



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Rio de Janeiro

Disciplina
Programação Front-End I

PROF. FERNANDO TAMBERLINI ALVES

Atividade Prática I

Objetivo: Criar um formulário com validação igual a imagem abaixo

Exercício de Formulário

Pressione Alt + tecla sublinhada para acessar os campos rapidamente. Use Tab para navegar entre os campos.

Nome Completo *

Nome deve conter apenas letras e ter no mínimo 3 caracteres

E-mail *

Insira um e-mail válido (exemplo: usuario@dominio.com)

CPF *

Digite um CPF válido (ex: 123.456.789-00 ou 12345678900)

Tipo de Usuário *

☐ Aluno ☐ Professor ☐ Visitante

Telefone *

Digite um telefone válido (ex: (11) 98765-4321 ou 11987654321)

Data de Nascimento

Website

Estado

Senha *

A senha deve conter pelo menos 8 caracteres, incluindo maiúsculas, minúsculas, números e caracteres especiais

Confirmar Senha *

As senhas devem ser iguais

Cor Preferida

Nível de Conhecimento em HTML (1-5) *

☐ Concordo com os termos e condições *

Mensagem

Limpar Formulário

Enviar Cadastro



Atividade Prática I

1. Criar um diretório para o formulário `.\nome_aluno\`
2. Clonar o repositório do formulário da atividade a partir do GITHUB

```
git clone https://github.com/ftamberlinidev/formulario.git
```

3. Abrir a pasta `.\nome_aluno\formulario` no VSCODE
4. Incluir as seguintes regras de validação do formulário:

- Atributo `"required"` para os campos obrigatórios;
- Atributo `"minlength"` para os campos com tamanho mínimo;
- Atributo `"maxlength"` para os campos com tamanho máximo;
- Atributo `"accesskey"` para as teclas de atalho;
- Atributo `"min"` para os valores mínimos;
- Atributo `"pattern"` para máscara de valor;

Atividade Prática I

- Atributo "placeholder" para os campos com sugestão de valor;
- Atributo "tabindex" para a ordem dos campos na tecla TAB;

5. Faça a associação dos patterns abaixo aos seus respectivos campos:

- `pattern="(?.*\d)(?.*[a-z])(?.*[A-Z])(?.*[!@#$%^&*]).{8,}"`
- `pattern="\([0-9]{2}\)\s?[0-9]{4,5}-?[0-9]{4})|([0-9]{10,11})"`
- `pattern="[0-9]{3}\.?[0-9]{3}\.?[0-9]{3}-?[0-9]{2}"`
- `pattern="[A-Za-zÀ-ÖØ-öø-ÿ\s]+"`

Atividade Prática I

- Incluir a tag para referência externa do CSS

```
<link rel="stylesheet" type="text/css" href="estilo.css">
```

- Definir os estilos dos elementos utilizando corretamente os seletores

- Corpo

```
font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif  
line-height: 1.6;  
max-width: 600px;  
margin: 0 auto;  
padding: 20px;
```

- Título 1 – Exercício de Formulário

```
text-align: center;  
color: #4285f4
```

- Definir os estilos dos elementos utilizando corretamente os seletores

- Descrição do Formulário

```
text-align: center;  
margin-bottom: 20px;  
color: #666;
```

- Formulário

```
background-color: #f9f9f9;  
padding: 20px;  
border-radius: 8px;  
box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
```

- Rótulos

```
display: block;  
margin-bottom: 5px;  
font-weight: bold;
```

- Definir os estilos dos elementos utilizando corretamente os seletores

- Elementos do tipo input, select e textarea

```
width: 100%;  
padding: 10px;  
border: 1px solid #ddd;  
border-radius: 4px;  
font-size: 16px;  
box-sizing: border-box;
```

- Botões

```
background-color: #4285f4;  
color: white;  
border: none;  
padding: 12px 20px;  
border-radius: 4px;  
cursor: pointer;  
font-size: 16px;  
width: 40%;
```

- Botões com mouse em cima

```
background-color: #3367d6;
```

- Demais elementos

```
fieldset {  
  border: 1px solid #ddd;  
  border-radius: 4px;  
  padding: 10px 15px;  
  margin: 0;  
}  
legend {  
  font-weight: bold;  
  padding: 0 5px;  
}  
.radio-group {  
  display: flex;  
  flex-wrap: wrap;  
  gap: 15px;  
  margin-top: 10px;  
}  
.radio-option {  
  display: flex;  
  align-items: center;  
}
```

```
.radio-option input[type="radio"] {  
  width: auto;  
  margin-right: 5px;  
}  
.radio-option label {  
  display: inline;  
  margin: 0;  
  font-weight: normal;  
}  
.form-group {  
  margin-bottom: 15px;  
}  
.form-button {  
  display: flex;  
  justify-content: center;  
  gap: 20px;  
}
```

Atividade Prática II

Objetivo: Elaborar um card centralizado e de layout responsivo com animações simples usando CSS



Atividade Prática II

1. Criar um diretório e repositório para o card

```
git init card
```

2. Abrir a pasta `.\nome_aluno\card` no VSCODE

3. Identifique os elementos HTML

- `body`
- `div card`
- `div profile-image`
- `heading card-title`
- `paragraph description`
- `social icons`

Atividade Prática II

1. Estrutura HTML

```
<> card.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Card Simples - Exercício CSS</title>
7      <link rel="stylesheet" type="text/css" href="style.css">
8  </head>
9  <body>
10     <div class="card">
11         <div class="profile-image">
12             
13         </div>
14         <h2 class="card-title">Prof. Fernando</h2>
15         <p class="card-description">
16             Analista, programador e professor de Informática do Instituto Federal do Rio de Janeiro.
17         </p>
18         <div class="social-icons">
19             <a href="#" class="icon website"></a>
20             <a href="#" class="icon github"></a>
21             <a href="#" class="icon tiktok"></a>
22         </div>
23     </div>
24 </body>
25 </html>
```

Atividade Prática II

1. Estilo do body, card, card: hover, .profile-image, profile-image: hover, img e *

```
# style.css > ...
1  * {
2      margin: 0;
3      padding: 0;
4      box-sizing: border-box;
5  }
6
7  body {
8      font-family: Arial, sans-serif;
9      min-height: 100vh;
10     background: linear-gradient(135deg, #74b9ff, #1f09e3);
11     display: flex;
12     justify-content: center;
13     align-items: center;
14     padding: 20px;
15 }
16
17 .card {
18     background: white;
19     border-radius: 15px;
20     padding: 30px;
21     box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
22     width: 350px;
23     text-align: center;
24     transition: transform 0.3s ease;
25 }
```

```
.card: hover {
    transform: translateY(-10px);
}
.profile-image {
    width: 100px;
    height: 100px;
    border-radius: 50%;
    margin: 0 auto 20px;
    overflow: hidden;
    border: 3px solid #74b9ff;
    transition: transform 0.3s ease;
}
.profile-image: hover {
    transform: scale(1.1);
}
.profile-image img {
    width: 100%;
    height: 100%;
    object-fit: cover;
}
```

Atividade Prática II

1. Estilo do card-title, card-title:hover, card-description, social-icons, icon, icon:hover

```
.card-title {  
  font-size: 22px;  
  color: #333;  
  margin-bottom: 10px;  
  transition: color 0.3s ease;  
}
```

```
.card-title:hover {  
  color: #74b9ff;  
}
```

```
.card-description {  
  font-size: 14px;  
  color: #666;  
  line-height: 1.5;  
  margin-bottom: 25px;  
}
```

```
.social-icons {  
  display: flex;  
  justify-content: center;  
  gap: 15px;  
}
```

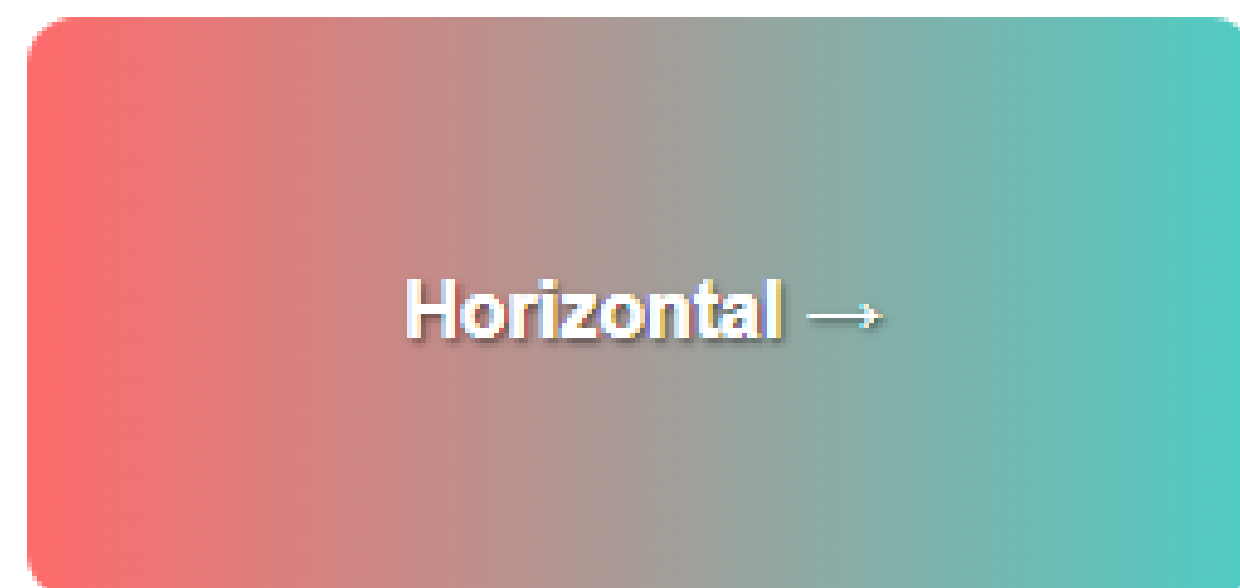
```
.icon {  
  width: 45px;  
  height: 45px;  
  border-radius: 50%;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  font-size: 18px;  
  color: white;  
  text-decoration: none;  
  transition: transform 0.3s ease;  
}
```

```
.icon:hover {  
  transform: translateY(-5px);  
}
```

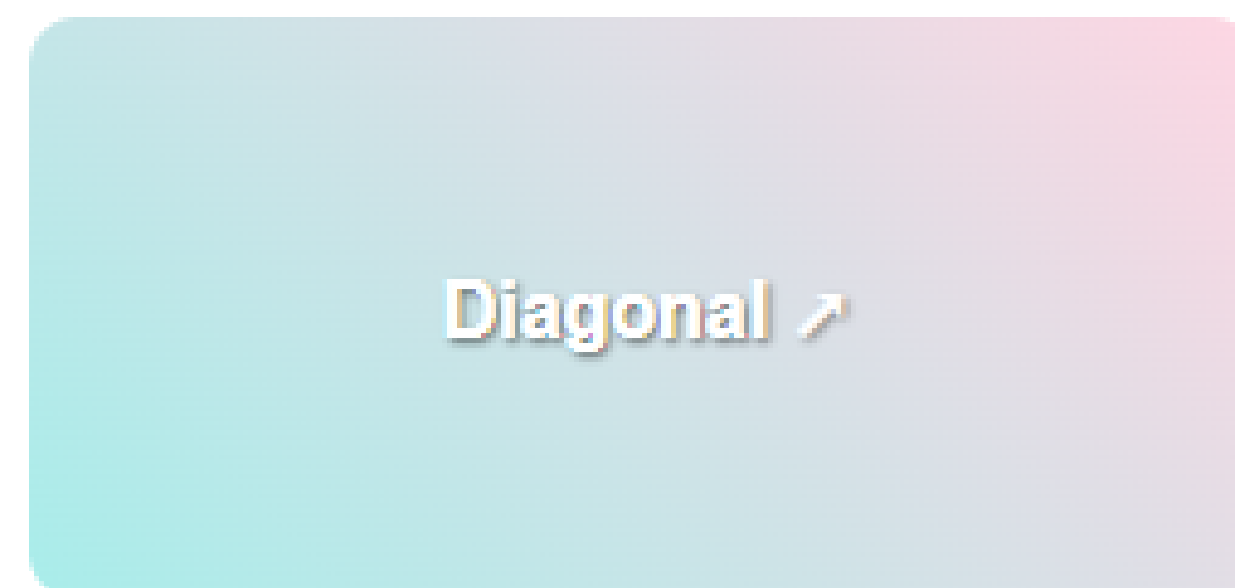
Atividade Prática II

Gradientes são transições suaves entre duas ou mais cores. Em vez de ter uma cor sólida, você cria um efeito onde uma cor se transforma gradualmente em outra.

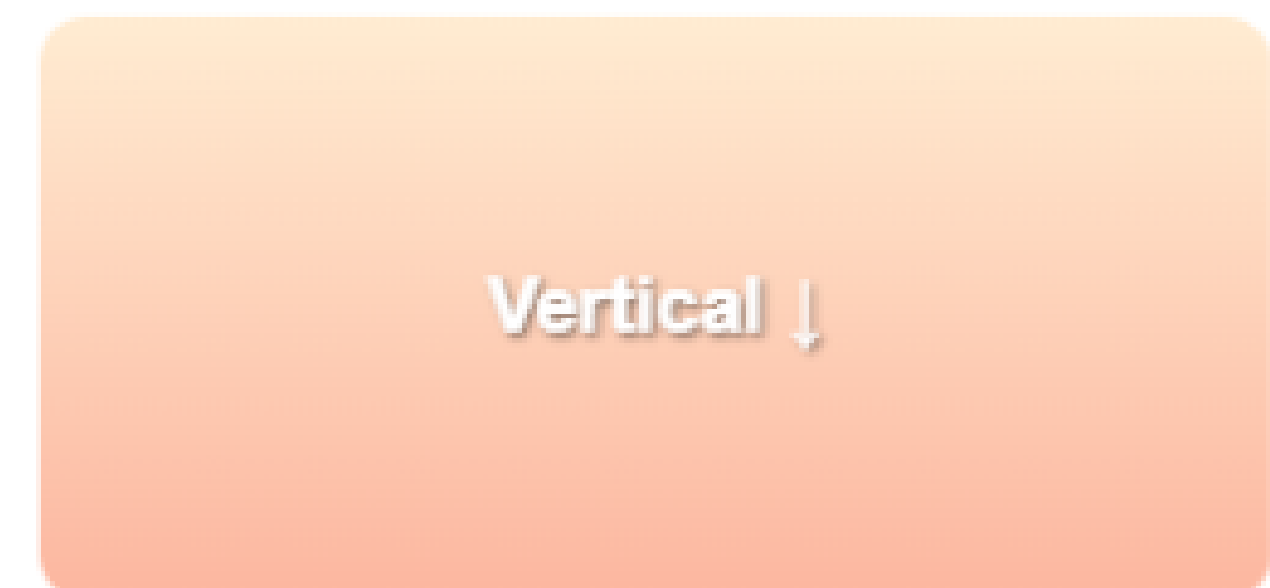
- Gradiente Linear - As cores mudam em uma linha reta



```
linear-gradient(to right,  
#ff6b6b, #4ecdc4)
```



```
linear-gradient(45deg,  
#a8edea, #fed6e3)
```



```
linear-gradient(to bottom,  
#ffecd2, #fcb69f)
```

Atividade Prática II

- Gradiente Radial - As cores mudam em círculos ou elipses

Círculo

```
radial-gradient(circle,  
#ff9a9e, #fecfef)
```

Elipse

```
radial-gradient(ellipse,  
#667eea, #764ba2)
```

Posicionado

```
radial-gradient(circle at  
top left, #ff6b6b,  
#4ecdc4)
```

Atividade Prática II

A propriedade *transition* do CSS permite criar animações suaves quando uma propriedade CSS muda de valor.

- Propriedade: O que vai animar (background, transform, all)
- Duração: Quanto tempo dura (0.3s, 500ms)
- Timing: Como acontece (ease, linear) – opcional
- Delay: Atraso para começar (0.1s) – opcional

```
transition: propriedade duração timing delay;  
transition: background 0.3s ease 0.1s;
```

Atividade Prática II

A propriedade *transform* do CSS permite modificar a aparência e posição de um elemento sem afetar o layout da página. *Transform* "transforma" o elemento visualmente - move, gira, redimensiona ou distorce - mas não ocupa espaço extra no layout.

```
transform: rotate(45deg);  
transform: rotate(-90deg);
```

```
transform: scale(1.5);          /* Aumenta 50% */  
transform: scale(0.8);          /* Diminui para 80% */  
transform: scale(2, 0.5);       /* Largura 2x, altura 50% */
```

Atividade Prática II

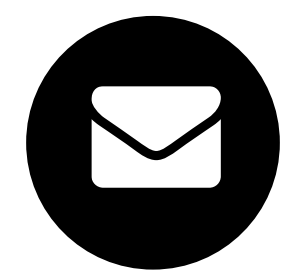
```
transform: translateX(50px);          /* Move 50px para direita */
transform: translateY(-20px);         /* Move 20px para cima */
transform: translate(30px, 40px);     /* Move X e Y */

transform: skewX(20deg);              /* Inclina horizontalmente */
transform: skewY(15deg);              /* Inclina verticalmente */
```

Dúvidas?



Dúvidas?



fernando.alves@ifrj.edu.br



ftamberlini.dev.br