

Nome: _____

Data: __ / __ / ____

Bairro: _____

Pretensão Salarial: **MEI/PJ R\$** _____ (bruto)

Nível de Inglês: () Nenhum () Básico () Intermediário () Avançado () Fluente

Marque o seu nível de conhecimento em cada uma das tecnologias abaixo, onde **0 (ZERO)** indica que você **NUNCA OUVIU FALAR** e **10 (DEZ)** que você conhece e tem **TOTAL DOMÍNIO E CONHECIMENTO** sobre a mesma:

- | | |
|------------------------|---|
| 1) PHP | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 2) POO | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 3) MVC | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 4) AWS | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 5) SQL para MYSQL | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 6) HTML | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 7) XHTML | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 8) CSS | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 9) Javascript | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 10) Xajax (não é Ajax) | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 11) Zend Framework | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 12) Code Igniter | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 13) Laravel | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 14) JQuery | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 15) Java | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 16) Android | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 17) Bootstrap | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 18) Ionic | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 19) Electron | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 20) NodeJS | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 21) Angular | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 21) React | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 |
| 21) Vue | ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 1 |

Conhecimentos de PHP (1/2)

1) Analise a função a seguir.

```
function xyz($data)
{
1.    if (empty($data)){
2.        return false;
    }

3.    if (str_replace("/", "", $data)) {
4.        $data=explode("/", $data);
5.    } else {
6.        $data=explode("-", $data);
    }

7.    if (count($data) != 3 ) {
8.        return false;
    }

9.    if (strlen($data[0])==2) {
10.        if (!checkdate($data[1], $data[0], $data[2])) {
11.            return false;
        }
12.    } else {
13.        if (!checkdate($data[1], $data[2], $data[0])) {
14.            return false;
        }
    }

15.    return true;
}
```

Para que server a função? _____

Descreva todo o procedimento utilizando a numeração da linha como referencia:

Conhecimentos de PHP (2/2)

2) A partir da tabela cliente mostrada abaixo escreva um trecho de código PHP que, consulte TODOS os registros, percorra a lista de registros retornados e escreva na tela somente os clientes com situação igual a 1, e ao final mostre o total de registros geral e o total com situação igual a 1.

Obs:

1. Considere que a conexão no banco já foi estabelecida e que você já tem a conexão do banco em uma variável/objeto, exemplo \$Db ou \$Model.
2. Construa a QUERIE em SQL, ou usando PDO, ou Framework – **Escreva a Querie**, sem usar ORM ou elementos facilitadores do Framework que você usar de exemplo.
3. Considere que a Array / Objeto com o conjunto dos dados foi recuperada com sucesso.
4. Construa o loop responsável por fazer o código dar certo de acordo com os requisitos do enunciado.

cliente
id nome cpf telefone situacao

3) Explique a função e a expressão regular usada abaixo:

```
if (preg_match("/teste/i", "Esta é minha funcao de Teste")) {  
    exemplo ... faz algo;  
}  
else { faz qualquer outra coisa }
```

4) Dentro de uma função preg_replace, qual o uso desta expressão regular?

```
preg_replace("/[^0-9]/", "", $valor);
```

5) Para que serve a função 'substr'. Dê um exemplo de uso?

6) Dentro de um loop, qual a diferença entre 'continue' e 'break'?

7) Digite as funções em PHP responsáveis por:

7.a) Deixar a primeira letra maiuscula de cada palavra

7.b) Capitalizar todas as letras da string

7.c) Deixar em letras minusculas todas as letras da string

7.d) Deixar a primeira letra maiuscula de toda a string

7.e) Remover os espacos em branco no inicio e no fim da string

8) Qual o resultado impresso pelo seguinte código?

```
<?php  
  
    $string = "14302";  
  
    $string[$string[2]] = "4";  
  
    print $string;  
  
?>
```

9) Considere o seguinte código e informe porque ele não irá funcionar.

```
<html>  
<head>... conteudo do head </head>  
<body>  
...  
<?php  
header("Location: http://www.dinsey.com");  
exit();  
?>  
...  
</body>  
</html>
```

10) O código abaixo irá imprimir qual resultado?

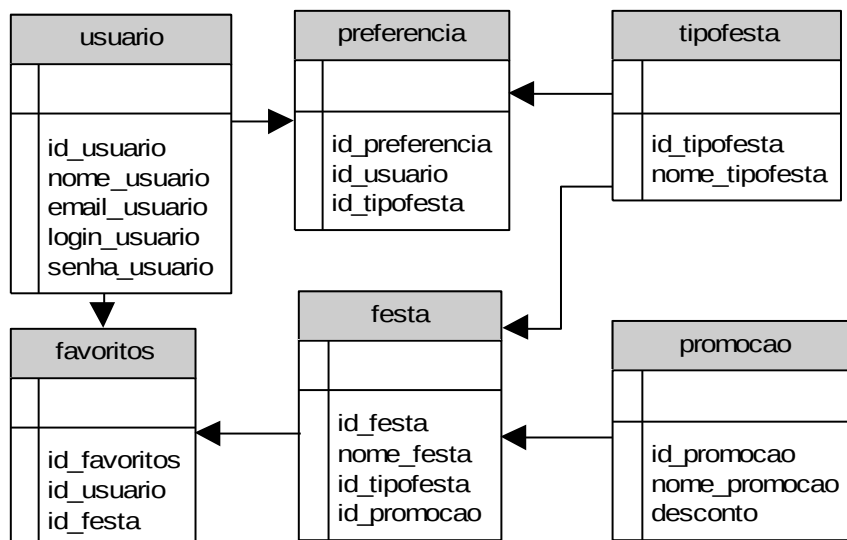
```
<?php  
  
$a = array(1 => 0, 3 => 2, 4 => 6);  
$b = array(3 => 1, 4 => 3, 6 => 4);  
  
print_r(array_intersect($a, $b));  
?>
```

11) O que é arquitetura MVC? Explique cada uma de suas camadas e a função delas?

12) Qual a função dos namespaces e porque eles devem ser utilizados?

13) O que é PSR? Você saberia comentar algumas das recomendações do PSR-0 ao PSR-4?

Conhecimentos de SQL (1/2)



1) Crie uma SQL onde um usuário se autentique usando o login "admin" e a senha "1234".

2) Crie UMA SQL utilizando o comando INNER JOIN, que liste todas as festas favoritas do usuário de ID igual a "10".

3) Considerando os seguintes requisitos, o programador construiu o seguinte código que "funciona" – como diria ele, mas que na prática tem uma série de inconsistências.

- Deve-se exibir uma **tabela (em html)** que mostra o ID, NOME e E-MAIL dos clientes. **Todas estas informações estão na tabela 'cliente'.**
- Deve-se poder aplicar diversos filtros para o resultado, estes filtros podem estar na tabela 'cliente' OU em tabelas relacionadas.
- Os filtros são acumulativos – e neste exemplo foram mostrados apenas alguns filtros, mas tenha em mente que a tela terá cerca de 30 filtros diferentes (envolvendo 30 ou mais tabelas) e a tabela de clientes tem 1 milhão de registros.
 - Filtrar pelo ID
 - Filtrar pelo NOME usando LIKE
 - Filtrar pelo E-MAIL usando LIKE
 - Filtrar pelos clientes que tem pedidos (True/False)
(Relacao: 1 para muitos)
 - Filtrar pelos clientes que tem favoritos (True/False)
(Relacao: 1 para muitos)
 - Filtrar pelos clientes cujo o favorito tenha um NOME do FAVORITO seja igual a algo (Usar LIKE)
(Relacao: 1 para muitos)

[SQL]

```
SELECT * FROM cliente  
LEFT JOIN pedido ON pedido.id_cliente = cliente.id  
LEFT JOIN favorito ON favorito.id_cliente = cliente.id  
WHERE 1=1
```

```
if (filtro_id) 'AND cliente.id = filtro_id'
```

```
if (filtro_nome) 'AND cliente.nome LIKE '%filtro_nome%'
```

```
if (filtro_email) 'AND cliente.email LIKE '%filtro_email%'
```

```
if (filtro_pedido) 'AND pedido.id > 0'
```

```
if (filtro_favorito) 'AND favorito.id > 0'
```

```
if (filtro_favorito_nome) 'AND favorito.nome LIKE '%filtro_favorito_nome%'
```

```
EXECUTE QUERY (SQL)
```

3.a) Essa query apresenta uma série de inconsistências. Uma delas diz respeito ao desconhecimento das relações de 1 para 1, 1 para muitos e muitos para muitos. Por que?

3.b) Além de representar um problema de inconsistência mencionado acima, existe um desperdício de recursos na construção dos filtros.

Reescreva esta lógica (query e programação) com o intuito de aumentar a velocidade de processamento.

4) Ainda com relação ao caso anterior, por ser uma tabela com 1 milhão de registros e várias opções de filtragem, que estratégias você utilizaria do ponto de vista do banco de dados e também da programação, para facilitar a manipulação desses dados. Como por exemplo, durante a paginação e filtragem de dados.

Escreva em tópicos suas ideias, de forma técnica e não vaga.

[Laravel]

	Cliente
	id
	Nome
	Razao_Social
	Nome_Fantasia
	CPF_CNPJ
	Created_at
	Updated_at
	Deleted_at

	Venda
	id
FK	Id_loja
FK	id_cliente
FK	Id_estoque
	Data
	Data_finalizado
	Created_at
	Updated_at
	Deleted_at

Para as questões 1 a 5, considere que temos o Model 'Cliente' e 'Venda' **estendendo o Eloquent**.

1) Busque todos os registros da tabela cliente:

- a. Usando o Eloquent
- b. Usando o Query Builder

2) Busque o cliente de ID = 3, e imprima a 'Razao Social' na tela usando echo.

- a. Usando o Eloquent
- b. Usando o Query Builder

3) Considerando o Eloquent, e que cada cliente pode ter "n" Vendas.

- a. Como o model "Cliente" poderia ser alterado para representar a relação de "um-para-muitos"?

```
Class Cliente extends Eloquent {  
    // Digite seu código um-para-muitas vendas. Atenção a FK, que não segue o  
    padrão do framework.  
}
```

- b. Ainda usando o Eloquent, como poderíamos buscar o cliente de ID=5, e suas vendas em um único comando?

4) O Eloquent permite o uso de dois atributos no Model para realizar "Mass Assignment".

- a. Quais são esses dois atributos?
- b. Qual a diferença entre eles?

5) No Model é possível definir Mutators. Digite um mutator para o Model Cliente, que retorne o campo Razao_Social em letras maiúsculas.

```
Class Cliente extends Eloquent {  
    // Digite seu mutator.  
}
```

6) Usando o Query Builder, liste todas as Vendas do Cliente ID=5, onde o estoque seja ID=13, e retorne o nome do cliente e a data da venda.