Implementação do site Best Free Games utilizando HTML, CSS e Java Script com consumo da API Game Power.

**Daniel S. Werner, Felipe O. Silva, Peterson R. Schroeder, Sofia Schmidt.**

Cadeira de Programação Web I, Curso Técnico Informática, Universidade de Santa Cruz do Sul (UNISC) – Santa Cruz do Sul, RS – Brasil Caixa Postal 188 –96.815 – Santa Cruz do Sul – RS – Brasil

**Resumo:** O site desenvolvido faz uso de diversos conhecimentos adquiridos ao longo da disciplina, como a marcação semântica com HTML5, a estilização de elementos por meio de CSS, a implementação de objetos utilizando Java Script, bem como a criação e o consumo de APIs. Assim, foi elaborada uma base capaz de consumir uma API externa, exibindo jogos gratuitos disponíveis em diversas plataformas.

**1. Introdução**

O desenvolvimento de uma página web a partir do zero pode ser considerado uma tarefa complexa. Entretanto, com a implementação adequada de tecnologias como HTML, CSS e JavaScript, é possível alcançar resultados eficientes e esteticamente agradáveis, como a criação de um site no qual gamers e entusiastas possam localizar, de forma prática, jogos que estão disponíveis gratuitamente em diversas plataformas.

O projeto baseia-se em uma interface simples e intuitiva, permitindo ao usuário o acesso a uma biblioteca de jogos gratuitos. Adicionalmente, o sistema possibilita que o usuário favorite determinados jogos, viabilizando um acesso posterior de maneira rápida e organizada.

Para a construção da página web, foram aplicados conceitos de design responsivo, técnicas de estilização por meio de classes e a implementação da comunicação cliente-servidor, por meio do consumo de uma API externa.

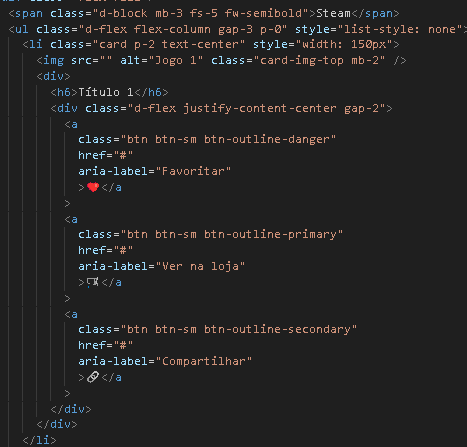
**2. Implementação**

Para a implementação do projeto, foram utilizadas as linguagens HTML, CSS e JavaScript, bem como a API Game Power, cada qual desempenhando sua respectiva função. O HTML e o CSS foram responsáveis pela estruturação e estilização da interface, proporcionando uma apresentação visual agradável e uma interação intuitiva com o usuário.

O JavaScript, por sua vez, foi empregado para adicionar funcionalidades dinâmicas ao sistema, bem como para realizar as requisições à API. A API Game Power é responsável por realizar a busca na web por jogos gratuitos, retornando os dados necessários para a exibição no site.

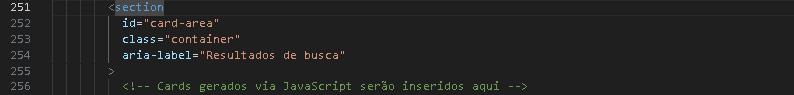
**2.1. HTML cards**

Esta seção do código HTML representa a estrutura de exibição de jogos na forma de **cards interativos**, permitindo que o usuário visualize o título e a imagem do jogo, além de oferecer ações rápidas como adicionar aos favoritos, **ver na loja** e compartilhar.

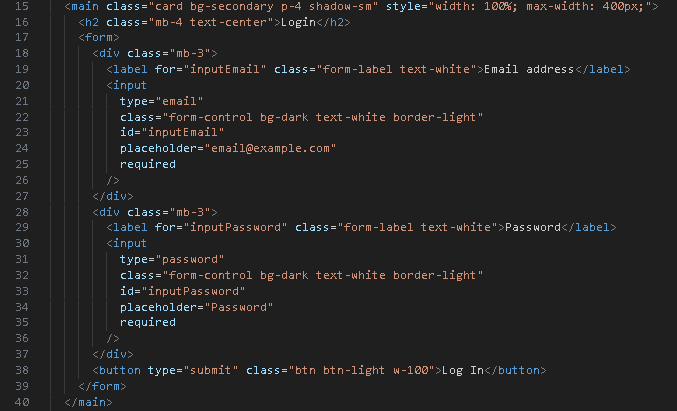


**2.2. HTML cards inserção**

O trecho de código a seguir serve como **ponto de integração com o JavaScript**, permitindo a **dinamização do conteúdo** dos cards. É por meio desse elemento que os resultados obtidos, como imagens, títulos e links, são inseridos dinamicamente na interface.

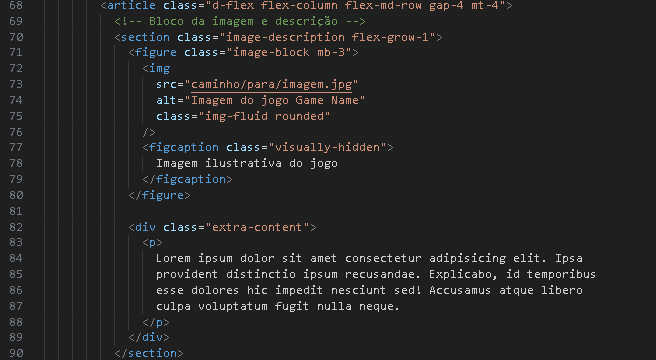
****

**2.2. HTML Login**

O trecho de código apresentado integra a aba de login do website, denominada login.html. Essa seção tem como principal objetivo coletar os dados do usuário, possibilitando que ele favorite os jogos disponíveis na plataforma.****

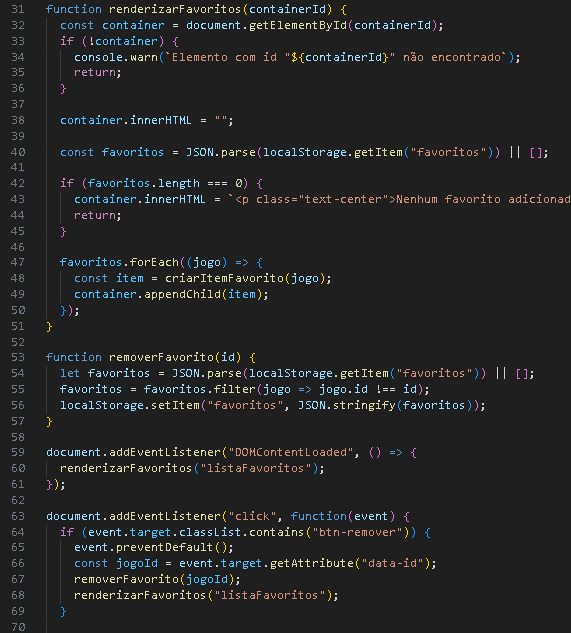
**2.3. HTML gamepage**

Este código HTML cria uma página responsiva para o site "Best Free Games", exibindo detalhes de um jogo com imagem, descrição e botões de interação. Com seu conteúdo principal com informações do jogo e um rodapé com links úteis, contatos e redes sociais. Utiliza Bootstrap para estilo e responsividade, além de um CSS personalizado.

****

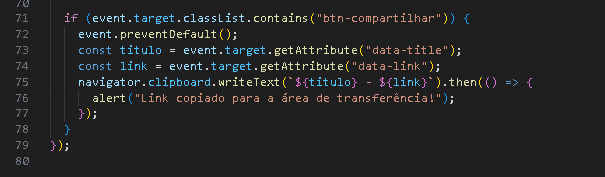
**2.4. 1. Função JS favoritos**

Este trecho implementa a função renderizar favoritos(containers), responsável por **exibir dinamicamente os jogos favorita dos** em um elemento HTML identificado por id. A função limpa o conteúdo atual, recupera os dados armazenados no localStorage e, caso existam favoritos, **cria e adiciona elementos** para cada um deles na interface. Se não houver favoritos, exibe uma mensagem informativa ao usuário.

****

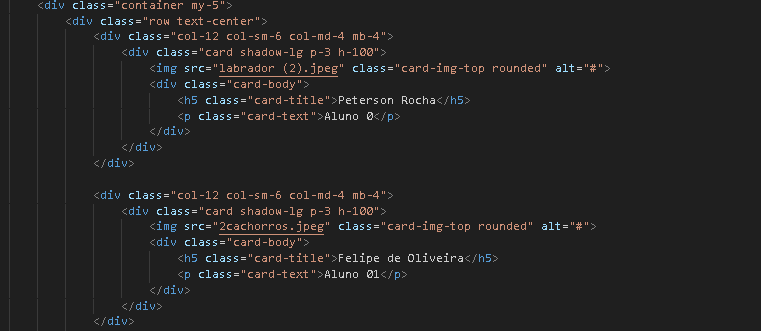
**2.4. 2. Função JS favoritos**

Esta segunda parte do código adiciona a funcionalidade de **compartilhar links**. Quando o usuário clica no botão de compartilhar, o código **impede a ação padrão**, recupera o **título e o link** do jogo, e copia essas informações para a **área de transferência** utilizando a API do navegador. Por fim, exibe um **alerta** confirmando que o link foi copiado com sucesso.

****

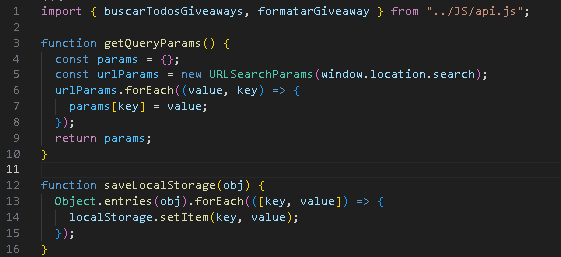
**2.5. Desenvolvedores**

A página dos desenvolvedores é **simples e estática**, composta apenas por **cards HTML** que apresentam a **equipe de desenvolvimento** com nome, imagem e descrição básica de cada membro, utilizando uma **estrutura responsiva** com **Bootstrap** para garantir a **boa exibição em diferentes dispositivos**.



**2.6.1 . requisição de URL e salvamento**

O código captura os parâmetros da URL e os salva no localStorage como pares chave-valor. Isso permite armazenar dados como preferências, identificadores ou referências de campanha, mantendo essas informações mesmo após recarregar a página. É útil para personalizar e rastrear a experiência do usuário.

****

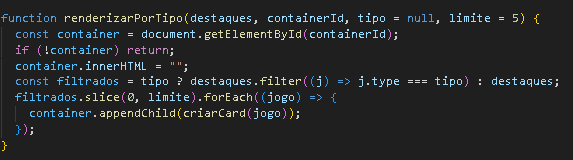
**2.6.2. Criação de cards dinâmicos**

A função criarCard(jogo) cria um elemento visual (card) para exibir informações de um jogo, como imagem e título. Dependendo do tipo (Game ou DLC), adiciona botões de favoritar, ver na loja e compartilhar. O HTML é gerado dinamicamente com base nos dados do objeto jogo. Por fim, retorna o card pronto para ser inserido na página.

****

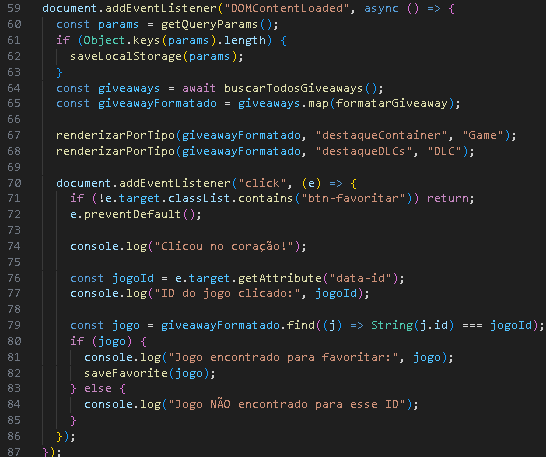
**2.6.3. renderização dos cards dinâmicos**

Renderiza no DOM até limite cards de jogos, filtrando por tipo (ex: "Game", "DLC") no container com ID especificado. Se o tipo for nulo, exibe todos os jogos disponíveis. Antes de inserir os cards, limpa o conteúdo anterior do container para evitar duplicações. Ideal para destacar diferentes categorias de jogos em seções separadas

****

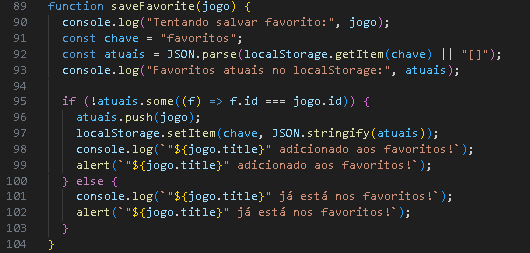
**2.6.4. Inicialização da pagina após o carregamento**

Ao carregar o DOM para executar ações. Ele coleta parâmetros da URL e os salva no localStorage, busca e formata giveaways, e os renderiza por tipo. Adiciona um listener de clique para lidar com a funcionalidade de favoritar jogos. Ao clicar no botão de favoritar, identifica o jogo correspondente e o salva como favorito.

****

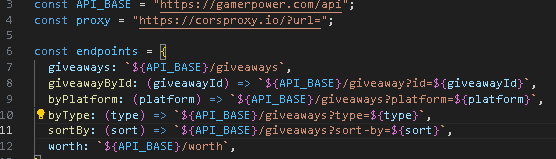
**2.6.5. Função salvar favoritos**

A função saveFavorite(jogo) salva um jogo nos favoritos no localStorage, garantindo que ele não seja duplicado. Se o jogo ainda não estiver na lista, ele é adicionado e uma mensagem de confirmação é exibida; caso já exista, apenas uma mensagem informativa é mostrada.



**2.7.1. Configurações api.js**

Esse trecho define a URL base da API do GamerPower e um proxy para evitar problemas de CORS. Também cria um objeto endpoints com funções e URLs para acessar diferentes recursos da API, como todos os giveaways, por ID, por plataforma, tipo, ordenação e valor estimado.



**2.7.2. utilitarios api.js**

Esse trecho de código define funções utilitárias para manipulação de dados de giveaways. A função normalizarTexto remove espaços extras e converte o texto para letras minúsculas, útil para comparações. A fetchJson busca dados JSON de uma URL, utilizando um proxy para evitar problemas de CORS, e lança um erro caso a resposta não seja bem-sucedida. A função formatarGiveaway simplifica os objetos recebidos da API, mantendo apenas as propriedades relevantes como título, imagem, descrição e tipo. Por fim, exibirGiveaways percorre uma lista de giveaways, formata cada item e imprime seu título no console.

****

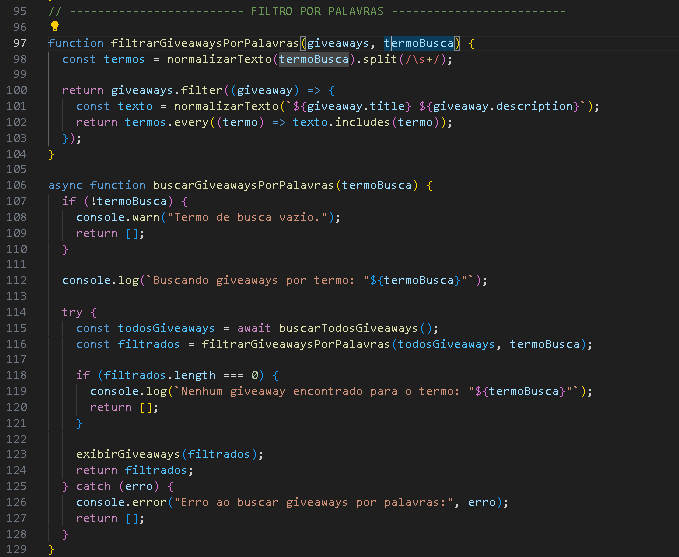
**2.7.3. buscas ­­­ api.js**

Essas funções assíncronas são responsáveis por buscar dados de giveaways na API do GamerPower. A função buscarTodosGiveaways recupera todos os giveaways disponíveis e exibe seus títulos no console. Já buscarGiveawayPorId busca um giveaway específico com base no ID fornecido. A função buscarGiveawaysPorPlataforma retorna os giveaways filtrados por plataforma, como PC ou Steam, enquanto buscarGiveawaysPorTipo busca os giveaways por tipo, como Game ou DLC, também exibindo os resultados. Todas as funções possuem tratamento de erro e, em caso de falha, retornam valores seguros como array vazio ou null.

****

**2.7.4. filtro de palavras ­ api.js**

As funções buscam giveaways por palavras-chave, filtrando aqueles cujo título ou descrição contenham todos os termos informados. Se não houver resultados ou ocorrer erro, retornam um array vazio.

****

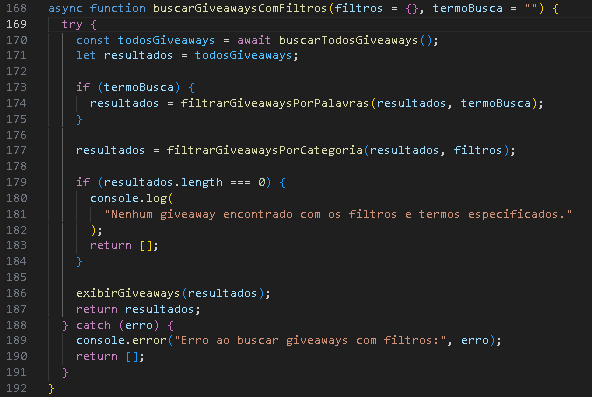
**2.7.5. filtro de categorias api.js**

Essa função filtra giveaways com base em critérios de plataforma, tipo e status. Ela retorna apenas os giveaways que atendem a todos os filtros informados.



**2.7.6. busca completa api.js**

A função buscarGiveawaysComFiltros busca todos os giveaways, filtra por palavras-chave (se fornecidas) e aplica filtros de categoria (plataforma, tipo e status). Se não encontrar resultados ou ocorrer erro, retorna um array vazio.

****

**2.7.7. Salvar no local storage api.js**

A função salvarGiveawayNoLocalStorage formata um giveaway e o salva no localStorage usando o título como chave. O bloco export disponibiliza várias funções relacionadas à busca, filtragem, formatação e salvamento de giveaways para uso em outros módulos do projeto.

