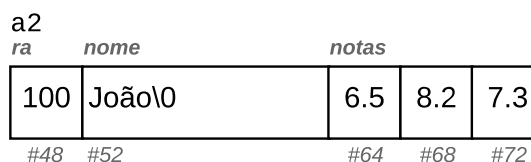
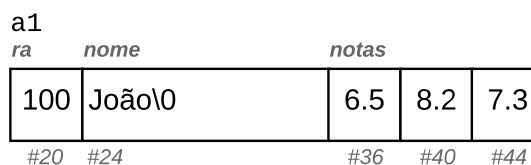


Parte 1

```
typedef struct aluno{
    unsigned int ra;
    char nome[12];
    float notas[3];
} Aluno;
```

```
Aluno a1 = {100, "Joao", {6.5, 8.2, 7.3}};
Aluno a2;
a2 = a1;
```

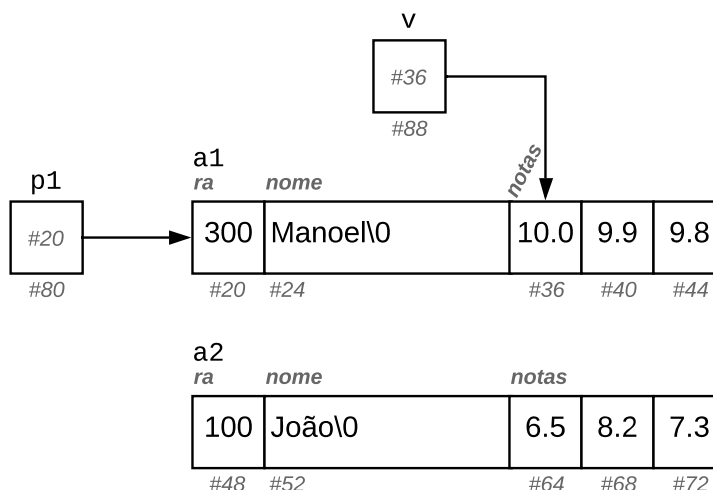


Parte 2

```
Aluno* p1 = &a1;

(*p1).ra = 200;
p1->ra = 300;
strcpy((*p1).nome, "Joaquim");
strcpy(p1->nome, "Manoel");
(*p1).notas[0] = 9.0;
p1->notas[0] = 10.0;

float* v = p1->notas;
v[1] = 9.9;
*(v+2) = 9.8;
```



Formas de Acesso

Acesso por meio de um ponteiro de struct

Acesso por meio da variável declarada como struct



A seta é um açúcar sintático para facilitar o acesso de structs por meio de ponteiros

Ponteiro 1	Ponteiro 2	Variável	Valor	Tipo
p1->ra	(*p1).ra	a1.ra	300	int
p1->nome	(*p1).nome	a1.nome	#24	char*
p1->notas	(*p1).notas	a1.notas	#36	float*
p1->notas[0]	(*p1).notas[0]	a1.notas[0]	10.0	float

Sinônimos

Cada linha representa três formas diferentes de acessar a mesma informação na memória

Exemplo

O padrão que aprendemos na manipulação de vetores e ponteiros se aplica também nas structs. Veja o exemplo abaixo, existem diferentes formas de se acessar a região de memória correspondente a segunda nota do estudante a1.

A partir da obtenção do endereço de memória, a forma de acesso é a mesma: notação de colchetes ou notação de ponteiros.

