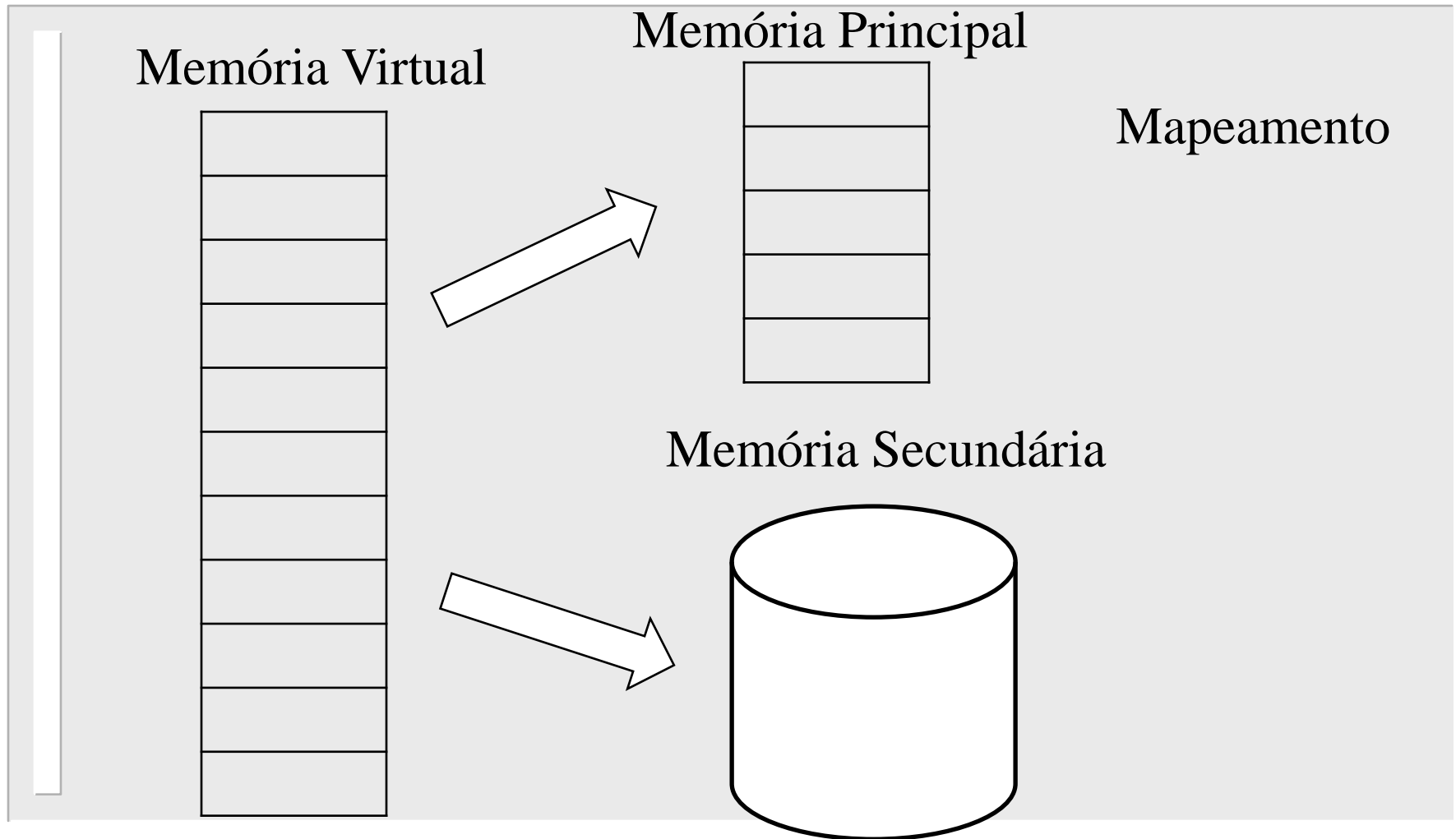
# *Memória Virtual*

End. Físico		Vetor
500		Vetor[1]
501		Vetor[2]
700		Vetor[3]
720		Vetor[4]
350		Vetor[5]
100		Vetor[6]
502		Vetor[7]

# *Memória Virtual*

- ❑ **Mapeamento**
- ❑ Traduzir um endereço localizado no espaço virtual para um espaço real.
- ❑ Um programa não precisa estar contíguo na memória real para ser executado.

# *Memória Virtual*

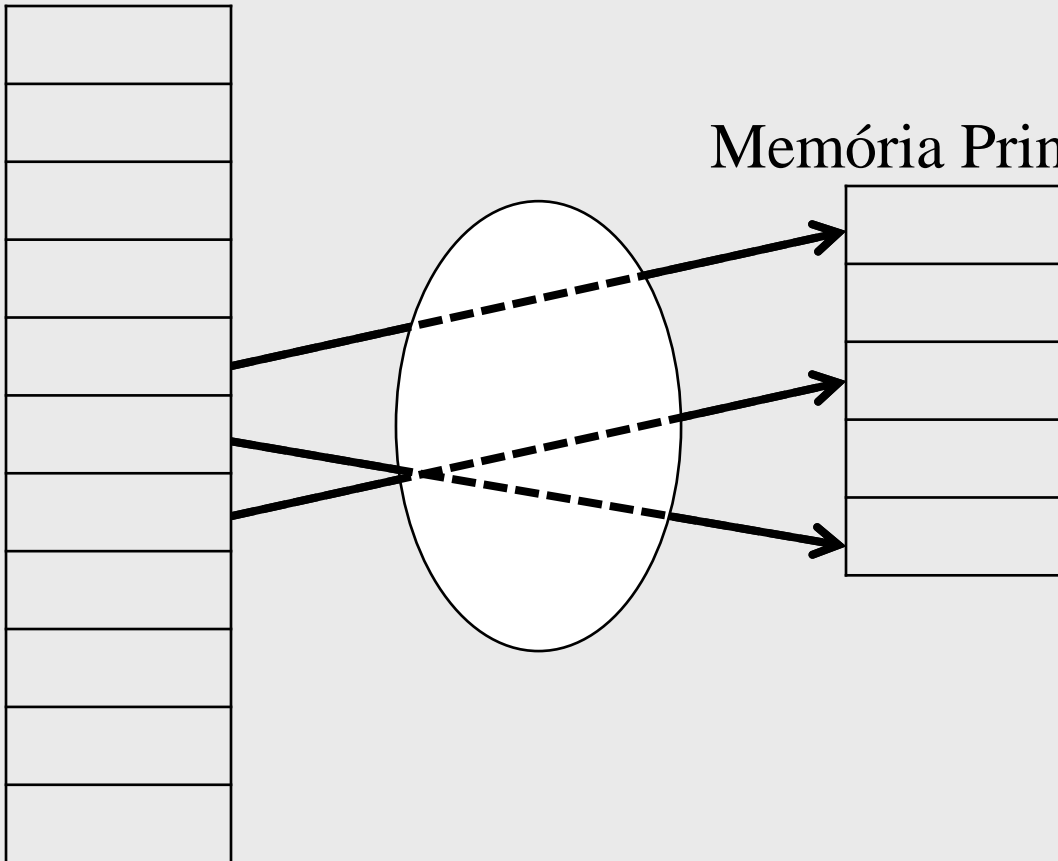


# *Memória Virtual*

Memória Virtual

Mapeamento

Memória Principal



# *Memória Virtual*

- Nos sistemas atuais a tradução é realizada por hardware em conjunto com o SO.
- Memória Associativa ou Lookside buffer.
- Cada processo passou a implementar a sua própria tabela de mapeamento (tabelas exclusivas de mapeamento).

# *Memória Virtual*

Espaço virtual de A

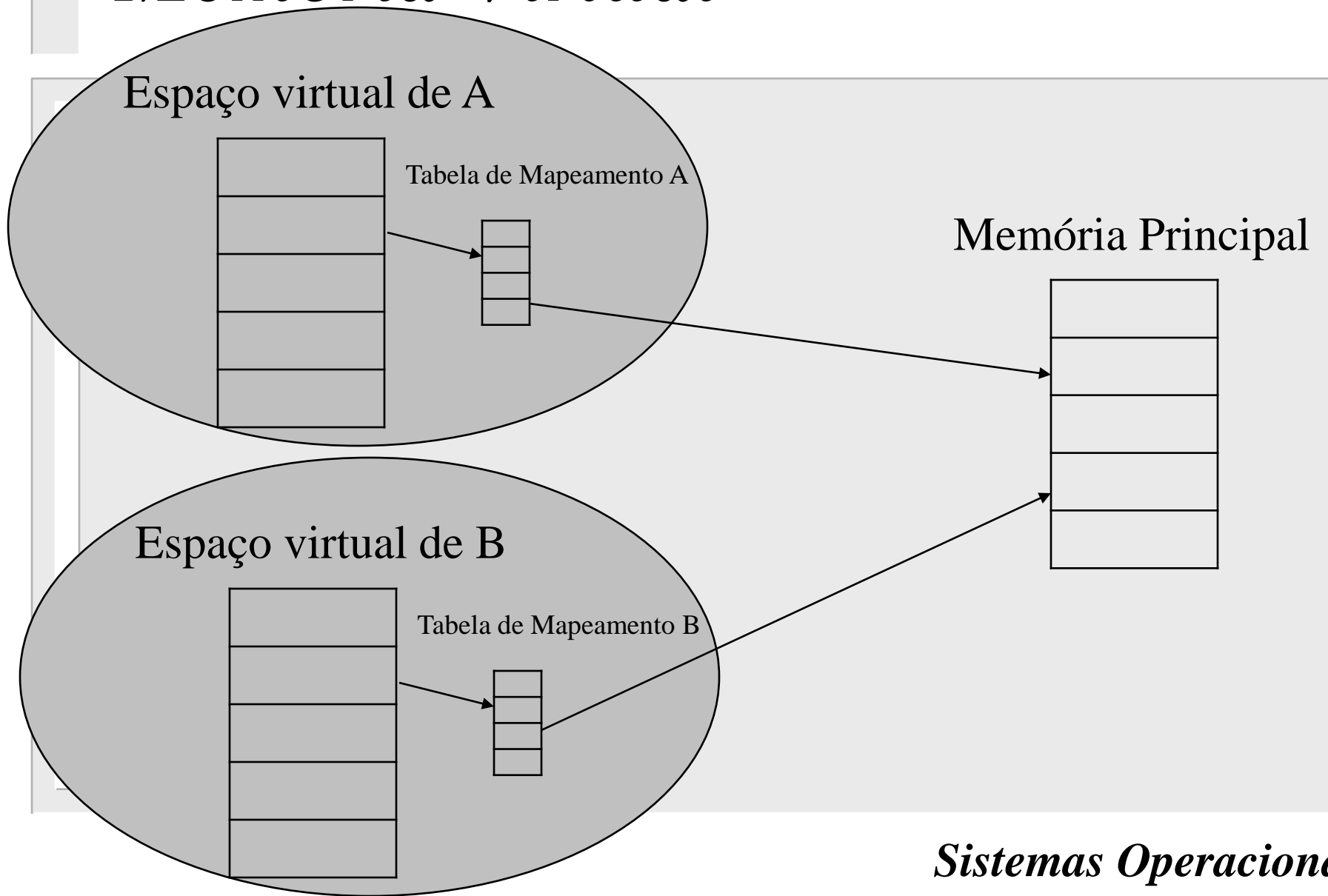
Tabela de Mapeamento A

Memória Principal

Espaço virtual de B

Tabela de Mapeamento B

*Sistemas Operacionais*



# *Memória Virtual*

- ❑ **Paginação (paging)**
- ❑ É a técnica de gerência de memória onde o espaço de endereçamento virtual e o espaço de endereçamento real são divididos em blocos do mesmo tamanho (chamados de páginas). Geralmente os blocos variam de 512 bytes a 64KB.
- ❑ No espaço virtual as páginas são chamadas de páginas virtuais, no espaço real são chamadas de páginas reais ou frames (molduras).



# *Memória Virtual*

- ❑ Todo o mapeamento é realizado em nível de página (tabela de páginas).
- ❑ Cada processo possui uma ETP (entrada na tabela de páginas).
- ❑ Quando um programa executa, as páginas virtuais são carregadas e colocadas em frames.

# *Memória Virtual*

- A tabela de páginas (ETP) possui:
  - A localização da página virtual (número e deslocamento).
  - O bit de validade indicando se a página está ou não na memória física (0 – não está; 1 – está).

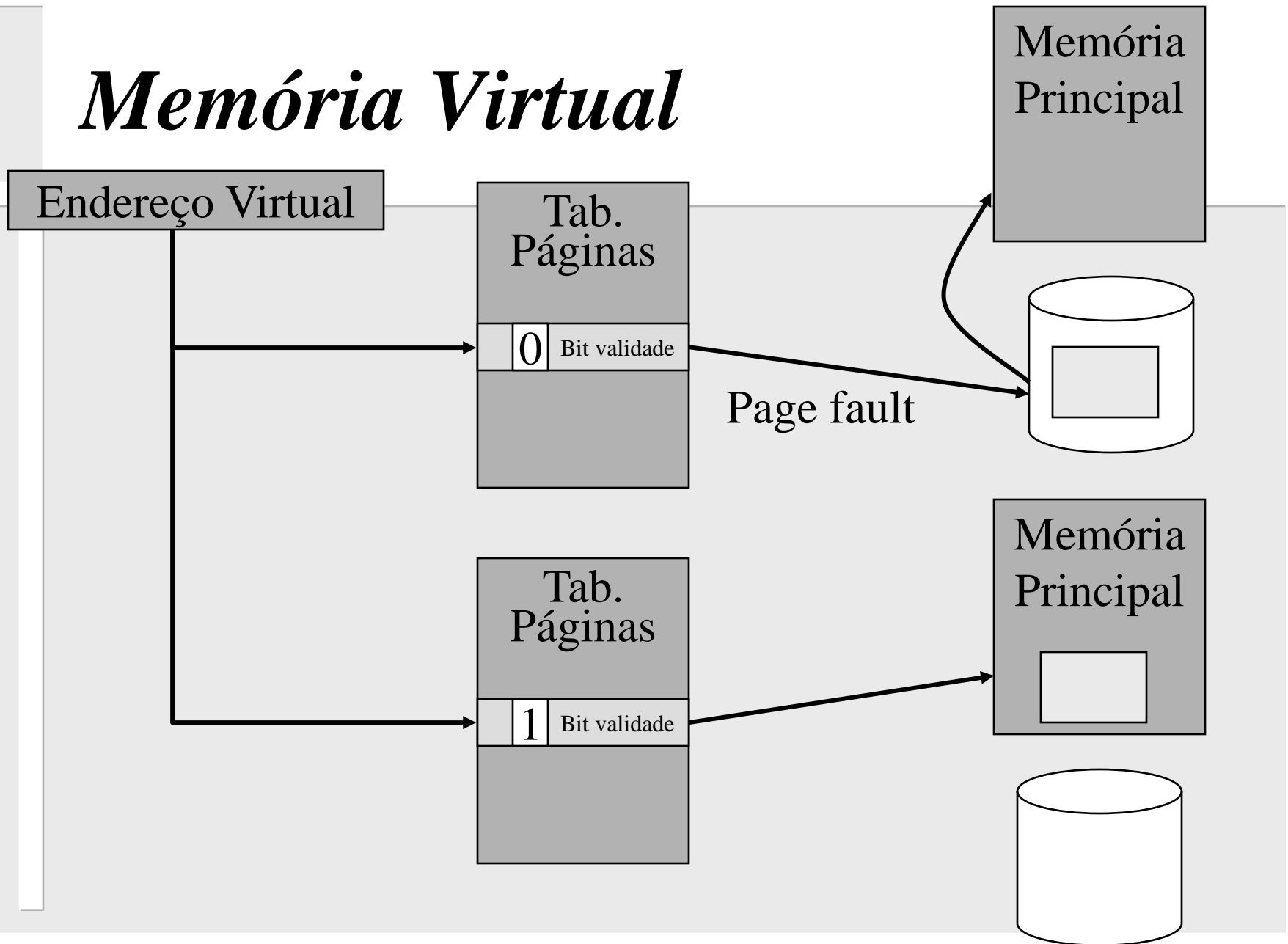
# *Memória Virtual*

- Conceitos da paginação:
  - **Page fault:** ocorre sempre que o sistema é solicitado para verificar se a página está ou não na memória principal, devendo então carregá-la.
  - **Paginação por demanda (demand paging):** as páginas são transferidas para a memória principal na medida em que são referenciadas (é possível que partes do programa nunca sejam carregadas na memória).

# *Memória Virtual*

- Conceitos da paginação:
  - **Paginação antecipada (anticipatory paging)**: o sistema tenta prever as páginas que serão necessárias. Em caso do sistema errar a previsão, haverá perda de tempo e espaço de memória alocado sem necessidade.

# *Memória Virtual*



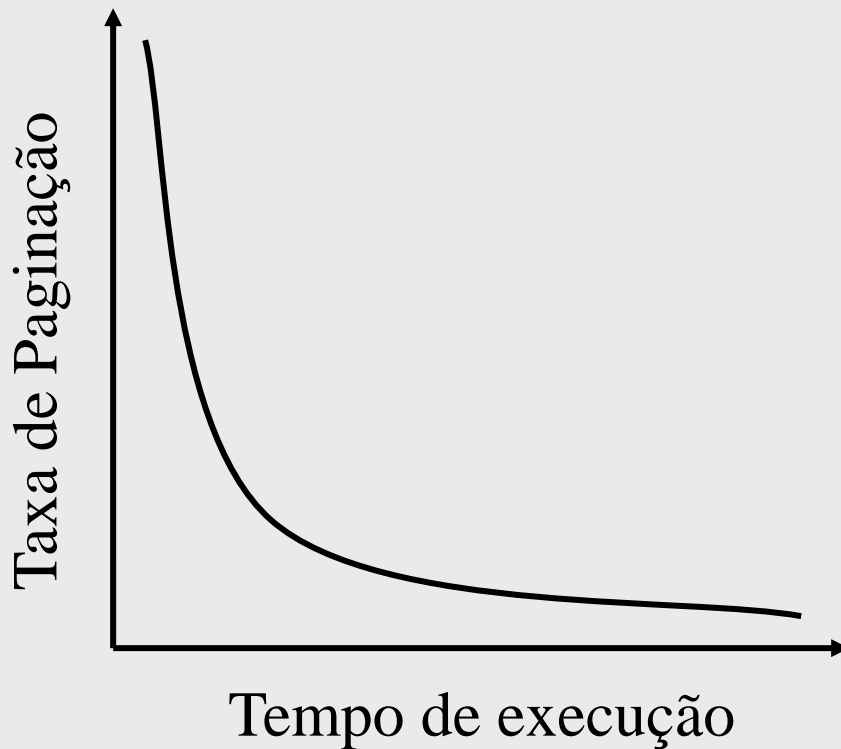
# *Memória Virtual*

- **Working Set:** Conjunto de páginas constantemente referenciadas pelos processos que permanecem na memória principal para que execute de forma eficiente.
- Sempre que o processo faz referência a uma página que não está na memória (page fault), o SO necessita realizar ao menos uma I/O, que deve sempre ser evitada.
- Grande (pouca page fault), poucos processos;  
Pequeno (muita page fault), muitos processos.

# *Memória Virtual*

- **Localidade:** tendência que existe em um programa de fazer referências a posições de memória de forma quase uniforme, ou seja, o processo tende a concentrar suas referências em um mesmo conjunto de páginas.
- Normalmente programas estruturados possuem o conceito de localidade bem definido.
- Não existe ferramentas que permita prever quais páginas serão as próximas a serem referenciadas.

# *Memória Virtual*



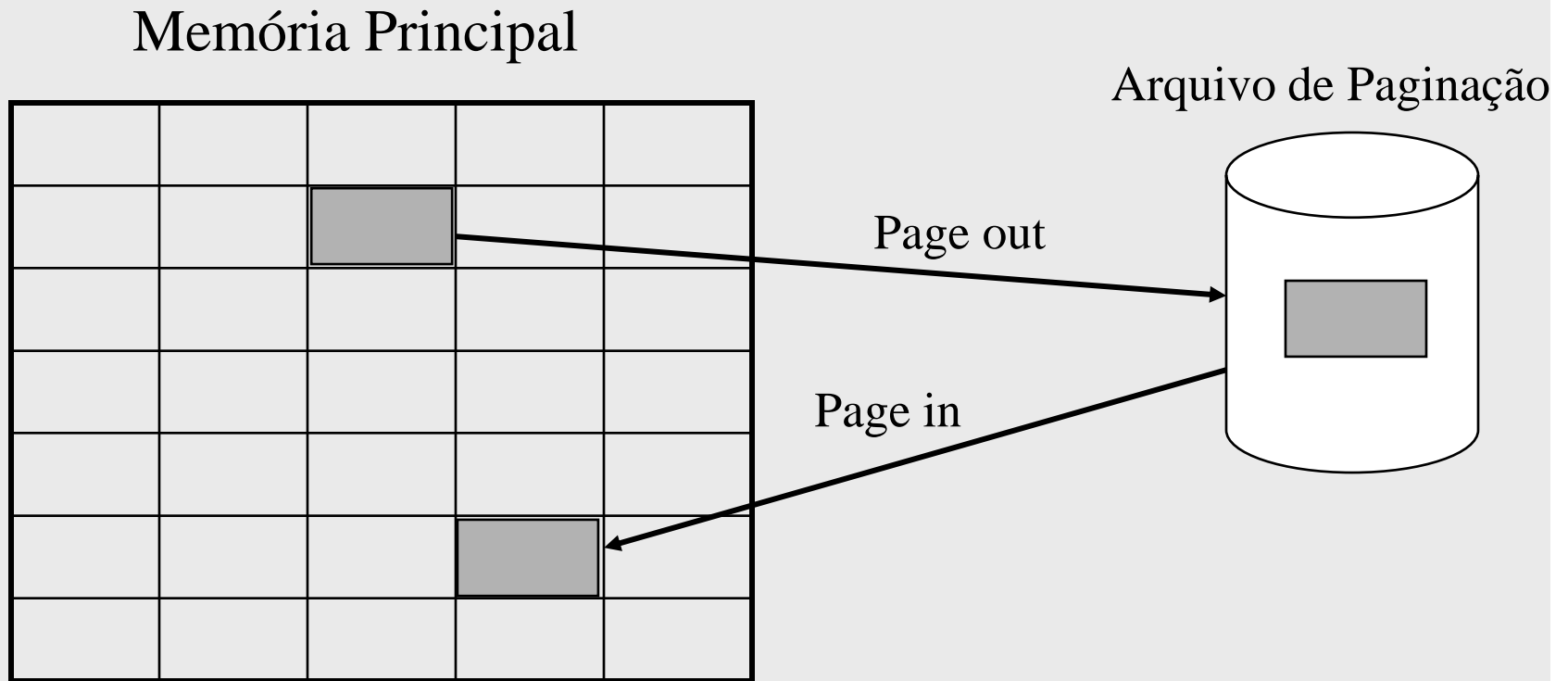
Inicialmente um processo possui grande número de page faults (carga do programa), após a execução esse número tende a baixar, pois os processos tem suas páginas carregadas em seus working set.



# *Realocação de Páginas*

- ❑ Problema em decidir quais páginas deverão sair.
- ❑ Quando o working set de um programa é alcançado, o SO deve escolher quais páginas serão liberadas.
- ❑ O SO deve levar em conta se a página foi ou não modificada (evitar a perda de dados), para isso utiliza-se o bit de modificação (dirty ou modify bit).

# *Realocação de Páginas*



# *Realocação de Páginas*

## □ Estratégias principais:

- **Aleatória (Random):** Todas as páginas do working set podem ser selecionadas. Consome poucos recursos. É pouco utilizada.
- **FIFO (First-In-First-Out):** A primeira que foi utilizada é a primeira a ser escolhida. Fácil implementação. Uma fila simples (páginas antigas no início e recentes no final).

# *Realocação de Páginas*

## □ Estratégias principais:

- **LRU (Least-Recently-Used):** Seleciona a página utilizada menos recentemente, ou seja, a que mais tempo esteve sem ser referenciada. Gera um grande overhead, pois as páginas precisam ser constantemente atualizadas no momento do último acesso.

# *Realocação de Páginas*

## □ Estratégias principais:

- **NRU (Not-Recently-Used):** Páginas que não foram recentemente utilizadas. Através de uma flag, o sistema indica quais são essas páginas, inicialmente todas iniciam com 0, a partir do momento que são utilizadas, sua flag é alterada para 1, a partir de um determinado tempo é possível saber quais páginas foram referenciadas ou não.

# *Realocação de Páginas*

## □ Estratégias principais:

- **LFU (Least-Frequently-Used):** Menos frequentemente utilizada, ou seja, a página menos referenciada. É mantido um contador do número de vezes que a página é referenciada. O algoritmo privilegia as páginas que são bastante utilizadas. Gera um problema, pois as páginas mais recentes serão justamente as que apresentam o menor valor no contador.

# *Realocação de Páginas*

## □ **Tamanho da Página**

- O problema da fragmentação também existe em sistemas com paginação, porém em menor escala.
- A maior ou menor fragmentação é consequência do tamanho da página.
- Páginas pequenas, usam grandes tabelas de mapeamento, aumentando o número de acessos à memória secundária, porém gera menos fragmentação.
- Tamanho da página está associado ao hardware.
- Estudos modernos apontam que o tamanho ideal é de páginas pequenas.