

## Exemplo 5 - Passagem de Valores

### Objetivos:

Explorar a passagem de valores de propriedades entre nós NCLua.

### Comportamento Esperado:

Um campo de entrada e dois de saída são exibidos na tela.

Enquanto o usuário preenche o campo de entrada, o primeiro campo de saída é automaticamente atualizado com o valor da entrada.

Quando o usuário aperta a tecla ENTER, o valor da entrada é copiado para o segundo campo de saída.

### Execução:

- Baixe o código do exemplo [aqui](#).
- Execute a aplicação e utilize o controle remoto (ou teclado numérico) para a entrada.

### Considerações:

Os nós de entrada e saída são implementados nos arquivos `input.lua` e `output.lua`, respectivamente.

Ambos tratam a propriedade `text` contendo o texto visualizado. No caso do NCLua de entrada, o campo `text` é controlado pelo próprio NCLua, e é alterado toda vez que o usuário pressiona uma nova tecla. No caso do NCLua de saída, o campo `text` deve ser controlado pelo documento NCL, através de elos de `set`.

Neste exemplo veremos os NCLua's como caixas pretas, não importando o conteúdo dos arquivos `.lua`. Para o autor do documento NCL basta saber a interface que cada um dos NCLua oferece com suas propriedades `text`. Dessa forma, os arquivos `.lua` podem ser reusados em outras aplicações.

O campo de entrada é representado pelo seguinte código:

```
<media id="input" src="input.lua" descriptor="dsInput">
  <area id="select"/>
  <property name="text"/>
</media>
```

Ele também possui a âncora `select` que é iniciada sempre que o usuário pressiona ENTER.

Os campos de saída são representados com o seguinte código:

```
<media id="output1" src="output.lua" descriptor="dsOutput1">
  <property name="text"/>
</media>

<media id="output2" src="output.lua" descriptor="dsOutput2">
  <property name="text"/>
</media>
```

A parte mais importante do documento é a que contém os elos responsáveis em copiar o conteúdo do campo de entrada para os de saída.

O primeiro campo de saída é atualizado sempre que a propriedade `text` do campo de entrada é alterada:

```
<link xconnector="onEndAttributionSet">
  <bind role="onEndAttribution" component="input" interface="text"/>
  <bind role="set" component="output1" interface="text">
    <bindParam name="var" value="$get"/>
  </bind>
  <bind role="get" component="input" interface="text"/>
```

```
</link>
```

O *bind* com `role="get"` permite consultar o valor de uma propriedade de qualquer nó, guardando-o na variável `$get`. No exemplo acima, a propriedade consultada é a `text` do nó `input`. A variável `$get`, por sua vez, é utilizada no *bind* com `role="set"`, fazendo com que o valor da entrada seja copiado para o nó `output1`.

O segundo campo de saída é atualizado somente quando o usuário pressiona a tecla `ENTER` no nó de entrada:

```
<link xconnector="onBeginSet">
  <bind role="onBegin" component="input" interface="select"/>
  <bind role="set" component="output2" interface="text">
    <bindParam name="var" value="$get"/>
  </bind>
  <bind role="get" component="input" interface="text"/>
</link>
```

O mecanismo de `role="get"` também é usado para a cópia do valor de entrada.