



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina Programação de Aplicações Web**  
**Profs. Cristina Nader e Daniel de Oliveira**  
**AP2 2º semestre de 2016.**

Nome: \_\_\_\_\_

**Questão 1. (5 pontos)**

Uma universidade mantém no banco de dados os Coeficientes de Rendimento (CR) dos alunos armazenados em uma tabela chamada ALUNO que contém as seguintes colunas: CODIGO (um número inteiro), NOME (uma string) e CR (um número decimal). Escreva em PHP uma mini-aplicação consistindo de três partes:

- a. **(1 ponto)** O programa consulta.php mostra um formulário com um campo, para entrada do nome de um aluno e um botão para submissão. Uma vez que o usuário preenche o formulário e o submete, o programa carrega o programa resposta.php.

```
<form id="form1" method="post" action="resposta.php">  
  
Name: <input type="text" name="name" value="<?php echo $name;?>">  
  
<button type="submit" form="form1" value="Submit">Submit</button>  
  
</form>
```

- b. **(2 pontos)** O programa resposta.php que consulta o banco de dados pelo aluno correspondente ao nome informado e, caso o encontre, exibe seu nome e CR. Caso o aluno não se encontre no banco, exibe a mensagem "não encontrado".

```
<?php  
global $conn;  
$nome = $_POST["name"];  
$sql = "SELECT CR FROM ALUNO WHERE NOME=$nome";  
$result = $conn->query($sql);
```

```

        if ((!$result) || $result->num_rows < 1){
            echo 'Busca PESSOA: Aluno inexistente<br>';
            return -1;
        }
        $row = $result->fetch_all();
        $cr = $row[0][0];
        echo "CR do aluno: ".$_POST["name"].": ".$cr;
    }
}

```

- c. **(2 pontos)** Implemente uma função em PHP que consulte o banco de dados e exiba a média do CR de todos os alunos constantes na base de dados.

```

function mediaDosAlunosNoBanco(){

global $short_connect;

//SQL statement for a table in the database
$sql = "SELECT CR FROM ALUNO;";
$result = $short_connect->query($sql);

if (($result) && ($result->num_rows > 0)){
    $media = 0;
    $cnt = 0;

    //convert query result into an associative array
    while ($row = $result->fetch_assoc())
    {
        $cr = $row['CR'];
        echo $cr.' ';
        $media = $media + $cr;
        $cnt = $cnt+1;
    }
    echo "<BR>".$media/$cnt;

    $result->free();
}

}

```

## Questão 2. (2 pontos)

Descreva a funcionalidade do código a seguir. Sua resposta deve descrever em profundidade o que o código está fazendo, para que serve um determinado trecho do código e, em especial, os trechos sublinhados.

```

<?php
    if(isset($_COOKIE['visitaRecente'])){
        $msg="Ficamosfelizescomseuretorno";
    }
    else{
        $msg="Sejabemvindo";
    }
    setcookie("welcomeBack",true,time()+60*60*24*30);
?>
<html>
<head><title>Site da Prova de Programação Web</title></head>
<body>
<h1><?php echo $msg; ?></h1>
</body>
</html>

```

A intenção do código é verificar a existência de um cookie denominado 'visitaRecente'. Em caso positivo, considera que o visitante voltou ao site após visita recente, em caso contrário o saúda como se fosse a primeira visita.

Nos dois casos, estabelece um novo prazo para a validade da visita atual, criando um cookie com o nome "welcomeBack" de expiração ativada (valor true do segundo parâmetro) e validade como sendo de 60 segundos \* 60 minutos \* 24 horas \* 30 dias a contar da data atual, obtida pela função time. Ou seja, a visita atual passaria a ser a nova marca para visitas recentes sendo válida por 30 dias corridos.

O problema é que na volta o cookie verificado seria o 'visitaRecente', e não o 'welcomeBack', logo, o resultado observado será sempre a msg 'Ficamosfelizescomseuretorno' já que o cookie testado não foi de fato criado.

### Questão 3. (3 pontos)

Você foi contratado para implementar uma aplicação Web que solicita o preenchimento de um formulário com os campos: nome, e-mail, idade e CEP.

Desenvolva um programa em PHP para apresentação do formulário e para validação do mesmo. Caso algum item não seja validado, o formulário deve ser reapresentado com os nomes dos campos incorretos e marcados com um asterisco.

É importante também que, nesse caso, os valores iniciais de todos os campos sejam aqueles originalmente informados pelo usuário.

As regras para validação são:

- Nome deve conter ao menos duas palavras com mais de dois caracteres, sendo que todos os caracteres devem ser alfabéticos (não é necessário testar caracteres acentuados). A primeira letra de cada nome deve ser maiúscula.

- E-mail deve conter exatamente um caractere '@', sendo que o código do usuário deve ser alfanumérico ou conter os caracteres '-', '.' ou '\_'.
- O domínio (parte após o '@') deve conter apenas caracteres alfanuméricos, '-' ou '.', sendo que ao menos um ponto é obrigatório e pontos não podem aparecer nem no início nem no fim do domínio.
- Idade deve ser numérica e superior a 18 (anos).

```

<!DOCTYPE HTML>
<html>
<body>

<?php
    $nomeErr = $emailErr = $idadeErr = $cepErr = "";
    $nome = $email = $idade = $cep = "";

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        if (empty($_POST["nome"])) {
            $nomeErr = "Nome deve ser informado";
        } else {
            $nome = test_input($_POST["nome"]);
            // check if name only contains letters and whitespace
            if (!preg_match("/^[a-zA-Z]{2}[a-zA-Z]*[ ]{1}[a-zA-Z]{2}[a-zA-Z]*$/", $nome)) {
                $nomeErr = "Apenas Letras permitidas, com no minimo dois nomes com
duas letras cada";
            }
        }

        if (empty($_POST["email"])) {
            $emailErr = "Email deve ser informado";
        } else {
            $email = test_input($_POST["email"]);

            if
(!preg_match("/^[a-zA-Z0-9\-. _]+@[a-zA-Z0-9-]{1}[a-zA-Z0-9\-.]*[.]{1}[a-zA-Z0-9-]{
1}$/", $email)) {
                $emailErr = "Email invalido";
            }
        }

        if (empty($_POST["idade"])) {
            $idadeErr = "Idade deve ser informada";
        } else {
            $idade = test_input($_POST["idade"]);
            if ( !preg_match("/^[0-9]*[0-9]$/", $idade) ) {
                $idadeErr = "O campo idade deve conter apenas números";
            }
            else if($idade <18){

```

```

        $idadeErr = "Idade deve ser maior 18 anos";

    }
}

if (empty($_POST["cep"])) {
    $cepErr = "CEP deve ser informada";
} else {
    $cep = test_input($_POST["cep"]);
    if ( !preg_match("/^[0-9]{5}([-][0-9]{3})$/", $cep) ) {
        $cepErr = "Formato esperado para o campo cep: 00000-000";
    }
}

}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

if($nomeErr != "" || $emailErr != "" || $idadeErr != "" || $cepErr != "" || $nome
== "" || $email == "" || $idade == "" || $cep == ""){

    echo ('<form method="post"
action="'.htmlspecialchars($_SERVER["PHP_SELF"]).'>' );

    echo('Nome: <input type="text" name="nome" value="'.$nome.'"><br>');
    if ($nomeErr!="")
        echo ("* ".$nomeErr);
    echo ("<br><br>");

    echo ('e-mail: <input type="text" name="email" value="'.$email.'"><br>');
    if ($emailErr!="")
        echo "* ".$emailErr;
    echo ("<br><br>");

    echo ('Idade: <input type="text" name="idade" value="'.$idade.'"><br>');
    if ($idadeErr!="")
        echo "* ".$idadeErr;
    echo ("<br><br>");

    echo('CEP: <input type="text" name="cep" value="'.$cep.'"><br>');
}

```

```
if ($cepErr!="")
    echo "* ".$cepErr;
echo ("<br><br>");

echo ('<input type="submit"></form>');

}
else{

    echo "<h2>Seus dados:</h2> Nome: ";
    echo $nome;
    echo "<br> E-mail: ";
    echo $email;
    echo "<br> Idade: ";
    echo $idade;
    echo "<br> Cep: ";
    echo cep;
}

?>

</body>
</html>
```