

Avaliação 2

Para a avaliação 2 você deverá implementar um sistema de "Mensagens" de acordo com os seguintes requisitos e protótipos:

Mais abaixo você vai encontrar as imagens completas do sistema que eu quero que você implemente. Assim como as informações de fontes e cores que devem ser utilizadas.

- Um cabeçalho com a logo do sistema e o título (vou disponibilizar a imagem)
 - A logo deve ter `80px` de largura e a altura automática



- No cabeçalho também deverá ter um menu com os itens distribuídos de forma uniforme na barra de menu:



Note que tem um fundo branco, cor do texto azul e uma borda azul na parte superior e que os cantos superiores estão arredondados em `5px`

- Ao posicionar o mouse sobre um dos itens de menu (`:hover`) o estilo do item de menu deve ser apresentado de acordo com a imagem abaixo:



Note que o texto agora está em branco e em negrito em um fundo azul, mesma cor da borda superior.

- Ao clicar em qualquer item do menu o usuário deverá ser redirecionado para uma outra página sendo que esta página é aberta em uma *nova aba* do navegador
- Esta nova página deverá ter o texto em um elemento `h3` na cor branca e centralizado na página (Ver **Imagem 5** mais abaixo) (use flexbox layout).
- O sistema será composto por vários "cards", sendo que ao carregar a página há apenas um card, cujo conteúdo é um formulário com campos para informar um título para uma mensagem e o texto para a mensagem que irá compor um novo card na página (Imagem 1 mai):

- Quando o usuário clicar em "Salvar" o sistema deverá adicionar na tela um novo card com o título e o texto informado.
- O formulário deverá ter a ação de "Salvar" e de "Limpar" (`submit` e `reset`)
- Conforme novas mensagens são salvas pelo usuário novos cards devem ser adicionados a página.
- Ao clicar em "Salvar", depois de adicionar o card com as informações, os campos do formulário devem ser resetados
 - Dica: `elementoFormulario.reset();`
- A página então será apresentada conforme a **Imagem 2** mais abaixo.
- Utilize `grid` layout para organizar os cards na tela
 - `grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));`
- Deverá ser possível remover um card criado pelo usuário
- Quando houver muitos card e criar barra de rolagem ao rolar a página o cabeçalho (logo, nome e menu) devem ficar sempre visíveis na tela, ou seja, devem ficar "grudados" no topo da página, conforme **Imagem 3** mais abaixo.
 - Dica: Coloque o `background-color` do header a mesma cor de fundo da página
- O sistema deverá ser responsivo e em tela menores do que `600px` (mobile), deverá ser apresentada conforme a **Imagem 4** mais abaixo
 - O menu não deve ser apresentado em telas mobile
 - O título da página não deve ser apresentado, apenas a logo, em telas mobile
 - A grid com os cards devem apresentar um card por linha (o código css que eu passei acima já faz isso)
 - Use media query para implementar a responsividade do cabeçalho

Você deverá implementar todo o estilo da página, a responsividade e também o comportamento utilizando JavaScript.

Para te ajudar vou te dar **de graça** alguns códigos:

- Código HTML do card com o formulário:

```
<div class="card">
  <p class="title">Adicionar mensagem</p>
  <div class="conteudo">
    <form id="criar-mensagem">
      <p>
```

```

        <input id="input-titulo" type="text" placeholder="Titulo">
    </p>
    <p>
        <textarea id="input-mensagem" cols="30" rows="10" placeholder="Mensagem"></textarea>
    </p>
    <button type="submit">Salvar</button>
    <button type="reset">Limpar</button>
    </form>
</div>
</div>

```

- Código CSS com o estilo dos campos de input:

```

#criar-mensagem input, #criar-mensagem textarea {
    border: none;
    appearance: none;
    background: #f2f2f2;
    padding: 10px;
    border-radius: 5px;
    width: calc(100% - 30px);
    font-size: 14px;
    resize: none;
}

#criar-mensagem input:focus-visible,
#criar-mensagem textarea:focus-visible {
    outline-color: #079ae4;
}

```

- Código JavaScript que adiciona a ação "Remover" ao card criado:

```

function adicionarAcaoRemover(card) {
    const btnRemover = document.createElement('a');
    btnRemover.className = 'btn-remover';
    btnRemover.innerText = 'Remover';
    btnRemover.onclick = event => {
        event.preventDefault();
        card.remove();
    }
    card.appendChild(btnRemover);
}

```

- Código CSS com a estilização do botão "Remover" do card:

```

.btn-remover {
    color: #fd6652;
    background-color: inherit;
    text-decoration: none;
    cursor: pointer;
    margin: 10px;
    display: inline-block;
}

.btn-remover:hover {

```

```
font-weight: bold;  
}
```

P.S: Para os cantos arredondados utilize `5px` de `radius`.

Fonte

A fonte que deve ser utilizada no sistema é a fonte **Roboto** do google, que pode ser encontrada aqui: <https://fonts.google.com/specimen/Roboto>

Cores

- `#075982`
 - Cor de fundo da página principal
 - Cor de fundo da página auxiliar
- `#079ae4`
 - Borda superior do menu
 - Cor dos itens de menu
 - Cor de fundo do título dos cards
 - Cor de fundo dos botões
- `#FFFFFF`
 - Cor branca

Logo

A logo do sistema foi disponibilizada em um arquivo chamado `m.png`

Imagens

Imagem 1 - Imagem inicial da página ao ser carregada

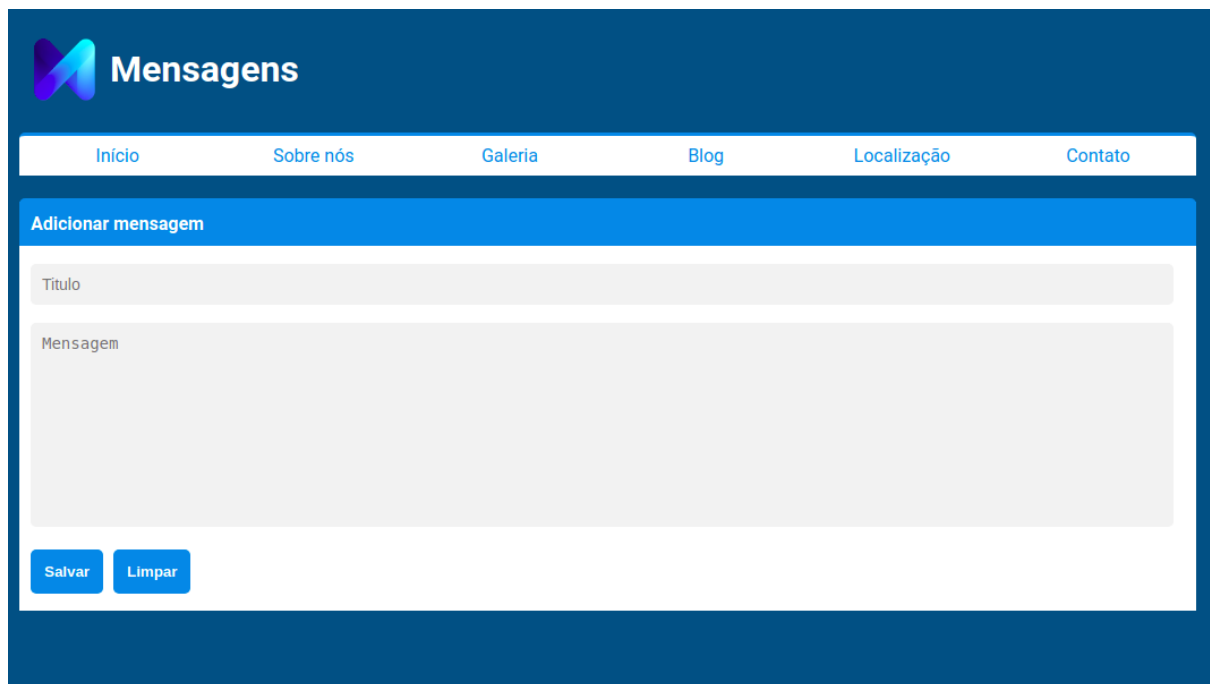


Imagem 2 - Duas mensagens adicionadas pelo usuário. Dois cards adicionado pelo sistema

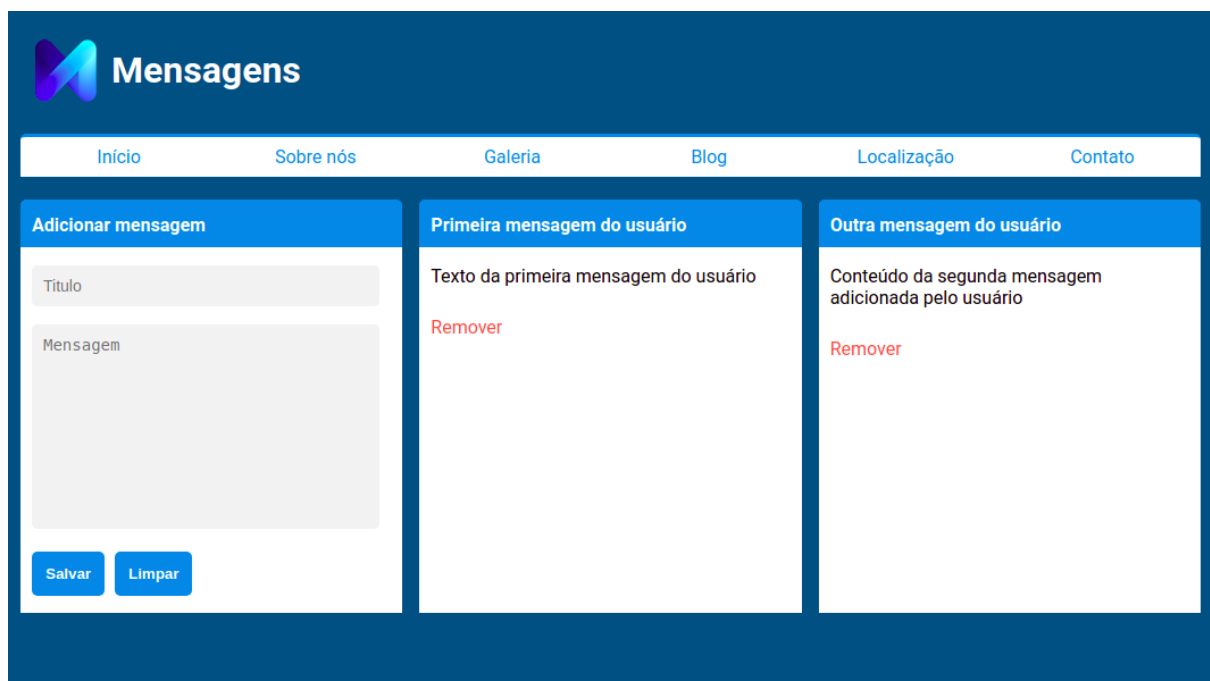


Imagem 3 - Cabeçalho fixo na pagina (grudado)

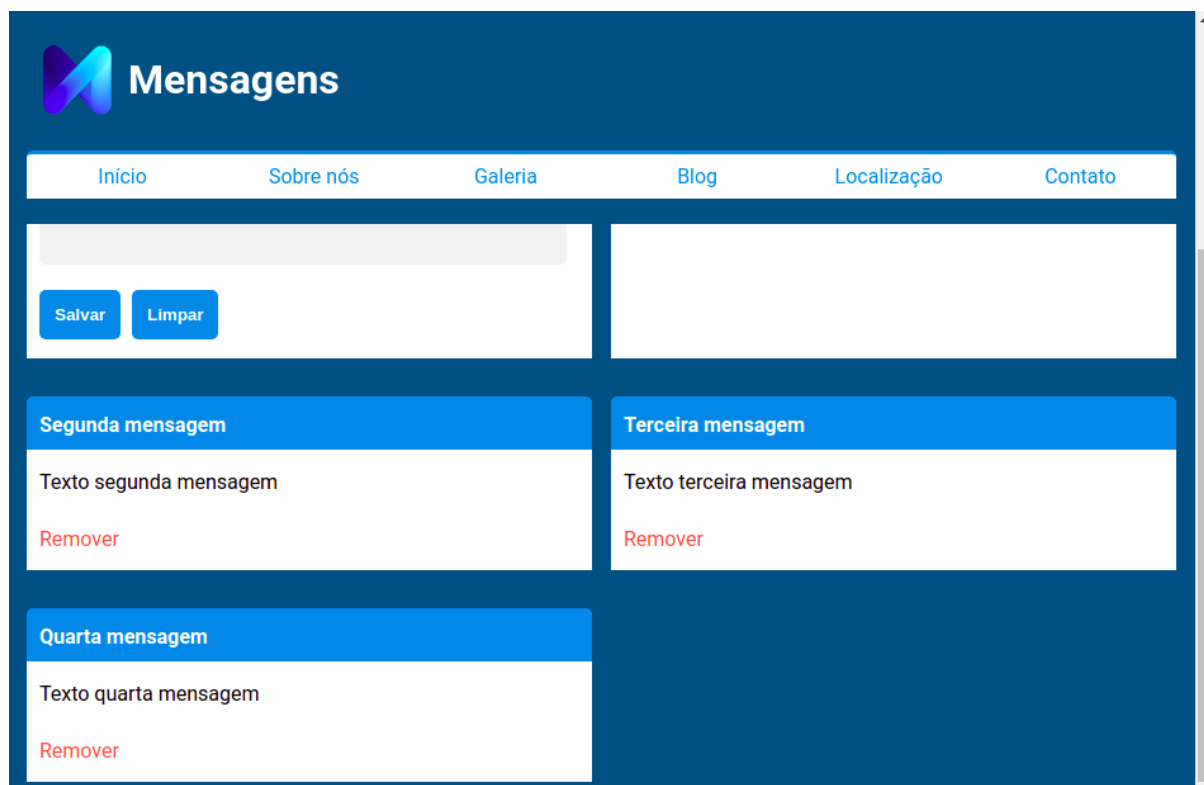



Imagem 4 - Página em tela menos de 600px



Adicionar mensagem

Primeira mensagem do usuário

Imagem 5 - Página auxiliar

Item de menu aberto em uma nova página

Essa avaliação está muito fácil... Divirta-se!