



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância  
Curso de Tecnologia em Sistemas de Computação – UFF

Disciplina: PROGRAMAÇÃO DE APLICAÇÕES WEB  
AD1 - 1º semestre de 2017

Nome: Marcus Lizander dos Santos Lugão - Matr.: 15213050325

## AVALIAÇÃO À DISTÂNCIA 1

### 1ª Questão (1,0 ponto)

Responda sucintamente às seguintes questões (atenção: use suas próprias palavras! Cópias da internet não serão consideradas!):

a. Por que PHP é considerada uma linguagem "fracamente tipada"?

**Resposta:**

*Porque em php não necessitamos de declarar uma variável juntamente com o seu tipo (string, int, float,...), além de que, em php, uma variável que num momento é de um tipo (uma string, por exemplo), pode passar a ser de outro tipo (uma variável inteira, por exemplo), apenas com um comando de atribuição, o que em uma linguagem tipada geraria um erro.*

b. Dentre as características apresentadas da linguagem PHP, quais você considera as mais importantes? Por que?

**Resposta:**

*Considero como a característica mais importante a biblioteca pré-definida para interação com diversos gerenciadores de bancos de dados, uma vez que em sistemas feitos para rodar na web, invariavelmente iremos necessitar de manipular dados em bancos de dados. Dada a variedade de gerenciadores de bancos de dados disponíveis no mercado, quanto mais compatível for a linguagem e de fácil interação com estes, melhor será.*

*Outra característica importante, em minha opinião, é o fato da linguagem ser Server-side, onde o código fonte da aplicação fica restrito ao Servidor. O cliente apenas recebe um HTML já processado, não podendo, portanto, visualizar o código fonte que gerou a página.*

### 2ª Questão (3,0 pontos)

Considere um tabuleiro de um jogo chamado "ache a soma". Esse jogo consiste em encontrar sequências de números na horizontal ou vertical que somados sejam iguais a um valor pré-escolhido. Implemente na linguagem PHP uma função chamada `acheSequencia($tabuleiro,$soma)` que permita encontrar as sequências de números que somados sejam iguais a \$soma. Nesta função, \$tabuleiro representa um array bidimensional de números. O resultado devem ser exibir na tela todas as sequências de números encontradas com a soma desejada, separando as verticais das horizontais. A seguir apresentamos um exemplo de tabuleiro e o resultado para valor de soma 9.

1	3	6	7	2	5
4	4	6	2	2	4
2	3	5	1	1	5
5	3	5	1	5	7
6	2	4	2	1	3
3	2	3	1	7	4

Exemplo de Saída:

Horizontal: 3,6 - 7,2 - 5,4 - 3,5,1 - 3,2,3,1

Vertical: 6,3 - 5,4 - 7,2 - 4,5

**Resposta:**

*Foi considerada a premissa de que cada linha subsequente do tabuleiro é continuação da linha anterior, para análise da sequência horizontal, assim como de que cada coluna subsequente é continuação da coluna anterior, para análise da sequência vertical, conforme conversado em sala de tutoria.*

```

<?php
Function acheSequencia($tabuleiro,$soma){
    $horizontal = "";
    $vertical = "";
    $linhas = [];
    $colunas = [];

    //Preenchendo Sequencias Horizontais
    for($li = 0; $li < sizeof($tabuleiro); $li++){
        for ($col = 0; $col < sizeof($tabuleiro[$li]); $col++){
            $linhas[] = $tabuleiro[$li][$col];
        }
    }

    //Preenchendo Sequencias Verticais
    for($col = 0; $col < sizeof($tabuleiro[0]); $col++){
        for ($li = 0; $li < sizeof($tabuleiro); $li++){
            $colunas[] = $tabuleiro[$li][$col];
        }
    }

    //Fazendo a verificação Horizontal
    for($li = 0; $li < sizeof($linhas); $li++){
        $total = 0;
        $ind = $li;
        $seq = "";
        while ($total < $soma and $ind < sizeof($linhas)){
            $total += $linhas[$ind];

            if ($seq<>""){
                $seq .= ",";
            }
            $seq .= $linhas[$ind];
            $ind++;
        }
        if ($total == $soma) {
            //Verifica se já existe a sequencia e caso não exista, adiciona
            if (strpos($horizontal, $seq) === false) {
                if ($horizontal <> "") {
                    $horizontal .= " - ";
                }
                $horizontal .= $seq;
            }
        }
    }

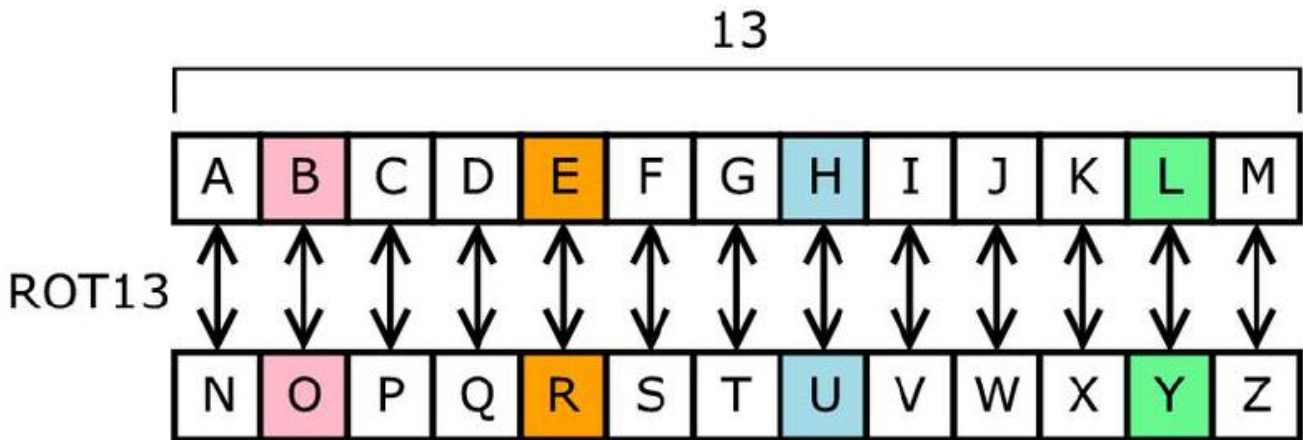
    //Fazendo a verificação Vertical
    for($col = 0; $col < sizeof($colunas); $col++){
        $total = 0;
        $ind = $col;
        $seq = "";
        while ($total < $soma and $ind < sizeof($colunas)){
            $total += $colunas[$ind];
            if ($seq<>""){
                $seq .= ",";
            }
            $seq .= $colunas[$ind];
            $ind++;
        }
        if ($total == $soma) {
            //Verifica se já existe a sequencia e caso não exista, adiciona
            if (strpos($vertical, $seq) === false) {
                if ($vertical <> "") {
                    $vertical .= " - ";
                }
                $vertical .= $seq;
            }
        }
    }

    return "Horizontal: ". $horizontal . "<br/>". "Vertical: ".$vertical;
}
?>

```

### 3ª Questão (2,0 pontos)

Uma tarefa muito importante em sistemas de informação atualmente é a segurança do sistema. Em geral, a segurança envolve criptografia, protocolos de segurança, etc. Vamos implementar nesse exercício um algoritmo de criptografia bem simples, chamado de substituição simples. Uma substituição simples pode ser expressa escrevendo o alfabeto em uma ordem diferente, que se designa **alfabeto de substituição**. Em geral, ele é deslocado de um passo fixo ou embaralhado de forma mais complexa. Nesse exercício utilizaremos a cifra ROT13, conforme apresentado a seguir, onde cada letra é substituída por sua "equivalente":



Implemente uma função em PHP chamada `cifraTexto($texto)` que recebe um texto como parâmetro de entrada e gera como saída o texto cifrado de acordo com a substituição de caracteres.

Por exemplo, se o texto "Meu nome é Daniel" fosse dado como entrada, a saída gerada seria "ZRH ABZR é QNAVRY".

**Resposta:**

```
<?php
function cifraTexto($texto){
    $trocaLetras = array ("A"=>"N", "B"=>"O", "C"=>"P", "D"=>"Q", "E"=>"R", "F"=>"S",
        "G"=>"T", "H"=>"U", "I"=>"V", "J"=>"W", "K"=>"X", "L"=>"Y",
        "M"=>"Z", "N"=>"A", "O"=>"B", "P"=>"C", "Q"=>"D", "R"=>"E",
        "S"=>"F", "T"=>"G", "U"=>"H", "V"=>"I", "W"=>"J", "X"=>"K",
        "Y"=>"L", "Z"=>"M");

    $textoCifrado = "";

    for ($letra = 0; $letra < strlen($texto); $letra++){
        if (!empty($trocaLetras[strtoupper(substr($texto, $letra, 1))])) {
            $textoCifrado .= $trocaLetras[strtoupper(substr($texto, $letra, 1))];
        } else {
            $textoCifrado .= substr($texto, $letra, 1);
        }
    }
    return $textoCifrado;
}

?>
```

### 4ª Questão (4,0 pontos)

O sítio Hortiflix (<http://hortiflix.com.br/>) foi criado como publicidade para um famoso mercado do Rio de Janeiro. Nesse sítio nomes de filmes famosos são reformulados de forma que façam referência a alguma fruta, legume ou verdura (produto vendido pelo mercado). Por exemplo, "O Poderoso Melão" em referência ao "O Poderoso Chefão" e os "Milhons" em referência aos "Minions". Como nem todas as pessoas conseguem entender de primeira a qual filme a propaganda se refere resolvemos pedir sua ajuda nessa tarefa. Escreva uma função em PHP chamada `comparacaoFilmes` que recebe dois arrays de strings contendo títulos de filmes. Cada array pode conter quantidades diferentes de strings. O primeiro array recebe o nome dos filmes originais e o segundo as versões do sítio Hortiflix. Sua função deve analisar o quão parecidas são as letras contidas em posições correspondentes para cada par de títulos de filmes fornecidos (um de cada array) e calcular um valor total que indica o índice de similaridade entre os títulos. É importante ressaltar que os títulos dos filmes nem sempre têm o mesmo comprimento ( strings de mesmo tamanho). Ao fim, sua função deve apresentar os 3 títulos (original e modificado) que mais se assemelham, ou seja, que tem os maiores índices de similaridade. Utilize a regra de pontuação a seguir para comparar cada letra do primeiro título com a letra correspondente no segundo título.

1. Letras Iguais – 10 pontos

2. Mesma letra, mas em um título ela é maiúscula e no outro minúscula – 5 pontos

3. Letras diferentes – 2 pontos

4. Letra em uma string e número ou outro símbolo na outra – 0

Exemplo:

Comparação entre "Minions" e "Milhons") -> calcula 10+10+2+2+10+10+10 = 54

**Resposta:**

*Para o desenvolvimento dessa função, foi necessário escrever funções auxiliares (que são chamadas de dentro da função comparacaoFilmes, a saber:*

- A função ehLetra, que recebe um caractere e retorna um valor booleano, que representa se o caractere é uma letra ou não;*
- A função pontuacaoTitulo, que compara duas strings, pontuando sua similaridade, retornando um inteiro.*

```
<?php
Function ehLetra($char) {
    $espectro = "ABCDEFGHIJKLMNOPQRSTUVWXYZÇÑÀÁÂÃÄÅÆÈÉÊËÌÍÎÏÐÓÔÕÖÙÚÛÜÝ";
    $espectro .= "abcdefghijklmnopqrstuvwxyzçñàáâãäåæèéêëìíîïðóôõöùúûüýÿ";

    if ( strpos($espectro, $char) === false ) {
        return false;
    } else {
        return true;
    }
}

Function pontuacaoTitulo($nomeFilme, $nomeFake){
    $totalPontos = 0;
    if (strlen($nomeFilme) > strlen($nomeFake)) {
        $tamanhoMaximo = strlen($nomeFilme);
    } else {
        $tamanhoMaximo = strlen($nomeFake);
    }

    for ($letra = 0; $letra < $tamanhoMaximo; $letra++){
        if (ehLetra(substr($nomeFilme,$letra,1)) && ehLetra(substr($nomeFake,$letra,1))) {
            if (substr($nomeFilme,$letra,1) == substr($nomeFake,$letra,1)){
                $totalPontos += 10;
            } else if (strtoupper(substr($nomeFilme,$letra,1)) ==
                strtoupper(substr($nomeFake,$letra,1))){
                $totalPontos += 5;
            } else {
                $totalPontos += 2;
            }
        }
    }
    return $totalPontos;
}

Function comparacaoFilmes($nomesFilmes,$nomesFakes){
    $compara = [];

    //Montando o array de comparações
    for($filme = 0; $filme < sizeof($nomesFilmes); $filme++){
        for ($fake = 0; $fake < sizeof($nomesFakes); $fake++){
            $compara[] = array($nomesFilmes[$filme],$nomesFakes[$fake],0);
        }
    }

    //Percorre o array de comparações, comparando todos os filmes com os fakes,
    //atribuindo uma pontuação
    for($li = 0; $li < sizeof($compara); $li++){
        $compara[$li][2] = pontuacaoTitulo($compara[$li][0], $compara[$li][1]);
    }

    //Compara os maiores, preenchendo um array com a classificação final
    $classificacao = array(array("", "", 0), array("", "", 0), array("", "", 0));
    for($li = 0; $li < sizeof($compara); $li++){
        if ($compara[$li][2] >= $classificacao[0][2]) {
            $classificacao[2] = $compara[$li][2];
            $classificacao[1] = $compara[$li][1];
            $classificacao[0][0] = $compara[$li][0];
            $classificacao[0][1] = $compara[$li][1];
            $classificacao[0][2] = $compara[$li][2];
        } elseif ($compara[$li][2] >= $classificacao[1][2]) {
```

```

        $classificacao[2] = $classificacao[1];
        $classificacao[1][0] = $compara[$li][0];
        $classificacao[1][1] = $compara[$li][1];
        $classificacao[1][2] = $compara[$li][2];
    } elseif ($compara[$li][2] >= $classificacao[2][2]) {
        $classificacao[2][0] = $compara[$li][0];
        $classificacao[2][1] = $compara[$li][1];
        $classificacao[2][2] = $compara[$li][2];
    }
}

$retorno = "";
for($li = 0; $li < sizeof($classificacao); $li++){
    $retorno .= ($li + 1) . "o. Lugar = ". $classificacao[$li][0] . " vs " .
        $classificacao[$li][1] . " => " . $classificacao[$li][2] . " pontos <br/>" ;
}

return $retorno;
}

?>

```