

Exemplo 6 - Consulta ao Google

Objetivos:

Introduzir o uso de conexões TCP.

Comportamento Esperado:

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

Após o usuário preencher o campo de entrada e pressionar ENTER, o valor da entrada é pesquisado no Google. O resultado da pesquisa (a URL retornada pelo *"I'm feeling lucky"*) é carregado no campo de saída.

Execução:

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

Considerações:

O nó de entrada possui a propriedade `text`, que guarda o conteúdo corrente, e a âncora `select`, iniciada sempre que o usuário pressiona ENTER:

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

O nó de saída possui somente a propriedade `text` cujo valor é exibido tela:

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

O código para ambos é o mesmo utilizado no [exemplo 5](#).

O nó `google` possui duas propriedades: `search` e `result`:

```
<media id="google" src="google.lua">
  <property name="search"/>
  <property name="result"/>
</media>
```

Quando o valor de `search` é alterado, o *script* faz a consulta com o valor passado e guarda o resultado na propriedade `result`.

O elo a seguir inicia a consulta assim que a âncora `select` do nó de entrada é iniciada, ou seja, quando o usuário pressiona ENTER. O valor da entrada é copiado para a propriedade `search` do nó `google`.

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

Quando a busca termina, o nó `google` altera a sua propriedade `result`, fazendo com que o elo abaixo atualize o campo de saída:

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

```
</bind>
<bind role="get" component="google" interface="result"/>
</link>
```

A comunicação via TCP é feita com o uso da **classe de eventos 'tcp'**. Seu uso não é trivial, uma vez que as chamadas são assíncronas e o programador deve controlar cada etapa da comunicação (conexão, envio, recebimento e desconexão) e chamadas desconexas.

Para facilitar a programação, disponibilizamos junto com este exemplo o arquivo `tcp.lua`. Esse módulo faz o controle das chamadas assíncronas, permitindo que a comunicação seja programada sequencialmente.

A principal função, `tcp.execute`, recebe uma função que pode conter os seguintes comandos:

- `tcp.connect`: recebe o endereço e porta de destinos
- `tcp.send`: recebe uma string com o valor a ser transmitido
- `tcp.receive`: retorna a string com o valor a ser recebido
- `tcp.disconnect`: fecha a conexão

Com o uso dessa API, o código que faz a busca no google fica como a seguir:

```
tcp.execute(
  function ()
    tcp.connect('www.google.com.br', 80)
    tcp.send('get /search?hl=pt-BR&btnI&q=' .. evt.value .. '\n')
    local result = tcp.receive()
    if result then
      result = string.match(result, '<A HREF="http://(.-)">') or 'nao encontrado'
    else
      result = 'error: ' .. evt.error
    end
    local evt = {
      class    = 'ncl',
      type     = 'attribution',
      property = 'result',
      value    = result,
    }
    evt.action = 'start'; event.post(evt)
    evt.action = 'stop' ; event.post(evt)
    tcp.disconnect()
  end
)
```