

Este teste deve ser respondido em até 3 dias.

As respostas devem ser adicionadas a um repositório público no GitHub ou no GitLab, e deverá nos disponibilizar o link para avaliarmos as respostas.

Critérios de avaliação do código:

- 1) Resultado coerente ao esperado
- 2) Praticidade do código
- 3) Qualidade do código

Exercício 1

Dado o seguinte formulário:

```
<form method="post">
  <div>
    <label for="name">Nome completo:</label>
    <input type="text" id="name" name="name">
  </div>

  <div>
    <label for="userName">Nome de login:</label>
    <input type="text" id="userName" name="userName">
  </div>

  <div>
    <label for="zipCode">CEP</label>
    <input type="text" id="zipCode" name="zipCode">
  </div>

  <div>
    <label for="email">Email:</label>
    <input type="text" id="email" name="email">
  </div>

  <div>
    <label for="password">Senha (8 caracteres mínimo, contendo pelo menos 1 letra
    e 1 número):</label>
    <input type="password" id="password" name="password">
  </div>

  <input type="submit" value="Cadastrar">
</form>
```

Crie um código com as seguintes especificações:

- Realizar validações para cada input do form (quantas e quais validações fica a seu critério)
- Crie um BD para armazenar os dados
- Crie um método para inserir os dados no BD

- Crie um método para consultar os dados do BD
- Fique a vontade para implementar qualquer outra classe e métodos que julgue necessários

Crie utilizando OO e fique a vontade para usar qualquer framework que preferir (pode ser feito com PHP puro também).

Métodos de exemplo:

```
/**
 * @param string $nome_da_tabela nome da tabela a ser consultada
 * @param array informacao é um array com os mesmos nomes dos elementos do form
 */
public function select(string $nomeDaTabela, array $informacao): bool
{
    // retorna se a consulta existe
}

/**
 * @param string $nome_da_tabela nome da tabela na qual vai inserir os dados
 * @param array informacao é um array com os mesmos nomes dos elementos do form
 */
public function insert(string $nomeDaTabela, array $informacao): int
{
    // retorna o id do registro
}
```

Exercício 2

Crie um código que realize as seguintes instruções:

- 1) Crie um array
- 2) Popule este array com 7 números
- 3) Imprima o número da posição 3 do array
- 4) Crie uma variável com todas as posições do array no formato de string separado por vírgula
- 5) Crie um novo array a partir da variável no formato de string que foi criada e destrua o array anterior
- 6) Crie uma condição para verificar se existe o valor 14 no array
- 7) Faça uma busca em cada posição. Se o número da posição atual for menor que o da posição anterior (valor anterior que não foi excluído do array ainda), exclua esta posição
- 8) Remova a última posição deste array
- 9) Conte quantos elementos tem neste array
- 10) Inverta as posições deste array

Exercício 3

A partir das tabelas USUARIO e INFO (que atualmente estão preenchidas com estas informações), crie um DML para o resultado esperado (Crie as tabelas, insira os dados, e exiba o resultado esperado).

USUARIO

Id	cpf	nome
1	16798125050	Luke Skywalker
2	59875804045	Bruce Wayne
3	04707649025	Diane Prince
4	21142450040	Bruce Banner
5	83257946074	Harley Quinn
6	07583509025	Peter Parker

INFO

Id	cpf	genero	ano_nascimento
1	16798125050	M	1976
2	59875804045	M	1960
3	04707649025	F	1988
4	21142450040	M	1954
5	83257946074	F	1970
6	07583509025	M	1972

Tabela de resultado esperado

usuário	maior_50_anos
Peter Parker - M	NÃO
Luke Skywalker - M	NÃO
Bruce Banner - M	SIM