

**Gabarito AD2**  
**Programação II**

1. (5 pontos) Suponha que lhe foi pedido implementar funcionalidades de um processador de texto usando a linguagem PHP.

a) Qual estrutura de dados você usaria para representar o texto de forma a poder indicar a linha e a coluna de um determinado caractere do texto? Descreva sua idéia em linhas gerais (sem código).

Resp.

Pode ser usado uma matriz, ou seja, um array de array de caracteres onde cada array representa uma linha do texto.

b) Escreva a função `linhacoluna ($l, $c)` que retorna o caractere encontrado na linha `$l` e coluna `$c` ou uma string vazia caso a linha ou a coluna não exista.

Resp.

```
function linhacoluna ($l, $c){  
    if (sizeof($this->data)>$l && sizeof($this->data[$l])>$c)  
        return $this->data[$l][$c];  
    return "";  
}
```

c) Escreva a função `insere ($l, $c, $s)` que insere a string `$s` na posição dada por linha `$l` e coluna `$c`.

Resp.

```
function insere ($l, $c, $s){  
if (sizeof($this->data)<=$l){  
    $l = sizeof($this->data);  
    $this->data[] = array();  
}  
  
if (sizeof($this->data)>$l && sizeof($this->data[$l])>=$c){  
    $i = 0;  
    $newline = array();  
    for(; $i<$c;$i++){  
        $newline[] = $this->data[$l][$i];  
    }  
    for($j =0;$j<strlen($s);$j++){  
        $newline[] = $s[$j];  
    }  
  
    for(; $i<sizeof($this->data[$l]);$i++){  
        $newline[] = $this->data[$l][$i];  
    }  
    $this->data[$l] = $newline;  
}  
  
}
```

d) Implemente com a função `busca ($s)` a funcionalidade de identificar a primeira ocorrência da string `$s` no texto. A função deve retornar um array com 2 valores: linha e coluna onde `$s` foi encontrada, ou false caso `$s` não seja encontrada.

Resp.

```
function busca($s){  
  
    for ($i =0;$i<sizeof($this->data);$i++){  
  
        $j = 0;  
        while ($j+strlen($s)<sizeof($this->data[$i]) &&  
            $s[0] != $this->data[$i][$j] ) $j++;  

```

```

        if ($j<sizeof($this->data[$i])){
            $found = true;
            for($k = 0; $k<strlen($s);$k++){
                if ($s[$k] != $this->data[$i][$j + $k])
                    $found = false;
            }
            if ($found)
                return array($i, $j);
        }
    }
    return false;
}

```

- e) Implemente com a função `trocar($s1, $s2)` a funcionalidade de substituir todas as ocorrências de `$s1` no texto por `$s2`. A função deve retornar o número total de ocorrências.

Resp.

```

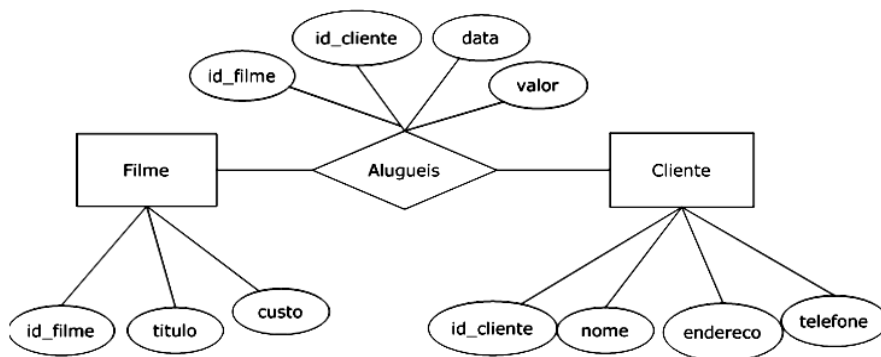
function remove ($l, $c, $s){
    if (sizeof($this->data)>$l && sizeof($this->data[$l])>=$c){
        $i = 0;
        $newline = array();
        for(; $i<$c;$i++){
            $newline[] = $this->data[$l][$i];
        }

        for(; $i<sizeof($this->data[$l])-strlen($s);$i++){
            $newline[] = $this->data[$l][$i+strlen($s)];
        }
        $this->data[$l] = $newline;
    }
}

function trocar($s1, $s2){
    $res = $this->busca($s1);
    if ( $res != false){
        $this->remove($res[0], $res[1], $s1);
        $this->insere($res[0], $res[1], $s2);
        $this->trocar($s1, $s2);
    }
}

```

2. O diagrama E-R (figura abaixo) corresponde a uma locadora de



Filmes

E o código SQL corresponde a alguns dados armazenados atualmente no BD

```

CREATE TABLE `cliente` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nome` varchar(200) NOT NULL,
  `endereco` varchar(200) NOT NULL,

```

```

    `telefone` varchar(40) NOT NULL,
    PRIMARY KEY (`id`)
);

CREATE TABLE `filme` (
    `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
    `nome` varchar(200) NOT NULL,
    `custo` decimal(5,0) unsigned NOT NULL,
    PRIMARY KEY (`id`)
);

CREATE TABLE `alugueis` (
    `id_filme` int(10) unsigned NOT NULL,
    `id_cliente` int(10) unsigned NOT NULL,
    `data` date NOT NULL,
    `valor` decimal(5,0) unsigned NOT NULL,
    PRIMARY KEY (`id_filme`,`id_cliente`)
);

INSERT INTO `cliente` (`id`, `nome`) VALUES
    ('1','Arturo'),
    ('2','Ccarlos'),
    ('3','Edgar'),
    ('4','Jonas'),
    ('5','Jhon'),
    ('6','Guilherme');

INSERT INTO `filme` (`id`, `nome`, `valor`) VALUES
    ('1','The Emperor',8),
    ('2','The Commander',12),
    ('3','The Mummy',12),
    ('4','Ring',12),
    ('5','500',12);

INSERT INTO `alugueis` (`id_filme`, `id_cliente`, `data`, `valor`) VALUES
    ('1','1','2009-10-12',2),
    ('5','2','2009-11-15',2),
    ('1','3','2009-12-16',2),
    ('3','2','2009-12-19',2),
    ('1','4','2009-12-22',2),
    ('1','6','2009-12-25',2),
    ('2','3','2010-01-13',3),
    ('2','4','2010-02-20',3),
    ('4','6','2010-02-22',3),
    ('1','5','2010-03-12',2);

```

Pede-se.

- a) (1 ponto) Escreva uma consulta SQL para obter os filmes que já têm lucro, isto é, cujo valor em alugueis supera seu custo.

Resp,

```

select * from filme as f
where f.custo < (select sum(a.valor) from alugueis as a where a.id_filme = f.id);

```

- b) (1 ponto) Escreva uma consulta SQL para saber os 3 filmes com maior demanda, isto é, os mais alugados.

Resp.

```

select *, count(*) as hits from alugueis as a, filme as f
where a.id_filme = f.id
group by f.nome order by hits desc limit 3

```

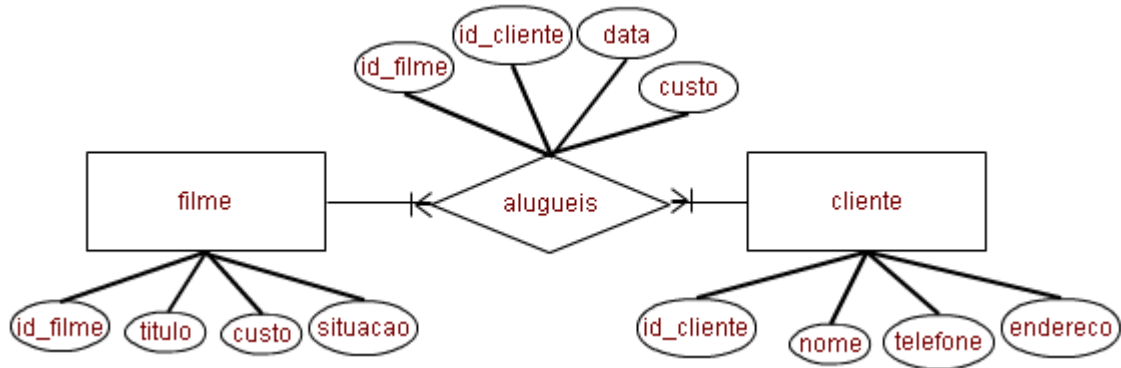
- c) (1 ponto) Escreva uma consulta SQL para saber o filme menos alugado.

Resp.

```
select *, count(*) as hits from alugueis as a, filme as f
where a.id_filme = f.id
group by f.nome order by hits limit 1
```

- d) (1 ponto) Na modelagem acima, não é possível saber se um filme alugado foi devolvido ou não. Altere a modelagem para contemplar esta circunstância (mostre o novo diagrama E-R e as alterações na modelagem física).

Resp.



```
CREATE TABLE `filme` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nome` varchar(200) NOT NULL,
  `custo` decimal(5,0) unsigned NOT NULL,
  `situacao` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
);
```

- e) (1 ponto) Usando a modelagem do item (d) acima, escreva uma consulta SQL para responder quantos filmes não foram devolvidos no momento atual.

Resp.

```
select * from filme where situacao = "alugado";
```