# Exercício Final

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamenteFaça uma lista de tarefas que salve as tarefas do usuário na tela, com funcionalidades de apagar uma tarefa criada e salvar tarefas criadas e excluídas em *localStorage*, para que, sempre ao abrir a página novamente, continuar sendo exibidas tarefas que não foram apagadas

Caso dúvidas, o código de resolução estará ao final do documento...

# Resolução

## HTML

<!DOCTYPE *html*>

<html *lang*="pt-BR">

<head>

  <meta *charset*="UTF-8">

  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

  <meta *http-equiv*="X-UA-Compatible" *content*="ie=edge">

  <title>Lista de Tarefas</title>

  <link *rel*="stylesheet" *href*="./assets/css/style.css">

</head>

<body>

  <section *class*="container">

    <h1>Lista de Tarefas</h1>

    <p>

      <input *type*="text" *class*="input-tarefa">

      <button *class*="btn-adicionar">Inserir Tarefa</button>

    </p>

    <ul *class*="lista">

    </ul>

  </section>

  <script *src*="./assets/js/main.js"></script>

</body>

</html>

## CSS

@import url('https://fonts.googleapis.com/css?family=Open+Sans:400,700&display=swap');

*:root* {

  --primary-color: rgb(17, 86, 102);

  --primary-color-darker: rgb(9, 48, 56);

}

\* {

  box-sizing: border-box;

  outline: 0;

}

body {

  margin: 0;

  padding: 0;

  background: var(--primary-color);

  font-family: 'Open sans', sans-serif;

  font-size: 1.3em;

  line-height: 1.5em;

}

*.container* {

  max-width: 640px;

  margin: 50px auto;

  background: #fff;

  padding: 20px;

  border-radius: 10px;

}

form input, form label, form button {

  display: block;

  width: 100%;

  margin-bottom: 10px;

}

form input {

  font-size: 24px;

  height: 50px;

  padding: 0 20px;

}

form input*:focus* {

  outline: 1px solid var(--primary-color);

}

form button {

  border: none;

  background: var(--primary-color);

  color: #fff;

  font-size: 18px;

  font-weight: 700;

  height: 50px;

  cursor: pointer;

  margin-top: 30px;

}

form button*:hover* {

  background: var(--primary-color-darker);

}

## JavaScript

const inputTarefa = document.querySelector('.input-tarefa')

const btnAdicionar = document.querySelector('.btn-adicionar')

const lista = document.querySelector('.lista')

function botaoApagar(*li*){

*li*.innerText += ' '

    const botaoApagar = document.createElement('button')

    botaoApagar.innerText = 'Apagar Tarefa'

    li.appendChild(botaoApagar)

    botaoApagar.setAttribute('class', 'apagar')

}

function criarTarefa(*tarefa*){

    const li = document.createElement('li')

    li.innerText = tarefa

    lista.appendChild(li)

    botaoApagar(li)

    salvarTarefa()

}

function limparTarefa(){

    inputTarefa.value = ''

    inputTarefa.focus()

}

function salvarTarefa(){

    const tarefas = lista.querySelectorAll('li')

    const listaDeTarefas = []

    for (let tarefa of tarefas){

        let tarefaTexto = tarefa.innerText

        tarefaTexto = tarefaTexto.replace('Apagar Tarefa', '').trim()

        listaDeTarefas.push(tarefaTexto)

    }

    const tarefasJSON = JSON.stringify(listaDeTarefas)

    localStorage.setItem('tarefas', tarefasJSON)

}

function adicionaTarefasSalvas(){

    const tarefas = localStorage.getItem('tarefas')

    const listaDeTarefas = JSON.parse(tarefas)

    for(let tarefa of listaDeTarefas){

        criarTarefa(tarefa)

    }

}

inputTarefa.addEventListener('keypress', function(*e*){

    if(e.keyCode === 13){

        if (!inputTarefa.value) return

        criarTarefa(inputTarefa.value)

        limparTarefa()

    }

})

btnAdicionar.addEventListener('click', function(*e*){

    if (!inputTarefa.value) return

    criarTarefa(inputTarefa.value)

    limparTarefa()

})

document.addEventListener('click', function(*e*){

    const el = e.target

    if(el.classList.contains('apagar')){

        el.parentElement.remove()

        salvarTarefa()

    }

})

adicionaTarefasSalvas()