

Objetivo

O objetivo deste trabalho é desenvolver uma aplicação web que funcione como uma **caderneta interativa de exercícios**, onde os desafios propostos serão resolvidos dentro de uma estrutura organizada e visualmente atrativa.

Este projeto integra conhecimentos já estudados (HTML e CSS) e introduz **novos conceitos com JavaScript**, incluindo manipulação do DOM, eventos, estruturas de decisão e lógica de programação. É uma oportunidade de aplicar e expandir seus conhecimentos, construindo um portfólio com soluções interativas.

Tecnologias utilizadas

- HTML5
 - CSS3
 - JavaScript (com foco em DOM e lógica)
 - Uso de <iframe> para carregar exercícios em páginas separadas
 - Organização de arquivos em pastas e estrutura de navegação
-

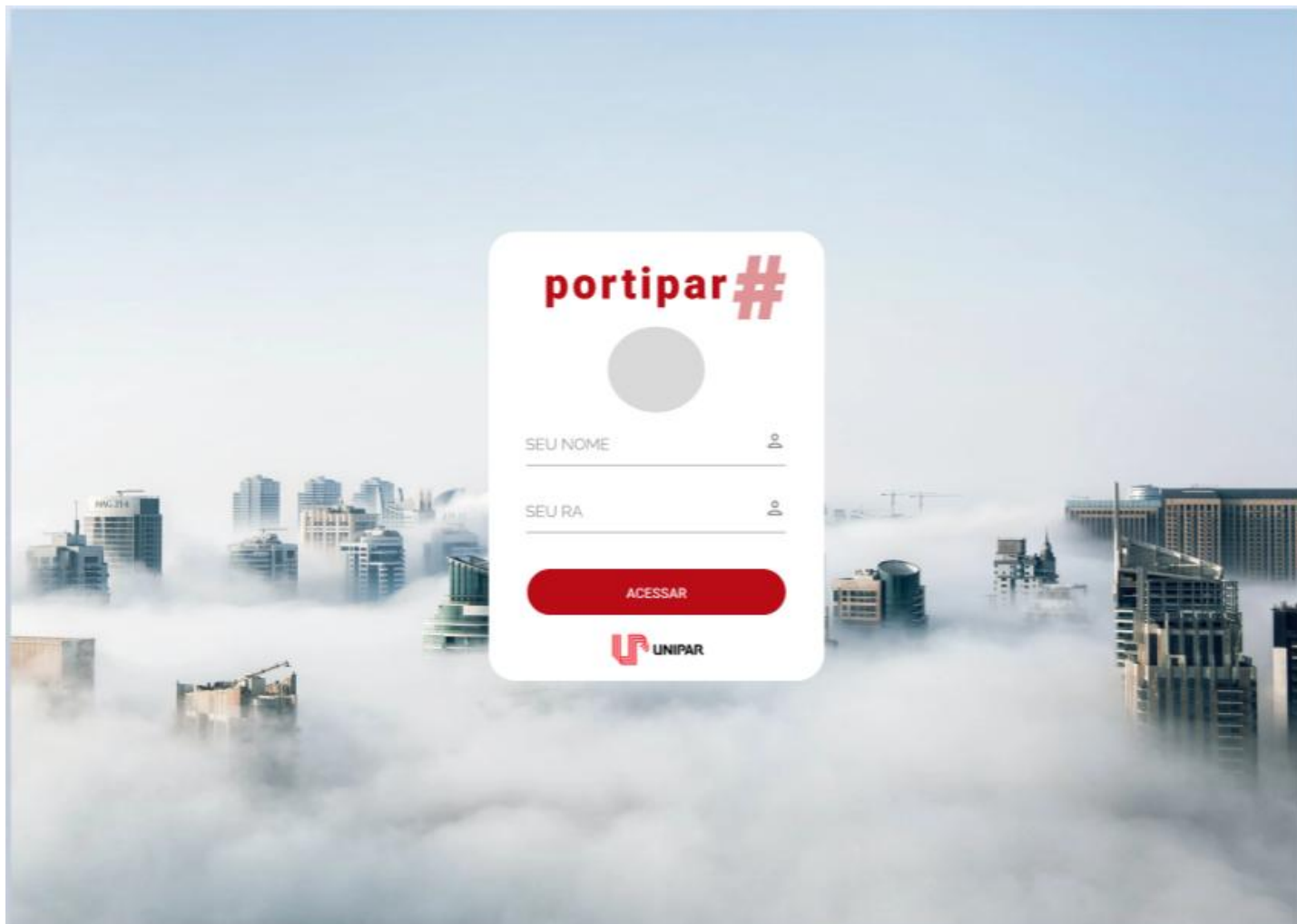
Descrição do projeto

Você deverá criar um site com:

- Uma **tela inicial**, com título, introdução do projeto e um botão "Acessar ", deve seguir o layout proposto na página 2.
- Uma **página de portfólio**, com **cards ou botões** representando os exercícios, após acessr dever ter uma página o qual o aluno vai ficar livre para montar um layout o qual precisa ter dois espaços, um espaço para a lista de atividade e outro para o iframe.
- Cada exercício deverá ser criado em um arquivo HTML, CSS, JAVASCRIPT
- Cada exercício será carregado em uma nova tela utilizando **iframe**;
- Os desafios devem ser resolvidos utilizando **JavaScript com manipulação do DOM** e lógica.

LAYOUT TELA INICIAL

- Uma **tela inicial**, com título, introdução do projeto e um botão "Acessar ", deve seguir o layout proposto na página 2. Não precisa exatamente ter as mesmas imagens, mas deve conter, seu nome, seu RA, o título portipar# e a logo da Unipar.
- Essa página deve ter obrigatoriamente os nomes de arquivo, index.html, Style.css e Scripts.js.



LAYOUT DAS ATIVIDADES

REGRA - Cada atividade deve conter o título, enunciado e uma explicação de como foi resolvida e data que foi resolvido. O LAYOUT deve ser bem-organi-zado, lembre-se do uso do iframe

1. Calculadora de IMC

Descrição:

Crie uma página onde o usuário possa digitar seu peso (kg) e altura (m). Ao clicar em um botão, o sistema deve calcular o Índice de Massa Corporal (IMC) usando a fórmula $\text{peso} / (\text{altura}^2)$ e mostrar o resultado na tela, indicando se o usuário está abaixo do peso, com peso normal, sobrepeso ou obesidade.

Skills JS: getElementById, addEventListener, operadores matemáticos, estruturas condicionais (if / else)

Dificuldade: 1

2. Calculadora de Média com Avaliação

Descrição:

Crie um formulário com três campos para entrada das notas de um aluno. Após clicar em "Calcular", o sistema deve calcular a média das notas e mostrar uma mensagem que indique se o aluno está aprovado (média ≥ 7), em exame (média entre 5 e 6.9) ou reprovado (média < 5).

Skills JS: manipulação de inputs, parseFloat, eventos de clique, lógica condicional,.textContent para atualizar a tela

Dificuldade: 1

3. Tabuada Interativa

Descrição:

Permita que o usuário insira um número e, ao clicar em um botão, gere e mostre a tabuada desse número de 1 a 10. A tabuada deve ser exibida de forma clara e organizada na página.

Skills JS: laços (for), manipulação do DOM com innerHTML, eventos, querySelector

Dificuldade: 1

4. Conversor de Temperatura (Celsius \leftrightarrow Fahrenheit)

Descrição:

Crie uma página que permita ao usuário inserir uma temperatura em Celsius e, ao clicar em "Converter", mostrar a temperatura convertida para Fahrenheit. Adicione também o caminho

inverso, de Fahrenheit para Celsius.

Skills JS: manipulação de formulários, eventos, operações matemáticas, DOM para mostrar resultado

Dificuldade: 1

5. Validador de CPF (simples)

Descrição:

Desenvolva uma página onde o usuário possa inserir um número de CPF (apenas dígitos). Ao clicar em "Validar", o sistema deve verificar se o CPF tem 11 dígitos e mostrar uma mensagem informando se é válido ou não (validação simples, apenas quantidade de números).

Skills JS: manipulação de strings, verificação de tamanho, eventos, condicionais

Dificuldade: 1

6. To-Do List

Descrição:

Crie uma lista de tarefas onde o usuário possa digitar uma nova tarefa em um campo de texto e clicar em "Adicionar". A tarefa deve aparecer na lista abaixo com opções para marcar como concluída (riscar o texto) e para remover a tarefa.

Skills JS: createElement, appendChild, manipulação do DOM, eventos de clique, classList para alterar estilos, remoção de elementos

Dificuldade: 2

7. Jogo da Adivinhação

Descrição:

Implemente um jogo onde o computador sorteia um número de 1 a 100, e o usuário deve tentar adivinhar. Após cada palpite, o sistema informa se o número digitado é maior ou menor que o sorteado. Quando o usuário acertar, uma mensagem de parabéns deve aparecer junto com o número de tentativas realizadas.

Skills JS: Math.random(), prompt e alert (ou inputs na página), loops (while ou recursão), condicionais, variáveis para contagem

Dificuldade: 2

8. Verificador de Números Primos

Descrição:

Crie uma página que permita ao usuário inserir um número e verificar se ele é primo ou não.

Mostre o resultado na tela com uma mensagem adequada.

Skills JS: laços (for), condicionais, manipulação de inputs e outputs no DOM

Dificuldade: 2

9. Lista de Compras

Descrição:

Desenvolva um sistema simples onde o usuário possa adicionar itens à lista de compras, visualizar os itens adicionados e excluir itens específicos. Os itens devem ser exibidos dinamicamente conforme adicionados ou removidos.

Skills JS: criação dinâmica de elementos, manipulação do DOM, eventos, arrays para armazenar itens

Dificuldade: 2

10. Sorteador de Nomes

Descrição:

Faça uma aplicação onde o usuário insira vários nomes em um campo e, ao clicar em "Sortear", o sistema escolha aleatoriamente um nome para ser exibido.

Skills JS: manipulação de arrays, eventos, uso de Math.random(), entrada e saída de dados no DOM

Dificuldade: 2

11. Simulador de Caixa Eletrônico

Descrição:

Crie um simulador simples que permita ao usuário inserir um valor para saque. O sistema deve calcular e mostrar a quantidade de notas necessárias para o valor solicitado, considerando notas de 100, 50, 20, 10, 5 e 2 reais.

Skills JS: operadores matemáticos, laços (while ou for), condicionais, manipulação de DOM

Dificuldade: 2

12. Quiz de Perguntas

Descrição:

Implemente um quiz com 3 a 5 perguntas de múltipla escolha. O usuário deve selecionar as respostas e, ao final, o sistema deve mostrar a pontuação total.

Skills JS: manipulação de formulários, arrays, condicionais, manipulação do DOM para mostrar

perguntas e resultado

Dificuldade: 2

13. Jogo da Velha (2 jogadores)

Descrição:

Construa o clássico jogo da velha para dois jogadores. Os jogadores devem poder clicar nas células para marcar X ou O. O sistema deve verificar vitória, empate e alternar turnos. Exiba mensagens no fim do jogo e possibilidade de reiniciar.

Skills JS: arrays multidimensionais, manipulação de eventos, condicionais complexas, DOM para atualizar a interface

-

14. Cronômetro com Iniciar/Pausar/Resetar

Descrição:

Desenvolva um cronômetro com botões para iniciar, pausar e resetar a contagem de tempo, exibindo minutos e segundos atualizados a cada segundo.

Skills JS: setInterval, clearInterval, manipulação de eventos, atualização dinâmica do DOM

Dificuldade: 3

15. Controle de Visibilidade (toggle)

Descrição:

Faça uma página onde o usuário possa esconder e mostrar diferentes seções de conteúdo clicando em botões. Utilize transições suaves para melhorar a experiência visual.

Skills JS: manipulação de classes com classList.toggle, eventos, CSS para transições, DOM para alterar estilos

Dificuldade: 3

16. Validador de Senha Forte

Descrição:

Crie um formulário onde o usuário digite uma senha e, ao enviar, o sistema verifica se a senha contém ao menos 8 caracteres, uma letra maiúscula, uma letra minúscula, um número e um caractere especial. Mostre mensagens indicando quais critérios foram ou não atendidos.

Skills JS: expressões regulares (RegExp), manipulação de strings, condicionais, DOM para mostrar mensagens

Dificuldade: 2

17. Galeria de Imagens com Filtro

Descrição:

Construa uma galeria de imagens com categorias (ex: natureza, tecnologia, pessoas). Crie botões para filtrar e mostrar apenas imagens da categoria selecionada.

Skills JS: manipulação do DOM, arrays, eventos, classList para mostrar/esconder elementos

Dificuldade: 3

18. Contador Regressivo (Timer)

Descrição:

Desenvolva um contador regressivo onde o usuário informe o tempo em minutos e segundos. Ao iniciar, o sistema deve mostrar a contagem regressiva até zero, emitindo uma mensagem ao final.

Skills JS: setInterval, manipulação de eventos, manipulação do DOM, lógica de decremento

Dificuldade: 3

19. Editor de Texto Simples

Descrição:

Crie um editor de texto simples na página, onde o usuário pode digitar texto e aplicar formatações básicas como negrito, itálico e sublinhado clicando em botões.

Skills JS: manipulação do DOM, eventos, uso de document.execCommand (ou alternativas), estilização CSS

Dificuldade: 3

20. Jogo de Memória (Cartas)

Descrição:

Implemente um jogo da memória simples com cartas viradas para baixo. Ao clicar em duas cartas, elas devem virar para cima. Se forem iguais, permanecem visíveis; se não, viram para baixo novamente. O jogo termina quando todas as cartas forem encontradas.

Skills JS: manipulação do DOM, arrays, lógica de comparação, temporizadores, eventos

Dificuldade: 3

