



Transcrição de aula

Disciplina	Programação Imperativa - 1º ano - LEI	
Secretário	Número: 36864	Nome: <i>Flávia Beatriz Camp</i>
Data: 2011-04-11 T4	Nº Página	

109 alunos

SUMÁRIO

Introdução às estruturas de dados em memória dinâmica.

Listas ligadas: conceito, modelo, declarações, utilização, listagem, contagem e inserções.

Aula TEÓRICA de PI - 11/04/2011

Listas Ligadas

Alunos. Programa em C

Obj: A função vai buscar à stack, parâmetros, resultados, etc.

Causa o controle da recursividade

()

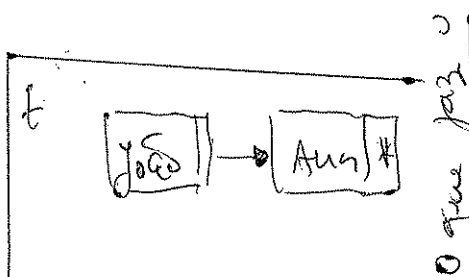
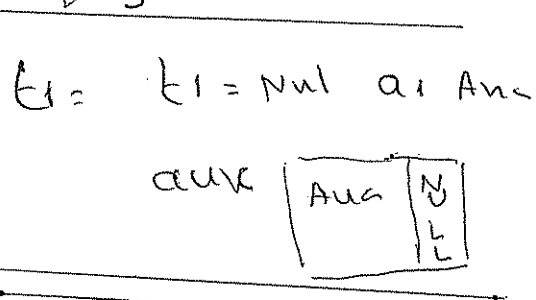
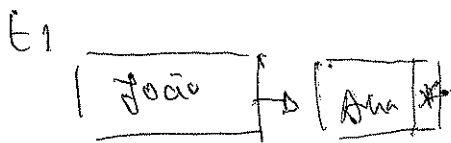
()



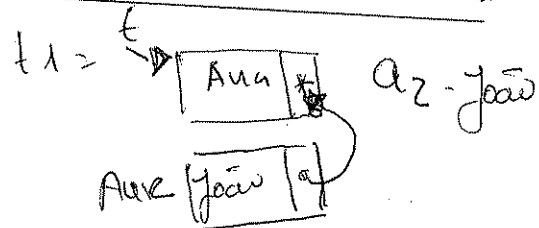
Transcrição (folha de continuação)

```
int sizeOf (<tipo>)
```

```
int main ( )
Aluno a1 = {"4140", "Aur"}
Aluno a2 = {"5355", "João"}
Turma t1 = Null, t2 = Null;
t1 = PushTurma (t1, a1);
t1 = PushTurma (t1, a2);
t2 = PushTurma (PushTurma (t2, a2), a1);
listaTurmas (t1);
} listaTurmas (t2);
}
```

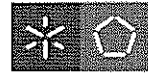


o que faz

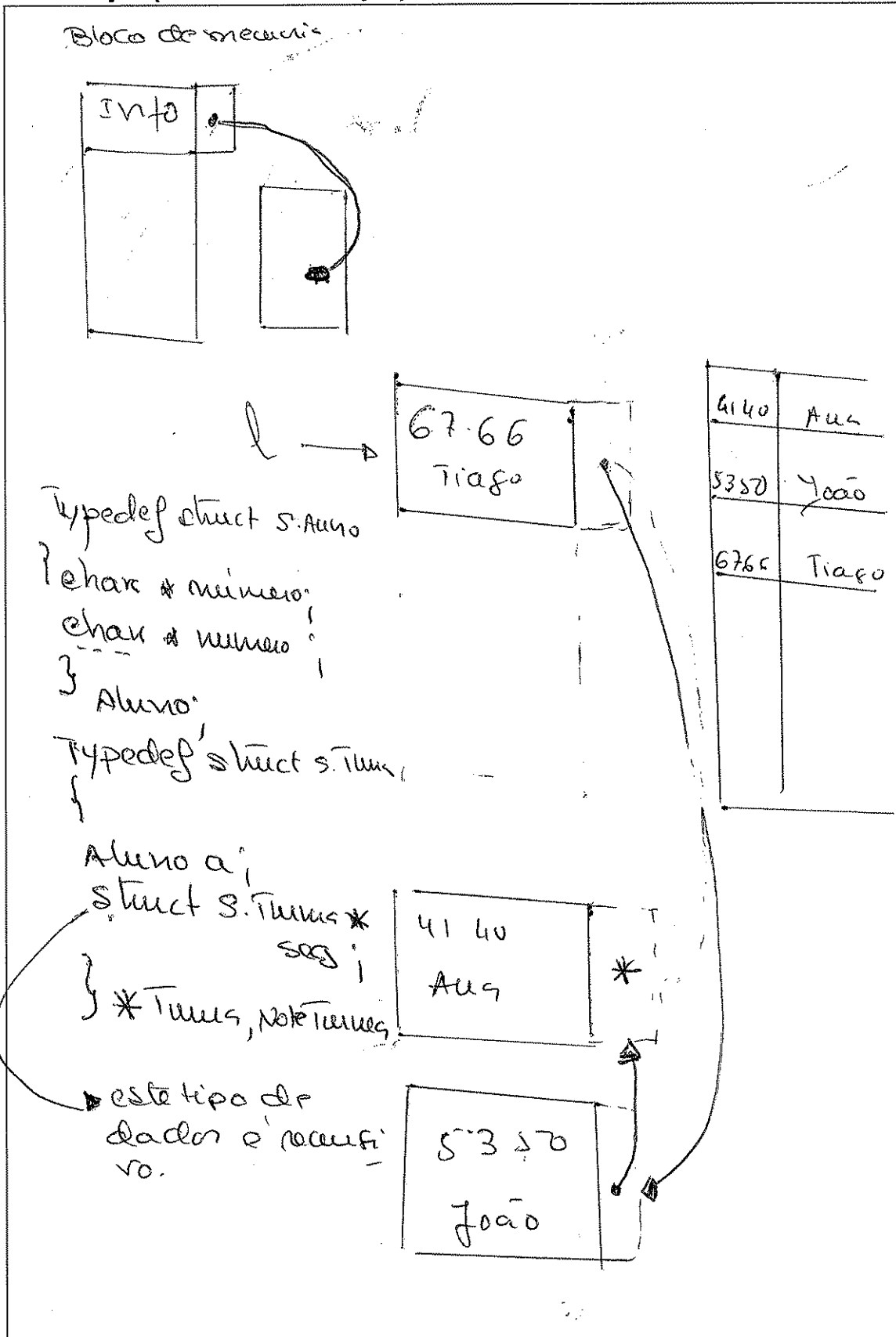


t -> Null

7. a lista LL: no momento Null quer dizer



Transcrição (folha de continuação)



int *a → ~~a~~ apontador a apontar para 1 inteiro
 *a = 5;

Turma t1;

*t1.a ... | t1 → a
 *t1.seg | t1 → seg

Função de listagem
 void listaTurmas (Turma t)

```

{
  if (t)
  {
    listaAluno (t → a);
    listaTurma (t → seg);
  }
}
  
```

Tipo de função

Turma listaTurmas (Turma t, Aluno a)

{
 *Turma aux
 *contínua

Estamos a trabalhar em memória dinâmica
 portanto vamos utilizar a função

Obs. quando a lista está vazia, qd. tratarmos com apontadores talvez devamos inicializar - 10 para os apontadores a inicializar cada e NULL

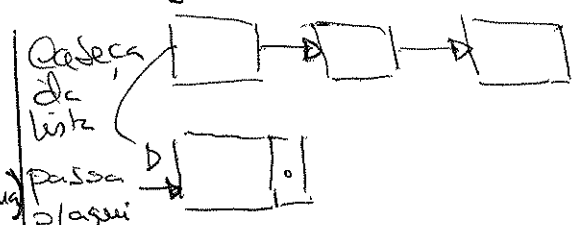
Quando o apontador apontar para NULL - lista vazia

Obs: lista Aluno recebe t de (no
 Obs: seg → seguinte

void * malloc (int n bytes)
 (tipo)

↓
 *contínua
 aux = (Turma)
 aux → a = a;
 aux → seg = t;

malloc (sizeof (Turma))



Obs: void * - apontador tipo do
 valores utilizar a função cast - só funciona em tipos compatíveis
 ex: float para um int. pode-se inferir cada a parte decimal despois