



# Transcrição de aula

Disciplina	Programação Imperativa – 1º ano - LEI	
Secretário	Número: 53988	Nome: Tija Sousa
Data: 2011-03-15	Nº Página 3	

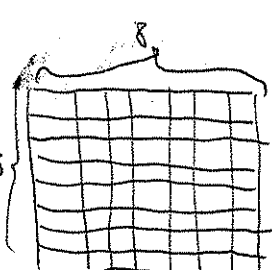
## SUMÁRIO

Introdução ao produto cartesiano: structs.  
 Definição de tipos abstractos de dados em C.  
 O modelo de dados struct.

Arrays:  
 multidimensionais

int tabuleiro [8] [8];

(pods. são omitida uma das dimensões)



# de linhas max linha 8  
 # de colunas max coluna 8

int tabuleiro [max linha] [max coluna];

int tabuleiro [max linha] [max col] = { { 1, 0, 2, 0, 3, 0, 4, 0, 5, 0, 6, 0, 7, 0 } }

Produto Cartesiano:

```

struct
{
    int numerador;
    int denominador;
}
int main()
{
    struct sfrac {
        int numerador;
        int denominador;
    };
}
  
```



## Transcrição (folha de continuação)

```
typedef struct >Proc
{
    int numerador;
    int denominador;
} Proc;

int main()
{
    Proc p1, p2 = (2, 3);
    printf("%d / %d", p1.numerador,
           p2.denominador);
}
```

```
Proc somaProc (Proc p, Proc q)
{
    int num = q.numerador * p.denominador + p.numerador * q.denominador / q.denominador * p.denominador;
    return num;
}
```

```
typedef struct >Frac
{
    int num;
    int denom;
} Frac, ListaFrac [5];
```

```
int main()
{
    Frac f1, f2, f3 = {1, 3}, f4 = {2, 5}, f5 = {6, 7}, f6;
    ListaFrac lFrac = {f3, f4, f5, f6, f1 = {4, 5}}
}
```

**Transcrição (folha de continuação)**

Definir um aluno e uma turma

```
typedef struct tAluno
{
    char numero[7];
    char nome[60];
    float notas[10];
} Aluno;

typedef struct tTurma
{
    Aluno Alunos[100];
    int nAlunos;
} Turma;
```

```
int main()
{
    Aluno a1 = {"4238", "Alberto", {0.0}};
    Turma t1 = {{a1, {"456", "Paulo", {0.0}}}, 2};
```

```
int ListaNome(Turma t)
```

```
{
    int i = 0;
    while (i < t.nAlunos)
    { printf("%s\n", t.Alunos[i].nome); }
    i++;
    return t.nAlunos;
}
```