

SISTEMA DE AGENDAMENTO ONLINE GENÉRICO: UMA PLATAFORMA WEB PARA GESTÃO DE HORÁRIOS EM PEQUENOS E MÉDIOS NEGÓCIOS

Victor Garcia Trennepohl, Kauê Reche, Douglas Finco, Maximiliano Pezzin
Curso de Análise e Desenvolvimento de Sistemas, UniSENAI Campus Chapecó

1. Introdução

A transformação digital tem impactado significativamente a forma como empresas e profissionais gerenciam seus atendimentos. O agendamento manual, realizado por telefone ou presencialmente, representa uma barreira operacional para pequenos e médios negócios, gerando ineficiências, conflitos de horários e perda de clientes [1]. Nesse contexto, sistemas de agendamento online surgem como ferramentas essenciais para modernizar a gestão de horários, reduzir custos operacionais e melhorar a experiência do cliente.

O presente trabalho descreve o desenvolvimento de uma plataforma web de agendamento online de caráter genérico, capaz de atender diferentes segmentos de mercado, como barbearias, clínicas, academias e prestadores de serviços em geral. A solução foi desenvolvida utilizando tecnologias modernas do ecossistema web: HTML e CSS no front-end, JavaScript para o back-end com Node.js e Express, PostgreSQL como sistema gerenciador de BD relacional, e Figma para prototipação das telas [2]. O objetivo central é fornecer uma ferramenta acessível, responsiva e funcional que permita a configuração de serviços e a realização de agendamentos de forma simples e eficiente.

2. Desenvolvimento

O sistema foi concebido seguindo a arquitetura cliente-servidor, com separação clara entre as camadas de apresentação, lógica de negócio e persistência de dados. A prototipação das interfaces foi realizada no Figma, permitindo validar o fluxo de navegação e a experiência do usuário antes da implementação. O protótipo contemplou as principais telas: página inicial com busca de estabelecimentos, tela de seleção de serviços, calendário interativo de horários disponíveis, confirmação de agendamento e painel administrativo.

O front-end foi desenvolvido com HTML5 semântico, CSS3 e JavaScript puro, garantindo compatibilidade com os principais navegadores e responsividade para dispositivos móveis. O back-end foi implementado com Node.js e o framework Express, expondo uma API RESTful que gerencia as operações de cadastro, consulta, criação e cancelamento de agendamentos. O banco de dados PostgreSQL foi modelado com as seguintes entidades principais: usuários, negócios, serviços, horários, agendamentos e avaliações. O ambiente de desenvolvimento utilizado foi o Visual Studio Code, com extensões de produtividade como ESLint, Prettier e Thunder Client para testes de API.

3. Resultados e Discussão

O sistema desenvolvido atende ao conjunto de requisitos funcionais levantados na etapa de análise, com destaque para a consistência do controle de disponibilidade de horários e a fluidez do fluxo de agendamento do ponto de vista do usuário final. Testes funcionais realizados manualmente com o auxílio da extensão Thunder Client confirmaram o correto funcionamento dos endpoints da API RESTful para todas as operações previstas: criação, consulta, atualização e cancelamento de agendamentos. A validação das interfaces junto a usuários representativos de diferentes segmentos —

barbearia e clínica estética — indicou boa aceitação da proposta visual e da organização das telas, aspectos diretamente influenciados pela etapa de prototipação no Figma [3].

Do ponto de vista da persistência de dados, o modelo relacional implementado no PostgreSQL demonstrou adequação ao domínio do problema. A normalização das entidades de negócios, serviços e horários permitiu consultas eficientes para a verificação de disponibilidade em tempo real, operação crítica para a experiência do usuário. A utilização de constraints de integridade referencial e validações em nível de banco de dados contribuiu para a confiabilidade dos dados armazenados, prevenindo inconsistências como agendamentos duplicados em um mesmo horário.

A separação entre as camadas do sistema revelou-se benéfica durante a fase de manutenção e incremento de funcionalidades: alterações na lógica de negócio, como a inclusão de regras de antecedência mínima para cancelamentos, foram realizadas no back-end sem impacto direto sobre as interfaces. Essa característica reforça a escalabilidade da arquitetura adotada e sua viabilidade para projetos que tendam a crescer em complexidade ao longo do tempo [4].

Como limitação, identificou-se a ausência de um sistema de notificações automáticas — via e-mail ou SMS — para confirmação e lembrança de agendamentos, funcionalidade apontada pelos usuários de teste como relevante para a adoção prática da plataforma. Tal recurso está previsto como incremento para versões futuras, com avaliação de integração a serviços de envio de e-mail transacional, como o Nodemailer em conjunto com provedores SMTP, e APIs de mensageria. A implementação de autenticação baseada em tokens JWT (JSON Web Token) também está listada como melhoria prioritária para reforçar a segurança das sessões de usuário.

4. Referências

- [1] A. P. Oliveira e M. R. Santos, Rev. Bras. Gestão e Negócios, 24, 312-328, (2022).
- [2] N. Duckett, "HTML & CSS: Design and Build Web Sites", John Wiley & Sons, USA, (2011).
- [3] T. Brown, Interaction Design Foundation, Figma Prototyping Guide, disponível em: <https://www.figma.com/resources>, (2023).
- [4] R. Fielding, "Architectural Styles and the Design of Network-based Software Architectures", PhD Thesis, University of California, USA, (2000).

Agradecimentos

Os autores agradecem à instituição de ensino pelo suporte técnico e pela infraestrutura disponibilizada durante o desenvolvimento deste projeto, bem como aos usuários voluntários que participaram das etapas de validação das interfaces e contribuíram com avaliações relevantes para o aprimoramento da plataforma.