

TRABALHA GESTÃO DO TEMPO

amigo bugs

Adicione documentação para API

Aumenta legibilidade
e estrutura do código

- Contribuição: Mostram evolução contínua
com correções
e melhoria no projeto

01: mensagens claras e objetivas

ex: eficazes

incluem mensagens específicas

02: Branch é uma linha de desenvolvimento
paralelo permite
trabalhar em funcionalidades sem afetar o
código estável

TRABALHO HTML

média: <https://www.bbc.com/portuguese>

tecnologia: <https://developer.mozilla.org/>

blog: <https://medium.com/>

<h?> ao <h?> - títulos e subtítulos

 - ênfase com importância

 - ênfase com entonação

<p> - parágrafo

 - linhas

2- - lista não ordenada

 - lista ordenada

<dl>, <dt>, <dd> - lista de definição

- reorganizar o conteúdo em itens claros e sequenciais

- facilitam as leituras e navegabilidade

3-<blockquote> citação longa

<q> citação curta

interatividade e Arquitetura Web Moderna

aluna: Tullyani

Introdução:

A World Wide Web, desde sua concepção, tem evoluído de um repositório estático de informações para um ambiente dinâmico e interativo, impulsionado por tecnologias que enriquecem a experiência do usuário e possibilitam a construção de aplicações complexas. O JavaScript emerge como uma linguagem fundamental nesse cenário, conferindo comportamento e interatividade às páginas web, complementando as linguagens de marcação e estilo. Paralelamente, a crescente sofisticação das aplicações online demanda metodologias robustas de organização de código, sendo os padrões de arquitetura, como o Model-View-Controller (MVC), cruciais para garantir a manutenibilidade e a escalabilidade desses sistemas. O presente ensaio tem como objetivo explorar a relevância do JavaScript na criação de efeitos visuais e na validação de dados no lado do cliente, bem como discorrer sobre o padrão arquitetural MVC e sua importância na estruturação de aplicações web modernas.

1. Efeitos Visuais Através de JavaScript:

O JavaScript desempenha um papel significativo na manipulação dinâmica do estilo dos elementos de uma página web. Através da interação com o Document Object Model (DOM), a linguagem possibilita a alteração programática de propriedades CSS, permitindo a criação de efeitos visuais em resposta a eventos iniciados pelo usuário. Simples alterações de cor, variações de tamanho e a manipulação da visibilidade de elementos são exemplos de interações que enriquecem a experiência de navegação.

Ademais, o JavaScript viabiliza a implementação de animações diretamente no navegador. Através de mecanismos como o controle temporal de alterações de estilo, é possível criar sequências visuais que conferem dinamismo e fornecem *feedback* intuitivo às ações do usuário. A utilização de bibliotecas especializadas em JavaScript para efeitos visuais simplifica o desenvolvimento de animações mais complexas e sofisticadas, oferecendo abstrações e otimizações que elevam a qualidade da apresentação visual sem demandar um esforço excessivo por parte do desenvolvedor. A integração criteriosa de efeitos visuais contribui para uma experiência do usuário mais engajadora e intuitiva.

2. Validação de Formulários com JavaScript:

A integridade dos dados inseridos em formulários online é um aspecto crucial para o correto funcionamento de aplicações web. O JavaScript possibilita a implementação de mecanismos de validação no lado do cliente, interceptando e verificando os dados fornecidos pelo usuário antes de seu envio ao servidor. Essa abordagem permite identificar e corrigir erros de preenchimento de forma imediata, otimizando a experiência do usuário e reduzindo a carga de processamento no servidor.

A validação implementada em JavaScript pode abranger desde verificações básicas, como o preenchimento de campos obrigatórios e a conformidade com tipos de dados esperados, até validações mais complexas, que empregam o uso de expressões regulares. As expressões regulares constituem uma ferramenta poderosa para a definição de padrões textuais, permitindo a verificação da conformidade de *strings* com formatos específicos, como endereços de correio eletrônico, números de telefone e códigos postais. O *feedback* imediato fornecido ao usuário sobre a validade dos dados inseridos contribui para a correção de erros e para a garantia da qualidade da informação submetida. A validação no lado do cliente, embora essencial para a experiência do usuário, deve ser complementada por validações robustas no lado do servidor para garantir a segurança e a integridade dos dados.

3. Arquitetura de Aplicações Web e o Padrão MVC:

A arquitetura de uma aplicação web define sua estrutura organizacional e a forma como seus diferentes componentes interagem. Em um modelo cliente-servidor, o navegador do usuário (cliente) comunica-se com um servidor através do protocolo HTTP para solicitar e receber recursos, como documentos HTML, folhas de estilo CSS e arquivos JavaScript. À medida que as aplicações web se tornam mais complexas, a adoção de padrões de arquitetura torna-se fundamental para a organização do código, facilitando a manutenção, a escalabilidade e a colaboração entre desenvolvedores.

O padrão Model-View-Controller (MVC) é uma metodologia arquitetural amplamente utilizada no desenvolvimento de aplicações web. Ele preconiza a separação das responsabilidades da aplicação em três partes distintas:

- **Model (Modelo):** Representa a camada de dados da aplicação e a lógica de negócios associada a esses dados. É responsável pela manipulação e pelo acesso às informações, frequentemente interagindo com sistemas de armazenamento de dados.
- **View (Visão):** Corresponde à camada de apresentação, responsável por exibir os dados ao usuário. A *view* geralmente é passiva e renderiza as informações fornecidas pelo *controller*.
- **Controller (Controlador):** Atua como um intermediário entre o *model* e a *view*. Ele recebe as requisições do usuário, invoca métodos no *model* para manipular os dados conforme necessário e, em seguida, seleciona a *view* apropriada para apresentar os resultados ao usuário.

O fluxo de interação em uma aplicação MVC tipicamente envolve a ação do usuário na *view*, que direciona uma requisição ao *controller*. O *controller*, por sua vez, interage com o *model* para obter ou modificar os dados e, posteriormente, escolhe a *view* adequada para exibir a resposta ao usuário. A adoção do padrão MVC oferece diversos benefícios, incluindo a separação de preocupações, a reutilização de código, o desenvolvimento paralelo e a melhoria da manutenibilidade e da testabilidade da aplicação.

Conclusão:

O JavaScript consolida-se como uma tecnologia essencial no desenvolvimento web moderno, transcendendo sua função inicial de manipulação do DOM para habilitar interatividade sofisticada e validação eficiente no lado do cliente. Paralelamente, a adoção de padrões arquiteturais como o MVC demonstra-se crucial para a organização e a escalabilidade de aplicações web complexas. A compreensão e a aplicação eficaz desses conceitos são fundamentais para a criação de experiências web dinâmicas, robustas e de fácil manutenção, moldando a forma como os usuários interagem com a informação e os serviços online. A contínua evolução dessas tecnologias e a exploração de novas abordagens arquiteturais certamente moldarão o futuro do desenvolvimento web.