

# Linguagem de Programação: Lua

Departamento de Sistemas de Informação  
Universidade Federal de Sergipe (UFS) – Itabaiana, SE – Brasil

Clovijan Bispo Rocha  
Gilmário Dos Santos Silva  
Magda Tainy Nunes Amaral

## Documentação Sintática da Linguagem LUA

### 1. Elementos Sintáticos.

#### Comandos

Lua oferece um conjunto quase convencional de comandos, similar ao conjunto de comandos disponíveis em Pascal ou C. Este conjunto inclui atribuições, estruturas de controle, chamadas de funções e declarações de variáveis

comando -> atribuicao  
| chamadaDeFuncao  
| comandoCondicional  
| comandoRepetitivo  
| comandoFor  
| comandoRepeat  
| comandoDeFuncao  
| comandoLocal

#### Trechos

Lua denomina trecho como uma sequência de comandos, os quais são executados sequencialmente. Cada Comando pode ser opcionalmente seguido por um ponto-e-vírgula.

trecho -> comando “;”

#### Blocos

Um bloco é uma lista de comandos; sintaticamente, um bloco é a mesma coisa que um trecho:

bloco -> trecho

Um bloco pode ser explicitamente delimitado para produzir um único comando:

comando -> **do** bloco **end**

Obs: Os blocos são úteis para controlar o escopo de declaração de variáveis. Pode ser usado também para adicionar um comando **return** ou **break** no meio de outro bloco.

## Atribuição

Lua permite atribuições múltiplas. Sendo assim, para atribuição define uma lista de variáveis no lado esquerdo e uma lista de expressões no lado direito. Os elementos em ambos os lados são separados por vírgulas:

```
comando -> listavar '=' listaexp
listavar -> var {' var}
listaexp -> exp {' exp}
```

## Estruturas de Controle

As estruturas de controle if, while e repeat possuem o significado usual e a sintaxe familiar:

```
comando -> while exp do bloco end
comando -> repeat bloco until exp
comando -> if exp then bloco {elseif exp then bloco} [else bloco] end
```

O comando While, inicia com a palavra reservada while seguido de uma expressão e de uma palavra reservada **do** que é o delimitador de código obrigatoriamente deve existir um bloco e outra palavra reservada o **end**.

O comando if em Lua começa com a palavra reservada "if", seguida de uma expressão. Depois disso, é esperado que haja a palavra reservada "then" para delimitar o bloco de código a ser executado se a expressão for avaliada como verdadeira. É possível ter um ou mais blocos de código adicionais utilizando a palavra reservada "elseif" seguida de outra expressão e do bloco de código correspondente. O comando if pode ser finalizado com a palavra reservada "else", que delimita o bloco de código a ser executado se nenhuma das expressões anteriores for verdadeira. O comando if deve terminar com a palavra reservada "end".

## Comando for

O comando **for** possui duas variações: uma numérica e outra genérica.

```
comando -> for nome '=' exp ',' exp[',' exp] do bloco end
| for listadenomes in listaexp do bloco end
| listadenomes '=' nome {',' nome}
```

A variação numérica começa com a palavra reservada "**for**", seguida pelo nome da variável que será usada para iterar, a palavra reservada "=", a expressão inicial da iteração, a palavra reservada ",", a expressão final da iteração e, opcionalmente, outra palavra reservada ",",

seguida pela expressão de passo. O bloco de código a ser executado é delimitado pela palavra reservada **"do"** e deve terminar com a palavra reservada **"end"**.

Já a variação genérica começa com a palavra reservada **"for"**, seguida por uma lista de nomes de variáveis separadas por vírgulas, a palavra reservada **"in"** e uma lista de expressões separadas por vírgulas. O bloco de código a ser executado é delimitado pela palavra reservada **"do"** e deve terminar com a palavra reservada **"end"**.

## Expressões

As expressões básicas em Lua são as seguintes:

exp -> expprefixo  
exp -> nil | false | true  
exp -> Numero  
exp -> Cadeia  
exp -> funcao  
exp -> construtortabela  
exp -> "..."  
exp -> exp opbin exp  
exp -> opunaria exp

expprefixo -> var | chamadadefuncao | "(" exp ")"

As expressões básicas em Lua são definidas pelos seguintes padrões sintáticos:

- Uma expressão pode começar com um prefixo, definido pelo padrão expprefixo.
- Uma expressão pode ser uma das palavras reservadas "nil", "false" ou "true".
- Uma expressão pode ser um número.
- Uma expressão pode ser uma cadeia de caracteres.
- Uma expressão pode ser uma função, definida pelo padrão sintático "funcao".
- Uma expressão pode ser um construtor de tabela, definido pelo padrão sintático "construtortabela".
- Uma expressão pode ser o operador de vararg, representado por três pontos ("...").
- Uma expressão pode ser uma operação binária, definida pelo padrão sintático "exp opbin exp", onde "opbin" é um operador binário e "exp" é uma expressão.
- Uma expressão pode ser uma operação unária, definida pelo padrão sintático "opunaria exp", onde "opunaria" é um operador unário e "exp" é uma expressão.
- O prefixo de uma expressão pode ser uma variável (definida por "var"), uma chamada de função (definida por "chamadadefuncao") ou uma expressão delimitada por parênteses "(" exp ")".

## **Referência**

Roberto Ierusalimschy, Luiz Henrique de Figueiredo, Waldemar Celes. Manual de Referência de Lua 5.1. Disponível em: <[Manual de Referência de Lua 5.1](#)>. Acessado em 28 de Março de 2023.