

Gabarito da AD1 de Programação II

Segundo Período

1. Responda sucintamente às seguintes perguntas:

- a. O que significa PHP ser uma linguagem “fracamente tipada”?

Resp.

A conversão entre tipos em geral é muitas vezes feita implicitamente

- b. Explique o funcionamento de uma arquitetura de 3 camadas.

Resp. A aplicação é dividida em uma camada de apresentação com a qual o usuário interage, uma camada de lógica onde os algoritmos são processados e uma camada de dados responsável pelo armazenamento e recuperação de dados.

- c. Enumere as características mais importantes da linguagem PHP.

Resp.

Código aberto

Multiplataforma

Acesso a banco de dados

Orientada a objetos

2. Exercícios de programação em PHP:

- a. Construa a função `cod($s)` que codifica uma string `s` contendo apenas caracteres alfabéticos de tal maneira que seqüências com n caracteres idênticos iguais a x (para $n > 1$) são substituídos por nx . Assim, por exemplo, `cod("aaabccddddefg")` retorna `"3ab2c4defg"`.

Resp.

```
function simplecrypt($str){
    $len = strlen($str);
    $i = 0;
    $result = "";
    while ($i < $len ){
        $char = $str[$i];
        $count = 1;
        while ($str[++$i] == $char)
            $count++;
        $result .= $count.$char;
    }
    return $result;
}
```

- b. Usando expressões regulares, escreva a função `substitui($s,$a,$b)` que permita substituir por `$b` palavras iguais a `$a` num texto `$s`. Sua função deve ter o cuidado de se assegurar que o primeiro caractere do texto e o primeiro caractere depois de cada ponto (“.”) é maiúsculo. Por exemplo, o código

```
$texto = TEXT<<<
```

Operadores internacionais disseram que a companhia nunca se recuperou e estimam uma queda na produção. Alguns operadores

```
também indicaram que a companhia deve ser vendida em breve.
TEXT;
$texto = substitui($texto, "operadores", "observadores");
echo $texto;
```

gera o seguinte resultado:

Observadores internacionais disseram que a companhia nunca se recuperou e estimam uma queda na produção. Alguns observadores indicaram também que a companhia deve ser vendida em breve.

Resp.

```
function substitui($s, $a, $b){
    // para palavras que começam em maiusculo
    $padr = ucfirst($a);
    $subs = ucfirst($b);
    $s = preg_replace("/$padr/", $subs, $s);

    // para palavras que começam em minusculo
    $padr = strtolower($a);
    $subs = strtolower($b);
    return preg_replace("/$padr/", $subs, $s);
}
```

- c. Usando loops, escreva a função `primosMenores($n)` que retorna uma lista de números primos menores que n .

Resp.

```
function primosMenores($n){
    for($i = 1; $i < $n; $i++){
        $a = false;
        for ($j = 2; $j < $i; $j++){
            if (!($i%$j)){
                $a = true;
                break;
            }
        }
        if ($a) continue;
        echo $i."\n";
    }
}
```

- d. A função `classificar($array)` recebe como parâmetro um array contendo dados de diferentes tipos e os classifica pelo tipo de dado (Dica: use a função `gettype`). O resultado retornado é um array onde cada elemento tem como chave o nome do tipo e o conteúdo é um array contendo elementos daquele tipo. Por exemplo:

```
$array = Array(5, 'c', "xpto", 5.5, 10, 'a', 15, true);
$dados = classificar($array);
```

```
print_r($dados);
```

gera o seguinte resultado:

```
Array(
    [integer] => Array (
        [0] => 5
        [1] => 10
        [2] => 15
    )

    [string] => Array (
        [0] => c
        [1] => xpto
        [2] => a
    )

    [double] => Array (
        [0] => 5.5
    )

    [boolean] => Array (
        [0] => 1
    )
)
```

Resp.

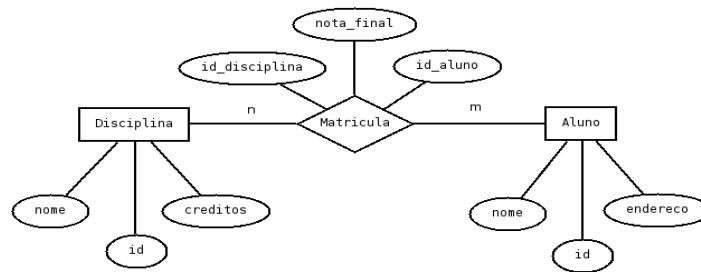
```
function clasificar($array){
    $clasificado = array();
    foreach ($array as $val){
        switch(gettype($val)){
            case 'integer': $clasificado['integer'][] = $val; break;
            case 'string': $clasificado['string'][] = $val; break;
            case 'double': $clasificado['double'][] = $val; break;
            case 'boolean': $clasificado['boolean'][] = $val; break;
            case 'array': $clasificado['array'][] = $val; break;
            case 'object': $clasificado['object'][] = $val; break;
        }
    }
    return $clasificado;
}
```

3. Exercícios com MySQL:

- a. Crie a modelagem completa de um banco de dados para um sistema de matrículas em uma faculdade. O sistema de matrículas permite que um aluno possa estar inscrito em várias disciplinas e em cada disciplina vários alunos podem se inscrever. Cada aluno é caracterizado por um número de matrícula e um nome. Cada disciplina é caracterizada por um nome e um código alfanumérico de 6 caracteres. Se um aluno está matriculado na disciplina, sua modelagem deve permitir associar uma nota final (número decimal entre 0 e 10). Pede-se:

i. Diagrama ER

Resp.



ii. Modelo físico MySQL

Resp.

```
create table disciplina(  
  id int not null auto_increment,  
  nome varchar(20),  
  credits tinyint,  
  primary key(id)  
);  
create table aluno(  
  id int not null auto_increment,  
  nome varchar(40),  
  endereco varchar(50),  
  primary key(id)  
);  
create table matricula(  
  id_aluno int not null,  
  id_disciplina int not null,  
  nota_final int not null default 0,  
  primary key(id_aluno, id_disciplina)  
);
```

b. Assumindo a modelagem física do exercício anterior, escreva comandos SQL para:

- i. Inserir a informação de que o aluno cujo número de matrícula é 123 está inscrito na disciplina cujo código é “ABC001”.

Resp.

```
INSERT INTO matricula (id_aluno, id_disciplina) VALUES (123,  
`ABC001`);
```

- ii. Atualizar a inscrição do item anterior de forma a constar que o aluno obteve nota final igual a 7,5.

Resp.

```
UPDATE matricula SET nota_final = 7.5 WHERE id_aluno = 123  
and id_disciplina = `ABC001`;
```