

Cache

→ Localidade:

- ↳ temporal: se o objeto foi referenciado, é provável que seja referenciado posteriormente
- ↳ espacial: se o objeto foi referenciado, é provável que os seus adjacentes sejam referenciados posteriormente.

```
ex: for (int i = 0; i < 5; i++) {  
    A[i] = i;  
}
```

A[i] = i;

* A é um array de 5 posições

Análise:

→ Dados

temporal

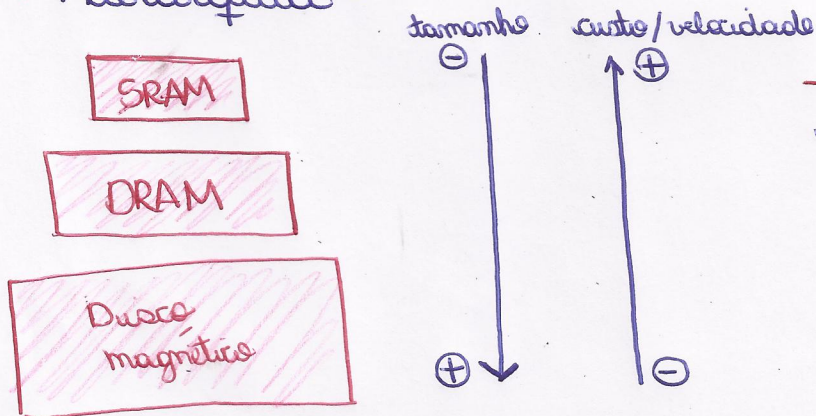
espacial

forte.

→ Instruções

fraca, são poucas instruções no loop (Blocs)

→ Hierarquia:



→ por ser extremamente cara, a cache (SRAM) é pequena e, por ser rápida, é a única a se comunicar com o processador

→ a transferência de dados entre níveis diferentes da memória ocorre apenas entre dois adjacentes

→ Estrutura Básica:

obs: a cache não trabalha com words!

↳ basta lembrar: 1w = 4 bytes!

↳ Bloco: unidade básica

↳ Mux: seleciona os dados conforme

a necessidade do processador

↳ Comparador: determina se hit

↳ se tag procurada = tag bloco → hit (hit = 1).

senão → miss (hit = 0)

→ Conceitos:

↳ hit: o dado procurado está na cache

↳ miss: o dado procurado não está na cache → buscar no nível inferior

↳ tag: um número para verificar se o dado procurado é o dado do bloco.