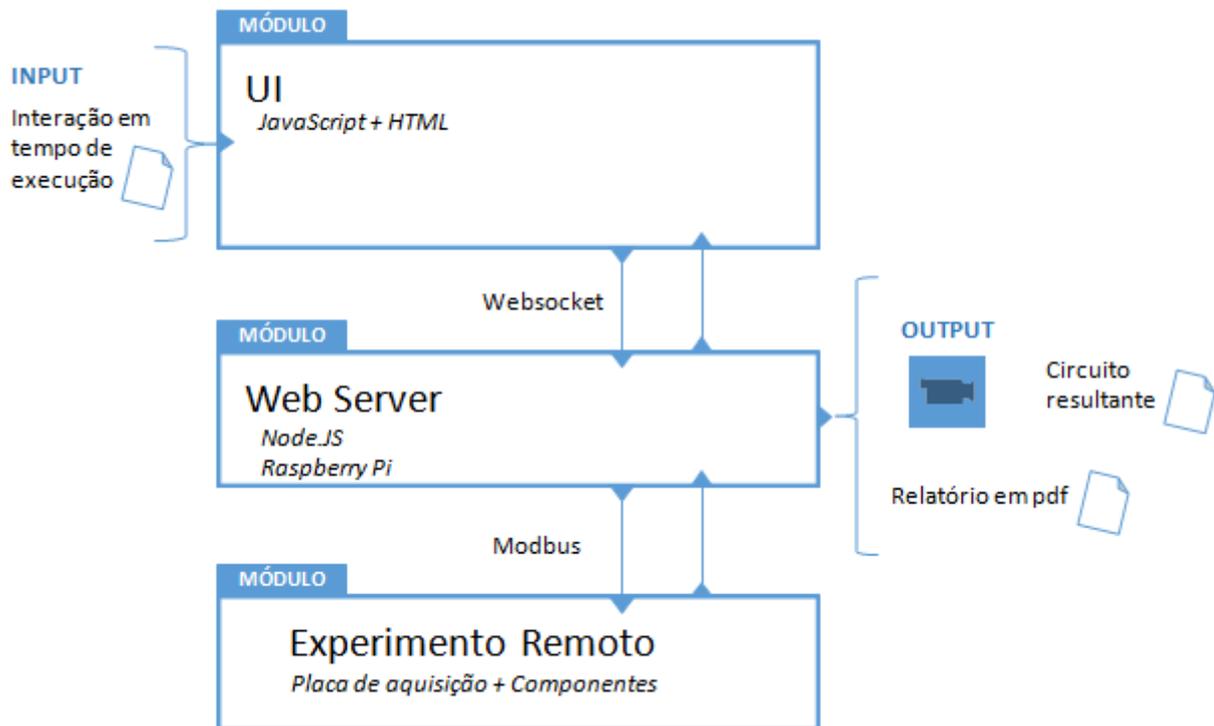


# Componentes de um laboratório no RELLE

Um experimento é composto de três módulos: a interface web, o servidor de experimento e o próprio experimento físico, como ilustrado na figura abaixo. Esta página traz informações para desenvolvedores de laboratórios remotos criarem interfaces e hospedar seus experimentos no RELLE.



## Pacote de arquivos

Este pacote contém os arquivos para desenvolver a interface web de um experimento.  
[https://github.com/RExLab/lab\\_package](https://github.com/RExLab/lab_package)

## Dependências disponíveis para a interface de usuário

Obrigatórios - já são carregados na interface, não é necessário adicionar qualquer código para usar suas funcionalidades:

- Biblioteca jQuery 2.1.3 <https://jquery.com/download/>
- Framework CSS Bootstrap <http://getbootstrap.com/>
- Tema Flat UI <http://designmodo.github.io/Flat-UI/>

Opcionais - não são carregados na interface, é necessário adicionar códigos para usar suas funcionalidades:

- Biblioteca Shepherd. Tutoriais: <http://github.hubspot.com/shepherd/docs/welcome/>. Código: <http://relle.ufsc.br/js/shepherd.js>
- ChartJS: Esta biblioteca é usada para criação de diversos tipos de gráficos dinâmicos em HTML5

utilizando canvas. [Documentação Chart.js](#)

- Biblioteca mlens <http://mlens.musings.it/>: Ampliação de imagens Código: <http://relle.ufsc.br/js/lens.js>
- Biblioteca ZoomJS <https://github.com/marcaube/bootstrap-magnify>: Ampliação de imagens Código: <http://relle.ufsc.br/js/zoom.js>
- Biblioteca RoundSlider <http://roundsliderui.com/>

<http://relle.ufsc.br/js/> Como exemplo, observe o código abaixo, ele deve estar no arquivo exp\_script.js.

```
$.getScript('http://relle.ufsc.br/js/Chart.js', function () {
    // Ajustar parâmetros necessários, etc...

});
```

## pt.html

Este arquivo é carregado quando o usuário tem acesso ao experimento. Somente é obrigatório manter a div DivExp, o restante fica a critério de cada desenvolvedor.

[pt.html](#)

```
<div id='DivExp' class="container">
  <center>
    <h3>Nome do Experimento</h3>

    <div class="row"> <!-- Classes container, row e col-xx fazem
parte do framework bootstrap e são usados para organizar os elementos
na página -->
      <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12" >
        <img class="cam" src='http://relle.ufsc.br:8072'
width="100%"/> <!-- Inserir endereço da câmera IP -->
        <div class='row'>
          <div class='col-lg-2 col-lg-offset-2 col-md-3 col-
sm-3 col-xs-3'>
            <!-- Aqui podem ser inseridos quaisquer
controles do experimento que ficarão abaixo da imagem -->
          </div>
        </div>
      </div>
      <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12">
        <!-- Aqui podem ser inseridos controles e retorno de
sensores -->
      </div>
      <div id="report"></div> <!--remove tag se não houver
relatório de experiência -->
      <div id="csv"></div> <!--remove tag se não houver
exportação dos dados em formato csv-->
    </div>
```

```
    </center>
</div>

<script> <!-- Strings traduzidas que serão utilizadas no exp_script.js
-->
var lang =
    {
        showme: 'Mostre-me como funciona',
        showmeshort: 'Como funciona',
        next: "Próximo",
        previous: "Anterior",
        done: "Pronto",
        leave: "Sair",
        intro:"Insira texto para o tutorial" ,
        introcamera: "Esta transmissão de vídeo",
        report: "Salve os resultados de sua experiência..."

    };

</script>
```

## en.html

Este arquivo possui as mesmas características do pt.html, diferenciando-se do conteúdo para o usuário, que por sua vez deve estar em inglês

## exp\_style.css

É usado para definir como serão exibidos os elementos contidos no código de uma página da internet.

[exp\\_style.css](#)

```
h1{
    text-align: center;
    padding-bottom: 20px;
}

#cam{
    margin-top:7%;
}

// Insira aqui quantas regras css forem necessárias.
```

## exp\_script.js

Este é um template para aplicações de laboratório escritas em node.js.

### exp\_script.js

```
var rpi_server = "http://painelcc1.relle.ufsc.br"; // URL para conexão
com o servidor do experimento

$(function () {

    $.getScript(rpi_server + '/socket.io/socket.io.js', function () {

        socket = io.connect(rpi_server);

        socket.emit('new connection', {pass: $("#pass").html()});

        socket.on('new message', function (data) {
            console.log(data);
        });

        socket.on('data received', function (data) {

        });

        $.getScript('http://relle.ufsc.br/exp_data/{ID}/welcome.js',
function () {

            $('#return').append('<button id="btnIntro" class="btn btn-
sm btn-default"> <span class="long">' + lang.showme + '</span><span
class="short">' + lang.showmeshort + '</span></button>');
            var shepherd = setupShepherd();
            $('#btnIntro').on('click', function (event) {
                event.preventDefault();
                shepherd.start();
            });
        });
    });
});

function report(id) {
}
```

## Adicionando tutoriais

Em alguns casos pode ser necessário mostrar ao usuário como usar a interface de seu experimento. A biblioteca [Shepherd](#) permite criar guias que indicam o elemento da página com texto explicativo associado. Além disso, é possível inserir código javascript para serem chamados durante a transição entre uma mensagem e outra.

O pacote de arquivos traz um template para configuração dos guias no arquivo `welcome.js`. Este arquivo deve estar na pasta do seu experimento (**/exp\_data/{ID}**) e é carregado pelo arquivo `exp_script.js` (ver linhas 21 a 30).

Os tutoriais dependem de um arquivo CSS. A seguinte linha de código deve ser incluída na primeira linha do arquivo `exp_script.js` ou também adicionado outro arquivo equivalente customizado.

```
$('head').append('<link rel="stylesheet" href="http://relle.ufsc.br/css/shepherd-theme-arrows.css" type="text/css"/>');
```

`welcome.js`

```
var setupShepherd = function () {

    var shepherd = new Shepherd.Tour({
        defaults: {
            classes: 'shepherd-element shepherd-open shepherd-theme-arrows',
            showCancelLink: true,
            scrollTo: true
        }
    });

    shepherd.on('cancel', function () {
        // função chamada ao cancelar ou sair do tutorial
    });

    shepherd.on('start', function () {
        // função chamada ao iniciar o tutorial
    });

    // Primeira mensagem do tutorial

    shepherd.addStep('shep_intro', {
        text: [lang.intro],
        classes: 'shepherd shepherd-open shepherd-theme-arrows shepherd-transparent-text',
        buttons: [
            {
                text: lang.leave,
                classes: 'shepherd-button-secondary',
                action: function () {
```

```
        // Função chamada ao clicar no primeiro botão.
        // Se necessário, escrever seu código aqui
        shepherd.cancel();
    }, {
        text: lang.next,
        action: function () {
            // Função chamada ao clicar no segundo botão.
            // Se necessário, escrever seu código aqui
            shepherd.next();
        },
        classes: 'shepherd-button-example-primary'
    }
    ]
});

// Tour no meio
shepherd.addStep('shep_cam', {
    text: [lang.introcamera],
    attachTo: 'img.cam center', // A primeira string indica o
    elemento focado, isto é, uma tag img com classe cam. A string após o
    espaço indica a posição da mensagem será no centro deste elemento
    // Sem este campo attachTo, a mensagem aparecerá no meio da
    página

    classes: 'shepherd shepherd-open shepherd-theme-arrows
shepherd-transparent-text',
    buttons: [
        {
            text: lang.previous,
            classes: 'shepherd-button-secondary',
            action: function () {
                shepherd.back();
            }
        }, {
            text: lang.next,
            action: function () {
                shepherd.next();
            },
            classes: 'shepherd-button-example-primary'
        }
    ]
});

shepherd.addStep('shep_report', {
    text: [lang.report],
    classes: 'shepherd shepherd-open shepherd-theme-arrows
shepherd-transparent-text',
    buttons: [
        {
            text: lang.previous,
```

```
        classes: 'shepherd-button-secondary',
        action: function () {
            shepherd.back();
        }
    }, {
        text: lang.done,
        action: function () {
            shepherd.cancel();
        },
        classes: 'shepherd-button-example-primary'
    }
    ]
});

return shepherd;
};
```

Para configurações adicionais ou customização pode ser consultada documentação da biblioteca Shepherd e seguir as instruções lá presentes.

From:

<http://docs.relle.ufsc.br/> -

Permanent link:

<http://docs.relle.ufsc.br/doku.php?id=desenvolvendo-experimento>



Last update: **2016/09/30 19:23**