

# Sistemas Operacionais

## Gerenciamento de memória

# Gerenciamento de memória

- Uma das funções do Sistema Operacional é gerência de recursos
- Um dos recursos mais concorridos de um computador é a memória



# Ideal

- Em um mundo ideal a memória seria infinitamente **grande**, infinitamente **rápida**, consumiria **pouca energia** e seria **barata**.



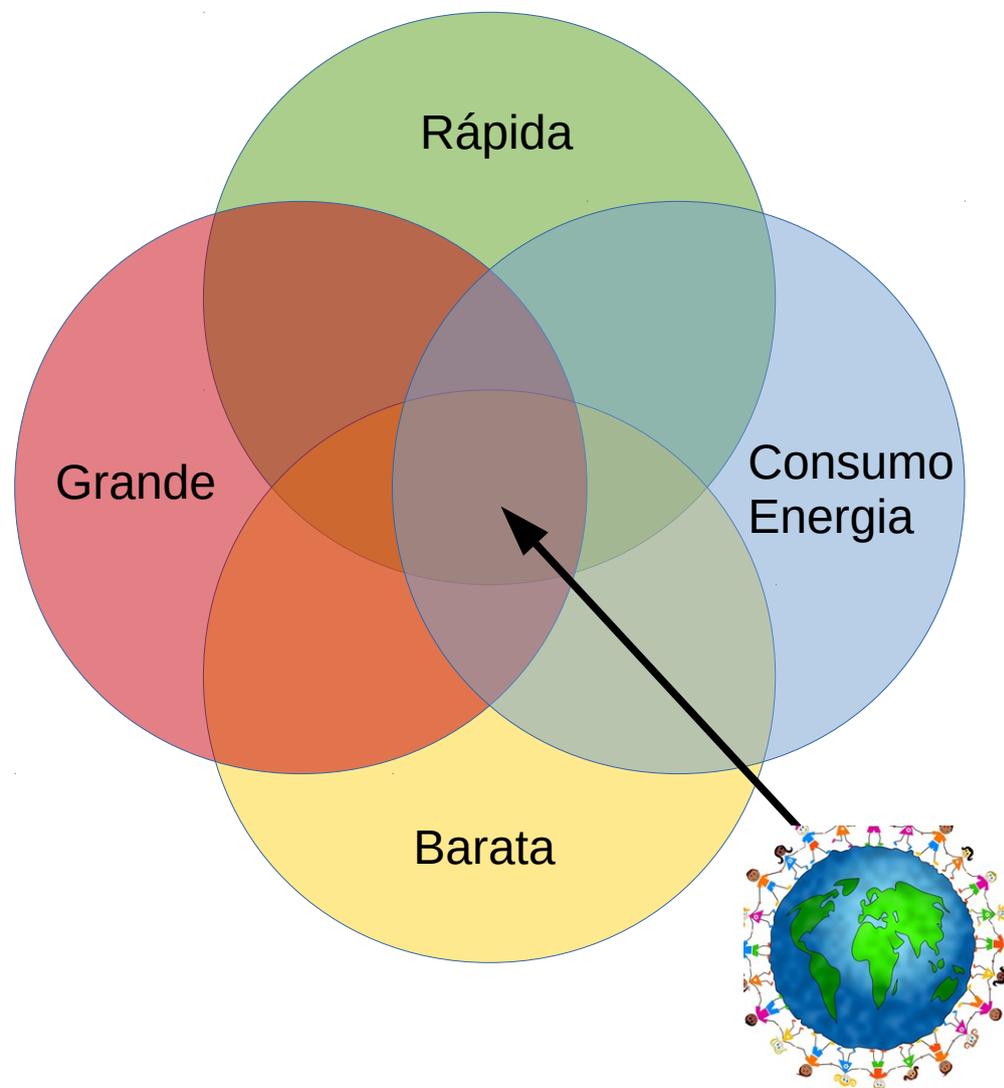
# Ideal

- Infelizmente o mundo ideal não existe
- Grande (HD)
- Rápida (Ram, Cache)
- Consome pouco (Flash?)
- Barata.(Fitas LTO)



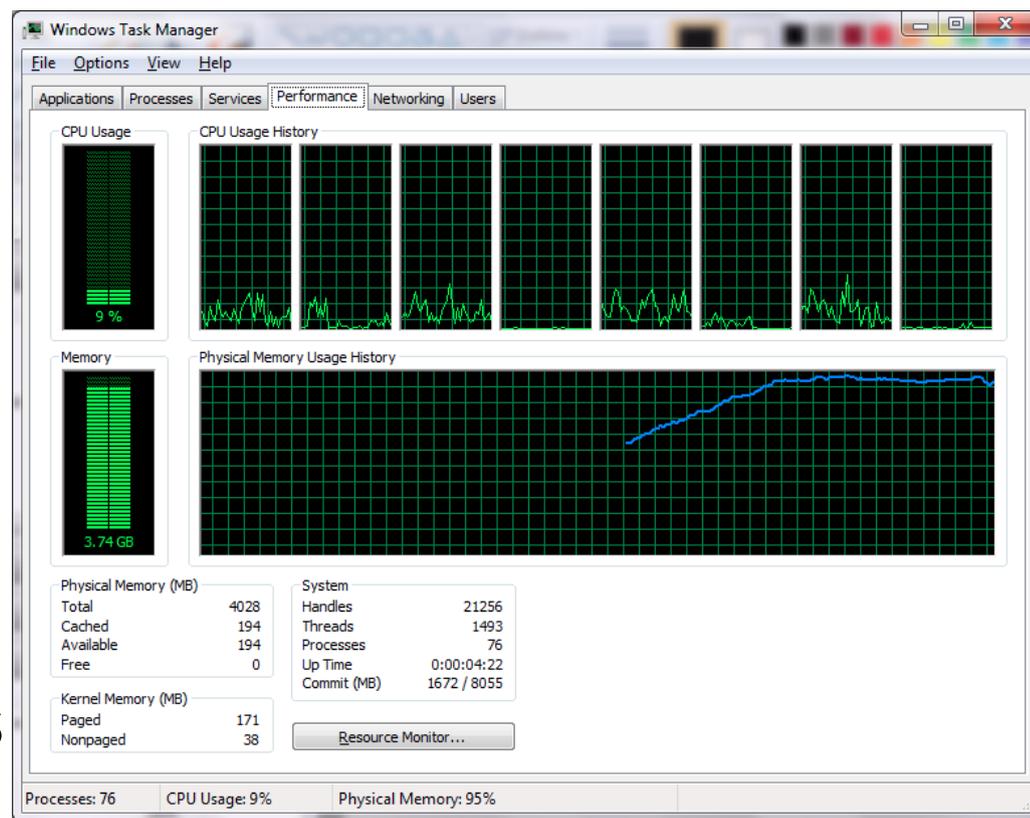
# O mundo das memórias

- Vários tipos de memória
- algumas mais rápidas
- algumas maiores
- algumas mais baratas
- algumas consomem pouca energia
- nenhuma Rápida, Grande e Barata



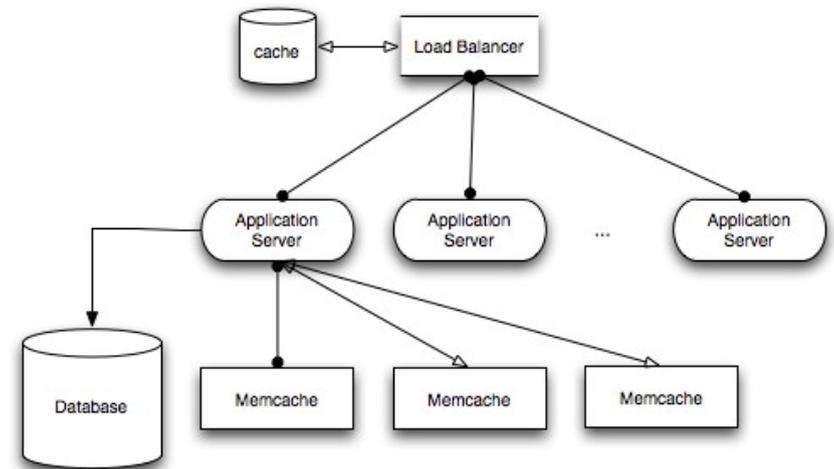
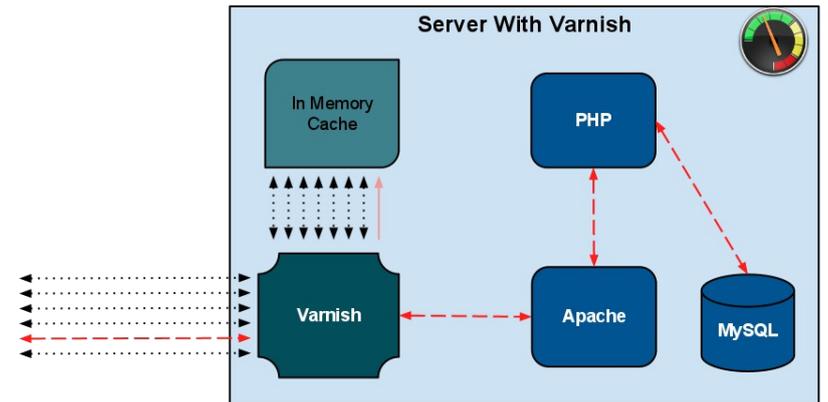
# Uso das memórias

- Programas são mais complexos
- Sistemas consomem cada vez mais memória
- Computadores executam cada vez mais múltiplas tarefas simultaneamente



# Uso das memórias

- Em ambientes de servidores a demanda por memória cresce à medida que clientes se juntam ao serviço.
- Muitas estratégias de escala em servidores se baseiam no cache de disco em memória primária.



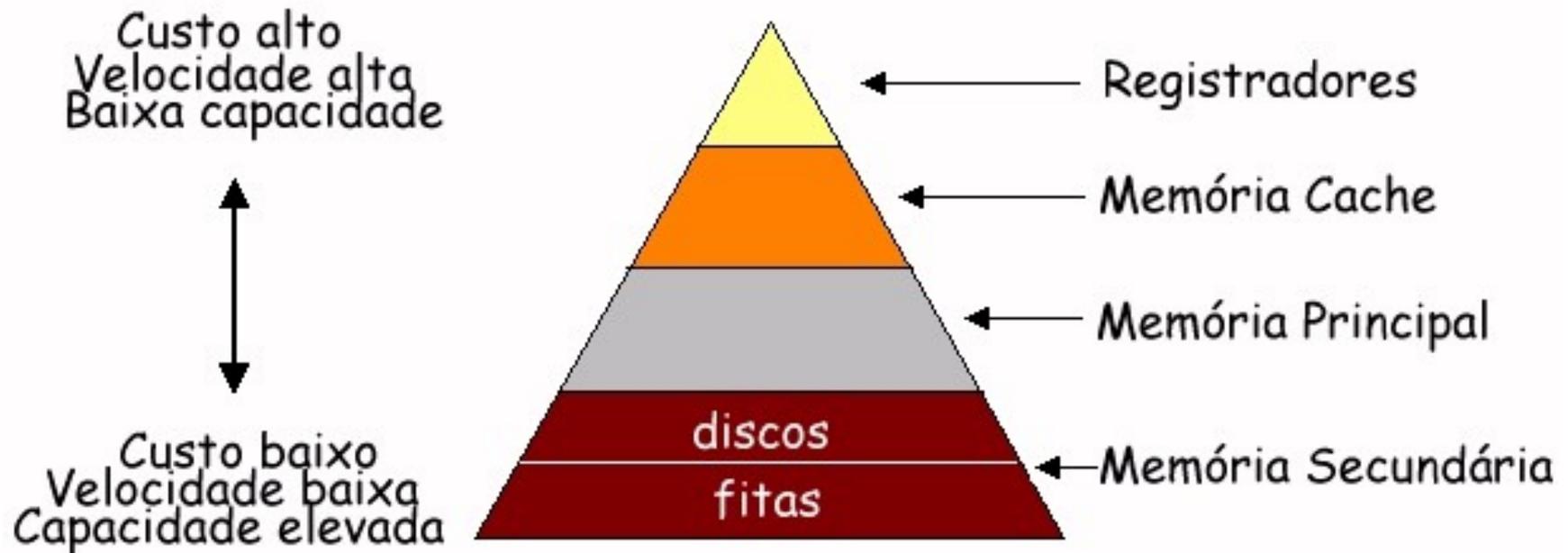
# Hierarquia de memória

- Todos já viram esta pirâmide:



# Hierarquia de memória

- Vocês já viram esta pirâmide:



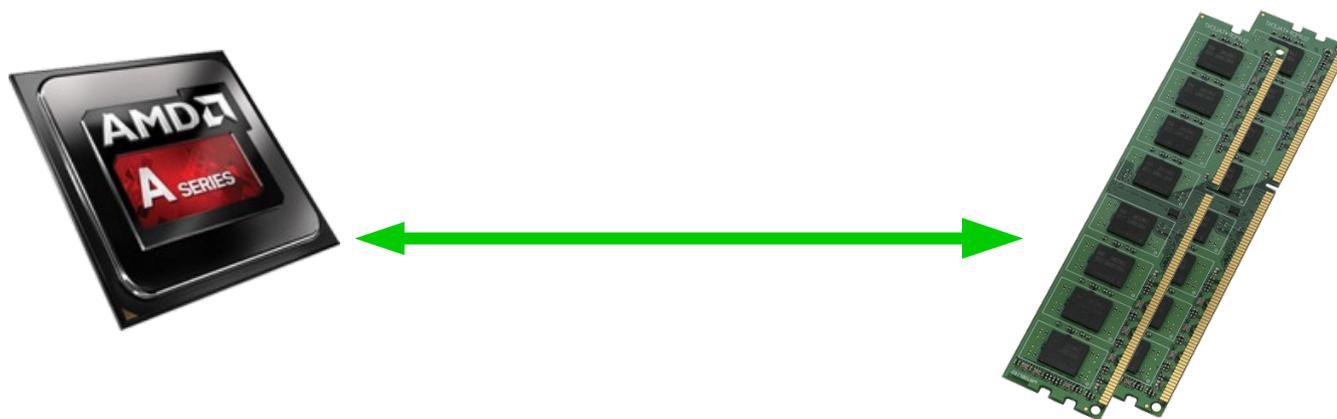
# A função do S.O.

- Nessa pirâmide a principal atividade do S.O. é:
  - Gerenciar memórias de velocidade e tamanhos diferentes
- Memórias cache são tratadas pelo hardware
- Para o nosso estudo o escopo será a memória Primária (RAM) e Secundária (HD/SSD)

# Acesso direto pelo endereço físico

- A estratégia mais simples
- Se um programa precisa acessar uma área de memória basta informar o endereço que deseja acessar

```
MOV R1,0xF34 ;Irá mover o conteúdo  
do endereço 0xF34 para o registrador R1
```



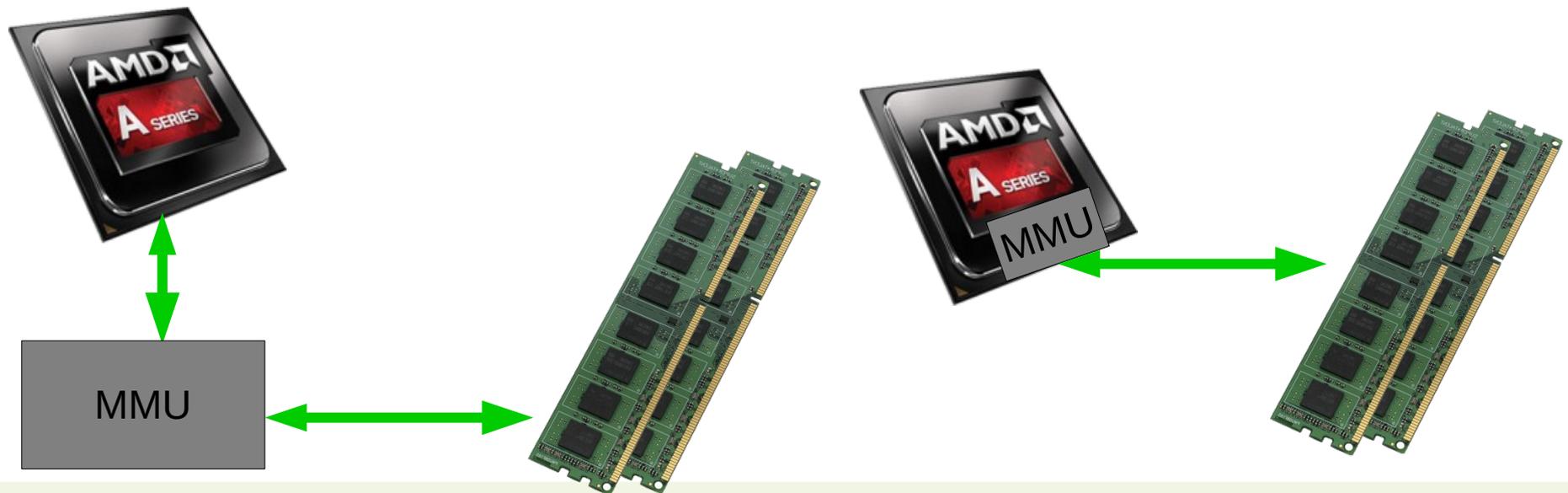
# Funciona para sistemas monotarefa

- Acessar diretamente pode permitir que o processo atual altere a área de memória de outro processo.
- Mesmo para ambientes monotarefa, pode ser inseguro já que o S.O. também não está protegido.
- Como garantir que o programa A não gravará no espaço de B?



# MMU

- Adicionar uma peça de hardware específica para proteger a memória
- Esse hardware comumente conhecido como Memory Management Unit pode ser adicionado ao processador ou pode ser externo a ele.



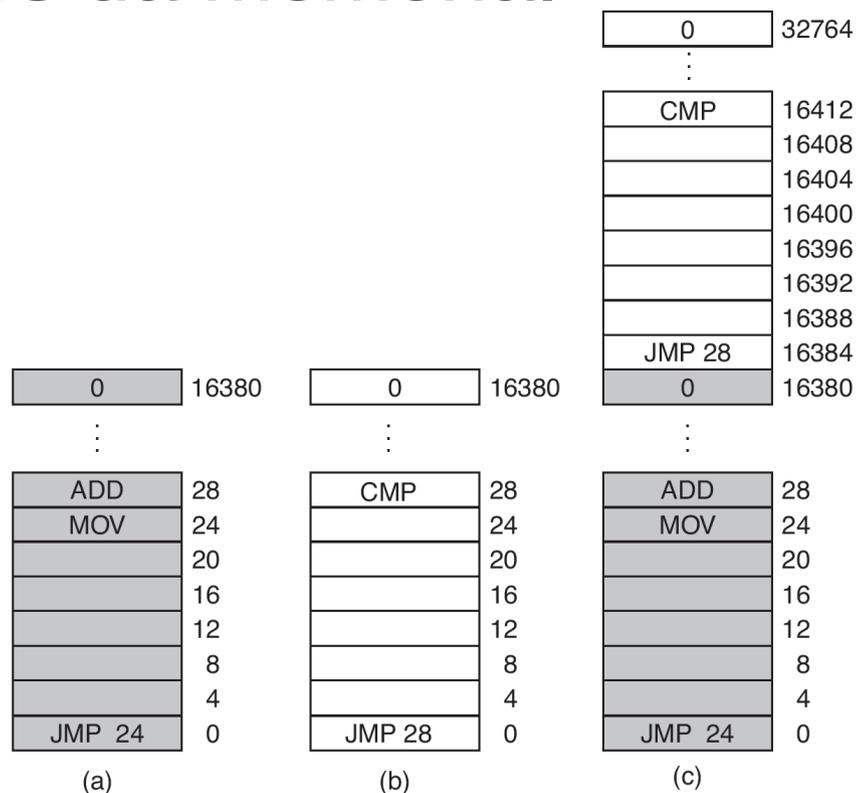
# MMU

- Todos os acessos à memória precisam passar por esse hardware
- Precisa conhecer as área que estão alocadas para cada processo



# Endereços relocáveis

- Para garantir que o processo A não altere dados do processo B, devemos alocá-los em locais diferentes da memória.



# Solution

- Como resolver o problema da relocação de instruções do software?



# Espaços de endereçamento

- **É o conjunto de endereços que um processo pode usar para acessar a memória.**
- Não significa que o processo tem disponível toda a memória do espaço de endereçamento

# Espaço de endereços

- Números de telefone são um exemplo de espaço de endereçamento

345-2233

2345-2233

92345-2233

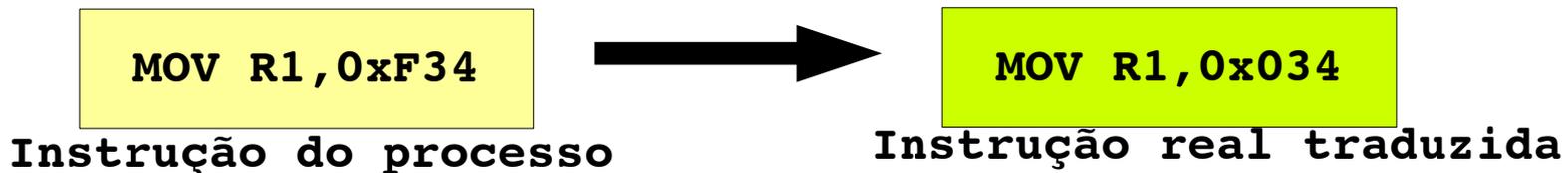
00000-0000.....99999-9999

- Números IP também são

0.0.0.0.....255.255.255.255

# Espaço de endereços

- Os espaços de 2 processos diferentes são separados.
- Isso evitará que um acesse os dados de outro.
- É necessário que algum mecanismo (hardware ou software) traduza o endereço do espaço de endereços para o endereço real (que o controlador de memória entende)



# Estratégias de implementação

- Registradores especiais Base e Limite
- Uma estratégia bastante simples é definir para cada processo:
  - onde começa o espaço de memória alocado para ele.
  - qual o maior endereço que o processo pode acessar
- Sempre que ocorrer uma troca de contexto os registradores Base e Limite são atualizados
- Sempre que o processo tenta acessar memória a CPU ou MMU checa os limites



# Registrador Base e Limite

Processo	Registrador Base	Registrador Limite
S.O.	0	0
A	1	4
B	5	9

Endereço	Processo
9	B
8	B
7	B
6	B
5	B
4	A
3	A
2	A
1	A
0	S.O.

# Atividade

**Endereço para entrega: <https://goo.gl/A2e1Hg>**

- Implementar a gerência de memória baseada em registradores Base e Limite
- Será carregado de um arquivo o espaço de memória que cada processo irá usar(máximo de 5 processos 1 por linha)
- Um vetor de 50 posições de `char` simulará a memória
- Para cada processo exibir o valor do registrador base e limite, se a soma dos processos for maior que 50 parar e emitir mensagem de erro
- Exibir para o usuário uma grid com as 50 posições do vetor
- Após carregado o usuário poderá selecionar um processo e um endereço de memória (do espaço de endereços do processo) e clicar em gravar
- O sistema deve validar se o processo selecionado tem permissão para gravar no endereço informado.



# Atividade

Carregar

Processo A ▾ Reg. Base: 0  
Reg. Limite: 12

Endereço: 0  
Valor: X Gravar

Endereço	Conteúdo
0	X
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
...	