



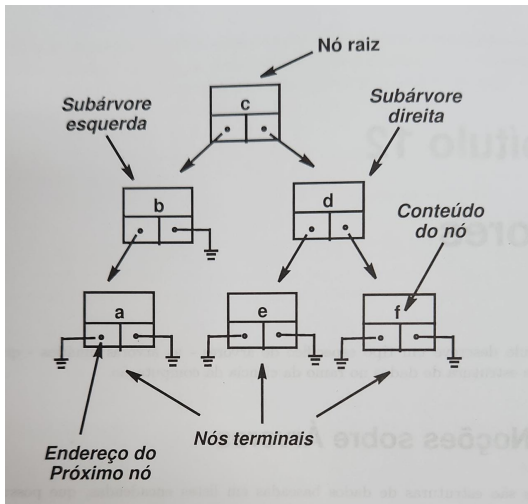
PROGRAMAÇÃO FUNCIONAL: Árvores em Haskell

Professor Rafael Kingeski

Departamento de Ciência da Computação
Centro de Ciências Tecnológicas - CCT
UDESC - Joinville.

- As árvores são estruturas de dados baseadas em listas encadeadas, que possuem um nó superior, comumente definido como raiz ou nó *pai*. Este por sua vez, aponta para outros nós denominados como *filhos*.
- Em Haskell, podemos representar árvores usando tipos de dados algébricos.
- Vamos explorar a criação e manipulação de árvores em Haskell.

Estrutura de uma árvore binária





```
data ArvoreBin_Int = Nil_Int | Nodo_Int Int
  ArvoreBin_Int ArvoreBin_Int deriving Show
```

- A árvore binária é definida por um tipo de dado algébrico.
- Possui dois construtores: `Nil` (representa uma árvore vazia) e `Nodo` (representa um nó com um valor e dois filhos).
- O tipo do valor armazenado na árvore é para este exemplo é inteiro e é representado por `Int`.

arvoreB =

```
Nodo_Int 4 (Nodo_Int 2 Nil_Int Nil_Int)
            (Nodo_Int 10 (Nodo_Int 5 Nil_Int Nil_Int)
                (Nodo_Int 15 Nil_Int Nil_Int))
```

- Podemos criar árvores binárias atribuindo valores a variáveis.
- Os nós e folhas são criados usando o construtor `Nodo` e `Nil`, respectivamente.
- Os valores e subárvores são especificados como argumentos do construtor.

-  C. C. de Sá; M. F. da Silva
Haskell: Uma Abordagem Prática
(Novatec, 2006)
-  Lipovaca, M.
Learn You a Haskell for Great Good!
(No Starch Press, 2011)