



Arbor Language

Uma linguagem de programação inovadora, inspirada na estrutura e crescimento das árvores. Perfeita para quem busca clareza e uma compreensão visual do código.

 por Thiago Victoriano

1. Motivação

Por que Arbor?

- Sintaxe intuitiva e clara.
- Visualiza a estrutura do código.
- Ideal para compiladores.

Filosofia

- "Código que cresce como uma árvore".
- Estrutura hierárquica natural.
- Foco na clareza e simplicidade.

2. Características

Palavras-chave Naturais

- **seed:** Declaração de variáveis.
- **branch:** Estruturas condicionais.
- **grow:** Loops de repetição.
- **print:** Saída de dados.

Tipagem e Estruturas

- Tipagem dinâmica flexível.
- Suporte a tipos básicos: int, string, list.
- Blocos de código por chaves.
- Operadores aritméticos e lógicos.

Estrutura em Blocos

```
branch x > 5 then { print "Maior que 5" } else { print  
"Menor ou igual a 5" }
```


3. Curiosidades



AST Visível

A linguagem mostra a Árvore Sintática Abstrata, auxiliando no entendimento da estrutura do código e servindo como ferramenta educacional valiosa.



Pré-processador

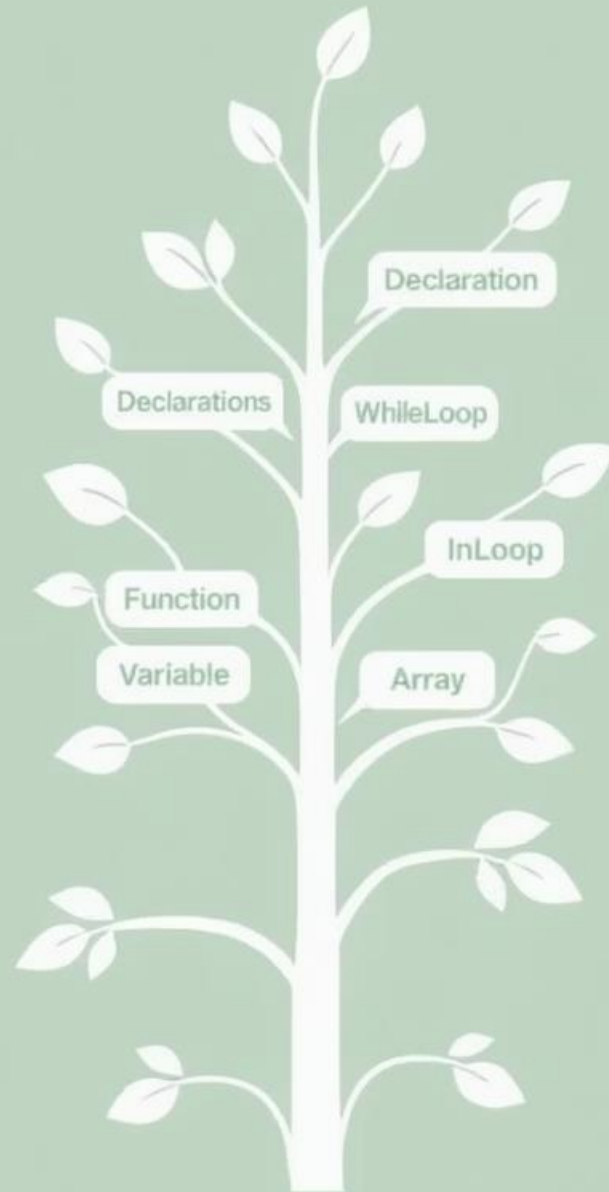
Remove comentários e normaliza espaços, mantendo o código limpo e consistente para uma leitura fácil e organizada.



Tabela de Símbolos

Gerencia o escopo hierárquico, suportando variáveis locais e globais, e verificando a existência de variáveis durante a compilação.





4. Exemplos

Exemplo Básico

1

```
seed x = 10  
print x
```

Declara e imprime uma variável simples.

Loops e Listas

2

```
seed lista = [1, 2, 3, 4, 5]  
grow i in lista {  
  print i  
}
```

Itera sobre uma lista e imprime cada elemento.

Estruturas de Controle Aninhadas

3

```
seed x = 1  
grow while x < 5 {  
  branch x == 3 then {  
    print "Encontrei o 3!"  
  }  
  x = x + 1  
}
```

Demonstra a combinação de loops e condicionais.

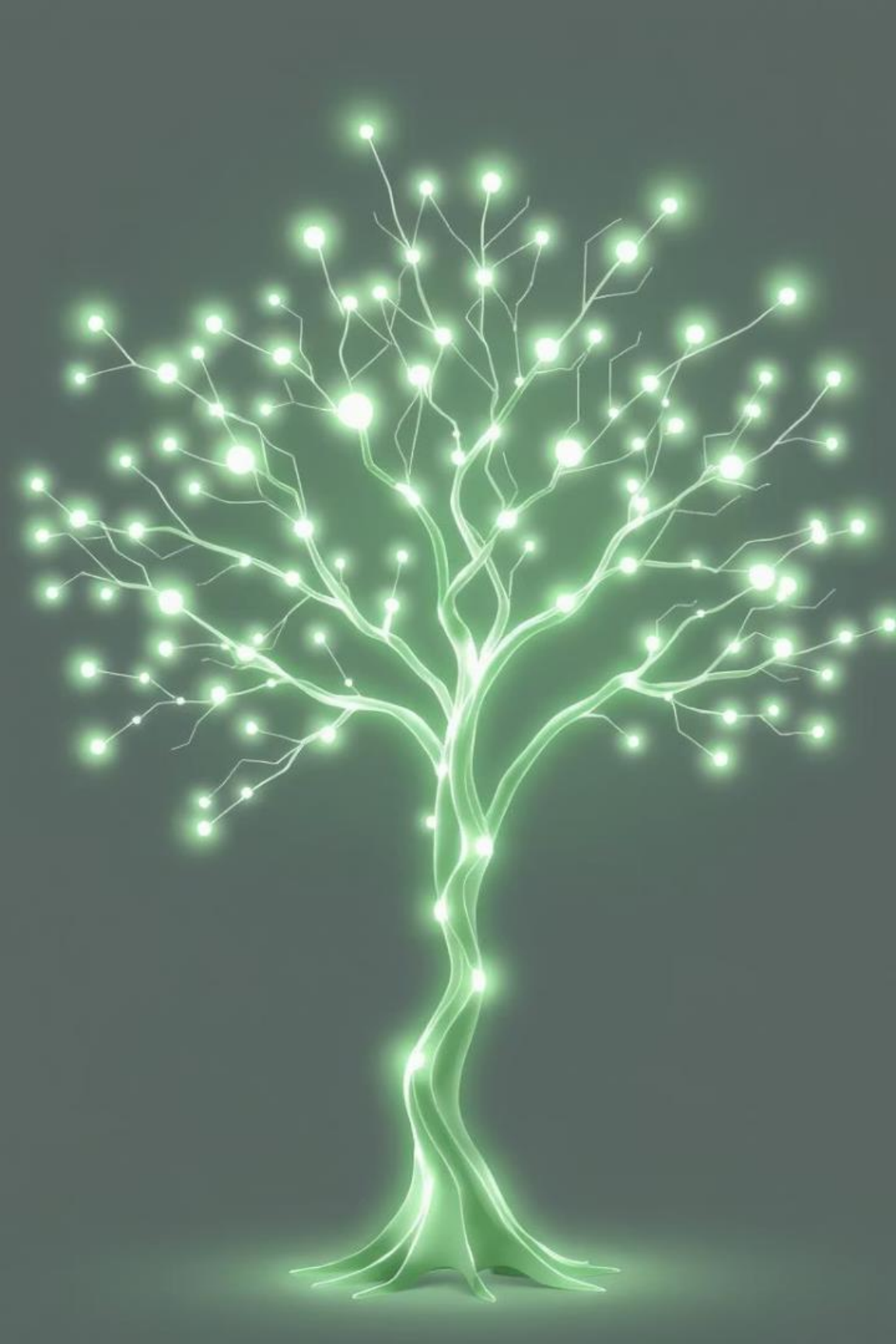
Exemplo de AST

Código Fonte:

```
seed x = 10 branch x > 5 then { print x }
```

AST Gerada:

```
Block
  Declaration
    Identifier(x)
    IntVal(10)
  Conditional
    Condition: BinOp(>)
      Identifier(x)
      IntVal(5)
    Then: Block
      Print
        Identifier(x)
```



Conclusão

4

Pontos Fortes

Sintaxe intuitiva e natural, visualização da estrutura do código, ferramenta educacional, simplicidade e clareza.

2

Aplicações

Ensino de conceitos de compiladores, prototipagem rápida e demonstração de estruturas de dados em árvore.

Obrigado!

Documentação Completa

Acesse o README para detalhes da linguagem.

Código Fonte Comentado

Explore o código para entender a implementação.

Exemplos Práticos

Experimente os exemplos para ver Arbor em ação.

