



Além de jogar, eu faço jogos!

Uma parceria de sucesso entre o CPDI e o
Instituto Cooperforte



Comitê para Democratização
da Informática

Professora Juliana Oliveira



Pauta – Aula 2

O que vamos aprender hoje

01

Índice do módulo jogos:
conhecendo Lua



Conhecendo o Sublime Text



Conhecendo o Solar 2d/Corona SDK

The Lua logo consists of a large dark blue circle with a white circle inside it, forming a stylized face. The word "Lua" is written in white inside the large circle. A dashed grey line forms a larger circle around the logo. A yellow wavy line is positioned above the logo, and a smaller dark blue circle is to its right. The background features green curved shapes.

Lua

Linguagem utilizada

- Projetada pela PUC-Rio em 1993.
- Primeiro jogo Grim Fandango (1997)



Jogos Famosos

Jogos que utilizam Lua em seu processo de programação

- Angry Birds
- Counter-Strike Online
- Roblox
- Street Fighter IV



- Linguagem de programação de extensão projetada para suportar facilidade na descrição de dados.
- Os identificadores podem ser qualquer sequência de letras ou dígitos, exceto números. Algumas palavras são reservadas e não podem ser usadas como variáveis.

and	break	do	else	elseif	end	false
for	function	if	in	local	nil	not
or	repeat	return	then	true	until	while

- Há diferenciação entre maiúsculas e minúsculas.
- Linguagem tipada dinamicamente.
- Todos os valores são de primeira classe.
- Mecanismo de estruturação por tabelas.

Variáveis

São lugares que armazenam valores. Existem três tipos de variáveis em Lua: **global** , **local** e **table fields(campos da tabela)** (propriedades). Qualquer variável não inicializada nil é por padrão.

Globais

Variáveis globais não precisam de declarações. Você simplesmente atribui um valor a um para criá-lo.

Local

As variáveis locais são declaradas usando a instrução **local**.

Campos da tabela

Os campos da tabela são apenas os elementos da própria tabela.
Você indexa na matriz para atribuir os valores a um campo.

Operadores aritméticos

Lua suporta os operadores aritméticos binários usuais:

Operador	Propósito
<code>+</code>	Adição
<code>-</code>	subtração
<code>*</code>	multiplicação
<code>/</code>	divisão
<code>%</code>	módulo
<code>^</code>	exponenciação

Operadores relacionais

Os operadores relacionais em Lua são:

Operador	Propósito
<code>==</code>	igual a
<code>~=</code>	não igual a
<code><</code>	Menor que
<code>></code>	Maior que
<code><=</code>	menos que ou igual a
<code>>=</code>	Melhor que ou igual a

Operadores lógicos

Os operadores lógicos em Lua são and, or e not. Todos consideram falsos false e nil e qualquer outra coisa como verdadeiro.

- **and** - se o valor do primeiro argumento for igual a false ou nil ele retorna o mesmo, caso contrário retorna o segundo argumento.
- **or** - se o primeiro argumento for diferente de false ou nil ele retorna o primeiro, caso contrário retorna o segundo.
- **not** - sempre retorna false ou true.

Exemplos:

<code>10 or 20</code>	<code>--> 10</code>
<code>10 or error()</code>	<code>--> 10</code>
<code>nil or "a"</code>	<code>--> "a"</code>
<code>nil and 10</code>	<code>--> nil</code>
<code>false and error()</code>	<code>--> false</code>
<code>false and nil</code>	<code>--> false</code>
<code>false or nil</code>	<code>--> nil</code>
<code>10 and 20</code>	<code>--> 20</code>