Documentação

Laboratório Interação e Audiovisuais – Plataforma de reservas

Conteúdo

[Requisitos 2](#_Toc524800337)

[Modelo de dados 3](#_Toc524800338)

[Permissões 4](#_Toc524800339)

[Guardar permissões 4](#_Toc524800340)

[Esconder elementos do utilizador 5](#_Toc524800341)

[Verificar permissões 6](#_Toc524800342)

[Mensagens 8](#_Toc524800343)

[Menu de mensagens 8](#_Toc524800344)

[Número de mensagens novas 8](#_Toc524800345)

[Apresentar mensagens 10](#_Toc524800346)

[Permissão para ver mensagem 12](#_Toc524800347)

[Reservas 13](#_Toc524800348)

[Número de reservas totais, “Pendentes” e “Em atraso” 13](#_Toc524800349)

[Alteração automática de reservas “Em progresso” para “Em atraso” 15](#_Toc524800350)

[Botão página anterior 17](#_Toc524800351)

[Datatables 19](#_Toc524800352)

[Datepicker 21](#_Toc524800353)

[Descrição dos ficheiros 23](#_Toc524800354)

[Localização de ficheiros 23](#_Toc524800355)

[Páginas do site na pasta \pa\production\ 23](#_Toc524800356)

[Ficheiros dos API na pasta \myslimsite\app\api\ 25](#_Toc524800357)

[Ficheiros dos API na pasta \myslimsite\public\ 25](#_Toc524800358)

# Requisitos

Para o funcionamento desta plataforma são **obrigatórios** os seguintes elementos na base de dados com os seguintes dados:

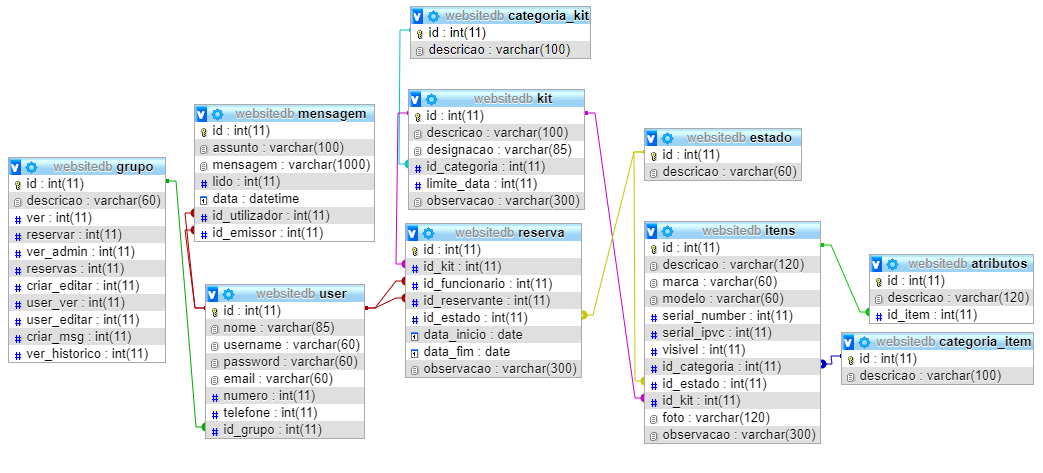
* Tabela grupo:
  + Um grupo com a descrição “Visitante” em que a única permissão será “ver”.
* Tabela user:
  + Um utilizador com o username “Sistema” e um email também.
* Tabela estado:
  + Estados com a descrição “Cancelado”, ”Pendente”, “Aceite”, “Recusado”, “Em atraso” e “Em progresso”.
* Tabela kit:
  + Kit com a descrição “Sem kit”.
* Tabela categoria\_kit:
  + Categoria kit com a descrição “Sem categoria”.
* Tabela categoria\_item:
  + Categoria item com a descrição “Sem categoria”.

Também é necessário alterar o limite de tamanho de upload para 10 mb e a quantidade de upload de ficheiros para 20 no ficheiro de configuração de php do servidor.

Por fim é preciso uma fotografia com o nome “default.png” estar presente na pasta “images”, este ficheiro será utilizado como fotografia por defeito caso um item seja inserido na base de dados sem nenhuma fotografia.

Só **depois** de ter estes elementos na base de dados é que a plataforma fica funcional.

# Modelo de dados

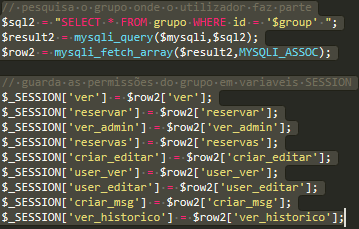


# Permissões

* **Ver** – permite ver mensagens do próprio utilizador e os kits disponíveis para reserva;
* **Reservar** – permite reservar kits e ver o histórico de reservas do próprio utilizador;
* **Ver Admin** – permite ver itens escondidos na pesquisa, ver ficheiro de admin na página sobre e também, caso esteja a editar itens, editar itens escondidos;
* **Reservas** – permite editar, aceitar e recusar reservas;
* **Criar Editar** – permite a criação e edição de itens, kits, estados, categorias de itens e categorias de kits;
* **User ver** – permite a visualização de uma lista de utilizadores;
* **User editar** – permite a criação de grupos e a edição de utilizadores e de grupos;
* **Criar msg** – permite a criação de mensagens;
* **Ver histórico** – permite a visualização do histórico de mensagens e de reservas de todos os utilizadores.

## Guardar permissões

As permissões na plataforma são todas verificadas através de variáveis SESSION que ficam guardadas sempre que o utilizador faz login. Depois de fazer login é feita uma query para verificar as permissões que o grupo, ao qual o utilizador pertence, tem. Na base de dados vai ter os campos referidos acima e vão estar guardados os valores “1” e ”0”, se o valor for “1” então é considerado como se o utilizador tem essa permissão, se for “0” então o utilizador não tem permissão.



Nos ficheiros “login\_check.php” e “register\_user.php” vai ter este código, em que é efetuada a query e depois guarda as permissões do utilizador em variáveis SESSION.

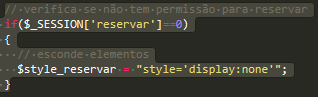
## Esconder elementos do utilizador

Em todas as páginas existem elementos que dependendo das permissões do utilizador, estar visíveis ou não. No ficheiro “session.php”, são feitas as verificações e mudanças para ocultar os elementos do utilizador.

Vamos verificar só elementos que não vão estar presentes porque o utilizador não tem permissão para reservar.



No início é definida a variável “style\_reservar”, onde o valor da variável vai ser “ ”.



Depois é feita a verificação para se saber se o utilizador tem permissão para reserva, se o utilizador não tiver permissão então o valor da variável passa a ser “ style = ’display:none’ ”.



Em todas as páginas vai ter o código apresentado acima, e como se pode verificar estamos a fazer um echo da nossa variável “style\_reservar”.

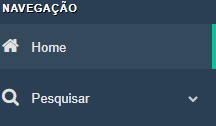
Caso o utilizador não tenha permissão o código html vai ficar assim:

“ **<li style = ’display:none’ ><a href="form\_search\_reserva.php"><i class="fa fa-archive"></i> Reservar kit </a></li>** ”

O que o código em vermelho faz é ocultar este elemento. Caso o utilizador tivesse permissão a variável ia manter o valor “ ” e o código html iria ter este aspecto:

“ **<li ><a href="form\_search\_reserva.php"><i class="fa fa-archive"></i> Reservar kit </a></li>** ”

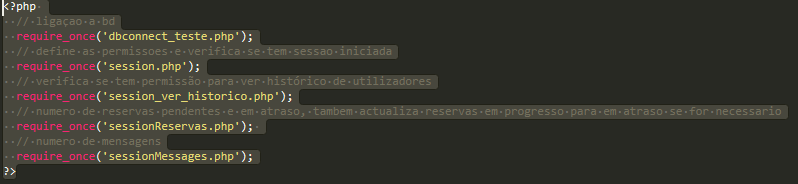
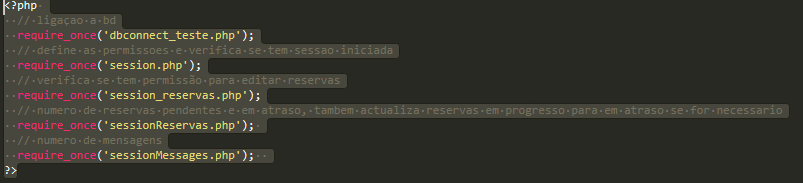
Desta forma o elemento está visível.



Como se pode verificar, numa imagem o botão “Reservar kit” está presente e na outra imagem não. Para todos os elementos que sejam para ficar visíveis ou não, dependendo da permissão, é sempre utilizado este método.

## Verificar permissões

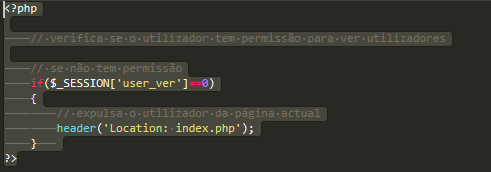
Em todas as páginas vai ter no início um excerto de código PHP que vai chamar uns ficheiros, e nesses mesmo ficheiros é verificado se o utilizador tem permissão para estar na página actual.



Os ficheiros “session\_” são os responsáveis por verificar se tem permissão para estar na página, como se pode verificar nas duas imagens, numa página chama o ficheiro “session\_reservas.php” e na outra “session\_ver\_historico.php”.

Os ficheiros em que verifica se o utilizador tem permissão para estar na página actual são os seguintes:

* session\_criar\_editar.php;
* session\_criar\_msg.php;
* session\_reservar.php;
* session\_user\_editar.php;
* session\_user\_ver.php;
* session\_ver.php;
* session\_ver\_admin.php;
* session\_ver\_historico.php.



Todos os ficheiros “session\_” tem o código idêntico a segunda imagem, onde é feita uma verificação a uma permissão em específico e se não tiver essa permissão, então o utilizador é enviado para a homepage.

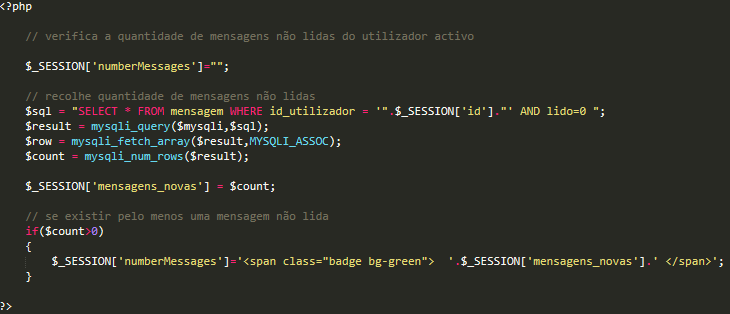
O único ficheiro “session\_” com o código ligeiramente diferente é o “session\_ver\_admin.php”, como se pode verificar na primeira imagem, o ficheiro é utilizado na página de editar itens, e neste caso a permissão permite ao utilizador ver e editar itens “escondidos”, sendo assim o código verifica se o item que está a ser editado é um desses itens “escondidos”, se for um desses itens e o utilizador não tiver a permissão “ver admin” então o utilizador é expulso da página actual.

# Mensagens

## Menu de mensagens

### Número de mensagens novas

Tal como no caso de verificar permissões, no início de todas as páginas vai ser chamado um ficheiro chamado “sessionMessages.php” onde é feita a verificação de novas mensagens.



A maneira como é feito para apresentar o número de mensagens novas é igual ao método utilizado para esconder elementos.

Primeiro a variável “$\_SESSION[‘numberMessages’]” é inicializada com o valor “ ”, depois é feita uma query onde pesquisa por todas as mensagens que pertençam ao utilizador com o mesmo id que o valor guardado no “$\_SESSION[‘id’]”, o utilizador activo, e com o estado de “lido” igual a 0.

Guarda o número de resultados na variável “$\_SESSION[‘mensagens\_novas’]”. Se o número de resultados da query for maior do que 0 então a variável “$\_SESSION[‘numberMessages’]” fica com o valor “**'<span class="badge bg-green"> '.$\_SESSION['mensagens\_novas'].' </span>'**”.

Caso o utilizador tenha, por exemplo, 14 mensagens por ler, o valor da variável iria ficar com este aspecto “**'<span class="badge bg-green"> 14 </span>'**”, caso não tenha nenhuma mensagem a variável iria manter este valor “ ”.



Em todas as páginas vai ter este código que representa o envelope que serve de menu das mensagens, como se pode ver pela imagem, é feito um echo da variável “$\_SESSION[‘numberMessages]”.

Voltando aos dois exemplos anteriores este código iria ficar assim para as 14 mensagens:

**“<a href="javascript:;" class="dropdown-toggle info-number" data-toggle="dropdown" aria-expanded="false">**

**<i class="fa fa-envelope-o"></i>**

**<span class="badge bg-green"> 14 </span>**

**</a>**”

E assim para as 0 mensagens novas:

**“<a href="javascript:;" class="dropdown-toggle info-number" data-toggle="dropdown" aria-expanded="false">**

**<i class="fa fa-envelope-o"></i>**

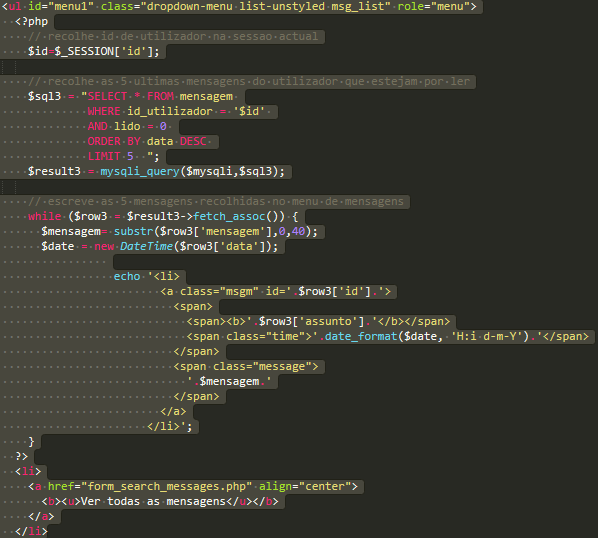
**</a>**”

Tal como se pode verificar nas imagens acima, uma delas indica 14 mensagens novas e a outra não indica nenhuma mensagem nova.

### Apresentar mensagens

Em todas as páginas vai ter este código presente para apresentar as últimas 5 novas mensagens.



Primeiro guarda o id do utilizador activo na variável “id”, depois é feita uma query onde pesquisa pelas 5 últimas mensagens que pertençam ao utilizador com o mesmo id que o valor guardado no “$\_SESSION[‘id’]”, o utilizador activo, e com o estado de “lido” igual a 0.

Faz um ciclo while em que enquanto existirem resultados da query, executa o código dentro do ciclo. Este código iria apresentar o assunto, a data e os primeiros 40 caracteres da mensagem.

Caso não existirem mensagens novas, simplesmente não corria o código dentro do ciclo while. No final é criado o botão para levar a uma página onde tem todas as mensagens do utilizador activo.

Nestas imagens pode verificar-se a diferença entre um utilizador com mensagens novas e outro sem mensagens novas.

## Permissão para ver mensagem

Na página “form\_open\_message.php” é feita uma verificação para saber se o utilizador tem permissão para ver a mensagem que vai ser apresentada, nessa página é chamado o ficheiro “sessionMessagesVer.php” onde vai ser feita essa verificação.



Para se efectuar esta verificação, compara-se o valor do “$\_SESSION[‘id’]”, o utilizador activo, e o valor do “id\_utilizador” da mensagem aberta, se os valores forem diferentes então o utilizador é enviado para a homepage.

# Reservas

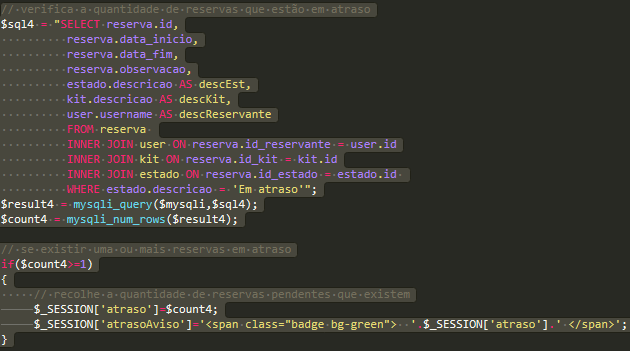
Em todas as páginas vai ter no início um excerto de código PHP que vai chamar o ficheiro “sessionReservas.php”, esse ficheiro tem como função verificar quantas reservas “pendentes” e “em atraso” existem, e também alterar as reservas de “em progresso” para “em atraso” caso a data de entrega passe.

## Número de reservas totais, “Pendentes” e “Em atraso”

A maneira como conta as reservas “pendentes” e “em atraso” é feita da mesma maneira, por isso só vai ser explicada para uma das reservas e do número total de reservas.

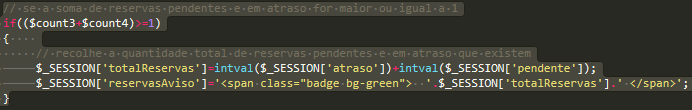


Primeiro são definidas as variáveis “$\_SESSION[‘atrasoAviso’]” e “$\_SESSION[‘reservasAviso’]”, com o valor de “ ”. Depois são definidas “$\_SESSION[‘atraso’]” e “$\_SESSION[‘totalReservas’]” com o valor de “0”.



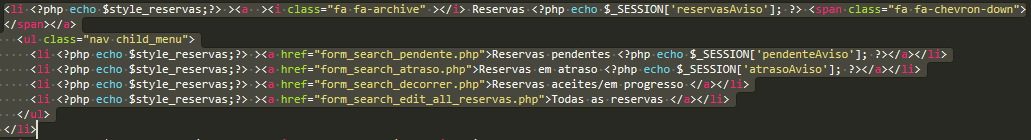
Depois é feita uma query para verificar todas as reservas que existem “Em atraso”, define-se a variável “$count4” como o número de resultados da query. Se o número de resultados for maior do que 0, então a variável “$\_SESSION[‘atraso’]” fica com o valor do número de resultados e a variável “$\_SESSION[‘atrasoAviso’]” fica com o valor “ **'<span class="badge bg-green"> '.$\_SESSION['atraso'].' </span>'** ”.

Tal como nas mensagens o valor de “$\_SESSION[‘atrasoAviso’]” vai depender do número de resultados, se o número de resultados for 8 então a variável fica com este valor “**'<span class="badge bg-green"> 8 </span>'**” e se não tiver nenhuma reserva em atraso então mantem o valor “ ”.



No caso das reservas totais, se a soma do número de reservas “pendentes” e “Em atraso” for maior do que 0, então a variável “$\_SESSION[‘totalReservas’]” fica com o valor dessa soma e a variável “$\_SESSION[‘reservasAviso’]” fica com o valor “**'<span class="badge bg-green"> '.$\_SESSION['totalReservas'].' </span>'**”.

Tal como nas reservas em atraso o valor da variável “$\_SESSION[‘reservasAviso’]” vai depender do número de reservas totais.



Em todas as páginas vai ter este código que representa os botões para editar reservas, como se pode verificar, são utilizadas as variáveis “$\_SESSION[‘atrasoAviso’]” e “$\_SESSION[‘reservasAviso’]” que foram definidas anteriormente.

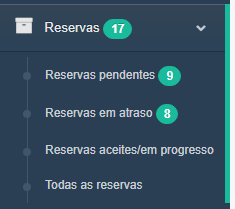
Voltando ao exemplo anterior vamos dizer que existem 8 reservas “Em atraso”, 9 reservas “Pendentes”, logo temos 17 reservas no total. O código iria ficar com o seguinte aspecto.

**“ <li ><a ><i class="fa fa-archive" ></i> Reservas <span class="badge bg-green"> 17 </span> <span class="fa fa-chevron-down"></span></a>**

**<ul class="nav child\_menu">**

**<li ><a href="form\_search\_pendente.php">Reservas pendentes <span class="badge bg-green"> 9 </span> </a></li>**

**<li ><a href="form\_search\_atraso.php">Reservas em atraso <span class="badge bg-green"> 8 </span> a></li> “**

****

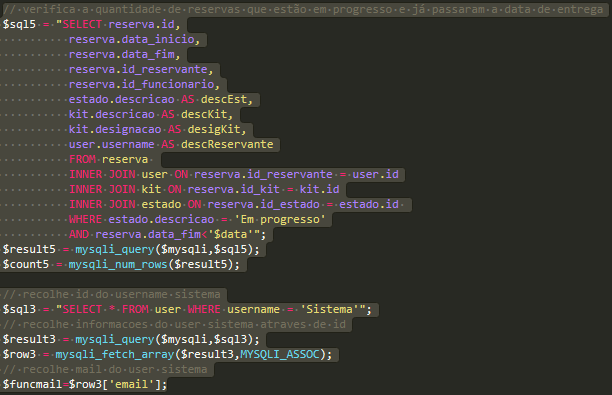
Como se pode verificar na imagem o aviso total de reservas está no topo, indicando 17 reservas no total, o número de reservas pendentes está no botão das pendentes, indicando 9 reservas pendentes, e por fim o número de reservas em atraso está no botão de reservas em atraso, indicando 8 reservas em atraso.

## Alteração automática de reservas “Em progresso” para “Em atraso”

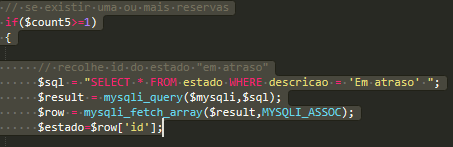
Esta alteração também ocorre no ficheiro “sessionReservas.php”.



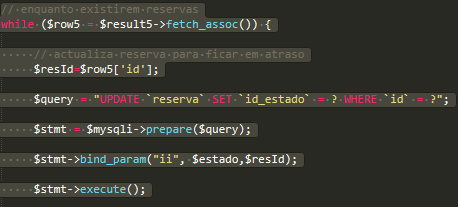
Primeiro define uma variável com a data actual.



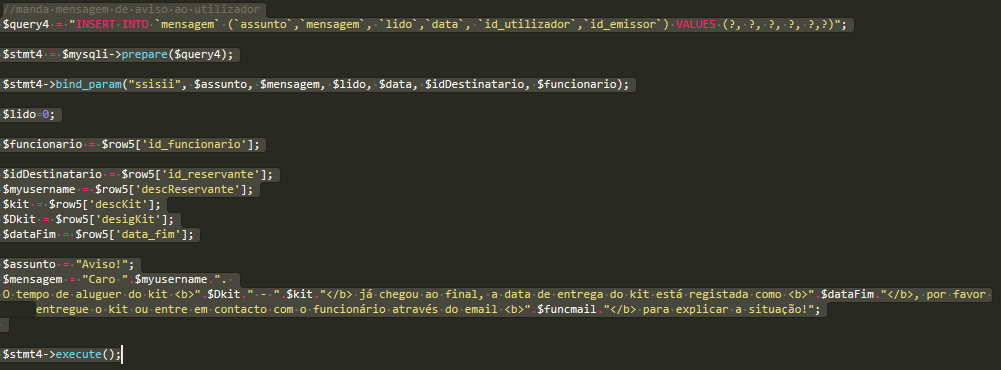
Depois faz uma query onde pesquisa por reservas com o estado “Em progresso” e que a sua data final já tenha passado a data definida na variável anterior. Depois é definida uma variável com o email do utilizador “Sistema”.



Se o resultado da query anterior for maior do que 0, primeiro define uma variável com o id do estado “Em atraso”.



De seguida começa um ciclo em que percorre todos os resultados da query e altera o estado da reserva para “Em atraso”, utilizando o id definido anteriormente.



Por fim envia uma mensagem ao utilizador a qual pertence a reserva, a avisar o utilizador da situação e para entrar em contacto com o funcionário.

# Botão página anterior

A maioria dos botões página anterior tem um link fixo, mas em alguns casos, esse link é definido por uma variável definida noutra página.

Como exemplo vai-se usar este diagrama, pode-se chegar a uma página genérica através de duas páginas diferentes, e na página genérica existe um botão para voltar a página anterior, a página genérica tem de saber qual foi a página de onde o utilizador veio para o botão funcionar de maneira correcta.

Uma dessas páginas é a página “form\_send\_messages.php”, que pode ser acedida através da página “form\_search\_send\_messages.php” e “form\_info\_user.php”.



Este código está presente na página “form\_search\_send\_messages.php” e como se pode verificar está a definir a variável da seguinte maneira **“$\_SESSION['pageMessage'] = 'form\_search\_send\_messages.php';**”.



Este código está presente na página “form\_info\_user.php” e como se pode verificar está a definir a variável da seguinte maneira **“$\_SESSION['pageMessage'] = 'form\_info\_user.php?var='.$id;**”.

Como se pode verificar quando o utilizador está numa dessas páginas, a variável “$\_SESSION['pageMessage']” é definida automaticamente.



Este código está presenta na página “form\_send\_messages.php”, que representa o código do botão da página anterior. Como se pode verificar estamos a indicar que a página alvo desse botão é a variável “$\_SESSION['pageMessage']”.

**“<a href="<?php echo $\_SESSION['pageMessage']; ?>" id="button" type="button" class="btn btn-primary botao" ><i class="fa fa-arrow-left"></i> Voltar pagina anterior</a>**”

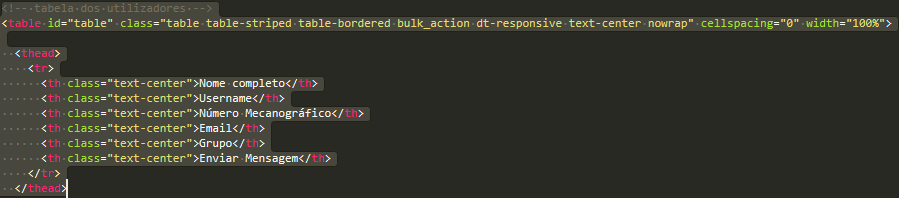
Desta maneira, quando se esta numa das páginas anteriores, a variável é automaticamente definida e na página seguinte, vai sempre voltar a página anterior correcta.

Todos os casos que utilizam este método:

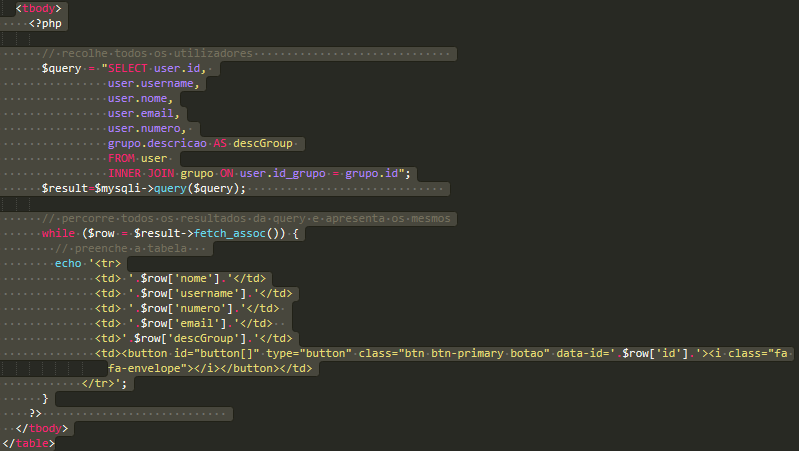
* “form\_info\_user.php” pode ser acedida a partir de:
  + “form\_edit\_all\_reservas.php”;
  + “form\_edit\_atraso\_reservas.php”;
  + “form\_edit\_decorrer.php”;
* “form\_view\_item.php” pode ser acedida a partir de:
  + “form\_view\_kit.php”;
* “form\_send\_messages.php” pode ser acedida a partir de:
  + “form\_info\_user.php”;
  + “form\_search\_send\_messages.php”.

# Datatables

Nas páginas onde existem tabelas é sempre utilizado datatables.

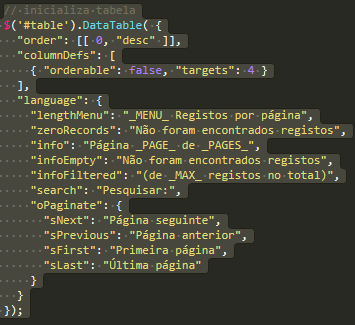


Primeiro é definido a tabela e os cabeçalhos da mesma.



Para preencher o corpo da tabela a maioria das vezes é utilizada uma query, neste caso é uma tabela com todos os utilizadores, onde é recolhida a informação necessária. Depois é preenchida a tabela através do código “**<td>’.$row[nome].’</td>**”, neste exemplo é o nome do utilizador.

No final da página é feita a inicialização da tabela.



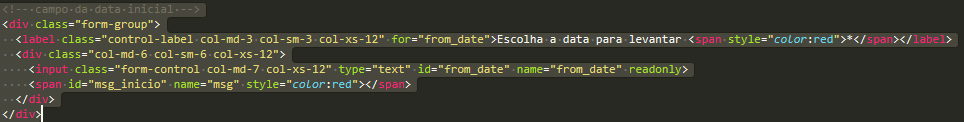
No código acima é inicializada a tabela, dentro da função tem várias opções para ter uma tabela mais personalizada.

Por exemplo, o “order” está a indicar que a coluna “0” tem ordem “desc”, ou seja, a primeira coluna vai estar ordenada por ordem descendente por defeito. Outro exemplo é o “orderable” que indica que a coluna “4” não pode ser ordenada, logo a tabela fica com uma quinta coluna que nunca se pode mudar a ordenação dos resultados.

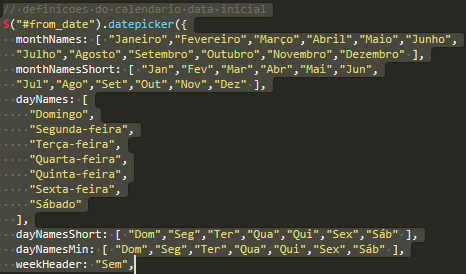
No resto do código é simplesmente tradução dos botões e textos apresentados na tabela.

# Datepicker

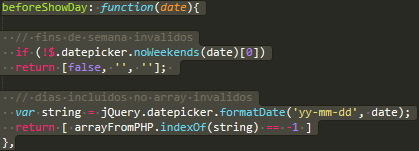
Nas páginas “form\_search\_reserva.php” e “form\_reserva.php” é utilizado um calendário para indicar as datas em que os kits estão ocupados, na página “form\_reserva.php” o calendário tem mais funções por isso vai ser utilizado esse calendário como exemplo.



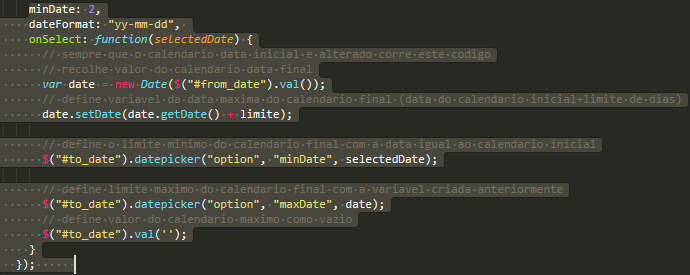
Na página é definido um campo com o “from\_date”, a data de início de reserva, e depois outro com o nome “to\_date”, a data de fim de reserva.



Depois o “from\_date” é inicializado e tal como os datatables são definidas várias opções quando é inicializado, e nesta parte do código é simplesmente tradução dos vários textos que são apresentados no datepicker.



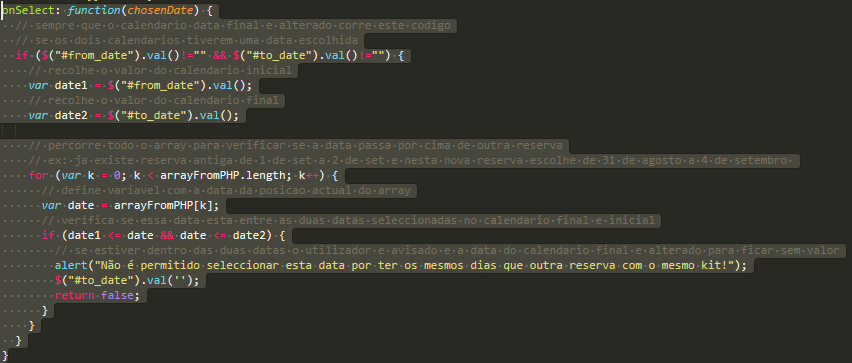
Nesta parte são definidos os dias inválidos, na primeira parte é indicado que os fins de semana contam como dias inválidos, na segunda parte é indicado que todas as datas presentes no array “arrayFromPHP”, que é um array com todas as datas em que o kit está numa reserva, também são considerados como dias inválidos.



Nesta última parte está a ser indicado que a data mínima que se pode escolher é 2 dias depois da data actual, o formato da data e por fim uma função que acontece sempre que se muda a data escolhida no datepicker “from\_date”.

Nessa função indica que a data mínima do “to\_date” vai ser igual á data que foi escolhida no “from\_date”, indica que a data máxima do “to\_date” vai ser a data escolhida mais o limite de reserva do kit, se o limite do kit for 3 dias, então a data máxima vai ser a data escolhida mais 3 dias. Por fim o “to\_date” fica com o valor “ ”, isto é para o caso de já ter sido escolhida uma data no “to\_date” e depois se escolher uma data no “from\_date” que seja depois do valor do “to\_date”.

No caso do “to\_date” o código a inicialização é igual excepto a função que aconteceu quando se escolhe a data.



Quando é selecionada uma data no “to\_date”, é verificado se já foi escolhida uma data no “from\_date” e “to\_date”, se for esse o caso então são definidas duas variáveis em que cada uma delas tem a data do “from\_date” e “to\_date”.

Depois tem um ciclo em que percorre todas as datas do “arrayFromPHP” e verifica se alguma das datas do array está entre a data do “from\_date” e “to\_date”, por exemplo existe uma reserva de 1 a 2 de Setembro e o utilizador está a tentar reservar de 31 de Agosto a 4 de Setembro, se for esse o caso então o utilizador é avisado e o “to\_date” fica com o valor de “ ”.

# Descrição dos ficheiros

## Localização de ficheiros

* **\pa\** – Ficheiros do site;
* **\myslimsite\** – Ficheiros de API;
* **\images\** – Imagens dos itens.

## Páginas do site na pasta \pa\production\

* **dbconnect\_test.php** – Ficheiro onde é guardada a configuração da ligação a base de dados;
* **form\_categoria\_item.php** – Página de criação de categorias de itens;
* **form\_categoria\_kit.php** – Página de criação de categorias de kits;
* **form\_edit\_all\_reservas.php** – Página de edição da reserva de qualquer estado;
* **form\_edit\_atraso\_reservas.php** – Página de edição de reservas com estado “Em atraso”;
* **form\_edit\_categoria\_item.php** – Página de edição de categorias de itens;
* **form\_edit\_categoria\_kit.php** – Página de edição de categorias de kits;
* **form\_edit\_decorrer.php -** Página de edição de reservas com estado “Em progresso” e “Aceite”;
* **form\_edit\_estado.php** - Página de edição de estados;
* **form\_edit\_group.php** - Página de edição de grupos;
* **form\_edit\_item.php** - Página de edição de itens;
* **form\_edit\_kit.php** - Página de edição de kits;
* **form\_edit\_user.php** - Página de edição de utilizadores;
* **form\_estado.php** – Página de criação de estados;
* **form\_grupo.php** – Página de criação de grupos;
* **form\_history\_messages.php** – Página informação de mensagem do histórico;
* **form\_history\_reservas.php** – Página informação de reserva do histórico;
* **form\_info\_user.php** – Página de informação do utilizador;
* **form\_item.php** – Página de criação de itens;
* **form\_kit.php** – Página de criação de kits;
* **form\_myhistory\_reservas.php** – Página de histórico de reservas do utilizador activo;
* **form\_open\_message.php** – Página de informação de mensagem do utilizador activo;
* **form\_reserva.php** – Página de criação de reserva;
* **form\_search\_atraso.php** – Página de pesquisa de reservas com estado “Em atraso” para edição;
* **form\_search\_decorrer.php** – Página de pesquisa de reservas com estado “Em progresso” e “Aceite” para edição;
* **form\_search\_edit\_all\_reservas.php** – Página de pesquisa de todas as reservas para edição;
* **form\_search\_edit\_categoria\_item.php** – Página de pesquisa de categorias de itens para edição;
* **form\_search\_edit\_categoria\_kit.php** – Página de pesquisa de categoria de kits para edição;
* **form\_search\_edit\_estado.php** – Página de pesquisa de estados para edição;
* **form\_search\_edit\_group.php** – Página de pesquisa de grupos para edição;
* **form\_search\_edit\_item.php** – Página de pesquisa de itens para edição;
* **form\_search\_edit\_kit.php** – Página de pesquisa de kits para edição;
* **form\_search\_edit\_user.php** – Página de pesquisa de utilizadores para edição;
* **form\_search\_history\_messages.php** – Página de pesquisa de histórico de mensagens;
* **form\_search\_history\_reservas.php** – Página de pesquisa de histórico de reservas;
* **form\_search\_history\_user.php** – Página de pesquisa de históricos de utilizadores;
* **form\_search\_messages.php** – Página de pesquisa de mensagens do utilizador activo;
* **form\_search\_pendente.php** – Página de pesquisa de reservas com estado “Pendente”;
* **form\_search\_reserva.php** – Página de pesquisa de kits para reservar;
* **form\_search\_send\_messages.php** – Página de pesquisa de utilizadores para mandar mensagem;
* **form\_search\_user.php** – Página de pesquisa de utilizadores;
* **form\_search\_view\_cat\_kit.php** – Página de pesquisa de categorias de kits;
* **form\_search\_view\_kit.php** – Página de pesquisa de kits para ver mais informação;
* **form\_send\_messages.php** – Página de criação de mensagem;
* **form\_sobre.php** – Página de contactos e pdf “Como utilizar”;
* **form\_utilizador.php** – Página de edição do perfil do utilizador activo;
* **form\_view\_item.php** – Página de informação do item;
* **form\_view\_kit.php** – Página de informação do kit
* **index.php** – Homepage;
* **login.php** – Página de login;
* **login\_check.php** – Ficheiro onde verifica se o login é válido;
* **logout.php** – Ficheiro onde efectua o logout;
* **print.php** – Página de impressão com detalhes da reserva;
* **register\_user.php** – Ficheiro onde efectua o registo de utilizador;
* **session.php** – Ficheiro onde são efectuadas as permissões e verifica se existe sessão iniciada;
* **session\_criar\_editar.php** – Ficheiro de verificação da permissão “criar editar”;
* **session\_criar\_msg.php** – Ficheiro de verificação da permissão “criar msg”;
* **session\_reservar.php** – Ficheiro de verificação da permissão “reservar”;
* **session\_reservas.php** – Ficheiro de verificação da permissão “reservas”;
* **session\_user\_editar.php** – Ficheiro de verificação da permissão “user editar”;
* **session\_user\_ver.php** – Ficheiro de verificação da permissão “user ver”;
* **session\_ver.php** – Ficheiro de verificação da permissão “ver”;
* **session\_ver\_admin.php** – Ficheiro de verificação da permissão “ver admin”;
* **session\_ver\_historico.php** – Ficheiro de verificação da permissão “ver historico”;
* **sessionMessages.php** – Ficheiro de verificação de mensagens novas;
* **sessionMessagesVer.php** – Ficheiro de verificação da permissão para ver mensagem actual;
* **sessionReservas.php** – Ficheiro de verificação de quantidade de reservas e verificação de novas reservas “Em atraso”;
* **sessionUpdate.php** – Ficheiro de update de informações ao editar utilizadores;
* **sessionUpdateGroup.php** – Ficheiro de update de permissões ao editar grupos;
* **sessionUpdateProfile.php** – Ficheiro de update de informações ao editar o perfil de utilizador activo;

## Ficheiros dos API na pasta \myslimsite\app\api\

* **dbconnect\_teste.php** – Ficheiro onde é guardada a configuração da ligação a base de dados;
* **formCat.php** – Ficheiro de criação de categorias de itens e de kits;
* **formCatEdit.php** – Ficheiro de edição de categorias de itens e de kits;
* **formEst.php** – Ficheiro de criação de estados;
* **formEstEdit.php** – Ficheiro de edição de estados;
* **formGroup.php** – Ficheiro de criação de grupos;
* **formGroupEdit.php** – Ficheiro de edição de grupos;
* **formItem.php** – Ficheiro de criação de itens e atributos;
* **formItemEdit.php** – Ficheiro de edição de itens e atributos, remoção de atributos;
* **formKit.php** – Ficheiro de criação de kits;
* **formKitEdit.php** – Ficheiro de edição de kits e de remoção e adição dos itens aos kits;
* **formMessageEdit.php** – Ficheiro de alterar campo “lido” das mensagens;
* **formMsg.php** – Ficheiro de criação de mensagens;
* **formResEdit.php** – Ficheiro de edição de reservas;
* **formUserEdit.php** – Ficheiro de edição de utilizadores;
* **reserva.php** – Ficheiro de criação de reservas.

## Ficheiros dos API na pasta \myslimsite\public\

* **índex.php** – Ficheiro onde são incluídos os ficheiros API para serem usados.