

A.16

Validação de Formulários e Expressões Regulares

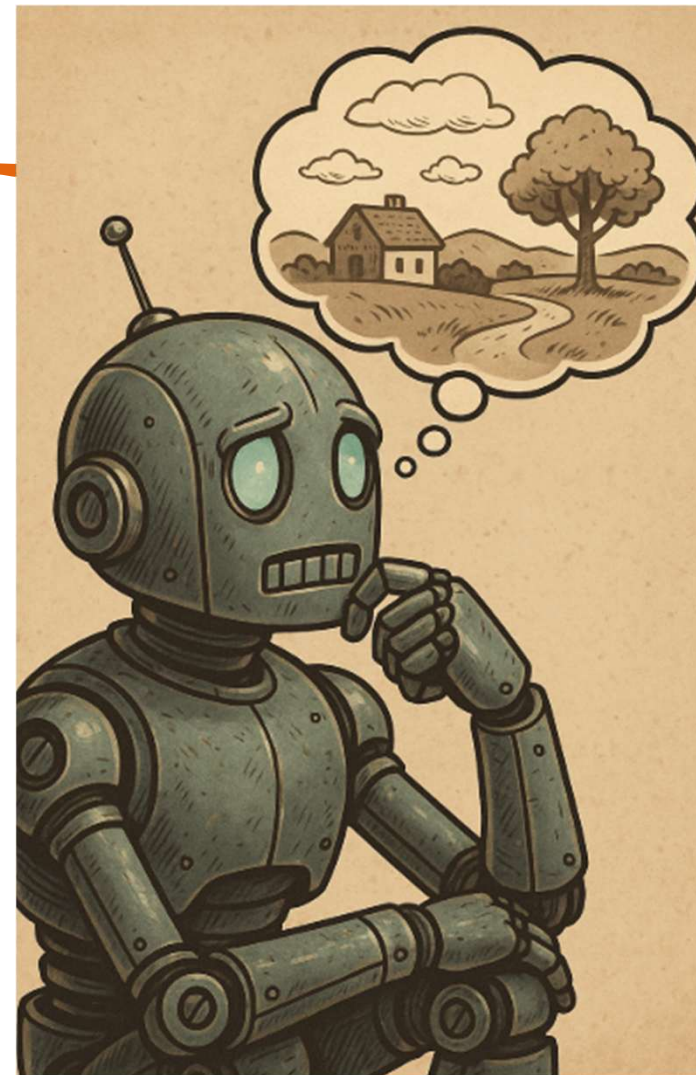
M4. Interatividade e Estruturação
Avançada em HTML

Objetivos

- Entender a importância da validação de dados no front-end
- Explorar atributos HTML5 que ajudam a validar formulários
- Conhecer elementos auxiliares em formulários
- Introduzir expressões regulares para validação personalizada

Na última aula...

- Formulários `<form>`
- Tipos de Campos de Entrada



Na aula de hoje...

- Introdução
- Elementos de Formulário
- Atributos Importantes
- Validação HTML5
- Expressões Regulares (Regex)



Introdução

- Imagine preencher um formulário online e conseguir enviar com **campos vazios** ou com **dados errados**
 - Exemplos: CPF com letras, e-mails sem @
- Isso pode causar erros em **cadastros**, **perdas de informações** e até **mesmo falha de segurança**
- A **validação** de formulários garante que os dados inseridos sejam **corretos**, **completos** e **seguros** antes de serem enviados para o servidor

Por que isso é importante?

- ❖ Evita erros no sistema e melhora a experiência do usuário
- ❖ Aumenta a segurança contra ataques simples (ex.: injeções de dados inválidos)
- ❖ Facilita o trabalho do desenvolvedor back-end, pois os dados já chegam mais limpos
- ❖ É exigido em **qualquer aplicação web profissional**

Estruturando Formulários no HTML

Elementos Fundamentais

1. `<fieldset>`

- Cria um **agrupamento lógico** de campos dentro do formulário
- Facilita a **organização visual** e a **acessibilidade**
- Geralmente usados em cadastros longos, separando categorias
- Exemplo: cadastro dividido em *Dados Pessoais*, *Endereço*, *Informações de Login*

2. `<legend>`

- Serve como **título do grupo de campos** criado pelo `<fieldset>`
- Ajuda o usuário entender rapidamente o que deve ser preenchido
- Também melhora a leitura por softwares de acessibilidade

3. `<label>`

- Texto descritivo associado a um campo do formulário
- Facilita a navegação: ao clicar no texto, o foco vai direto para o campo
- Melhora a **usabilidade** e a **acessibilidade**
- Sempre associado a um campo via `for="idDoCampo"`

```
<fieldset>
  <legend>Dados Pessoais</legend>
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome">
</fieldset>

<fieldset>
  <legend>Endereço</legend>
  <label for="rua">Rua:</label>
  <input type="text" id="rua" name="rua">
</fieldset>

<label for="email">E-mail:</label>
<input type="email" id="email" name="email" required>
```

Atributos Essenciais em Campos de Formulários

Atributos Principais e seus usos

1. **name** - Identifica o campo no **back-end** (servidor). É o que será enviado no corpo da requisição (POST/GET)
2. **id** - Identificador único dentro da página. Usado pelo **label**, JavaScript e CSS
3. **value** - Define um valor **padrão inicial** do campo. Útil em formulários de edição ou sugestões
4. **placeholder** - Texto de **dica** que aparece dentro do campo. Some quando o usuário digita
5. **required** - Torna o preenchimento **obrigatório**. O navegador impede o envio sem esse campo
6. **readonly** - Campo **somente leitura**. Útil para mostrar valores fixos, ex.: número de matrícula
7. **disabled** - Campo **desativado** (nem leitura, nem envio)

```
<input type="text" name="usuario">

<label for="cpf">CPF:</label>
<input type="text" id="cpf" name="cpf">

<input type="text" name="cidade" value="São Paulo">

<input type="email" placeholder="exemplo@email.com">

<input type="password" required>

<input type="text" value="12345" readonly>
```

Validação Automática com HTML5

O que é Validação HTML5?

- O próprio navegador já consegue validar alguns tipos de dados **sem precisar de JavaScript**
- Essa validação é chamada de **Client-side Validation**
- Ajuda a garantir que os dados estão corretos **antes do envio**

Validação de e-mail

fdf

! Inclua um "@" no endereço de e-mail. "fdf" está com um "@" faltando.

Name

Please fill out this field.

Address

WW

SSN:

! Please match the requested format.
The Social Security Number

Validação Automática com HTML5

Atributos e Tipos de Campos

1. **type** – define que tipo de dado é esperado
 - **email** – precisa ter o @ e domínio
 - **number** – aceita apenas números
 - **url** – deve começar com http:// ou https://
 - **date** – calendário de seleção de data
2. **required** – torna o preenchimento obrigatório.
3. **min** e **max** – define limites para números ou datas
4. **maxlength** e **minlength** – controla a quantidade de caracteres no texto
5. **step** – define saltos em valores numéricos
6. **pattern** – usa **expressão regular** para validar

```
<form>
  <label>Email:</label>
  <input type="email" required><br><br>

  <label>Idade:</label>
  <input type="number" min="18" max="99" required><br><br>

  <label>Usuário:</label>
  <input type="text" minlength="5" maxlength="12" required><br><br>

  <label>CEP:</label>
  <input type="text" pattern="\d{5}-\d{3}" placeholder="00000-000" required><br><br>

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

Benefícios

- ✓ Fácil de implementar (sem JS)
- ✓ Funciona em todos os navegadores modernos
- ✓ Mensagens de erro automáticas

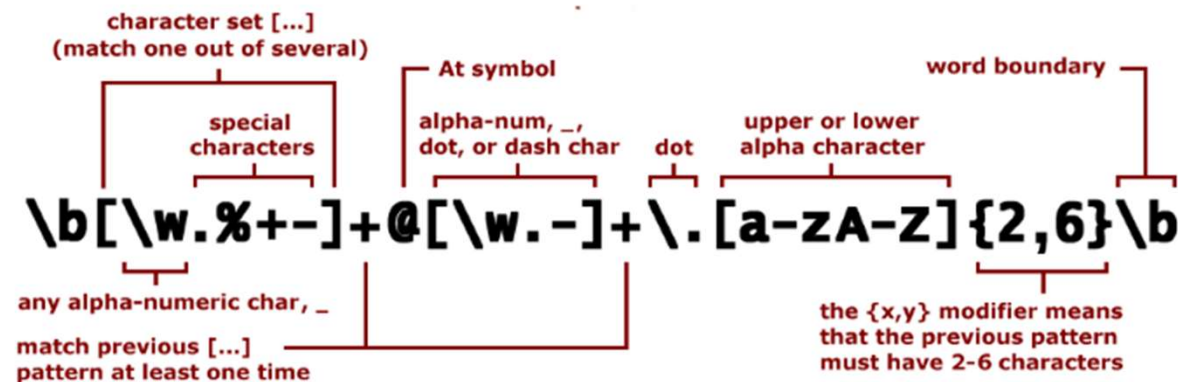
Limitações

- Cada navegador exibe mensagens diferentes (não personalizadas)
- Apenas regras simples
- Não substitui a validação no servidor

Expressões Regulares (Regex)

O que é Regex?

- **Regex** = Regular Expression (expressão regular)
- É um **padrão de busca** usada para verificar se um texto segue uma determinada regra
- Em HTML5, é usado no atributo **pattern** dos inputs
- Ajuda a **validar dados** diretamente no navegador



Parse: username@domain.TLD (top level domain)

Regex em HTML5:

Estrutura Básica

- `[]` – conjunto de caracteres permitidos
 - Ex.: `[0-9]` – apenas números de 0 a 9
- `{}` – quantidade de repetições
 - Ex.: `{3}` – exatamente 3 vezes
 - Ex.: `{2,4}` – entre 2 a 4 vezes
- `^` – início da string
- `$` – final da string
- `\d` – qualquer dígito (igual a `[0-9]`)
- `\w` – letras, números ou underline

```
<input type="text" pattern="[0-9]+" placeholder="Somente números">
<input type="text" pattern="\d{5}-\d{3}" placeholder="CEP (00000-000)">
<input type="text" pattern="\(\d{2}\)\d{5}-\d{4}" placeholder="(11)91234-5678">
<input type="text" pattern="\d{3}\.\d{3}\.\d{3}-\d{2}" placeholder="000.000.000-00">
<input type="email" pattern="^[^@\s]+@[^@\s]+\.[^@\s]+$" placeholder="usuario@email.com">
```

Perguntas?! Dúvidas?!?

