

ÁRVORES

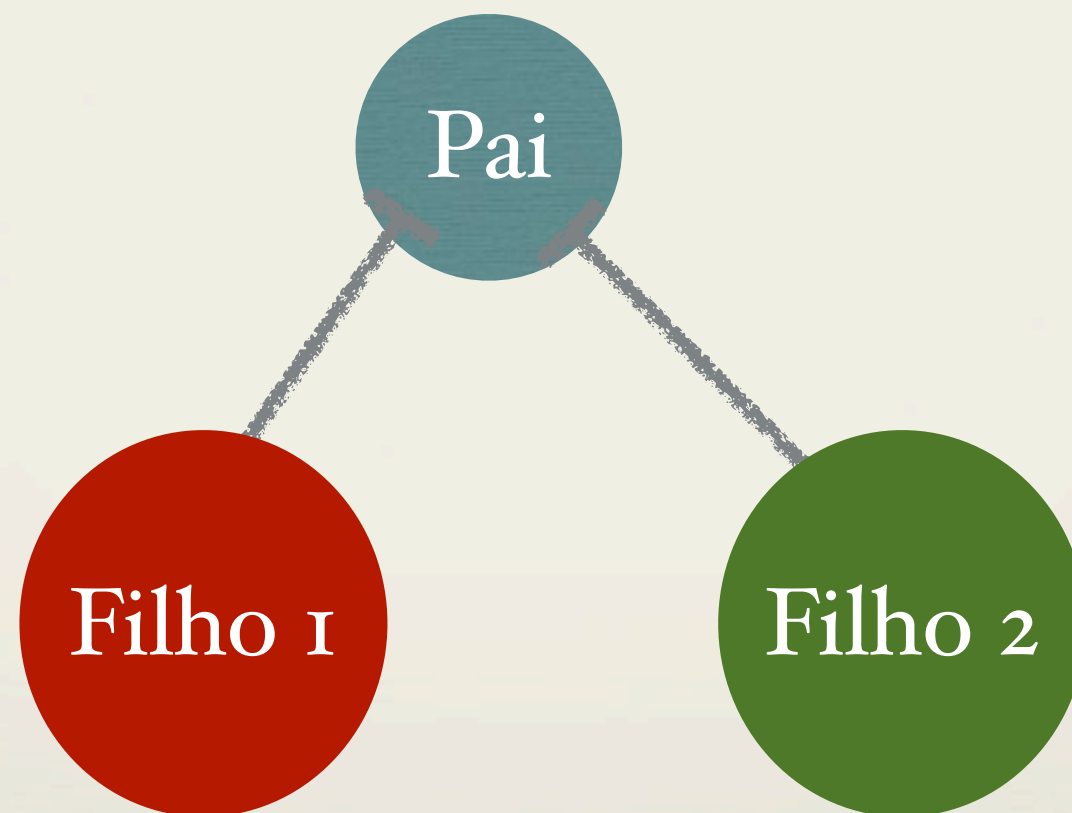
if63c - Estrutura de Dados
prof. Robinson Vida Noronha

Pilhas, Filas e Listas ?

- * Elementos poderosos para representar conhecimento e para realizar buscas de informação.
- * Porém, são estruturas limitadas a uma dimensão.

Árvores

* Para contornar essa limitação, utiliza-se Árvores.



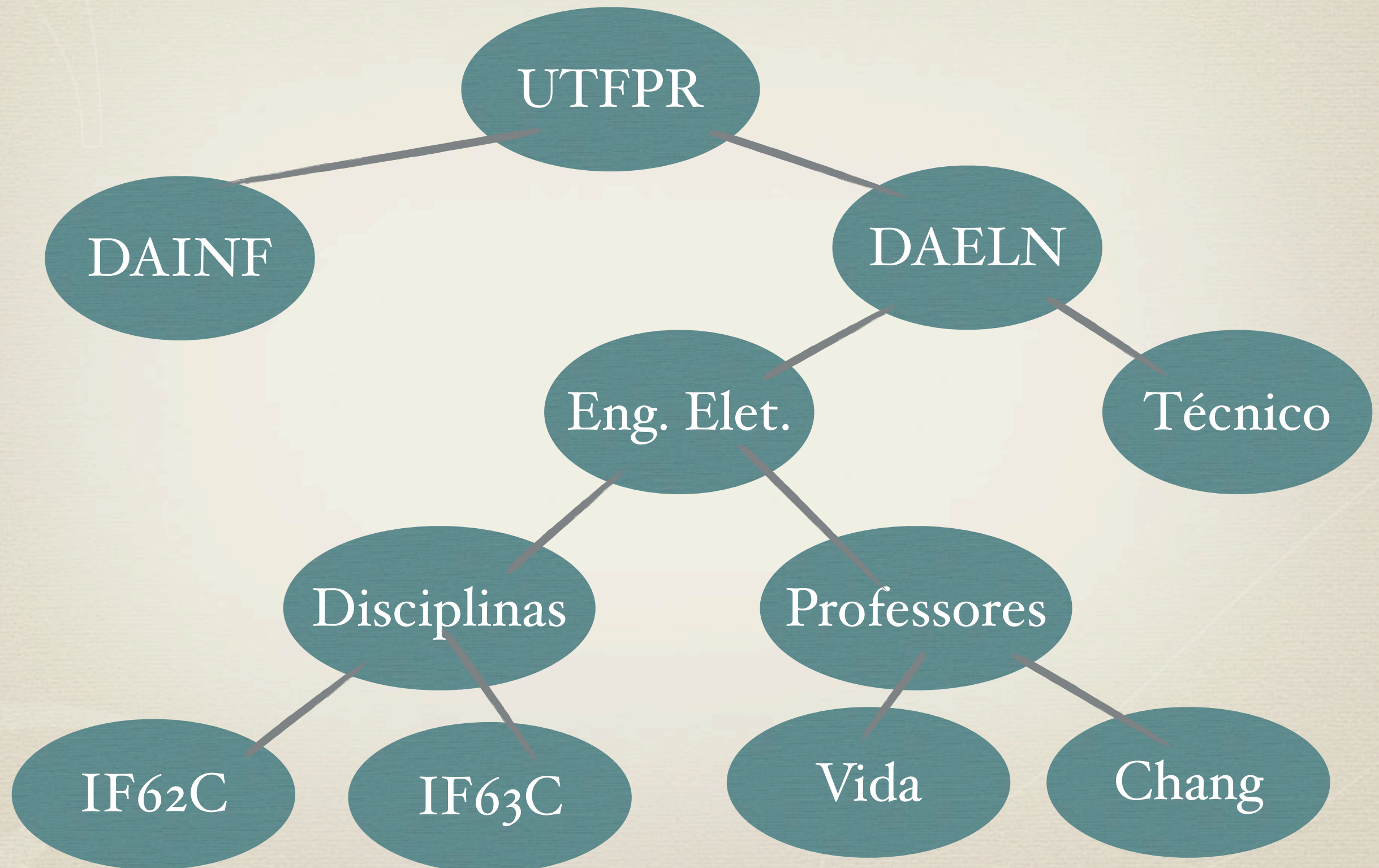
Exemplo de Árvore



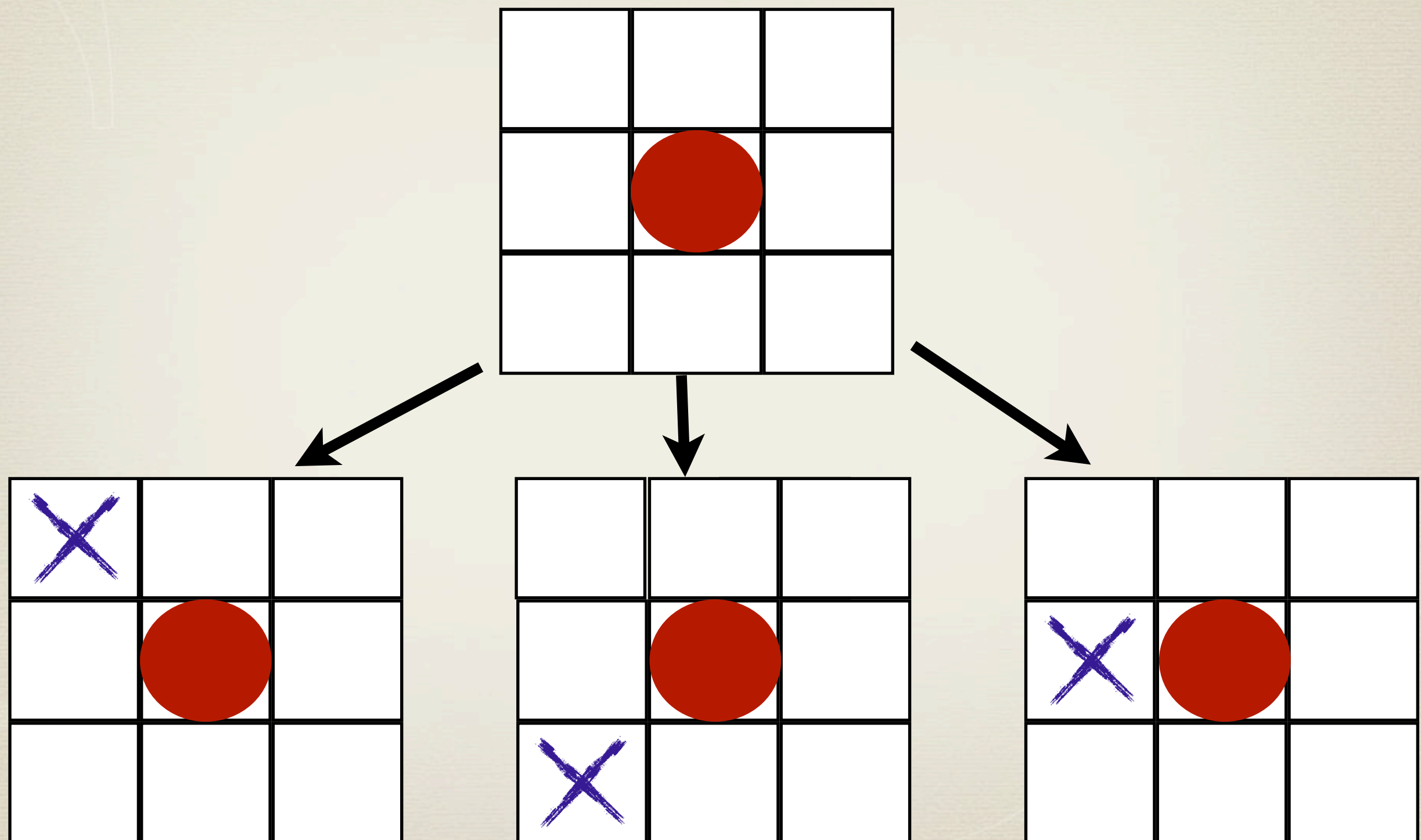
Exemplo de Árvore



Exemplo de Árvore



Exemplo - Jogos



Implementação de Árvores

- * como matrizes
- * estruturas ligadas

Implementação : Matrizes

índice	info	esquerda	direita
0	UTFPR	1	2
1	DAINF	null	null
2	DAELN	3	5
3	Eng. Elet.	4	6
4	Disciplinas	...	...
5	Técnico	null	null
6	Professores	...	

Estrutura

```
template <class T>
class Node
{
    public:
        T info;
        Node *right;
        Node *left;
        Node(){ info=0; left = 0; right=0;}
        Node(const T *v) { info = v; left=0; right = 0;}
        Node(const T &v) { info = v; left = 0; right = 0;}

        void insertLeft( Node *v) { left = v;}
        void insertRight( Node * v) { right = v;}
        void insertLeft( Node v) { left = &v;}
        void insertRight( Node  v) { right = &v;}

        Node * getRight(){return right;}
        Node * getLeft(){return left;}

        T getInfo(){ return info; }

};
```


Busca Cega na Árvore

```
template <class T>
T * search(Node <T> *p, T dado)
{
    T* saida = 0;
    if ( p == 0 ) return 0;
    else
    {
        if ( dado == p->info ) return &p->info;
        saida = search(p->right, dado);
    }
    if ( saida == 0)
        {saida = search( p->left, dado);}

    return saida;
};
```


Exemplo de Uso

```
#include "Tree.h"
#include <string>
#include <iostream>

using namespace std;

int main()
{
    Node<int> n1(10);
    Node<int> n2(20);
    Node<int> n3(30);
    Node<int> n4(40);
    Node<int> n5(50);

    n1.insertLeft(&n2);
    n1.insertRight(&n3);

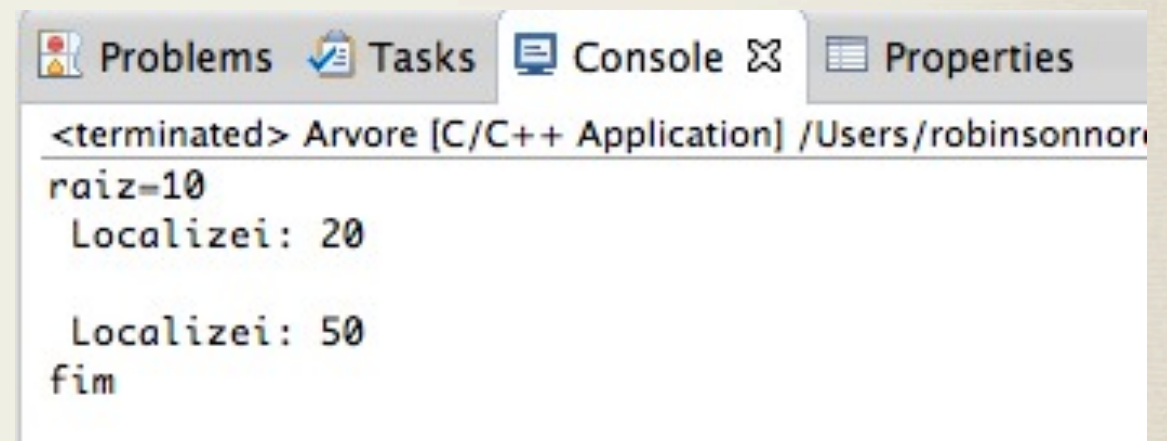
    n2.insertLeft(&n4);
    n3.insertLeft(&n5);

    std::cout << "raiz=" << n1.getInfo();

    std::cout << "\n Localizei: ";
    std::cout << *search(&n1, 20) << std::endl;

    std::cout << "\n Localizei: ";
    std::cout << *search(&n1, 50) << std::endl;

    std::cout << "fim";
}
```



The screenshot shows a console window with the following output:

```
<terminated> Arvore [C/C++ Application] /Users/robinsonnor...
raiz=10
  Localizei: 20

  Localizei: 50
fim
```