

Sistema de Agendamento para Salão de Beleza - SASBApp

Claudio Caueh Oliveira Xavier
UFERSA

Pau dos Ferros, Brasil
claudio.xavier@alunos.ufersa.edu.br

Dayvison Eryc de Moura Silva
UFERSA

Pau dos Ferros, Brasil
dayvison.silva@alunos.ufersa.edu.br

Heitor Claudino Dantas
UFERSA

Pau dos Ferros, Brasil
heitor.dantas@alunos.ufersa.edu.br

Jhoan Fernandes de Oliveira
UFERSA

Pau dos Ferros, Brasil
jhoan.oliveira@alunos.ufersa.edu.br

Murilo Lacerda de Souza Fontes
UFERSA

Pau dos Ferros, Brasil
murilo.fontes@alunos.ufersa.edu.br

Resumo—Na área do empreendedorismo, a satisfação do cliente é um dos principais pontos a se considerar. Sistemas lentos, pouco intuitivos e que não atendem necessidades de seus utilizadores podem afetar negativamente um negócio. A necessidade de criar um software para garantir uma melhor experiência para os seus usuários e, consequentemente, o sucesso do empreendedor, se torna urgente, principalmente em regiões interioranas com baixo nível de presença tecnológica em serviços diversos. Considerando esses pontos, este artigo busca descrever a criação de um sistema de agendamentos para salões de beleza, criado a partir da confecção de uma rede de Petri. O resultado foi uma aplicação web permitindo que tanto administradores quanto colaboradores lidem com as demandas do estabelecimento de forma mais organizada, enquanto sua estrutura modular facilita a reutilização da documentação em sistemas com funcionalidades semelhantes.

Index Terms—desenvolvimento de software, redes de Petri ordinárias, sistema de agendamentos, salão de beleza

I. INTRODUÇÃO

Considerando a relevância crescente da Tecnologia da Informação (TI) na gestão empresarial, tornou-se imprescindível que organizações de distintos portes promovam a atualização contínua de suas práticas administrativas [7]. Oliveira [3] destaca que a informação funciona como um recurso estratégico, permitindo às instituições atingir seus objetivos mediante a utilização eficiente dos ativos disponíveis, tais como tecnologia e dados.

Ademais, o empreendedorismo configura-se como um dos principais vetores de dinamização econômica e fortalecimento do desenvolvimento nacional. Contudo, para que o empreendedor alcance êxito, é imprescindível valorizar a experiência do cliente, assegurando um atendimento de excelência pautado na criatividade e na inovação [1].

Tal premissa aplica-se de modo particular aos proprietários de salões de beleza, uma vez que os consumidores dispõem, atualmente, de amplo acesso a informações acerca de produtos, marcas, fornecedores e preços [1]. Nesse contexto, a convergência entre soluções tecnológicas e conhecimentos práticos revela-se fundamental para a competitividade do setor.

Diante desse cenário, o presente artigo apresenta o Sistema de Agendamento para Salão de Beleza (SASBApp) como uma solução de agendamento voltada ao segmento de salões de beleza, oferecendo recursos específicos para dois perfis de usuários: o administrador (proprietário do estabelecimento) e os colaboradores (profissionais do salão). A plataforma permite ao administrador gerir horários, serviços e colaboradores, enquanto fornece aos profissionais uma interface intuitiva para consulta e confirmação de atendimentos.

Este artigo encontra-se organizado em quatro seções. Na Seção II, apresenta-se a fundamentação teórica que embasa o desenvolvimento do sistema; na Seção III, detalha-se a implementação das funcionalidades principais; e, por fim, na Seção IV, expõem-se as conclusões decorrentes do trabalho, bem como as diretrizes para pesquisas e aprimoramentos futuros.

II. FUNDAMENTAÇÃO TEÓRICA

Esta seção destaca a relevância das fases iniciais do desenvolvimento de qualquer sistema. Inicialmente, aborda-se o papel da Engenharia de Software como etapa precursora a qualquer outro estágio, uma vez que contempla as reais necessidades do(s) usuário(s). Por conseguinte, ressalta-se a importância da especificação formal do sistema, a fim de se obter melhor compreensão de seu comportamento. Por fim, expõem-se as tecnologias empregadas para a modelagem.

A. Engenharia de Software

Em sua obra, Sommerville [8] menciona que existe uma pressão cada vez maior para desenvolver sistemas de forma mais rápida, com custos reduzidos e alta confiabilidade. Ademais, é sabido que a engenharia de software se preocupa com todos os aspectos de produção, isto é, dos estágios iniciais da especificação do sistema até a manutenção do mesmo após a entrega [8]. Nesse sentido, é de suma importância que o desenvolvimento de um software tenha como ponto de partida a discussão dos requisitos com o cliente, para que este alinhe com a equipe desenvolvedora suas necessidades [9].

B. Método Formal Escolhido

Uma vez que as necessidades do cliente são compreendidas, é necessário realizar uma especificação formal dos requisitos. Os métodos formais, segundo Woodcock [10], são técnicas matemáticas para especificar, desenvolver e verificar sistemas de software e hardware. Assim, para a realização do SASBApp, foi decidido pelo uso de redes de Petri ordinárias, que são especialmente úteis para modelar, descrever e entender o comportamento dinâmico de sistemas [5], [6]. Essa abordagem permitiu a criação de um aplicativo com fluxo de funcionamento claro e sem ambiguidades, facilitando o processo de implementação a partir dos documentos gerados na fase de planejamento.

C. Tecnologias Utilizadas

A rede de Petri do sistema foi modelada com ajuda do software PIPE (*Platform Independent Petri Net Editor*), que permite análises rigorosas de um sistema a partir de uma interface intuitiva [2]. Entre os benefícios de se usar a ferramenta, podem ser citados a agilidade na modelagem, a possibilidade de validação — uma vez que ela indica possíveis erros na rede — e a simulação personalizada — na qual fatores como quantidade e velocidade das simulações podem ser alterados.

III. ABORDAGEM

Nesta seção, é detalhada a abordagem utilizada para o desenvolvimento do Sistema de Agendamento para Salão de Beleza (SASBApp). Primeiramente, apresenta-se a arquitetura geral do sistema (III-A). Por fim, descreve-se o modelo formal baseado em Redes de Petri (III-B), que define o fluxo principal de agendamento.

A. Arquitetura do Sistema

A arquitetura do projeto foi composta pelos seguintes componentes:

- **Backend:** Desenvolvido com o **Django REST Framework**.
- **Banco de Dados:** Utilizou-se o **SQLite3**, banco de dados padrão do Django.
- **Frontend:** A interface do usuário foi construída com **React**.
- **Estilização:** O visual do sistema foi elaborado com o auxílio do **TailwindCSS**, permitindo uma personalização rápida e responsiva.

Essa arquitetura proporcionou uma boa separação de responsabilidades entre *front-end* e *back-end*, além de ter permitido um desenvolvimento ágil e uma boa experiência para o usuário.

B. Modelagem do Fluxo de Agendamento com Redes de Petri

Conforme introduzido na Seção II, o comportamento dinâmico do SASBApp foi modelado utilizando Redes de Petri (RdP) ordinárias. Essa abordagem foi empregada para representar de forma estruturada e visual o fluxo de interação do usuário durante o processo de agendamento, desde o

funcionário autenticado até a confirmação ou falha do agendamento. A utilização de RdPs permitiu identificar claramente os estados possíveis da aplicação e as ações que provocam transições entre esses estados, garantindo uma modelagem precisa e livre de ambiguidade. Além disso, o modelo serviu como um recurso útil para validação lógica do fluxo e como referência direta para a implementação do sistema. A rede pode ser visualizada na Figura 1.

Os elementos da rede possuem a seguinte semântica no contexto do sistema:

Lugares — Representam os estados ou telas da interface do usuário durante o processo:

- P1:** Estado inicial, indicando que um funcionário realizou o *login* com sucesso.
- P2:** Tela principal do sistema.
- P3:** Início do processo de agendamento.
- P4:** Lista de serviços exibida ao usuário.
- P5:** Serviço escolhido.
- P6:** Exibição das opções de data e horário.
- P7:** Data e horário selecionados.
- P8:** Lista de profissionais exibida.
- P9:** Profissional selecionado.
- P10:** Estado intermediário após envio da solicitação.
- P11:** Agendamento salvo com sucesso.
- P12:** Erro técnico durante o registro.

Transições — Representam ações do usuário ou do sistema que provocam mudanças de estado:

- T1:** Acessar o painel principal após *login*.
- T2:** Iniciar um novo agendamento.
- T3:** Exibir lista de serviços.
- T4:** Selecionar um serviço.
- T5:** Exibir as datas e horários disponíveis.
- T6:** Selecionar data e horário.
- T7:** Falha na escolha do horário, retornando à tela anterior.
- T8:** Exibir profissionais disponíveis.
- T9:** Selecionar um profissional.
- T10:** Registrar o agendamento.
- T11:** Confirmar sucesso do registro.
- T12:** Tratamento de erro técnico.
- T13:** Retornar ao painel após agendamento bem-sucedido.
- T14:** Retornar ao painel após falha no agendamento.

Este modelo formal permitiu validar a lógica de interação e serviu como um guia preciso para a equipe de desenvolvimento, garantindo que a implementação do software seguisse fielmente o fluxo de negócio especificado, prevenindo estados inválidos na interface do usuário. As figuras 2, 3, 4 e 5 demonstram visualmente o principal fluxo do SASBApp.

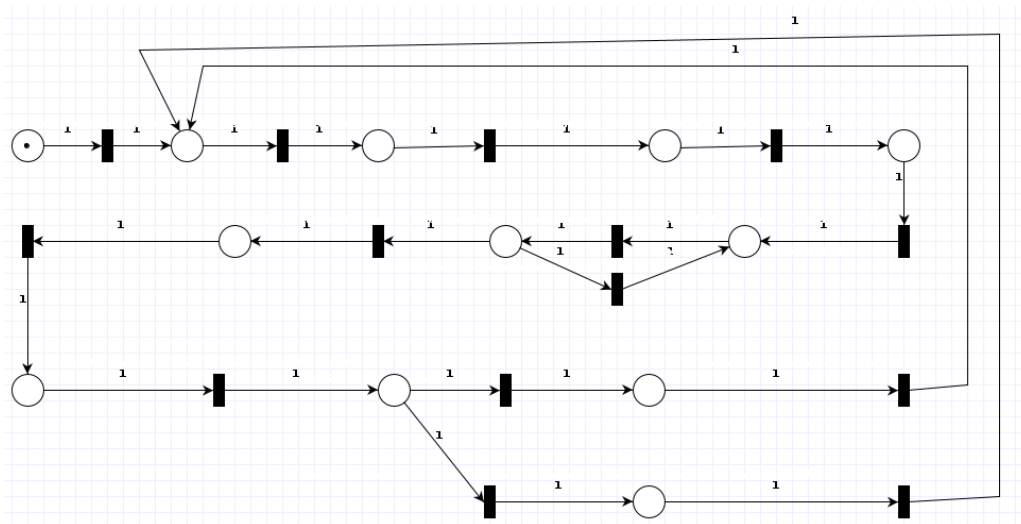


Figura 1. Modelo em Rede de Petri representando o fluxo de criação de um agendamento no SASBApp.

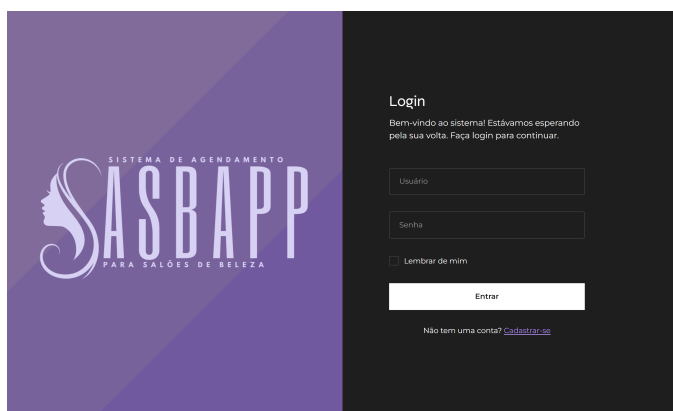


Figura 2. Tela de login

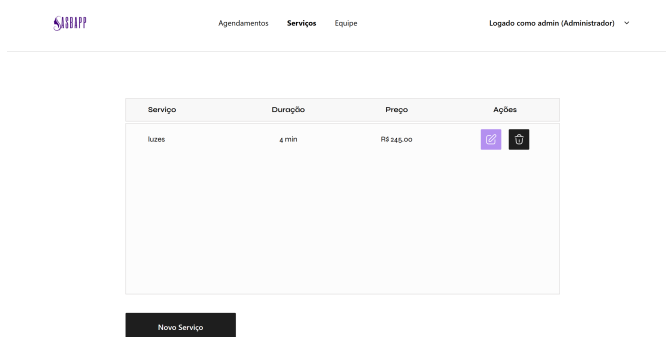


Figura 4. Tela de serviços

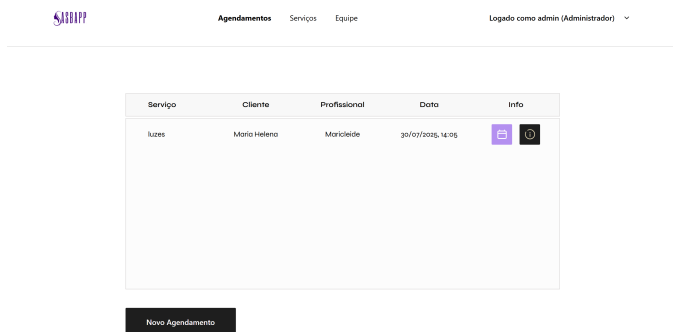


Figura 3. Tela de agendamentos

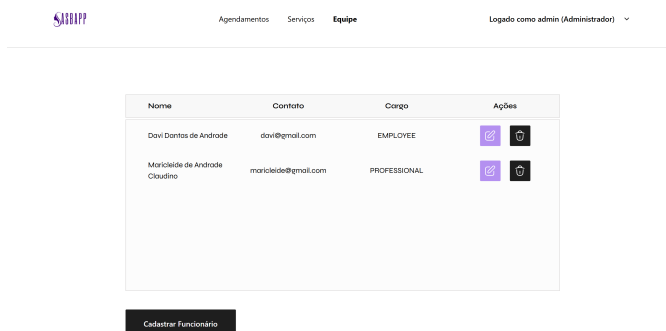


Figura 5. Tela de equipe

IV. CONCLUSÕES E TRABALHOS FUTUROS

Em um contexto no qual a tecnologia adentra cada vez mais a vida das pessoas, os salões ainda mantêm certa mecanicidade no que diz respeito ao gerenciamento de clientes, sobretudo em regiões interioranas onde sistemas e aplicativos para serviços não essenciais ainda não é tão difundida pela falta de soluções escaláveis [4]. Portanto, ao final da implementação, obtivemos um sistema que pode ser viável para suprir a necessidade de facilitar, dinamizar e controlar melhor os agendamentos desse tipo de serviço. A implementação do SASBApp traz mais agilidade tanto para a equipe profissional quanto para os clientes que se beneficiam de um serviço pautado em um sistema bem construído.

A criação de um sistema de agendamentos possibilita a modularidade do produto, uma vez que é possível adaptá-lo para qualquer outro estabelecimento que funcione de forma similar. Além disso, a utilização de uma rede de Petri publicamente disponível pode auxiliar na escalabilidade, permitindo que o funcionamento do SASBApp seja testado com níveis maiores de demanda ou carga de trabalho. Dessa forma, a documentação do projeto pode ser acessada por outros pesquisadores que tenham interesse na construção de um sistema de agendamentos.

REFERÊNCIAS

- [1] BASTOS, B. C., FREITAS, D. F. D., GOMES, I. D. A., LOPES, J. R., FELIPE, J. D. A., MELO, L. S. A. D., RAMOS, P. D. O., SOUZA, R. D., AND COSTA, S. M. F. Sistema informatizado para agendamento de serviços: estudo de caso em um salão de beleza no município de franco da rocha. In *Open Science Research XII*. Editora Científica, 2023, ch. 37, pp. 430–450.
- [2] CLARK, T., TIMMIS, J., AND HITCHCOCK, P. The platform independent petri net editor (pipe). In *Proceedings of the International Conference on Applications and Theory of Petri Nets* (2006), University of Kent. Available at: <https://pipe2.sourceforge.net/>.
- [3] DA CUNHA OLIVEIRA, A. C. M. Tecnologia de informação: competitividade e políticas públicas. *Revista de Administração de Empresas* (1996).
- [4] FERREIRA, L., SOUZA, A., AND MENEZES, C. Skills-based on technological knowledge in the digital economy activity: Evidence from brazilian microregions. arXiv preprint arXiv:2102.01711, 2021.
- [5] MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* 77, 4 (1989), 541–580.
- [6] REISIG, W. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer, Berlin, 2013.
- [7] SILVA, J. P., SOUZA, M. F., AND OLIVEIRA, R. C. O papel da tecnologia da informação na gestão empresarial. *RECIMA21* 4, 6 (2023), 2–3. Acesso em: 16 jul. 2025.
- [8] SOMMERVILLE, I. *Software Engineering*, 10 ed. Pearson, Boston, 2015.
- [9] VAZQUEZ, C. E., AND SIMÃO ES, G. S. *Engenharia de requisitos: software orientado ao negócio*. Brasport, 2016.
- [10] WOODCOCK, J., AND DAVIES, J. *Using Z: Specification, Refinement, and Proof*. Prentice Hall, London, 1996.