



**Engenharia de Software a Distância**

Carlos Francisco Silva Junior

**Estrutura de dados II**

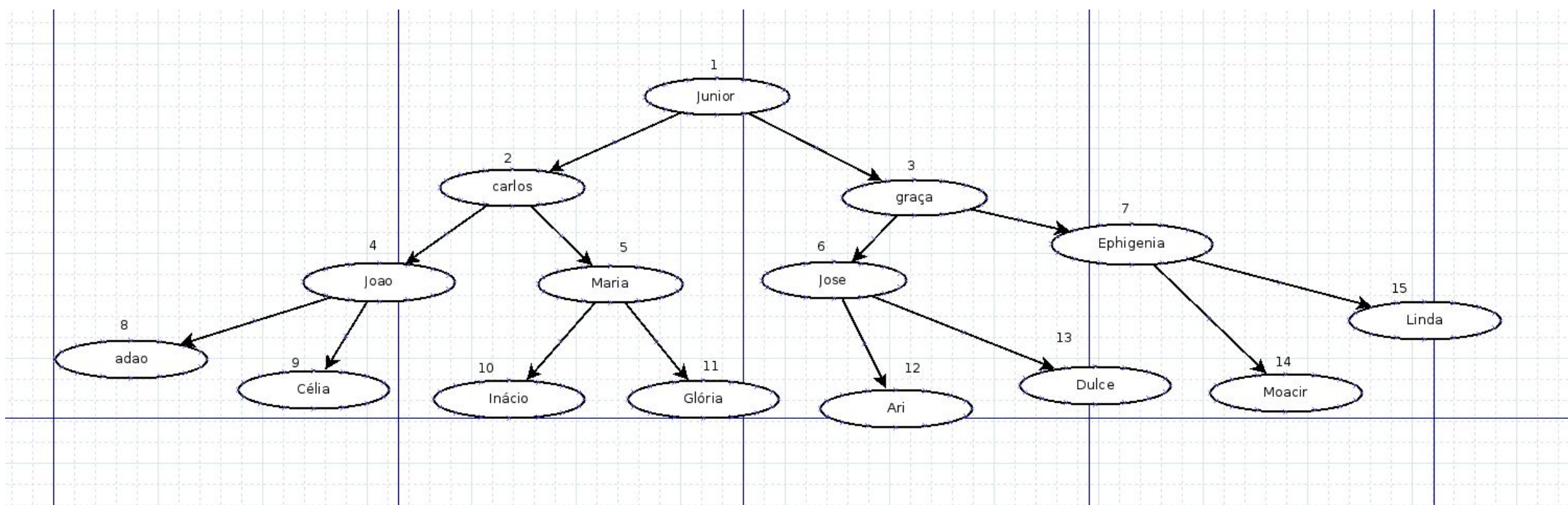
Maringá  
Junho de 2017

### Proposta do Mapa

Construa uma árvore binária genealógica em C. O nó raiz será você, os filhos desse nó serão seus pais e, por conseguinte, avós e bisavós (15 nós no total). Os rótulos dos nós serão o teu nome e os respectivos nomes dos parentes, podendo atribuir um nome fictício no caso de desconhecer o nome de alguém ou se alguém não tiver pai ou mãe.

Faça um programa que leia os 15 nomes e depois faça a construção da árvore genealógica. Depois de construída, faça a impressão dos nomes na tela.

### Desenho árvore genealógica.



Print do software em execução.

```
carlos.junior@CpdAdmin:~/C$ ./mapa
Informe o numero de pessoas com pais e mães
7
Informe os dados dos elementos:
Digite o nome do filho:
junior
informe o nome do pai:
carlos
informe o nome da mae:
graca
Digite informe os dados do proximo filho 2:
Digite o nome do filho:
carlos
Digite o nome do pai:
joao
Digite o nome da mae:
maria
Digite informe os dados do proximo filho 3:
Digite o nome do filho:
graca
Digite o nome do pai:
jose
Digite o nome da mae:
ephigenia
Digite informe os dados do proximo filho 4:
Digite o nome do filho:
joao
Digite o nome do pai:
adao
Digite o nome da mae:
celia
```

```
Digite informe os dados do proximo filho 5:
Digite o nome do filho:
maria
Digite o nome do pai:
inacio
Digite o nome da mae:
gloria
Digite informe os dados do proximo filho 6:
Digite o nome do filho:
jose
Digite o nome do pai:
ari
Digite o nome da mae:
dulce
Digite informe os dados do proximo filho 7:
Digite o nome do filho:
ephigenia
Digite o nome do pai:
moacir
Digite o nome da mae:
Linda

Selecione:
1) Mostrar geracao
2) Sair do programa

```

Print do resultado.

```
Selezione:
1) Mostrar geracao
2) Sair do programa
1

GenArvore genealogica e suas gerações:
junior
carlos graca
joao maria jose ephigenia
adao celia inacio gloria ari dulce moacir Linda
```

Código fonte:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct GenArvore{
    char nome[50];
    int relacao;
    struct GenArvore* lado1;
    struct GenArvore* lado2;
};

struct GenArvore* buscar(struct GenArvore* humano, char *filho){

    struct GenArvore *addr = NULL;
    if(humano!=NULL){
        if(strcmp(humano->nome, filho)==0){
            addr=humano;
        }
        if(addr==NULL){
            addr= buscar(humano->lado2, filho);
        }
    }
}
```

```

        }
        if(addr==NULL){
            addr= buscar(humano->lado1, filho);
        }
    }
    return addr;
}

```

```

void Cadastrar(struct GenArvore*humano, char *filho, char *pai, char *mae){
    struct GenArvore *aux=humano;
    struct GenArvore *atual=NULL;
    int incr;
    atual=buscar(aux, filho);
    atual->lado2=(struct GenArvore*)malloc (sizeof(struct GenArvore));
    atual->lado1=(struct GenArvore*)malloc (sizeof(struct GenArvore));
    incr=atual->relacao;
    if(atual!=NULL){
        strcpy(atual->lado2->nome, pai);
        atual->lado2->relacao=incr+1;
        atual->lado2->lado2=NULL;
        atual->lado2->lado1=NULL;

        strcpy(atual->lado1->nome, mae);
        atual->lado1->relacao=incr+1;
        atual->lado1->lado2=NULL;
        atual->lado1->lado1=NULL;
    }
}

```

```

void showGeneration(struct GenArvore* humano, int heght){
    if(humano!=NULL){
        if(humano->relacao==heght){
            printf("%s ", humano->nome);
        }
        showGeneration(humano->lado2, heght);
        showGeneration(humano->lado1, heght);
    }
}

```

```

int Menu(){

```

```

    int opt;
    printf("\nSelecione:\n");
    printf("1) Mostrar geracao\n");
    printf("2) Sair do programa\n");
    scanf("%d", &opt);
    return (opt);
}

```

```

void main(){
    int nMagic, i=2, j, maior=0;
    char filho[50], pai[50], mae[50];
    char nome1[50], nome2[50];
    struct GenArvore *pessoa0=(struct GenArvore*)malloc (sizeof(struct GenArvore));
    struct GenArvore *pessoa1=(struct GenArvore*)malloc (sizeof(struct GenArvore));

    printf("Informe o numero de pessoas com pais e mães\n");
    scanf("%d", &nMagic);
    while(nMagic<1){
        printf("Informe um valor maior ou igual a 1\n");
        scanf("%d", &nMagic);
    }
    printf("Informe os dados dos elementos:\n");
    printf("Digite o nome do filho:\n");
    struct GenArvore* humanoraiz= (struct GenArvore*)malloc (sizeof(struct GenArvore));
    humanoraiz->lado1=(struct GenArvore*)malloc (sizeof(struct GenArvore));
    humanoraiz->lado2=(struct GenArvore*)malloc (sizeof(struct GenArvore));
    scanf("%s", &filho);
    printf("informe o nome do pai:\n");
    scanf("%s", &pai);
    printf("informe o nome da mae:\n");
    scanf("%s", &mae);
    strcpy(humanoraiz->nome, filho);
    humanoraiz->relacao=0;
    humanoraiz->lado1=humanoraiz->lado2=NULL;
    Cadastrar(humanoraiz, filho, pai, mae);
    if(nMagic>1){
        while(i<=nMagic){
            printf("Digite informe os dados do proximo filho %d:\n", i);
            printf("Digite o nome do filho:\n");

```

```

        scanf("%s", &filho);
        printf("Digite o nome do pai:\n");
        scanf("%s", &pai);
        printf("Digite o nome da mae:\n");
        scanf("%s", &mae);
        Cadastrar(humanoraiz, filho, pai, mae);
        i++;
    }
}

    int aux, reg=1;
while(reg==1){
    switch(Menu()){
        case 1:
            printf("\nGenArvore genealogica e suas gerações:\n");
            for(j=0;j<=nMagic;j++){
                showGeneration(humanoraiz, j);
                printf("\n");
            }
            break;
        case 2:
            exit(0);
        default:
            printf("\nValor incorreto\n");
    }
}
}

```