

# Uma Introdução à PICAT

Miguel Alfredo Nunes  
`miguel.nunes@edu.udesc.br`

# Sumário

- O que é PICAT.
- Quais são seus usos “no mundo real”.
- Como será usada por vocês.
- Sintaxe Básica.
- Variáveis e Tipos.
- Predicados e *index*.
- Exemplos de Código.

# O que é

- Uma linguagem de programação Multiparadigma, criada por Neng-Fa Zhou e Jonathan Fruhman em 2013.
- A lógica matemática é base de toda linguagem.
- Foi desenvolvida para ser uma linguagem de uso geral, que incorpora características de linguagens de paradigmas lógico, funcional, e scripting.
- Pretende expandir e melhorar o que Prolog faz(ia).

# O que é ser Multiparadigma

- Uma linguagem de programação Multiparadigma é uma linguagem que possui as características de diversos paradigmas de programação ao mesmo tempo.
- PICAT é uma linguagem de paradigmas lógico, declarativo e funcional.

# Usos Práticos

- Resolução de Problemas de:
  - Busca Exaustiva
  - Pesquisa Combinatória
  - Planejamento
  - Satisfação de Restrições.

# Como Usarão

- Trabalho Final da Matéria.
  - Pesquisar em um domínio pré-definido um conjunto de valores que satisfaz uma série de condições.
  - Ou seja, irão associar uma série de informações que serão dadas e terão que montar um algoritmo que deduza uma resposta.

# Sintaxe Básica

- Conjunção : ,
- Disjunção : ;
- Negação : **not**
- Implicação, Bi Implicação, Disjunção Exclusiva : Não tem
- Linhas em um programa são como proposições lógicas.
- Portanto são separados por conectivos lógicos.

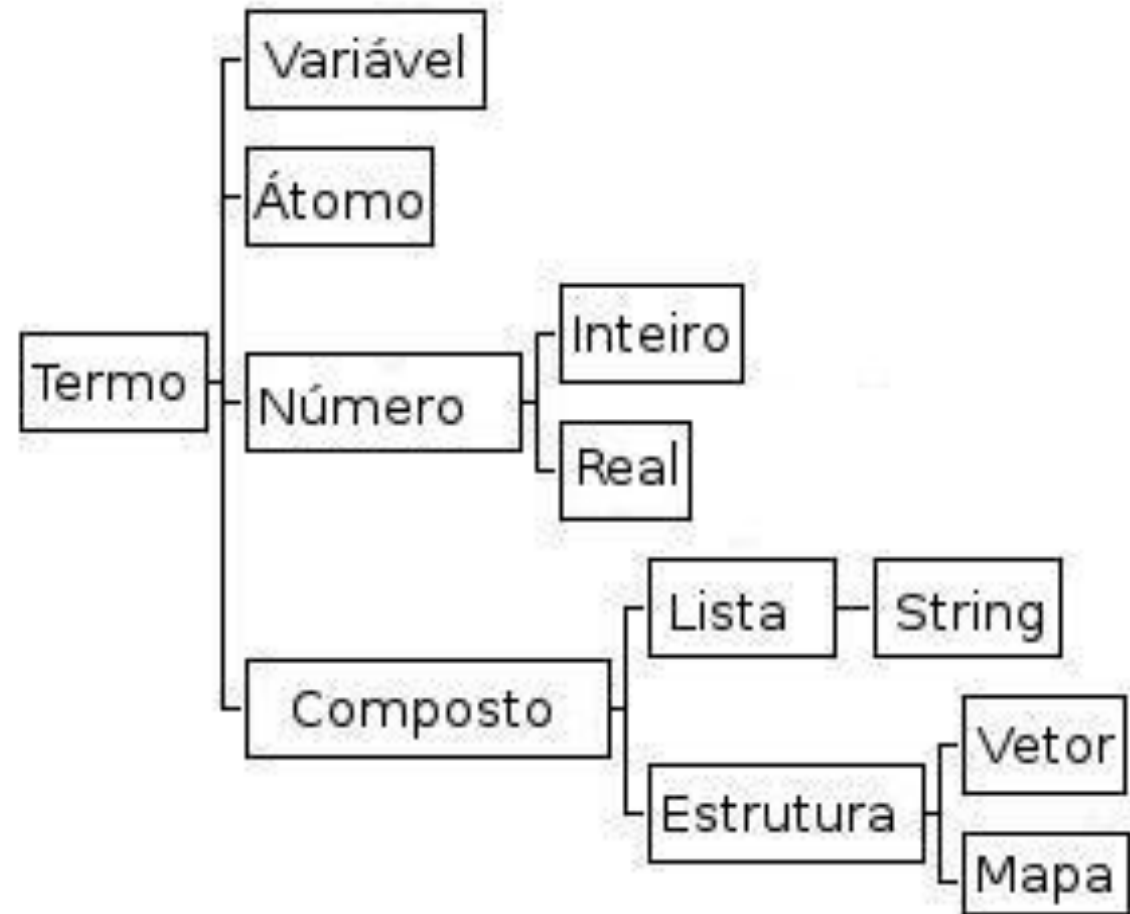
# Variáveis e Tipos

- Variáveis são termos “seguradores de valores”, que possuem dois estados, Instanciados ou Não Instanciados.
- Variáveis sempre são declaradas com a primeira letra do seu nome sendo maiúscula.
- O tipo de uma variável é definido quando ela recebe um valor.
- Toda variável deve ser atribuída um valor quando é inicializada.



# Variáveis e Tipos

- Hierarquia de Dados.
- Variáveis podem ser receber qualquer Termo.
- Átomos são termos especiais, são constantes que representam caracteres únicos ou strings imutáveis.



# Variáveis e Tipos

- Particularização:
  - Definido pelo Símbolo  $=$  .
  - Pode ser feita somente uma vez por variável.
  - Quando uma variável é particularizada ela terá a mesma identidade e comportamento do valor a qual ela foi particularizada.
  - Pode ser revertido com backtracking.
- Atribuição:
  - Definido pelo Símbolo  $:=$  .
  - Equivalente a atribuição de outras linguagens.
  - Pode ser revertido com backtracking.

# Variáveis e Tipos

- Exemplos de Declaração:
  - Nome = “Miguel”
  - A = 22
  - B = 8320
  - C = 3.1415
  - Lista = [1, 2, 3, 4]
  - String = “Isso e uma string”
  - String2 = [o, u, t, r, a, “ ”, s, t, r, i, n, g]
  - X
  - abc = “alguma coisa”

# Variáveis e Tipos

- Exemplos de Atribuição e Particularização:
  - $X = \text{“Isso foi uma particularização”}$
  - $X := \text{“Isso foi uma atribuição, posso repetir isso quantas vezes quiser”}$
  - $X = \text{“Não posso particularizar uma variável múltiplas vezes”}$
  - $\text{Numero} = 14$
  - $\text{Numero} := 42$

# Predicados e *index*

- Predicados são regras que definem relações entre os termos do domínio sobre os quais eles foram definidos.
- O domínio de um predicado é o domínio de seus argumentos.
- Um predicado é do formato  $\text{predicado}(T1, \dots, Tn), C \Rightarrow Cp$ .
  - “predicado” é o nome do predicado.
  - $T1$  até  $Tn$  são seus argumentos, onde  $n$  é a aridade.
  - $C$  é uma condição para a execução do predicado.
  - $Cp$  é o corpo do predicado, onde é escrito o código.

# Predicados e *index*

- Predicados são capazes de fazer backtracking, ou seja, de voltar atrás caso falhem durante sua execução.
- Para que um predicado possa fazer backtracking, ele deve ser do formato  $\text{predicado}(T1, \dots, Tn), C \text{ ?}=\> C_p$ .
- A única diferença de um predicado normal é o operador “ $\text{?}=\>$ ”
- Predicados podem ser recursivos.

# Predicados e *index*

- O predicado *index* é um predicado especial que permite a definição de domínios para seus programas.
- Domínios são definidos criando predicados fatos.
- É do formato `index(-)` então é seguido por fatos do seu domínio, todos terminados com “.”
- O símbolo – indica que uma variável vai receber um valor do domínio que foi definido.

# Predicados e *index*

- Exemplos:

```
index(-)  
professor(claudio)  
professor(karina)
```

```
main() =>  
    professor(X),  
    println(X)
```

•



# Exemplos

- Implicação.
- Bi Implicação.
- Xor ou Disjunção Exclusiva.
- Modus Ponens.
- Modus Tolens.
- Silogismo Hipotético.
- Silogismo Disjuntivo.
- Definição de *pai* e *avo* a partir de um domínio.