UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR
SISTEMAS PARA INTERNET - SI

MARIA EDUARDA GUEDES DE OLIVEIRA

APLICAÇÃO DE UMA LOJA ONLINE SOBRE VENDA DE CERVEJAS ARTESANAIS ROTA 277

GUARAPUAVA
2025

SUMÁRIO

INTRODUÇÃO

A presente aplicação chamada Rota277 tem como objetivo desenvolver, de forma progressiva e didática, um sistema de vendas digitais especializado na comercialização de itens de cervejaria artesanal. A plataforma oferece uma experiência intuitiva para os clientes, permitindo a navegação por um catálogo de produtos, a realização da cotação de orçamento, ofertas exclusivas e o contato direto com a empresa pelas redes sociais.

1 INSPIRAÇÃO

O design e a estrutura visual da aplicação foram inspirados no template do site Rota34, com a garantia de uma interface moderna e atrativa. A aplicação será desenvolvida com foco em usabilidade, escalabilidade e segurança, proporcionando uma plataforma eficiente tanto para clientes quanto para administradores.

2 PROTOTIPAÇÃO

A prototipação das telas foi desenvolvida com o auxílio da ferramenta Figma, disponível em:
https://www.figma.com/design/C6Qk709ic6fdhwfC9Kf1jG/rota-277?node-id=103-117&p=f&t=TBJUq9xonupbCsHV-0

Com isso, foram desenvolvidas as seguintes páginas:

2.1 DESIGN-SYSTEM

Trata-se de um conjunto de padrões que o projeto deve seguir, como componentes reutilizáveis, guias de estilo e princípios de design que servem como uma referência central para a criação e manutenção de produtos digitais consistentes dentro de uma organização. Ele unifica a linguagem visual e funcional de uma marca, facilitando a colaboração entre designers e desenvolvedores e garantindo uma experiência coesa para o usuário final.

2.2 DESKTOP

É a prototipação das telas no tamanho de 1920 x 1080 pixels. Foram desenvolvidas três telas: tela principal, tela de orçamento e tela de produto.

2.3 MOBILE

A prototipação mobile refere-se à resolução de telas no tamanho de um smartphone. As resoluções mais comuns variam, mas frequentemente incluem tamanhos como 360 x 640, 375 x 667, 360 x 720 e 375 x 812 pixels.

3 Checklist | Indicadores de Desempenho (ID) dos Resultados de Aprendizagem (RA)

RA1 - Utilizar Frameworks CSS para estilização de elementos HTML e criação de layouts responsivos.

ID0 - Prototipa interfaces adaptáveis para no mínimo os tamanhos de tela mobile e desktop, usando ferramentas de design como Figma, Quant UX ou Sketch.

ID01 - Implementar um layout responsivo de uma página web utilizando um Framework CSS, como Bootstrap, Materialize ou Tailwind (com DaisyUI), aproveitando as técnicas de Flexbox ou Grid oferecidas pelo próprio framework, garantindo que o layout se adapte adequadamente a diferentes tamanhos de tela e dispositivos.

ID 02 - Utiliza técnica de responsividade nativa de CSS, como Flexbox ou Grid Layout, para criar layouts responsivos e fluidos em diferentes resoluções de tela.

ID 03 - Utiliza componentes CSS (ex. card, button ou outros) e JavaScript (ex. modal, carrousel ou outro) oferecidos por um Framework CSS.

ID 04 - Implementa um layout fluido e responsivo utilizando unidades relativas (vw, vh, %, em ou rem) em vez de unidades fixas (px) em diferentes dispositivos e tamanhos de tela.

ID 05 - Implementa animações em elementos da página, como fadeIn/fadeOut, slideIn/slideOut, utilizando CSS Animations ou bibliotecas de animação, como o Animate.css ou JQuery, para fornecer feedback visual ao usuário e criar uma experiência interativa.

~~[ ] ID 06 - Cria transições personalizadas entre diferentes estados da página ou elementos, como mudanças de layout, alterações de cor ou exibição/hide de elementos, usando CSS Transitions ou CSS Animation, para melhorar a usabilidade e a aparência da aplicação.~~

ID 07 - Aplica um Design System consistente, definindo diretrizes de estilo, cores, tipografia e padrões de componentes que são seguidos em toda a aplicação, garantindo uma experiência de usuário uniforme e atraente.

ID 08 - Implementa pré-processadores CSS, como o Sass, em conjunto com um Framework CSS ou de forma isolada, para organizar e modularizar o código CSS, aplicando variáveis, mixins e funções para facilitar a manutenção e escalabilidade dos estilos.

ID 09 - Aplica tipografia responsiva utilizando media queries ou a função clamp(), em conjunto com unidades relativas como rem, em ou vw, para ajustar o tamanho da fonte de acordo com diferentes tamanhos de tela.

ID 10 – Aplica técnicas de responsividade de imagens usando CSS, como object-fit e containers com unidades relativas (vh, %, rem) para uniformizar tamanhos, garantindo boa exibição em diferentes tamanhos de tela.

ID 11 – Otimiza imagens com uso de formatos modernos como WebP e carregamento adaptativo, utilizando atributos como srcset, elemento.

RA2 - Realizar tratamento de formulários e aplicar validações customizadas no lado cliente, utilizando a API do HTML e expressões regulares (REGEX).

ID 10 - Implementa tratamento de formulários no lado cliente com apresentação de mensagens de erro (texto próximo dos campos de entrada ou balões com mensagens) ou sucesso, utilizando os recursos da API do HTML, como validação de campos obrigatórios, tipo de entrada e limites de caracteres, garantindo que os dados inseridos sejam válidos antes de serem enviados para o servidor (via tratador de evento submit).

ID 11 - Aplica expressões regulares (REGEX) de forma eficiente para realizar validações customizadas nos campos de formulários, como formatos específicos de e-mail, telefone, data ou outros padrões personalizados definidos pelos requisitos do projeto.

ID 12 - Incorpora elementos de listagem, como checkbox, radio ou select, de maneira eficiente em formulários web, possibilitando a seleção e coleta precisa de dados pelos usuários.

ID 13 - Realiza a escrita e leitura de dados no Web Storage, permitindo a persistência de informações entre sessões de usuário e fornecendo uma maneira eficaz de armazenar dados localmente no navegador.

RA3 - Aplicar ferramentas para otimização do processo de desenvolvimento web, incluindo Node.js, NPM e linters para garantir a qualidade do código, juntamente com boas práticas de versionamento e organização de projetos.

ID 14 - Configura adequadamente um ambiente de desenvolvimento usando Node.js e NPM para gerenciar pacotes e dependências do projeto, facilitando a instalação e o uso de bibliotecas e ferramentas de terceiros.

ID 15 - Utiliza linters, como ESLint ou Stylelint, para analisar e corrigir automaticamente problemas de código, incluindo erros de sintaxe, estilo e boas práticas, garantindo a qualidade e consistência do código do projeto.

ID 16 - Adota boas práticas de versionamento utilizando sistemas como Git e GitHub, criando e gerenciando repositórios com branches adequados ou pelo menos o branch main.

[ ] ID 17 - Utiliza técnicas de minificação e otimização de recursos, como minificação de CSS e JavaScript e otimização de imagens, para melhorar o desempenho e o tempo de carregamento do site ou aplicação.

ID 18 - Organiza o arquivo README.md conforme o template exigido na disciplina, contendo informações claras e estruturadas sobre o projeto, principalmente o checklist de tópicos devidamente preenchidos.

ID 19 - Organiza os arquivos do projeto em uma estrutura coerente, lógica e modular, conforme projeto de exemplo, facilitando a localização, manutenção e escalabilidade.

[ ] ID 20 - Utiliza as metodologias BEM (Block Element Modifier) ou SMACSS (Scalable and Modular Architecture for CSS) para organizar e estruturar os estilos CSS de forma eficiente, garantindo a reutilização de estilos, a legibilidade do código e a manutenção sustentável do projeto.

RA4 - Aplicar bibliotecas de funções e componentes em JavaScript para aprimorar a interatividade de páginas web.

[ ] ID 21 - Utiliza a biblioteca jQuery para manipular o DOM e aprimorar a interatividade das páginas web, implementando funcionalidades como eventos, animações e manipulação de elementos HTML de forma eficiente.

[ ] ID 22 - Seleciona e integra com sucesso um plugin jQuery, como o jQuery Mask Plugin ou outro plugin relevante para o projeto, a fim de melhorar a funcionalidade ou a aparência de elementos específicos em uma página web.

[ ] ID 23 - Utiliza bibliotecas de web components, como Lit, para criar componentes reutilizáveis e encapsulados, melhorando a modularidade e a manutenibilidade das páginas web.

[ ] ID 24 - Utiliza uma biblioteca de componentes prontos, como Material Web Components ou outra de escolha, ou então, algum componente independente (standalone) a fim de oferecer funcionalidades específicas sem a necessidade de estar integrado a uma biblioteca completa.

RA5 - Efetuar requisições assíncronas para uma API fake e APIs públicas, permitindo a obtenção e manipulação de dados dinamicamente.

ID 25 - Realiza requisições assíncronas para uma API fake utilizando adequadamente conceitos como AJAX, Fetch API ou bibliotecas, para persistir os dados originados de um formulário.

ID 26 - Realiza requisições assíncronas para uma API fake utilizando adequadamente conceitos como AJAX, Fetch API ou bibliotecas, para exibição dos dados na página web.

4 Passos para executar a aplicação

Clonar o repositório com:

```
None
git clone
```

Fazer checkout no branch develop que contém as modificações mais recentes.

Abrir o projeto no editor Visual Studio Code (VS Code).

Abrir um terminal pelo VSCode ou qualquer terminal do seu Sistema Operacional apontando para o diretório raiz do projeto.

Instalar as dependências contidas no *package.json*.

```
None
npm i
```

(Opcional) Instalar o JSON Server globalmente disponível em https://www.npmjs.com/package/json-server

```
None
npm i -g json-server
```

É opcional porque a dependência já vem cadastrada no arquivo *package.json* para instalação local na pasta *node_modules*.

Executar a API Fake (JSON Server) via um dos seguintes comandos:

Execução via script registrado no *package.json*:

```
None
npm run json:server:routes
```

Ou via Execução explícita: **json-server --watch db.json --routes routes.json**

O comando para execução do JSON Server deve ser aplicado no diretório raiz do projeto, ou seja, que contém o arquivo db.json e routes.json.

Por padrão, a aplicação JSON Server executa no endereço localhost:3000

Executar o projeto frontend.