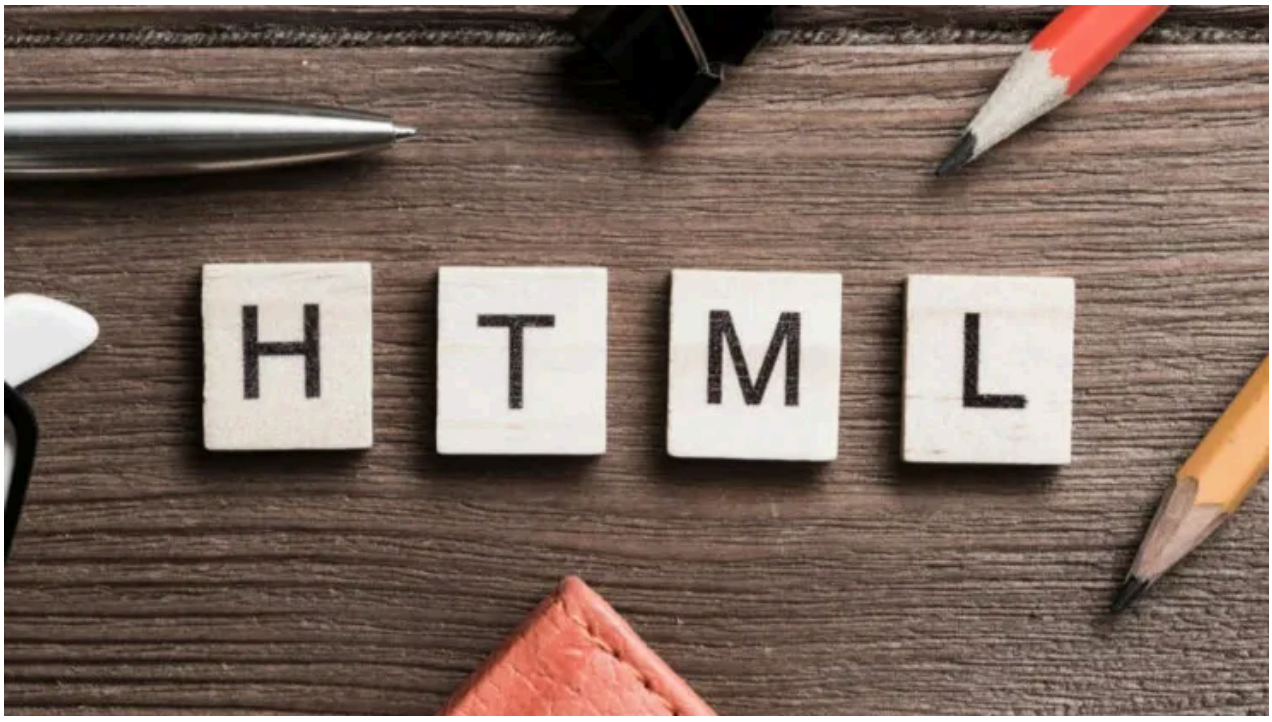

Construção de uma RichTextBox em HTML5, CSS3 e JavaScript

Igor Felix
Computer Scientist - Mobile Developer - .Net(Fullstack) - .Net(MAUI), Linux Developer and Python(Django) - NodeJS(Express) - BAT/BASH -...

10 de maio de 2025

Introdução: A História do HTML e a Evolução até o HTML5 e CSS3

O HTML (HyperText Markup Language) surgiu no início da década de 1990 como uma linguagem de marcação para estruturar páginas na Web. Desenvolvido por Tim Berners-Lee, o HTML permitia a criação de documentos interligados por hiperlinks, o que revolucionou o acesso à informação. Com o tempo, novas versões foram lançadas até culminar no HTML5, publicado em 2014, com suporte a multimídia nativo, APIs avançadas e uma sintaxe mais limpa e semântica. Paralelamente, o CSS (Cascading Style Sheets) evoluiu até o CSS3, trazendo transições, animações, layout flexível e design responsivo para a Web moderna.

Construção de uma RichTextBox em HTML5, CSS3 e JavaScript

Uma RichTextBox é uma caixa de texto avançada que permite ao usuário inserir e formatar conteúdo em HTML. Com HTML5 e JavaScript, podemos construir uma RichTextBox sem a necessidade de tabelas (tableless), utilizando apenas s e elementos modernos.

Estrutura HTML

```
<div id="editor-toolbar">
  <button onclick="execCmd('bold')">Negrito</button>
  <button onclick="execCmd('italic')">Itálico</button>
  <button onclick="execCmd('underline')">Sublinhado</button>
  <button onclick="execCmd('insertImage')">Imagem</button>
  <button onclick="execCmd('createLink')">Link</button>
  <button onclick="execCmd('insertVideo')">Vídeo</button>
  <button onclick="execCmd('insertAudio')">Áudio</button>
</div>
<div id="richtextbox" contenteditable="true"></div>
```

Estilo com CSS3

```
#editor-toolbar button {
  margin: 2px;
  padding: 5px;
}

#richtextbox {
  border: 1px solid #ccc;
  min-height: 300px;
  padding: 10px;
  margin-top: 10px;
  background-color: #fff;
}
```

Scripts JavaScript de Formatação

```
function execCmd(command) {
  if (command === 'insertImage') {
    const url = prompt("Insira o URL da imagem:");
    document.execCommand('insertImage', false, url);
  } else if (command === 'createLink') {
    const url = prompt("Insira o URL do link:");
    document.execCommand('createLink', false, url);
```

```
  } else {
    document.execCommand(command, false, null);
  }
}
```

Listeners para Captura de Digitação e Comandos

Para capturar eventos de digitação e comandos de atalho:

```
document.getElementById("richtextbox").addEventListener("keyup", (e) =>
{
  console.log("Texto atualizado:", e.target.innerHTML);
});

document.addEventListener("keydown", (e) => {
  if (e.ctrlKey && e.key === "c") {
    console.log("Comando Ctrl+C detectado");
  } else if (e.ctrlKey && e.key === "v") {
    console.log("Comando Ctrl+V detectado");
  }
});
```

Compartilhamento em Redes Sociais com JavaScript

Botões e scripts para compartilhamento:

```
<div id="share">
  <button onclick="share('facebook')">Compartilhar no Facebook</button>
  <button onclick="share('linkedin')">Compartilhar no LinkedIn</button>
  <button onclick="share('github')">Ir para o GitHub</button>
  <button onclick="share('instagram')">Instagram</button>
</div>

<script>
function share(platform) {
  const url = encodeURIComponent(window.location.href);
```

```
  let shareURL = "";
  switch (platform) {
    case "facebook":
      shareURL = `https://www.facebook.com/sharer/sharer.php?u=${url}`;
      break;
    case "linkedin":
      shareURL = `https://www.linkedin.com/shareArticle?
mini=true&url=${url}`;
      break;
    case "github":
      window.open("https://github.com/seu-usuario/seu-repositorio",
"_blank");
      return;
    case "instagram":
      alert("O Instagram não permite compartilhamento direto via web.");
      return;
  }
  window.open(shareURL, "_blank");
}
</script>
```

Script em JavaScript com Textbox e Return

Abaixo está um exemplo simples de um script JavaScript <jsia.js> que cria uma caixa de texto. O usuário digita algo, e a função buscarTexto() retorna o resultado diretamente.

HTML + JavaScript

```
<!DOCTYPE html>
<html>
<head>
  <title>Busca com Return</title>
  <style>
    body { font-family: Arial, sans-serif; padding: 20px; }
    input, button { padding: 10px; margin: 5px 0; }
    #resultado { margin-top: 15px; font-weight: bold; }
  </style>
</head>
<body>
```

```
  <h2>Busca Simples</h2>
  <input type="text" id="entrada" placeholder="Digite algo..." />
  <button onclick="mostrarResultado()">Buscar</button>
  <div id="resultado"></div>

  <script>
    function buscarTexto(texto) {
      // Simula um "banco de dados" com respostas simples
      const respostas = {
        "openai": "A OpenAI é uma empresa de pesquisa em inteligência
artificial.",
        "gpt": "GPT é um modelo de linguagem criado pela OpenAI.",
        "llm": "LLM significa Large Language Model, ou Modelo de
Linguagem de Grande Escala.",
        "chatgpt": "ChatGPT é uma aplicação do modelo GPT voltada para
conversas.",
      };

      // Converte a entrada para minúsculas e remove espaços
      const chave = texto.toLowerCase().trim();

      // Retorna a resposta ou uma mensagem padrão
      return respostas[chave] || "Nenhum resultado encontrado.";
    }

    function mostrarResultado() {
      const entrada = document.getElementById("entrada").value;
      const resultado = buscarTexto(entrada);
      document.getElementById("resultado").innerText = resultado;
    }
  </script>
</body>
</html>
```

Publicando o Protótipo no GitHub

1. Crie um repositório no GitHub.
2. Adicione os arquivos index.html, style.css, e script.js, jsia.js.
3. Faça o commit inicial:

```
git init
git add .
git commit -m "Protótipo de RichTextBox com HTML5, CSS3 e JS"
git remote add origin https://github.com/seu-usuario/seu-repositorio.git
git push -u origin master
```

1. Habilite o GitHub Pages nas configurações do repositório.
2. Acesse via: **https://seu-usuario.github.io/seu-repositorio**