Aula prática 1

Docente – Carlos Gonçalves

Tiago Oliveira – 45144

Tiago Gil – 46296

Artur Paredes – 46347

Grupo 2 61D

2.

a. Para validar os dados introduzidos pelo utilizador tendo em consideração igual validação por parte do cliente tal como por parte do servidor foram identificados 3 ficheiros para sofrerem modificações:

- Form.js
- formUpdateProfile.php
- formLogin.php

Para este efeito foram definidas as duas expressões regulares por parte do servidor que quando é criado o JavaScript são enviadas por parte do servidor na sua criação assim mantendo a certeza de que as expressões se encontram iguais e inalteráveis por parte do cliente. Podemos ver nas imagens seguintes a definição em PHP do *regex* e o seu envio para a função de JS onde será usado para a validação.

```
<?php
      $output = "/^[a-zA-Z0-9_.-]*$/";
?>
<form
  enctype="multipart/form-data"
  action="processFormUpdateProfile.php"
  method="POST"
  onsubmit="return FormUpdateProfileValidator(this,<?php echo $output?>)"
  name="FormUpdateProfile">
```

```
// Validate the login form
function FormLoginValidator(theForm, regex) {
  // Check to see if name isn't blank
  if ( theForm.name.value === "" ) {
    alert("You must enter a VALID name.");
    theForm.name.focus();
    return false;
  }

  if (!regex.test( theForm.email.value)) {
    alert('Please provide a valid e-mail address');
    theForm.email.focus();
    return false;
  }

  return true;
}
```

b.

i. No processo de obter os zipCodes para completar o nosso formUpdateProfile tomou-se por base o exemplo que tínhamos perante os Counties sendo assim mudamos depois o SQLQuery e o tipo de informação que estávamos a inserir no array de Results, e por fim no forms.js fomos alterar as referências de atributos para estar de acordo com as referências definidas no getZips.php.

```
$queryString =
    "SELECT `idLocation`, `postalCode`, `postalCodeExtension` FROM `$dataBaseName`.`forms-zips` " .
    "WHERE `idCounty`=$county AND `idDistrict`=$district";

$queryResult = mysqli_query( $GLOBALS['ligacao'], $queryString );

if ( $queryResult ) {
    $result[] = array( 'idLocation'=>0, 'postalCode'=>"" );

    while ($registo = mysqli_fetch_array($queryResult)) {
        $result[] = array(
            'idLocation'=>$registo['idLocation'],
            'postalCode'=>$registo['postalCode'].'-'.$registo['postalCodeExtension']);
    }
}
```

```
var value  = currentZip.idLocation;
var option = currentZip.postalCode;
```

ii. Para o ponto final do trabalho foi criada uma tabela para configurações de utilizadores para que possa ser usada aqui na definição de um max upload size com base na nossa base de dados. Com isso temos o SQL Script usado para a criação dessa tabela e o seu insert em baixo e também por fim o código PHP utilizado para a busca desta informação na tabela que criamos.

```php
<?php
  dbConnect( ConfigFile );
  $dataBaseName = $GLOBALS['configDataBase']->db;
  mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

  $queryString = "SELECT `maxUploadSize` FROM `$dataBaseName`.`user-configs` ".
  "WHERE `idUser`='1'";

  $queryResult = mysqli_query( $GLOBALS['ligacao'], $queryString );

  if($queryResult){
      $registo = mysqli_fetch_array($queryResult);
      $MaxFileSize = $registo['maxUploadSize'];
  } else {
      $MaxFileSize = 64*1024;
  }
?>
```