Projeto 1: Pichu – Pokelist TCG

Erick Cauã Ferreira Gomes – 10238824

Guilherme de Arruda Leme – 10396048

João Victor Pacheco Ferreira - 10390689

Victor Junqueira Colombaro – 10395711

Durante o processo de ideação, o grupo se reuniu para explorar e definir as melhores abordagens para o projeto. Inicialmente, realizamos uma sessão de brainstorming onde cada integrante trouxe ideias e referências que pudessem atender aos requisitos propostos. Durante essas discussões, identificamos que o universo Pokémon oferecia um tema cativante, repleto de elementos interativos que poderiam ser explorados para demonstrar o uso de HTML5, CSS3 e JavaScript. Como ideia final, resolvemos criar um site onde o usuário pode criar um marcador de quantas cartas possui.

Como protótipo inicial, criamos um template no figma:

Interface gráfica do usuário, Aplicativo, Site

O conteúdo gerado por IA pode estar incorreto.

A ideia era ter algo simples e que fosse fácil do usuário final entender como funciona.

Explicação do código:

Index.html:  
Texto

O conteúdo gerado por IA pode estar incorreto.

Código html básico, linkando o arquivo style.css e definindo título da página e icon. No corpo, definimos o header onde tem o logo do Pokémon TCGP e a tag main com a tag section, que será populada com o conteúdo do site via Javascript.

Script.js:

O código contém um dicionário com as informações dos pokémons exibidos no site. Após o dicionário, obtemos o elemento com o id “cards-container”, onde iremos adicionar o conteúdo do site.



Depois, declaramos a função que renderiza os cards, criando para cada objeto na lista inicial, a div com a imagem e os botões para adição e subtração da quantidade de cards. Também adicionamos um listener para clicks na imagem do card, que envia o usuário para a página de detalhes do pokémon, passando o nome do pokémon como parâmetro na URL.

function renderCards() {

    // Limpa o container para evitar duplicações

    cardsContainer.innerHTML = "";

    cardsData.forEach((*card*, *index*) => {

        const cardDiv = document.createElement("div");

        cardDiv.classList.add("main\_\_section\_\_card");

        // Cria a imagem

        const img = document.createElement("img");

        img.src = *card*.src;

        img.alt = *card*.alt;

        // Adiciona o event listener para redirecionar ao clicar na imagem

        img.addEventListener("click", () => {

            // Redireciona para a página de detalhes passando o nome do Pokémon na URL

            window.location.href = `detalhes.html?pokemon=${encodeURIComponent(*card*.pokemon)}`;

        });

        cardDiv.appendChild(img);

        const info = document.createElement("div");

        info.classList.add("main\_\_section\_\_card\_\_info")

        // Exibe o nome do Pokémon

        const name = document.createElement("h3");

        name.textContent = *card*.card\_name;

        cardDiv.appendChild(name);

        const btnMenos = document.createElement("button");

        btnMenos.textContent = "-";

        btnMenos.classList.add("main\_\_section\_\_card\_\_info\_\_btn")

        btnMenos.addEventListener("click", () => {

            if(cardsData[*index*].quantidade > 0) {

              cardsData[*index*].quantidade--;

              quantidadeDisplay.textContent = `${cardsData[*index*].quantidade}`;

            }

        });

        const quantidadeDisplay = document.createElement("p");

        quantidadeDisplay.textContent = `${*card*.quantidade}`;

        const btnMais = document.createElement("button");

        btnMais.textContent = "+";

        btnMais.classList.add("main\_\_section\_\_card\_\_info\_\_btn")

        btnMais.addEventListener("click", () => {

          cardsData[*index*].quantidade++;

          quantidadeDisplay.textContent = `${cardsData[*index*].quantidade}`;

        });

        info.appendChild(btnMenos);

        info.appendChild(quantidadeDisplay);

        info.appendChild(btnMais);

        cardDiv.appendChild(info);

        cardsContainer.appendChild(cardDiv);

    });

}

renderCards();

Detalhe.html:

O arquivo segue o mesmo padrão do index.html, com a diferença da troca de id no section. Todo o resto se mantém.

Detalhe\_script.html:

O código declara a função que, após o código HTML ser renderizado, exibe a palavra “Carregando” enquanto o resto do código é executado. Após isso, o código obtém o nome do pokémon passado na URL, realiza a consulta na API de pokémons para obter mais detalhes sobre o pokémon e exibe as informações na página. As informações exibidas são o ID, altura, peso, tipos e habilidades. A página também exibe a imagem oficial do pokémon.

Com o desenvolvimento do projeto, aprendemos os conceitos básicos de HTML, CSS e Javascript. Aprendemos a manipular o documento com o uso de Javascript para deixar o site interativo. Também aprendemos a realizar consultas em APIs de terceiros e a trabalhar com requisições HTTP, utilizando Javascript.