



**Fundação CECIERJ – Vice Presidência de Educação Superior à Distância**  
**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Programação II**  
**Gabarito AD2 – 1º Semestre de 2013**

**Questão 1**

Uma das formas pelas quais se pode escrever letras utilizando-se as teclas numéricas de um celular é pressionando-as repetidas vezes até que a letra correspondente seja exibida. Os fabricantes geralmente usam a seguinte associação de teclas numéricas com letras:

Tecla numérica	1x	2x	3x	4x
2	a	b	c	
3	d	e	f	
4	g	h	i	
5	j	k	l	
6	m	n	o	
7	p	q	r	s
8	t	u	v	
9	w	x	y	z

Por exemplo, para se obter a letra **b**, deve-se pressionar duas vezes a tecla **2**; para a letra **m**, uma vez tecla **6** e assim por diante.

Escreva uma função em PHP chamada **ajuda\_digitacao** para auxiliar usuários com dificuldade neste sistema de digitação. Sua função deve receber como parâmetro de entrada uma palavra P e deve exibir na ordem correta as informações de quantas vezes e quais teclas terão que ser pressionadas para obter a palavra.

A saída deve obedecer a seguinte formatação (veja os exemplos abaixo): cada linha deve apresentar uma **trilha (#)**, a **tecla numérica**, um **sinal de igual (=)** e a **quantidade de vezes a ser pressionada**.

## Exemplo 1

### Entrada

internet

### Saída

```
#4=3
#6=2
#8=1
#3=2
#7=3
#6=2
#3=2
#8=1
```

## Exemplo 2

### Entrada

preconceber

### Saída

```
#7=1
#7=3
#3=2
#2=3
#6=3
#6=2
#2=3
#3=2
#2=2
#3=2
#7=3
```

## Gabarito:

```
1 <?php
2 function ajuda_digitacao($palavra) {
3     /*
4      * Separa as letras em cada linha da matriz de acordo com o dígito que gera a letra;
5      *
6      * O número da coluna representa quantas vezes o dígito precisa ser pressionado para
7      * gerar a letra.
8      */
9     $matriz = array();
10    $matriz[] = array('a', 'b', 'c'); //Número 2
11    $matriz[] = array('d', 'e', 'f'); //Número 3
12    $matriz[] = array('g', 'h', 'i'); //Número 4
13    $matriz[] = array('j', 'k', 'l'); //Número 5
14    $matriz[] = array('m', 'n', 'o'); //Número 6
15    $matriz[] = array('p', 'q', 'r', 's'); //Número 7
16    $matriz[] = array('t', 'u', 'v'); //Número 8
17    $matriz[] = array('w', 'x', 'y', 'z'); //Número 9
18
19    $encontrado = false;
20
21    //Para cada letra da palavra
22    for($i = 0; $i < strlen($palavra); $i++) {
23        //Procura em qual linha da matriz a letra se encontra
24        for($j = 0; $j < count($matriz); $j++) {
25            if($encontrado) $encontrado = false;
26            //Verifica cada coluna da linha atual da matriz
27            for($k = 0; $k < count($matriz[$j]); $k++) {
28                //Se a letra for encontrada...
29                if($matriz[$j][$k] == $palavra[$i]) {
30                    /*
31                     * Será impresso o número da linha acrescido de duas posições. Assim,
32                     * se uma letra for encontrada na posição 0, o programa imprime 2, se for
33                     * 1, o programa imprime 3, e assim por diante.
34                     *
35                     * O número da coluna será acrescido de uma unidade, visto que a coluna 0
36                     * refere-se ao número sendo pressionado 1 vez, e assim por diante.
37                     */
38                    echo "#".($j+2).".".$k+1)."<br />";
39                    //Se a letra for encontrada, não é necessário continuar procurando.
40                    $encontrado = true;
41                    break;
42                }
43            }
44            if($encontrado) break;
45        }
46    }
47 }
48 ?>
```

## Questão 2

Em sistemas Web, o Login é uma funcionalidade importante, pois, entre outras coisas, ajuda a regular o acesso a informações que podem ser sigilosas.

Um sistema em PHP é baseado em chamadas de arquivos com a terminação **.php** dentro de um servidor de aplicação que dê suporte à linguagem. Uma possível lógica de execução seria:



Suponha que um usuário mal intencionado tente burlar o login deste sistema digitando diretamente na barra de endereço do navegador a URL correspondente ao arquivo 'dadosSigilosos.php'. Explique, **a nível de programação**, o que pode ser feito para evitar que este usuário tenha acesso a esta página sem a devida autenticação.

## Gabarito

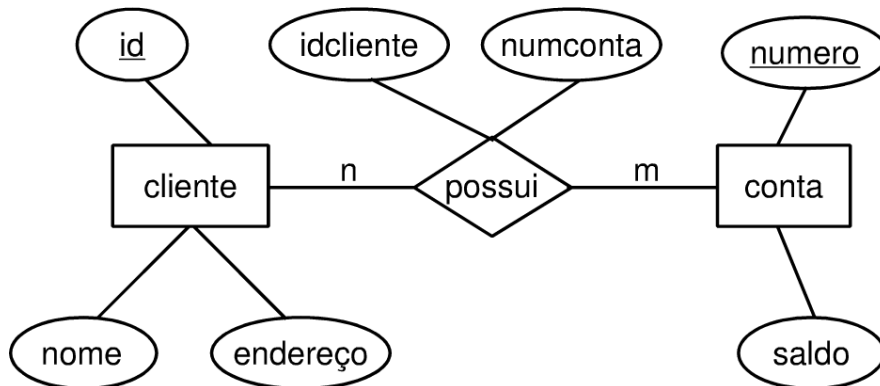
Uma possível solução para ajudar na segurança do sistema é colocar uma verificação das variáveis `$_SESSION['usuario']` e `$_SESSION['senha']`. A cada nova página que exigir autenticação, o par 'usuario' e 'senha' devem ser verificados junto ao banco de dados. Caso o usuário tente acessar a página sem ter feito login, as duas variáveis não estarão setadas ou estarão vazias. Isso deve ser verificado através da função `IS_SET()` e da comparação do conteúdo das `$_SESSION` com `null`.

O sistema não deve deixar que o usuário navegue nas páginas caso isso aconteça. Ele pode ser redirecionado para a página de login ou para alguma página que exiba o aviso de falta de autorização. É importante lembrar que as variáveis de sessão (`$_SESSION`) só serão válidas se, a cada página, a função `session_start()` for chamada.

Ao deslogar, o sistema deve setar o conteúdo dessas variáveis como nulo, ou utilizar a função `UNSET()`.

### Questão 3

Considere o Diagrama Entidade-Relacionamento a seguir, representado pela notação de Peter Chen:



Deseja-se programar um sistema que irá trabalhar com base nas tabelas geradas a partir desse diagrama. Para simplificar sua programação, a equipe de desenvolvimento decidiu que uma classe deve ser escrita a fim de organizar as operações que são feitas em conjunto com o Sistema Gerenciador de Banco de Dados.

Em cada item desta questão você deve completar o código correspondente a um dos métodos desta classe obedecendo ao nome do método e ao conjunto de parâmetros apresentados. Para cada método, escreva as operações que são realizadas dentro dele.

Para simplificar o problema, considere que todos os dados passados nos parâmetros estão escritos corretamente e que não é necessário validar o conteúdo das variáveis. Considere, também, que não existem conexões simultâneas no SGBD.

a) conectar(\$ip, \$porta, \$schema, \$usuario, \$senha) { }

b) inserirUsuario(\$nome, \$endereco) { }

c) inserirConta(\$numero, \$saldo) { }

d) removerUsuario(\$id) { }

e) alterarSaldo(\$numero, \$saldo){ }

f) associarConta(\$id, \$numero) { }

g) recuperarContasDeUsuario(\$id) { }

h) desconectar() { }

## Gabarito

```
1  <?php
2  class banco {
3      $conn = null;
4      //item a)
5      public conectar($ip, $porta, $schema, $usuario, $senha) {
6          $conn = mysql_connect($ip, $porta, $usuario, $senha);
7          mysql_select_db($schema, $conn);
8      }
9      //item b)
10     public inserirUsuario($nome, $endereco) {
11         $query = "INSERT INTO cliente VALUES ('$nome', '$endereco')";
12         mysql_query($query, $conn);
13     }
14     //item c)
15     public inserirConta($numero, $saldo) {
16         $query = "INSERT INTO conta VALUES ($numero, $saldo)";
17         mysql_query($query, $conn);
18     }
19     //item d)
20     public removerUsuario($id) {
21         $query = "DELETE FROM cliente WHERE id = $id";
22         mysql_query($query, $conn);
23     }
24     //item e)
25     public alterarSaldo($numero, $saldo){
26         $query = "UPDATE cliente SET saldo = $saldo WHERE numero = $numero";
27         mysql_query($query, $conn);
28     }
29     //item f)
30     public associarConta($id, $numero) {
31         $query = "INSERT INTO possui VALUES ($id, $numero)";
32         mysql_query($query, $conn);
33     }
34     //item g)
35     public recuperarContasDeUsuario($id) {
36         $query = "SELECT numconta FROM possui WHERE idcliente = $id";
37         $resultSet = mysql_query($query, $conn);
38
39         return $resultSet;
40     }
41     //item h)
42     public desconectar() {
43         mysql_close($conn);
44     }
45 }
46 ?>
```