

Faça o que eu fiz na aula

Vamos implementar o telefone da nossa tela de cadastro, como podemos fazer?

Primeiramente vamos colocar o telefone que recebemos do formulário da página a partir do método `POST`, como argumento da instanciação da classe.

```
$contato = new Contato($_POST['email'], $_POST['endereco'], $_POST['cep'], $_POST['telefone']);
```

Vamos adicionar a propriedade no construtor da classe `Contato`.

```
private $telefone;

public function __construct(string $email, string $telefone)
{
    $this->email = $email;
    $this->telefone = $telefone;
}
```

E podemos exibir o telefone criando um método `getTelefone`

```
public function getTelefone(): string
{
    return $this->telefone;
}
```

E se formos no arquivo, vamos chamar o método no HTML.

```
<li class="list-group-item">Telefone: <?php echo $contato->getTelefone(); ?></li class="list-group-i
```

E o nosso telefone vai ser impresso, no entanto, podemos digitar qualquer formato aí, inclusive letras!! e um telefone não tem letras.

Poderíamos tentar realizar algum tipo de filtro, o problema é que a função `filter_var` que usamos para o e-mail não tem uma maneira de validar um telefone, vamos ter que construir o nosso próprio filtro.

Uma solução nesse caso é utilizar uma expressão regular a partir da função `preg_match`, uma função feita para utilizarmos expressões regulares em nossas strings.

Como vamos lidar com telefones fixos, acho que 6785-4736 é um bom padrão, quatro números, um traço e mais quatro números. Se o telefone estiver dentro deste padrão, está certo, caso contrário, está errado.

Precisamos levar em conta também que é só isso que precisamos encontrar, ou seja, precisamos de um começo de linha, quatro números, hífen, mais quatro números e um fim de linha.

Para utilizarmos uma expressão regular no PHP, chamamos a função `preg_match`, que usa como argumentos a expressão regular e a string no qual deve ser feita a pesquisa.

Vamos criar uma função e escrever o `preg_match` dentro dela:

```
private function validaTelefone(string $telefone): int
{
    return preg_match('/^[0-9]{4}-[0-9]{4}/');
}
```

Agora vamos chamar essa função no construtor, no lugar da nossa atribuição:

```
if ($this->validaTelefone($telefone)) {
    $this->telefone = $telefone;
} else {
    $this->telefone = "Telefone inválido";
}
```

Como temos duas atribuições, vamos criar um método privado só pra fazer isso:

```
private function setTelefone(string $telefone): void
{
    $this->telefone = $telefone;
}
```