

Q1	2,0	
Q2	2,0	
Q3	4,0	
Q4	2,0	

Fundação CECIERJ – Vice Presidência de Educação Superior à Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação de Aplicações Web

Professores: Flávio L. Seixas e Miguel Elias M. Campista

APX2 – 1º Semestre de 2020

Orientações para a prova:

1. Mostre todos os passos para se chegar a solução encontrada.
2. Você pode: (A) escrever de forma legível as suas respostas em folhas em branco com caneta preta, escaneá-las e convertê-las para PDF, ou (B) escrever as suas respostas no computador usando um editor de texto, e converter o arquivo para PDF.
3. Procure numerar todas as folhas e indicar a questão que a resposta se refere.
4. As respostas serão individuais. Ou seja, não é permitido consultas ao tutor, material da disciplina, websites, ou outra pessoa.
5. O arquivo PDF deve ser anexado junto a entrega da tarefa APX2.
6. O prazo de entrega da tarefa é dia 06/06, até às 12h (horário de Brasília).

Questão 1. Para o desenvolvimento de uma aplicação de matemática básica, deseja-se apresentar um formulário para estudo de operações de soma e subtração. Para isso, dois operandos inteiros variando entre 1 e 9 e a operação + ou - são apresentados, cabendo ao usuário inserir o resultado no único campo do formulário disponível para preenchimento. Por padrão, o campo do formulário para inserção do resultado é inicializado com ?

Assumindo que o valor do primeiro e segundo operandos se encontram, respectivamente, nas variáveis \$operando1 e \$operando2, e que a operação se encontra na variável \$operacao, escreva em PHP/HTML o trecho de código para exibição do formulário.

Saiba que o seu código deve prever que a submissão do formulário é encaminhada a um *script* chamado `matematica.php`. Além disso, o código deve prever que a função `rand(min,max)` é usada para inicialização dos operandos (`$operando1` e `$operando2`) e da operação (`$operacao`). A função `rand(min,max)` retorna um inteiro aleatório pertencente ao intervalo `[min,max]`. A variável `$operacao` também pode ser definida no intervalo `[0,1]`, onde 0 e 1 denotam a operação de soma e subtração, respectivamente. Lembre-se que caso a operação seja de soma, o caractere + deve ser exibido para o usuário, assim como o caractere -, em caso de subtração.

Questão 2. Suponha que uma tabela `Exercicios` em MySQL contenha 05 colunas: `operando1`, `operando2`, `operacao`, `resultado` e `flag`. Escreva um *script* PHP/MySQL capaz de inserir elementos na tabela `Exercicios` levando em conta que a referida `flag` é inicializada com `TRUE`, caso o resultado esteja correto, e `FALSE`, caso contrário.

Questão 3. Deseja-se montar um serviço para armazenar estatísticas relacionadas ao desempenho dos alunos que utilizam uma aplicação matemática como a descrita de forma simplificada nas **questões 1 e 2**. Para isso, os alunos devem se inscrever para executar uma série de operações, cuja quantidade é limitada por usuário a no máximo 10 operações. Todas as operações realizadas por cada usuário, incluindo os dois operadores, operação, resultado e flag devem ser armazenados no banco de dados. Apenas as operações realizadas, aquelas em que o usuário digitou no resultado algo diferente do valor padrão ? são contabilizadas. O processo de inscrição requer nome, idade, CPF e a inicialização com zero de uma variável contadora a ser usada para armazenar o número de operações executadas por cada usuário. Dessa forma, pede-se:

1. Desenhe um diagrama de entidades e relacionamentos para este banco de dados.
2. Escreva uma modelagem física do banco de dados em MySQL.
3. Escreva uma consulta em MySQL que retorne o desempenho, ou seja, o número de operações respondidas corretamente sobre o número total de operações respondidas, de um determinado usuário.
4. Escreva uma consulta em MySQL que retorne todos os usuários inscritos que obtiveram um desempenho superior ao de um dado usuário em particular, informado usando parâmetro na consulta SQL.

Questão 4. Diga se é verdadeiro ou falso, justificando a resposta caso seja assinalado falso. Elabore a justificativa.

- () Em PHP, o método POST permite obter múltiplos valores passados no endereço URL do navegador (ex. `http://www.exemplo.com/index.php&a=5&b=6`).
- () Em PHP, só é possível fazer passagem de parâmetro por valor para funções.
- () Em SQL, caso dois registros tenham o mesmo valor para uma dada propriedade, não há como retornar de uma consulta apenas um dos registros.
- () Em SQL, a cláusula SELECT é usada para selecionar e alterar dados.

Tecnologia em sistemas de computação

Aluno: Glauber de Souza Faria

Matricula: 17213050160

Polo: Angra dos Reis - RJ

AP2X – Programação de Aplicações Web

1)

```
<!--ARQUIVO INDEX.PHP -->
<?php
    include("matematica.php");
    $operando1 = rand(1,9);
    $operando2 = rand(1,9);
    $operacao = rand(0,1);

    function convertSinal($operacao){
        if ($operacao == 0){
            return "+";
        }
        else{
            return "-";
        }
    }
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Q1</title>
</head>
<body>
    <form action="matematica.php" method="post">
        <?php
            echo("<input type='text' name='operando1' id='operando1' size
='1' readonly value='". $operando1. "'>");
            echo("<input type='text' name='operacao' id='operacao' size='
1' readonly value='". convertSinal($operacao). "'>");
            echo("<input type='text' name='operando2' id='operando2' size
='1' readonly value='". $operando2. "'>");
            echo("<input type='text' name='=' id='=' size='1' readonly va
lue='='>");
```

```

        echo("<input type='text' name='result' id='result' size='1' value='?'>");
    ?>
    <button type="submit" name="submit">Responder</button>
</form>
</body>
</html>

```

```

<!--ARQUIVO MATEMATICA.PHP -->

```

```

<?php
function adicao($a,$b){
    return ($a + $b);
}
function subtracao($a,$b){
    return ($a - $b);
}
function resultado($operacao, $operando1, $operando2){
    if($operacao == "+"){
        return adicao($operando1, $operando2);
    }
    else{
        return subtracao($operando1, $operando2);
    }
}

if(isset($_POST["submit"])){
    $operacao = $_POST["operacao"];
    $operando1 = $_POST["operando1"];
    $operando2 = $_POST["operando2"];
    $result = $_POST["result"];
    if($result == resultado($operacao, $operando1, $operando2)){
        echo("Resposta Correta");
    }
    else{
        echo("Resposta Errada!!! <br/>");
        echo("Resposta Certa é: ". resultado($operacao, $operando1, $operando2));
    }
}
?>

```

2)

```

<?php
    $servername = "localhost";
    $username = "root";
    $password = null;
    $db = "apx2";

    $conexao = new mysqli($servername, $username, $password, $db);
    $insertExercicios = ("INSERT INTO 'tbl_exercicios' ('id', 'operando1'
, 'operando2', 'operacao', 'resultado', 'flag') VALUES (NULL, '$operando1'
, '$operando2', '$operacao', '$resultado', '$flag')");

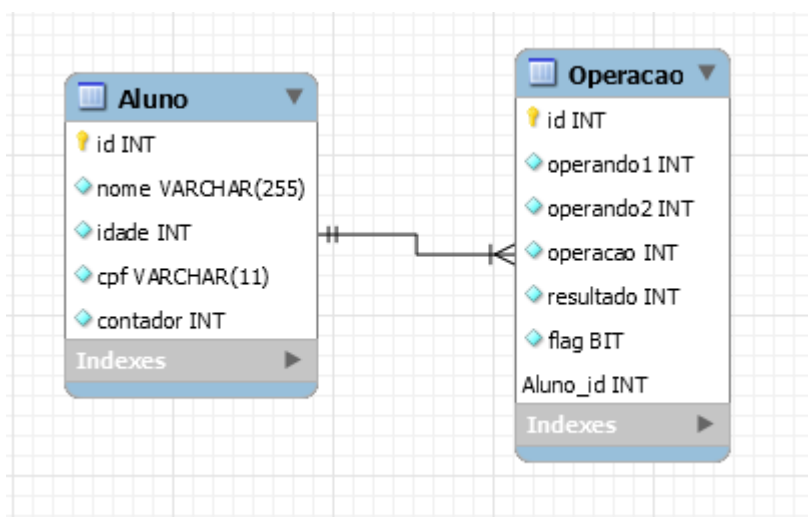
    $query = mysqli_query($conexao,$insertExercicios);

    if(mysqli_query($conexao,$insertExercicios)){
        //inserido
        echo("Inserido com sucesso !!!");
    }
    else{
        //erro
        echo("Erro ao inserir registro". mysqli_error($conexao));
    }
    mysqli_close($conexao);
?>

```

3)

i)



ii)

```

CREATE TABLE Aluno (
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,

```

```

nome VARCHAR(255) NOT NULL,
idade INT NOT NULL,
cpf VARCHAR(11) NOT NULL,
contador INT NOT NULL
);

```

```

CREATE TABLE Operacao (
id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
operando1 INT NOT NULL,
operando2 INT NOT NULL,
operacao INT NOT NULL,
resultado INT NOT NULL,
flag BIT NOT NULL,
id_Aluno INT NOT NULL,
FOREIGN KEY(id_Aluno) REFERENCES Aluno(id)
);

```

iii)

```

SELECT aluno.id, aluno.nome, aluno.cpf, (Count(operacao.flag)/aluno.contador) AS
Desempenho FROM aluno INNER JOIN operacao ON aluno.id = operacao.id_aluno GROUP BY
aluno.id, aluno.nome, aluno.cpf, aluno.contador, operacao.flag HAVING (((operacao.flag)=1))

```

iv)

```

SELECT aluno.id, aluno.nome, aluno.cpf, aluno.contador, operacao.flag, Count(operacao.flag)
AS ContarFlagTrue, (Count(operacao.flag)/aluno.contador) AS Desempenho FROM aluno
INNER JOIN operacao ON aluno.id = operacao.id_aluno GROUP BY aluno.id, aluno.nome,
aluno.cpf, aluno.contador, operacao.flag HAVING (((operacao.flag)=1) AND (aluno.id>1))

```

4)

False. O método utilizado para passagem de valores por endereço é chamado de GET e não POST.

False. Também é possível passarmos o parâmetro por referência.

False. Se tivermos dois registros com o mesmo valor para a mesma propriedade, poderemos selecionar apenas um deles através de uma condição relacionada a outro atributo, exemplo:

Tbl_Pessoa		
Id	Nome	Cpf
1	Glauber de S. Faria	16012485846
2	Glauber de S. Faria	13255548789
3	Glauber de S. Faria	11122200011

Logo teremos 3 pessoas com o mesmo nome porém seu CPF e ID são distintos, para selecionarmos um único basta dizermos através do WHERE:

//TEREMOS OS 3 REGISTROS:

```
SELECT * FROM Tbl_Pessoa WHERE (Nome="Glauber de S. Faria");
```

//TEREMOS APENAS 1 REGISTRO:

```
SELECT * FROM Tbl_Pessoa WHERE (Nome="Glauber de S. Faria" and Id = 1) ;
```

```
SELECT * FROM Tbl_Pessoa WHERE (Nome="Glauber de S. Faria" and Cpf =  
"16012485846") ;
```

False. A cláusula SELECT é utilizada para selecionar registros nas tabelas de um banco de dados, já a cláusula UPDATE é responsável pela alteração de registros no banco de dados.