Guia 1

AMETISTA

LANGUAGE DESIGN DOCUMENT (LDD)

v1.0

Criado em 14/09/2025

1. Objetivo da linguagem

Ametista é uma linguagem feita para tornar a programação mais leve e prazerosa. Ela usa a filosofia do “desenvolvedor feliz” do Ruby e leva adiante, mas sem abrir mão da performance de verdade.

A sintaxe é simples e direta, fácil de ler e de entender, pensada para que você consiga ser criativo sem se perder em detalhes desnecessários. Ao mesmo tempo, oferece recursos robustos e eficientes, funcionando bem tanto em projetos pequenos quanto nos maiores desafios.

No fundo, Ametista valoriza **legibilidade**, **simplicidade** e **felicidade**, mas sem deixar de lado eficiência e robustez. O objetivo é que escrever código limpo e elegante seja natural e prazeroso, sem sacrificar velocidade ou recursos avançados.

2. Entrada e Saída de dados

2.1. Saída

```
out "Olá, mundo!" ;; output sem quebra de linha  
print "Hello World!" ;; output com quebra de linha
```

2.2. Entrada

```
str nome = input "Digite seu nome: "
```

3. Tipos de dados

- `int`, `float`, `bool`, `str`, `nil`, `list`, `table`

3.1 Variáveis

```
int numero = 45  
str texto = "Ametista"  
bool status = true  
float valor = 0.0  
str frutas = ["maçã", "banana", "goiaba"]  
table numeros_extenso = {"zero": 0, "um": 1, "dois": 2}
```

3.2 Constantes

```
const str nome = "Ametista"
```

4. Estrutura condicional

4.1 Condição simples

```
if idade < 12 do  
  out "Criança"  
end
```

4.2 Condição composta

```
if idade < 12 do  
  out "Criança"  
else  
  out "Adulto"  
end
```

4.3 Condição aninhada

```
if idade <= 12 do
  out "Criança"
elif idade < 18 do
  out "Adolescente"
else
  out "Adulto"
end
```

4.4 Condição múltipla

```
choose letra begin
  case "A" do
    out "Você escolheu A!"
  end
  case "B" do
    out "Você escolheu B!"
  end
  other
    out "Você escolheu outra letra!"
end
```

5. Estrutura de repetição

5.1 Estrutura while

```
while true do
  out "Ametista é bom!"
end
```

5.2 Estrutura do while

```
do
  out "Ametista é legal"
while numero < 50
```

5.3 Estrutura for

```
for int i = 0; i < idade; i++ do
    out i
end
```

6. Funções

6.1 Função sem retorno

```
defun hello_world do
    out "Olá, mundo!"
end
```

6.2 Função com retorno

```
defun str say_hello do
    return "Olá!"
end
```

6.3 Função sem parâmetro

```
defun int isNumber do
    return 0
end
```

6.4 Função com parâmetro

```
defun int somar(a, b) begin
    return a + b
end
```