

SalonConnect

Links:

[Repositório do projeto GitHub](#)

[PORTFOLIO PI - SalonConnet](#)

[Figma](#)

[Behance](#)

[Notion](#)

[Jira](#)

Deploy

Membros do Grupo:

Vinicius Baldochi Cardoso

Designer

Contribuições:

Behance:

- Criação de uma página no Behance para o projeto, visando aumentar a visibilidade e facilitar a divulgação do trabalho. A página foi estruturada para apresentar os principais aspectos do projeto.

Paleta de Cores:

- Desenvolvimento de uma paleta de cores coerente e atrativa, levando em consideração a psicologia das cores e o público-alvo do projeto. A paleta foi usada em todos os elementos visuais do projeto, garantindo uma identidade visual consistente e agradável.

Protótipo de Alta Fidelidade - Navegável:

- Criação de um protótipo de alta fidelidade, totalmente navegável, para simular a experiência do usuário final. Utilizei co Figma para desenvolver

o protótipo, que inclui todas as principais funcionalidades do produto, permitindo testes de usabilidade e validação das propostas de design.

Pedro Gonçalves Moreira

Frontend

Contribuições:

1. Desenvolvimento da Página Home com Integração da API do Google Maps

- Criação de uma interface de usuário para a página principal do site Salon Connect.
 - **Integração com API do Google Maps:** Implementação de uma funcionalidade que utiliza a API do Google Maps para exibir a localização dos salões de beleza cadastrados. Isso incluiu configurar a API, lidar com chaves de acesso e garantir que os mapas sejam carregados corretamente na página.
 - **Funcionalidades Adicionais:** Adição de marcadores (pins) no mapa representando os salões, com informações básicas que aparecem ao clicar nos marcadores, como nome e endereço.

2. Criação da Aba de Serviços

- **Descrição Geral:** Desenvolvimento de uma seção dedicada a listar os serviços oferecidos pelos salões de beleza cadastrados no site.
 - **Listagem de Serviços:** Implementação de uma listagem dinâmica que exibe os serviços disponíveis, puxando dados de um backend ou API.
 - **Design Responsivo:** Garantir que a listagem seja acessível e visualmente agradável em diferentes dispositivos (desktop, tablet, mobile).
 - **Interatividade:** Adição de filtros ou categorias para facilitar a busca dos usuários pelos serviços desejados.

3. Desenvolvimento das Páginas de Registro e Login com Integração com Banco de Dados

- Implementação das funcionalidades de autenticação do usuário.
 - **Formulários de Registro e Login:** Criação de formulários interativos para que novos usuários possam se registrar e usuários existentes possam fazer login.
 - **Validação de Dados:** Implementação de validações tanto no frontend quanto no backend para garantir a integridade dos dados inseridos pelos usuários.
 - **Integração com Banco de Dados:** Configuração e integração com o banco de dados para armazenar e validar as credenciais dos usuários. Isso envolveu interagir com APIs RESTful ou GraphQL para enviar e receber dados.
 - **Mensagens de Feedback:** Adição de mensagens de erro e sucesso para informar os usuários sobre o status de suas ações (por exemplo, "Registro bem-sucedido" ou "Senha incorreta").

4. Desenvolvimento da Página Sobre (About)

- Criação de uma página dedicada a fornecer informações sobre o site Salon Connect
 - **Conteúdo Informativo:** Escrever e formatar o conteúdo que descreve o propósito do site, a equipe por trás do projeto, e os benefícios de usar o Salon Connect.
 - **Design Atraente:** Seguir o design especificado no Figma para garantir uma aparência profissional e consistente com o restante do site.
 - **Responsividade:** Garantir que a página seja visualmente agradável e funcional em dispositivos de todos os tamanhos.

5. Implementação de Designs Baseados no Figma

- Seguir e implementar os designs fornecidos no Figma para garantir a consistência visual e usabilidade do site

- **Fidelidade ao Design:** Garantir que todos os elementos visuais (cores, tipografia, espaçamento, etc.) estejam de acordo com os modelos de design do Figma.
- **Componentes Reutilizáveis:** Criação de componentes de UI reutilizáveis para manter a consistência em todo o site e facilitar futuras manutenções.
- **Teste e Ajustes:** Testar a implementação em diferentes navegadores e dispositivos, ajustando conforme necessário para resolver quaisquer discrepâncias ou problemas de usabilidade.

João Pedro Andrade Cintra

Desenvolvedor Back-end

Contribuições:

Fiquei responsável por toda a montagem do back-end e criação do banco de dados, desde a modelagem das entidades a criação e funcionamento.

Primeiramente foram definidas as linguagens de programação e qual tipo de banco de dados seria usado.

O projeto foi definido para ser feito em Node.js e MongoDB, um banco de dados NoSQL.

Todo o processo de criação inicial foi feito com base no conhecimento adquirido na aula de banco de dados não relacional, com o professor Fausto Cintra.

Criação do Back-End em Node.js

Após definir as tecnologias a serem utilizadas, comecei a estruturar o back-end utilizando Node.js, um ambiente de execução JavaScript orientado a eventos,

Principais Etapas:

1. Configuração Inicial do Projeto:

- Iniciei criando um novo projeto Node.js e configurando os arquivos básicos como package.json e os scripts necessários para rodar o servidor.
- Utilizei o express para gerenciar as rotas e middlewares, facilitando a criação de um servidor web robusto e modular.

2. Estrutura de Pastas:

- Organizei o projeto em pastas como routes, controllers, models e config, seguindo boas práticas de desenvolvimento para manter o código limpo e fácil de manter.
- routes: Contém todas as definições de rotas para a API.
- controllers: Contém a lógica de negócio e manipulação dos dados.
- models: Define a estrutura dos dados e interage com o banco de dados.

3. Configuração do Banco de Dados:

- Utilizei o mongoose, uma biblioteca ODM (Object Data Modeling) para MongoDB e Node.js, que proporciona uma solução simples e baseada em esquemas para modelar os dados da aplicação.
- Configurei a conexão com o MongoDB Atlas, um serviço de banco de dados na nuvem, que nos oferece alta disponibilidade, escalabilidade e segurança.

Criação do Banco de Dados NoSQL (MongoDB)

O MongoDB foi escolhido por ser um banco de dados NoSQL, que armazena dados em documentos flexíveis no formato JSON (BSON).

Principais Etapas:

4. Modelagem das Entidades:

- Baseado no conhecimento adquirido nas aulas com o professor Fausto Cintra, comecei a modelar as entidades do banco de dados.

- Cada entidade foi representada por um modelo Mongoose, definindo os esquemas que especificam a estrutura dos documentos, incluindo os campos e seus tipos de dados.

5. Criação dos Esquemas Mongoose:

- Criei os esquemas para as principais entidades, como User, Cliente, Empresa, Servico, Atendimento e Carrinho.
- Cada esquema inclui validações e métodos de pré-processamento (middlewares) para garantir a integridade e consistência dos dados.

6. Operações CRUD:

- Implementei as operações CRUD (POST, PUT, GET, DELETE) para cada uma das entidades.
- As rotas da API foram configuradas para expor essas operações, permitindo que o front-end e outros consumidores da API possam interagir com o banco de dados de maneira segura e eficiente.

Conclusão

O processo de criação do back-end em Node.js e do banco de dados MongoDB foi fundamental para estabelecer uma base sólida para o nosso projeto.

Utilizando o conhecimento adquirido nas aulas e as melhores práticas de desenvolvimento, conseguimos criar uma arquitetura escalável, eficiente e de fácil manutenção, que atende às necessidades do projeto e proporciona uma experiência robusta para os usuários finais.

Cláudio Matos

Product Owner (PO)

Elaboração da documentação do PI

Portfólio completo (IHC)

Jira

Notion

Overleaf

Contribuições: Interlocução com os membros, informações de evolução, de exigência e integração com informações e padronização do projeto. Elabora do portfólio que demandou muito estudo, pesquisa e tempo de organização.