**COLÉGIO ESTADUAL DE EDUCAÇÃO PROFISSIONAL**

**PEDRO BOARETTO NETO**

**CURSO TECNICO EM DESENVOLVIMENTO DE SISTEMAS**

**CRISTIANO ANTONIO MAROSTEGA**

**PROJETO-X**

**CASCAVEL - PR**

**2025**

**CRISTIANO ANTONIO MAROSTEGA**

**PROJETO-X**

Projeto de Desenvolvimento de Software do Curso Técnico em Informática do Colégio Estadual de Educação Profissional Pedro Boaretto Neto – Cascavel, Paraná.

Orientadores: Profª Aparecida S.Ferreira[[1]](#footnote-1)

Prof. Reinaldo C. da Silva2

Prof. Célia K.Cabral3

**CASCAVEL - PR**

**2024**

**CRISTIANO ANTONIO MAROSTEGA**

**PROJETO-X**

Este Projeto de Conclusão de Curso foi julgado e aprovado pelo Curso Técnico em Informática do Colégio Estadual de Educação Profissional Pedro Boaretto Neto.

Cascavel, Pr., xx de Xxxxx de 2024

**COMISSÃO EXAMINADOR**

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Profª. Aparecida da S. Ferreira1  Especialista em Tecnologia da Informação  *Faculdade de Ciências Sociais Aplicadas de Cascavel*  Orientadora | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Prof. Reinaldo  Web Design |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Profª. Célia Kouth Cabral  Pós-graduada em Sistemas Distribuídos JAVA.  Universidade Tecnológica Federal do Paraná - UTFPR  Banco de dados | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Profª Ana Cristina Santana  Especialista em Gestão e Docência no ensino superior, médio e técnico.  Coordenadora de curso |
|  |  |

Sumário

# INTRODUÇÃO

O presente projeto tem por objetivo o desenvolvimento de um aplicativo de agendamento de serviços diversos, tais como: Cabeleireiros, barbeiros, manicures, maquiadoras, etc.

Para ChatGpt ( 2021), nos últimos anos, a tecnologia tem desempenhado um papel crucial na transformação de diversos setores, facilitando a vida das pessoas e otimizando processos. No setor de serviços pessoais, como cabeleireiros, barbeiros, manicures e maquiadores, a inovação tecnológica tem o potencial de revolucionar a forma como os clientes agendam e gerenciam seus compromissos. A dificuldade em encontrar horários disponíveis, a falta de integração entre prestadores de serviços e a necessidade de uma gestão mais eficiente têm sido desafios recorrentes para tanto os profissionais quanto para os clientes.

A necessidade de um sistema mais eficiente e acessível para agendamento de serviços diversos é evidente. Muitos consumidores ainda enfrentam dificuldades para marcar horários, coordenar compromissos e lidar com a ineficiência de métodos tradicionais, como telefonemas e mensagens. Além disso, para os prestadores de serviços, a gestão manual e desorganizada dos agendamentos pode resultar em perda de tempo e, consequentemente, perda de receita. Portanto, a criação de uma solução digital que otimize este processo se torna não apenas desejável, mas essencial.

O objetivo deste projeto é desenvolver um aplicativo de agendamento de serviços diversos que facilite a marcação de compromissos para cabeleireiros, barbeiros, manicures, maquiadores e outros prestadores de serviços pessoais. O aplicativo será projetado para ser intuitivo, eficiente e acessível, atendendo tanto às necessidades dos usuários quanto às dos profissionais. Entre os principais objetivos do aplicativo estão:

**Facilidade de Uso**: Proporcionar uma interface amigável e de fácil navegação para que os usuários possam agendar serviços com rapidez e sem complicações.

**Gerenciamento Eficiente**: Permitir que prestadores de serviços gerenciem seus horários e compromissos de forma organizada, reduzindo a possibilidade de conflitos e melhorando a eficiência operacional.

**Integração e Acesso**: Oferecer uma plataforma centralizada onde clientes e profissionais possam interagir, visualizar disponibilidade e confirmar agendamentos de maneira prática e rápida.

**Notificações e Lembretes**: Implementar um sistema de notificações para lembrar os usuários e prestadores de serviços sobre os compromissos agendados, evitando faltas e melhorando a organização.

Este projeto visa preencher uma lacuna significativa no mercado de serviços pessoais, oferecendo uma solução digital que simplifica o agendamento e gerenciamento de compromissos. Acredita-se que a implementação deste aplicativo não só melhorará a experiência dos clientes, mas também otimizará o gerenciamento de tempo e recursos para os profissionais de serviços. Com uma abordagem centrada no usuário e uma metodologia ágil, o projeto tem o potencial de se destacar como uma ferramenta indispensável na modernização da forma como serviços pessoais são agendados e gerenciados. Para atingir os objetivos propostos, o projeto será desenvolvido seguindo uma abordagem ágil, que permitirá flexibilidade e adaptação ao longo do processo.

## Apresentação do Problema

Criar um site e um aplicativo de agendamento de serviços estéticos tanto para o público masculino como para o público feminino. Este projeto vem para suprir uma demanda com pouca ou quase nada de oferta que apresenta solução real na otimização de tempo e na facilidade de agendamento de horário em setores que se tornaram essenciais na vida das pessoas. Com o desenvolvimento completo e aberto aos usuários, os mesmos terão a possibilidade de agendar horários em salões de beleza, barbeiros, manicure e estética em geral, de forma rápida, segura e intuitiva. Assim que o horário é agendado, o sistema vai disparar 3 alertas de aviso: Um alerta com 12 horas de antecedência, um alerta com 6 horas de antecedência, e o ultimo alerta com 1 hora de antecedência, gerando assim uma maior assertividade e maior taxa de comparecimento.

Este projeto visa preencher uma lacuna significativa no mercado de serviços pessoais, oferecendo uma solução digital que simplifica o agendamento e gerenciamento de compromissos. Acredita-se que a implementação deste aplicativo não só melhorará a experiência dos clientes, mas também otimizará o gerenciamento de tempo e recursos para os profissionais de serviços. Com uma abordagem centrada no usuário e uma metodologia ágil, o projeto tem o potencial de se destacar como uma ferramenta indispensável na modernização da forma como serviços pessoais são agendados e gerenciados.

Análise de projetos e sistemas: É a disciplina que ajuda como entender, planejar e desenhar sistemas de software para entender as necessidades de um projeto.

Banco de dados: É a disciplina que ensina como criar, gerenciar e utilizar banco de dados para armazenar e organizar informações de forma eficiente.

Web design: É a disciplina que ensina com o criar, projetar páginas e sites para a web de forma que sejam visualmente atraentes, funcionais e fáceis de usar.

# 2 OBJETIVOS

O objetivo deste projeto é desenvolver um aplicativo de agendamento de serviços diversos que facilite a marcação de compromissos para cabeleireiros, barbeiros, manicures, maquiadores e outros prestadores de serviços pessoais. O aplicativo será projetado para ser intuitivo, eficiente e acessível, atendendo tanto às necessidades dos usuários quanto às dos profissionais.

**Facilidade de Uso**: Proporcionar uma interface amigável e de fácil navegação para que os usuários possam agendar serviços com rapidez e sem complicações.

**Gerenciamento Eficiente**: Permitir que prestadores de serviços gerenciem seus horários e compromissos de forma organizada, reduzindo a possibilidade de conflitos e melhorando a eficiência operacional.

**Integração e Acesso**: Oferecer uma plataforma centralizada onde clientes e profissionais possam interagir, visualizar disponibilidade e confirmar agendamentos de maneira prática e rápida.

**Notificações e Lembretes**: Implementar um sistema de notificações para lembrar os usuários e prestadores de serviços sobre os compromissos agendados, evitando faltas e melhorando a organização.

# 3 METODOLOGIA

Conforme CARVALHO (2007), a ciência moderna, com suas raízes no século XVII, representa uma ruptura com a tradição filosófica anterior, onde as fronteiras entre saberes eram mais fluidas. No entanto, a interdependência entre ciência, filosofia e arte persiste, unidas pelo desejo comum de questionar a realidade e buscar o florescimento humano. Em contraponto ao conhecimento do senso comum, caracterizado pela imediaticidade e superficialidade, a ciência se destaca por sua profundidade e rigor. A ciência não se limita a descrever fenômenos, mas busca explicá-los de forma sistemática e testável. A natureza crítica do método científico, que se manifesta na constante avaliação e revisão das teorias, é um dos seus pilares fundamentais.

[..]o conhecimento é obra da razão, é ela que garante a correção das descobertas e a relação real entre ideias e extensão. E é, sobretudo, de natureza matemática, saber, por definição, puramente dedutivo. Nessa perspectiva, o sujeito produtor de conhecimento se apresenta como um eu que valoriza a si mesmo, por dedução (todo o raciocínio feito para garantir a correspondência entre ideia e realidade mostra tal operação do intelecto humano). CARVALHO (2007).

CARVALHO (2007), analisa as obras transcritas de Paul Feyerabend, em oposição a Popper, argumenta que a ciência não se desenvolve de forma linear e racional, mas sim por meio de um processo mais livre e criativo. Inspirado nos estudos de Piaget, ele propõe o 'anarquismo epistemológico', no qual a imaginação e a criatividade são fundamentais para a geração de novas ideias. Para Feyerabend, não há um método científico único e absoluto, e a persuasão e a retórica desempenham um papel crucial na aceitação de novas teorias.

LAKATOS, por sua vez, apresenta uma visão mais estruturada do desenvolvimento científico, propondo o conceito de 'programas de pesquisa'. Esses programas são constituídos por um núcleo central de ideias, que é protegido por um cinturão de teorias auxiliares. A escolha de um programa de pesquisa é influenciada por fatores históricos e contextuais.

Embora as contribuições de Feyerabend e Lakatos tenham enriquecido o debate sobre a natureza da ciência, ambos os autores tendem a se concentrar nos aspectos internos do desenvolvimento científico, como a lógica e a metodologia. As influências externas, como os contextos sociais, culturais, políticos e econômicos, que moldam a produção científica, são frequentemente subestimadas em suas análises.

Para COELHO (2022), o método comparativo é uma estratégia de pesquisa que busca estabelecer paralelos entre diferentes fenômenos, com o objetivo de identificar padrões, similaridades e divergências. Ao comparar, os pesquisadores podem relacionar novos conhecimentos com o que já se sabe, além de diferenciar conceitos semelhantes. Essa abordagem metodológica é particularmente útil nas Ciências Sociais e Naturais, onde é empregada para construir e testar teorias, bem como para estabelecer leis gerais sobre fenômenos naturais. Ao comparar indivíduos ou grupos, por exemplo, os cientistas podem identificar padrões comportamentais ou características comuns que os unem, ou diferenças que os distinguem. Assim, para estabelecer a comparação, é preciso compreender muito bem os dois objetos comparados, caso contrário, o emprego do método comparativo não será científico. O método comparativo exige a compreensão de toda a complexidade dos fenômenos estudados, não apenas a parte que será objeto da monografia, da pesquisa científica. Novamente, é importante relembrar que o método comparativo de pesquisa pode ser empregado, juntamente, com outros métodos, já que se trata de um método de procedimento e, geralmente, é combinado com outras modalidades de métodos.

A modelagem de dados é uma etapa crucial no desenvolvimento de software, pois fornece uma representação abstrata dos dados que o sistema armazenará e manipulara. Essa representação, que inclui o fluxo de dados e o projeto do banco de dados, serve como um blueprint para a construção do software. Ao definir claramente as entidades, atributos e relacionamentos entre os dados, a modelagem garante que o sistema seja capaz de atender aos requisitos de negócio e fornecer informações precisas e confiáveis. Além disso, a documentação da modelagem facilita a comunicação entre os membros da equipe de desenvolvimento e a manutenção do sistema ao longo do tempo.

A modelagem de dados dá a você a chance de entender seus dados e fazer as escolhas tecnológicas certas para armazenar e gerenciar esses dados. Da mesma forma que um arquiteto projeta um esquema antes de construir uma casa, as partes interessadas da empresa projetam um modelo de dados antes de desenvolver soluções de banco de dados para suas organizações.(AWS, 2024).

# 4 REFERENCIAL TEÓRICO

De acordo com EICH (2024), o HTML (HyperText Markup Language) é a base estrutural da web, definindo a organização e o significado do conteúdo visualizado nas páginas da internet. Podemos comparar o HTML aos ossos de um corpo, pois ele fornece forma e sustentação às páginas. Para simplificar ainda mais, imagine uma página web como um livro: o HTML seria responsável por estabelecer a estrutura desse livro, organizando seus capítulos, parágrafos, títulos e imagens. Outras tecnologias, como o CSS, seriam encarregadas da apresentação visual, como a escolha de fontes, cores e o layout, enquanto o JavaScript adicionaria funcionalidades interativas, como menus suspensos ou animações dinâmicas.

O termo “hipertexto” refere-se à capacidade de interligar diferentes partes de um texto ou até mesmo páginas inteiras, por meio de links clicáveis que facilitam a navegação entre sites. Assim, o HTML não apenas estrutura o conteúdo, mas também permite a criação de conexões dinâmicas entre diferentes recursos da web, transformando-a em uma rede interativa e interconectada (EICH, 2024).

Conforme EICH (2024), o CSS (Cascading Style Sheets) é responsável pela apresentação visual das páginas da web, desempenhando um papel semelhante ao de “vestir” os elementos estruturais criados pelo HTML. O CSS define como os diferentes componentes de uma página – como títulos, parágrafos e imagens – serão exibidos no navegador. Ele controla aspectos como a cor do texto, o tamanho das fontes, o espaçamento entre elementos e até mesmo cria efeitos visuais, como animações.

Para compreender melhor sua função, podemos compará-lo novamente a um livro: se o HTML fornece a estrutura do livro (organizando capítulos, parágrafos e imagens), o CSS seria a diagramação, escolhendo as fontes, as cores e o layout de cada página. Além de estilizar elementos básicos, o CSS possibilita a criação de layouts complexos, transformando uma simples coluna de texto em um design visual mais atrativo, com múltiplas colunas e áreas distintas de conteúdo.

O CSS também permite adicionar efeitos visuais, como animações e transições, tornando a experiência do usuário mais envolvente e interativa (Eich, 2024).

Segundo Eich (2024), o JavaScript é uma linguagem de programação essencial para a criação de páginas web dinâmicas e interativas. Sempre que um site oferece mais do que apenas conteúdo estático, como atualizações periódicas, mapas interativos ou gráficos animados, o JavaScript é a tecnologia por trás dessas funcionalidades. Diferentemente do HTML e do CSS, que estruturam e estilizam as páginas, o JavaScript adiciona uma camada de interatividade, permitindo que os usuários interajam com os elementos da página em tempo real.

O JavaScript compõe a terceira camada da estrutura da web, complementando o HTML e o CSS. Ele é amplamente utilizado para tarefas como validar formulários, criar menus suspensos e implementar animações. Sua versatilidade o torna uma das linguagens mais populares e importantes no desenvolvimento web moderno, transformando páginas estáticas em experiências dinâmicas e responsivas para o usuário (EICH, 2024).

Conforme CRISTO (2024), o PHP (PHP: Hypertext Preprocessor) é uma linguagem de programação de código aberto amplamente utilizada no desenvolvimento de aplicativos web dinâmicos. Uma das suas principais características é a facilidade de integração com o HTML, o que permite que as páginas web gerem conteúdo dinâmico de forma eficiente. Ao contrário do JavaScript, que é executado no lado do cliente (no navegador), o PHP é processado no servidor. Isso significa que o navegador do usuário recebe apenas o resultado da execução do código, sem ter acesso ao código fonte original, o que contribui para a segurança da aplicação.

Por exemplo, o PHP pode ser utilizado para exibir mensagens dinâmicas em uma página ou interagir com bancos de dados para gerar conteúdo personalizado. Uma das grandes vantagens dessa linguagem é sua curva de aprendizado acessível, tornando-se uma excelente escolha tanto para iniciantes quanto para programadores mais experientes. Além disso, a comunidade ativa de desenvolvedores PHP oferece uma vasta gama de recursos e suporte contínuo (CRISTO, 2024).

Segundo CIRNE (2024), o MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto, amplamente utilizado no desenvolvimento de aplicações que exigem armazenamento e manipulação de grandes volumes de dados. O MySQL opera em um modelo cliente-servidor, onde múltiplos clientes podem se conectar ao servidor simultaneamente para acessar e modificar os dados armazenados.

Os dados são organizados em tabelas que consistem em linhas e colunas, permitindo operações de criação, modificação e consulta de forma eficiente, utilizando a Structured Query Language (SQL). O MySQL oferece uma ampla gama de recursos, incluindo suporte a transações, controle de acesso e segurança robusta dos dados. Além disso, é conhecido por sua confiabilidade e escalabilidade, sendo uma escolha popular tanto para pequenos projetos quanto para grandes sistemas corporativos.

Com uma comunidade ativa de desenvolvedores e usuários, o MySQL é continuamente aprimorado, tornando-se uma ferramenta fundamental no desenvolvimento de soluções de banco de dados robustas e flexíveis (CIRNE, 2024).

# 5 DOCUMENTAÇÃO do projeto

De acordo com MESQUITA (2024), a documentação de projetos desempenha um papel crucial para o sucesso e eficiência dos empreendimentos. Esse processo, que está intrinsecamente ligado à gestão empresarial, é o fator decisivo entre uma organização que opera de forma amadora ou profissional. Independentemente do porte ou da natureza do projeto, a documentação envolve o registro de todas as informações relevantes, desde o planejamento inicial até a sua conclusão. Assim, a documentação serve como um resumo abrangente e uma reunião organizada de todas as informações vitais do projeto.

A documentação de projetos pode ser entendida como o processo sistemático de registro e organização de todos os dados relacionados a um determinado projeto. Isso inclui o registro dos objetivos, escopo, cronograma, recursos, requisitos, estratégias, riscos, decisões e os resultados alcançados. O principal objetivo desse processo é criar um histórico completo e estruturado do projeto, o que facilita sua compreensão, promove a colaboração entre as partes envolvidas e serve como uma referência para futuras consultas (MESQUITA, 2024).

A importância da documentação de projetos não pode ser subestimada. Uma documentação clara e detalhada assegura que todas as partes interessadas compreendam o projeto de maneira uniforme e precisa. Além disso, a documentação funciona como uma fonte confiável de informações, facilitando a comunicação entre a equipe de projeto, os clientes, patrocinadores e outros stakeholders envolvidos. Com uma documentação bem elaborada, é possível evitar mal-entendidos, aumentar a eficiência e garantir que as interações entre as partes ocorram de forma mais fluida e eficaz (MESQUITA, 2024).

Outro ponto fundamental é o aprendizado organizacional. Ao documentar seus projetos, as organizações conseguem acumular conhecimento e experiência ao longo do tempo. Esse registro contínuo e detalhado permite que as empresas identifiquem boas práticas, aprendam com erros e melhorem a execução de projetos futuros (MESQUITA, 2024).

O Ciclo de Vida do Desenvolvimento de Software (SDLC) é uma metodologia que visa garantir que o desenvolvimento de software seja realizado dentro dos prazos e do orçamento previsto. O SDLC oferece um plano detalhado para planejar, construir e manter software com alta qualidade, guiando cada fase com seus próprios processos e entregáveis. Ele permite o controle das tarefas, facilita a comunicação com stakeholders, acelera o desenvolvimento, reduz riscos e melhora a experiência do cliente. Ao seguir um conjunto padrão de atividades, o SDLC ajuda a organizar e otimizar a produção.

O Ciclo de Desenvolvimento de Software (SDLC) é composto por sete fases: Análise de Requisitos, Estudo de Viabilidade, Design, Codificação, Testes, Instalação e Deploy e Manutenção. Cada fase desempenha um papel essencial no controle do cronograma e na garantia da qualidade do software, permitindo que as equipes se adaptem e executem o projeto conforme os objetivos do cliente. O SDLC proporciona flexibilidade, assegura o cumprimento dos prazos e facilita o gerenciamento ágil e preciso de todas as atividades envolvidas.



Fonte: MAROSTEGA, 2024

5.1 Análise de Requisitos

Descrição: Nesta fase, são identificadas as necessidades dos usuários e os requisitos do sistema. Isso envolve reuniões com stakeholders, entrevistas e coleta de feedback para entender o que o software deve realizar.

* Estudo de Viabilidade

Descrição: Avalia a viabilidade do projeto em termos técnicos, financeiros e operacionais. Essa fase ajuda a determinar se o projeto é viável e se deve seguir em frente.

* Design

Descrição: Com os requisitos definidos, a equipe cria a arquitetura do software, incluindo diagramas de fluxo, estrutura de dados e interfaces. Essa fase determina como o sistema funcionará.

* Codificação

Descrição: A fase de codificação envolve a implementação do design, onde os programadores escrevem o código-fonte do software. Essa etapa é fundamental para transformar os designs em um produto funcional.

* Testes

Descrição: Os testes são realizados para identificar e corrigir erros ou bugs no software. Isso pode incluir testes unitários, testes de integração e testes de sistema para garantir que o software atenda aos requisitos.

* Instalação e Deploy

Descrição: Nesta fase, o software é colocado em produção, sendo instalado nos ambientes dos usuários. Isso envolve treinamento, documentação e suporte inicial.

* Manutenção

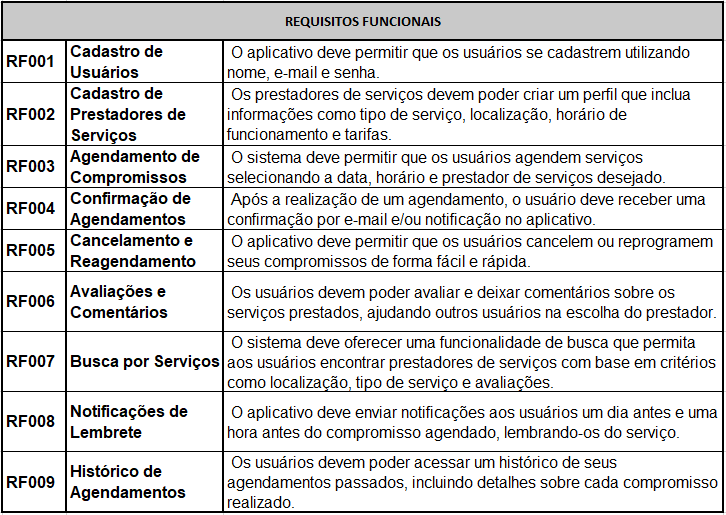
Descrição: Após a implantação, o software entra em uma fase de manutenção, onde é monitorado, atualizado e melhorado conforme necessário. Isso garante que o sistema continue a atender às necessidades dos usuários. Essas fases do SDLC são fundamentais para o desenvolvimento de software de qualidade e garantem que as expectativas dos clientes sejam atendidas.

## 5.1 Requisitos

Segundo SOMMERVILLE, a coleta de requisitos é uma fase fundamental na iniciação de um projeto. Nesse processo, são esclarecidas as razões que motivam o início da empreitada, bem como os objetivos e as estratégias que guiarão a execução das propostas. É também o momento em que se identificam as necessidades a serem atendidas, permitindo a antecipação e resolução de potenciais problemas futuros. Com as informações organizadas de maneira eficaz, é possível desenvolver um protótipo que assegure a conformidade com as especificações e a entrega adequada ao cliente.

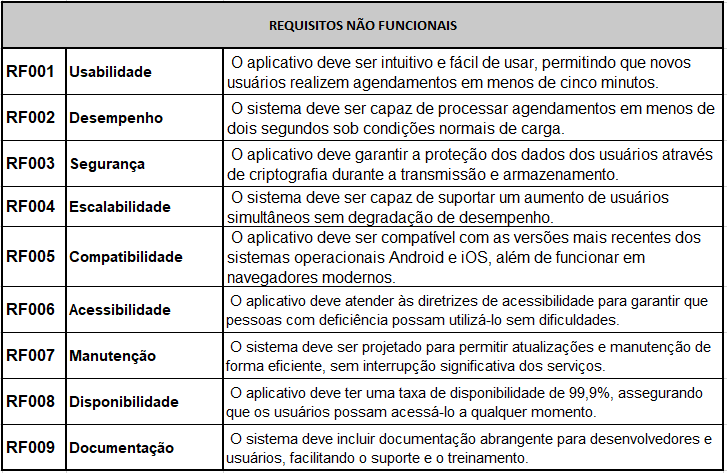
## 5.1.1 Requisitos funcionais

Requisitos funcionais são descrições detalhadas das funções que um sistema deve desempenhar. Eles definem o comportamento do software e especificam o que o sistema deve fazer, abrangendo interações do usuário, processamento de dados e requisitos de desempenho. Esses requisitos são cruciais para o desenvolvimento, pois orientam a equipe em todas as fases do ciclo de vida do software, desde a análise até os testes. A documentação adequada dos requisitos funcionais ajuda a garantir que as expectativas dos stakeholders sejam atendidas e facilita a comunicação entre desenvolvedores, gerentes e clientes. Em suma, requisitos funcionais são a base sobre a qual o sistema é construído, influenciando a qualidade e a eficácia do produto final (SOMMERVILLE, 2011).

Fonte: MAROSTEGA, 2024

### **5.1.2 Requisitos não funcionais**

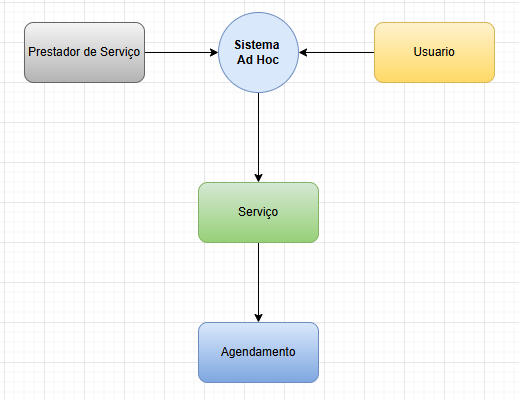
Requisitos não funcionais são atributos que definem a qualidade e o desempenho de um sistema, além de suas funcionalidades. Eles não descrevem o que o software deve fazer, mas sim como ele deve se comportar e quais são as suas restrições. Esses requisitos podem incluir aspectos como usabilidade, segurança, desempenho, escalabilidade e compatibilidade. Para um aplicativo, como o de agendamento de serviços, os requisitos não funcionais são essenciais para garantir uma experiência positiva ao usuário, assegurando que o sistema seja intuitivo, responsivo e seguro. Eles também influenciam a arquitetura do sistema e as decisões de design, impactando diretamente na satisfação do cliente e na eficiência do serviço (Pressman, 2014).



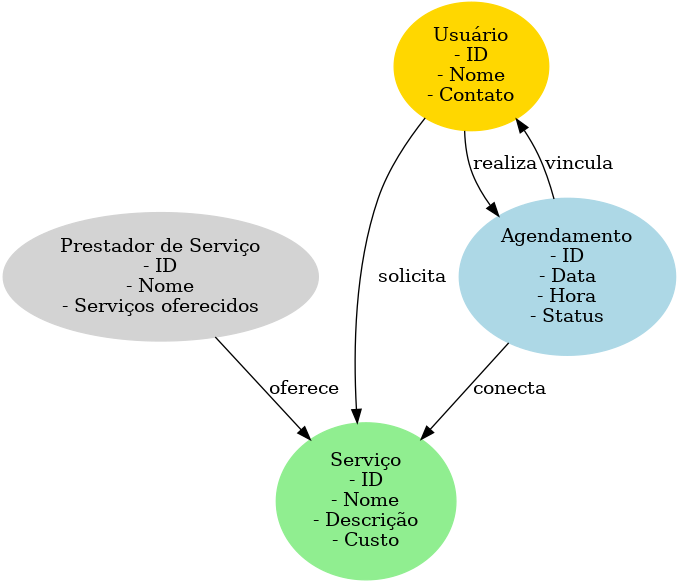
Fonte: MAROSTEGA, 2022

## Diagrama de Contexto

O diagrama de contexto representa o sistema de software como um único processo, destacando as interações entre as entidades externas e o sistema em questão. Essa ferramenta é valiosa para analistas de negócios e outras partes interessadas, pois proporciona uma visão abrangente do projeto, permitindo que gerentes realizem observações importantes antes do início do desenvolvimento.

Entre as principais vantagens do diagrama de contexto, destaca-se sua capacidade de oferecer uma representação visual clara do escopo do projeto, facilitando a identificação de possíveis erros. Além disso, o diagrama pode ser facilmente editado, permitindo a inserção e remoção de elementos conforme necessário. Outra característica benéfica é a possibilidade de personalização com diferentes formas e cores, o que significa que a criação do diagrama não exige conhecimentos técnicos ou experiência em programação (SOMMERVILLE, 2011).

Fonte: MAROSTEGA, 2022



Fonte: MAROSTEGA, 2022

## Diagrama de Fluxo de dados

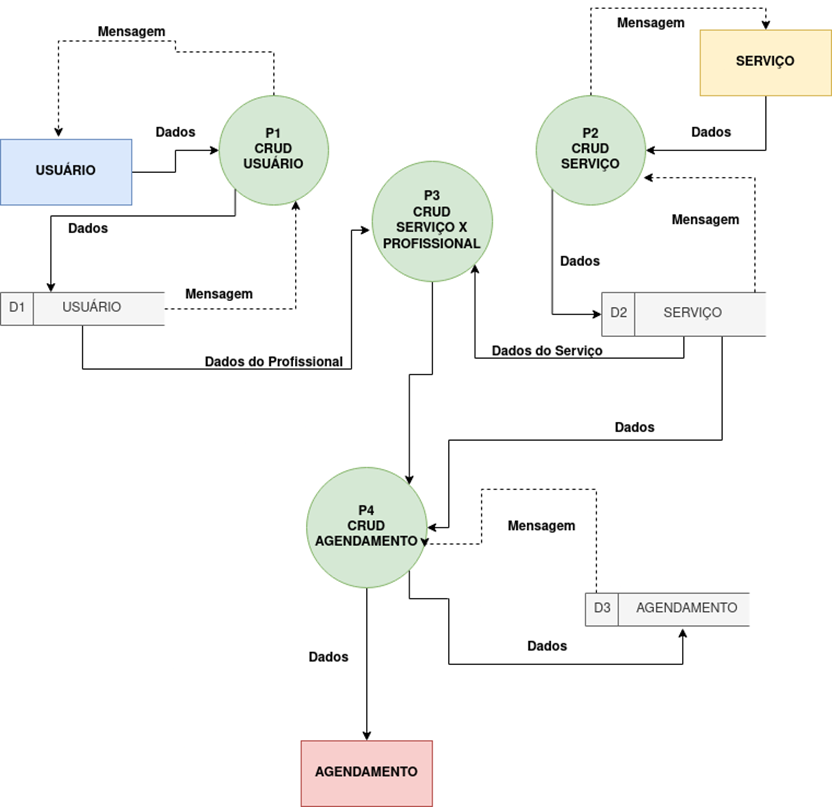
O Diagrama de Fluxo de Dados (DFD) é uma ferramenta essencial para a modelagem de sistemas, oferecendo uma representação visual do fluxo de informações e processos dentro de uma estrutura organizacional ou sistema específico. Ele é amplamente utilizado na análise de sistemas e no desenvolvimento de software, especialmente nas fases iniciais de análise e design. O DFD ajuda analistas e desenvolvedores a identificar como os dados circulam entre diferentes partes de um sistema e quais operações são realizadas ao longo desse fluxo.

Um DFD é composto por quatro elementos principais: processos, fluxos de dados, depósitos de dados e entidades externas. Os processos representam as operações executadas sobre os dados, convertendo entradas em saídas. Os fluxos de dados indicam o movimento dos dados entre processos, depósitos e entidades externas. Já os depósitos de dados armazenam informações, e as entidades externas representam fontes ou destinos dos dados fora do sistema modelado (Pressman, 2016). Esses elementos são conectados para criar uma visualização estruturada do sistema, facilitando a compreensão e o desenvolvimento de soluções eficazes.

A principal vantagem do uso de DFDs está em sua simplicidade e clareza. Ao utilizar representações gráficas, ele permite que até mesmo pessoas sem conhecimento técnico profundo compreendam a estrutura de um sistema. Segundo Kendall e Kendall (2014), o DFD é uma ferramenta poderosa para facilitar a comunicação entre desenvolvedores, analistas e stakeholders, garantindo que todos compartilhem uma compreensão comum dos processos de negócios ou das funcionalidades de um sistema. Esse recurso é particularmente útil em ambientes de desenvolvimento ágil, onde a comunicação clara e contínua é essencial para a entrega de valor rápido.

Existem diferentes níveis de DFDs, cada um com grau específico de detalhamento. O nível 0, ou diagrama de contexto, oferece uma visão geral do sistema, mostrando-o como um único processo com interações externas. À medida que se avança para níveis mais detalhados, como o nível 1 e o nível 2, os processos são decompostos em sub-processos, permitindo uma análise mais aprofundada. Essa decomposição hierárquica é essencial para a engenharia de requisitos e facilita a identificação de pontos críticos ou de gargalos no fluxo de informações (SOMMERVILLE, 2011).

Dada a sua importância, o DFD é uma ferramenta amplamente utilizada em metodologias de desenvolvimento estruturado, mas também pode ser adaptada para ambientes de desenvolvimento orientado a objetos. Embora o DFD não modele a lógica interna dos processos, ele fornece uma visão fundamental dos fluxos de dados e das interações, o que é um excelente ponto de partida para análises mais detalhadas, como o mapeamento de entidades e relacionamentos ou diagramas de classes em metodologias orientadas a objetos.

Em suma, o Diagrama de Fluxo de Dados é uma técnica de modelagem indispensável, que oferece uma visão clara e estruturada de como dados e processos interagem dentro de um sistema, facilitando tanto o entendimento inicial quanto a documentação e a comunicação entre os envolvidos no desenvolvimento do projeto.

Fonte: MAROSTEGA, 2024

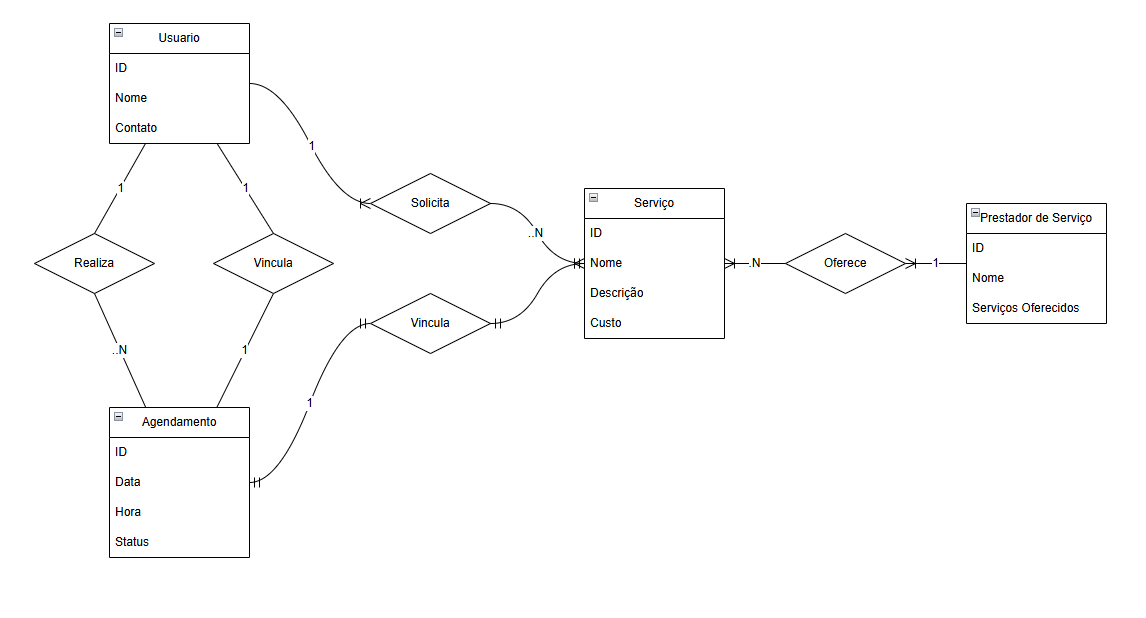
## Diagrama de Entidade e relacionamento

O Diagrama de Entidade e Relacionamento (DER) é uma ferramenta fundamental no processo de modelagem de banco de dados. Ele representa graficamente as entidades (como pessoas, objetos ou conceitos) e os relacionamentos entre elas, ajudando a estruturar e organizar os dados antes de sua implementação no sistema. Cada entidade em um DER possui atributos que a definem, e esses atributos são fundamentais para descrever as características únicas de cada uma. Essa modelagem é essencial para capturar os requisitos do sistema e garantir que o banco de dados atenda às necessidades de consulta e armazenamento de informações de maneira eficaz (ELMASRI E NAVATHE, 2011).

Em um DER, as entidades são representadas por retângulos, os relacionamentos por losangos, e os atributos por elipses. Essas formas e conexões fornecem uma maneira visual de entender como diferentes partes de um sistema interagem, tornando o DER uma ferramenta valiosa tanto para analistas quanto para desenvolvedores. Conforme ELMASRI E NAVATHE (2011) destacam, o uso de diagramas visuais facilita a comunicação entre os stakeholders, pois permite que pessoas de áreas diferentes compreendam a estrutura do sistema sem a necessidade de conhecimento técnico avançado.

Além das entidades e dos relacionamentos, o DER inclui o conceito de cardinalidade, que define a quantidade de instâncias de uma entidade que pode estar associada a outra entidade em um relacionamento. Por exemplo, uma entidade "Cliente" pode estar associada a várias instâncias de "Pedidos", caracterizando um relacionamento de um para muitos. Essa especificação é importante, pois permite ao desenvolvedor organizar o banco de dados de forma que ele represente com precisão as regras de negócio e as interações reais do sistema (SILBERSCHATZ, KORTH E SUDARSHAN, 2019).

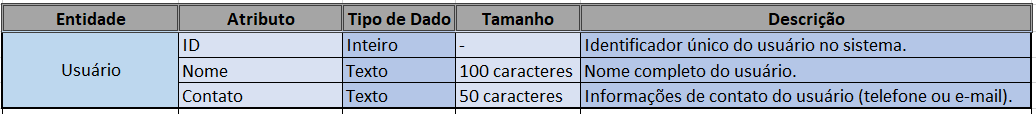
Em suma, o Diagrama de Entidade e Relacionamento é uma ferramenta essencial para a fase de modelagem de dados. Ao representar graficamente entidades, relacionamentos e cardinalidades, ele permite uma visão clara e estruturada dos dados, o que facilita o desenvolvimento de um banco de dados bem organizado e eficiente, capaz de atender às exigências do sistema.

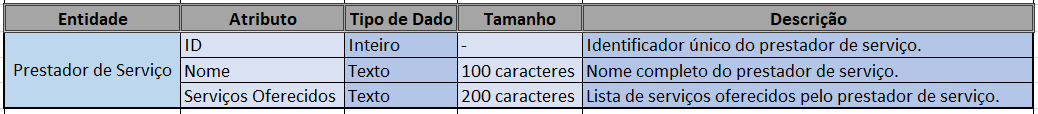
 Fonte: MAROSTEGA, 2022

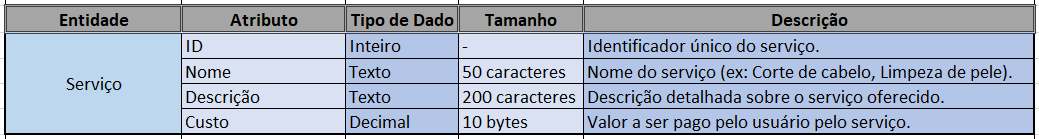
## Dicionário de Dados

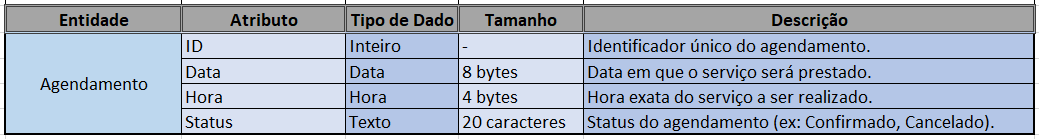
Um dicionário de dados é fundamental para a integridade e consistência de qualquer sistema de banco de dados, proporcionando um ponto central de referência sobre os dados armazenados e manipulados. (Elmasri & Navathe, 2015).

Segundo Elmasri & Navathe (2015), dicionário de Dados é um repositório que contém a descrição detalhada de todos os elementos de dados usados em um sistema ou base de dados. Ele serve como uma referência essencial para entender como os dados são estruturados, usados e manipulados no sistema. Cada item ou entidade no banco de dados é descrito de forma detalhada, incluindo seu nome, tipo, formato, relação com outras entidades e restrições. O dicionário é um componente crucial para garantir consistência e clareza durante o desenvolvimento, manutenção e utilização de sistemas de informação.









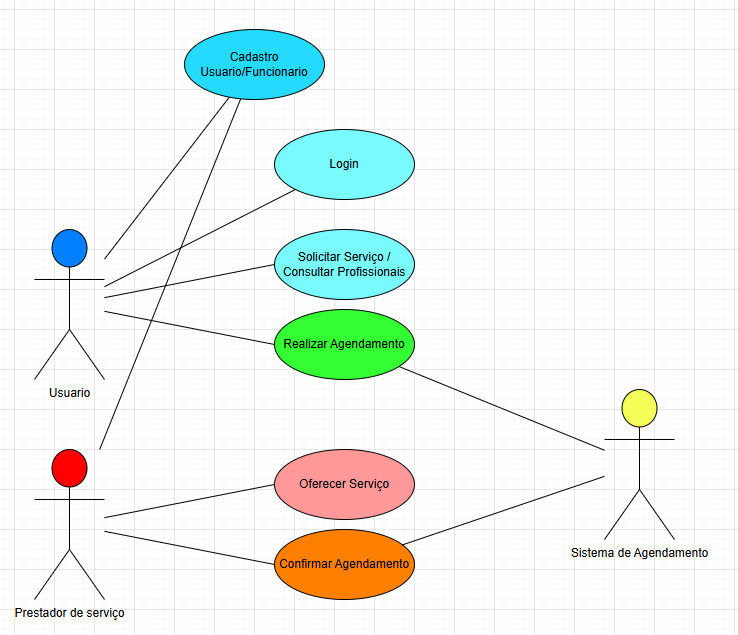
**Fonte: MAROSTEGA, 2022**

## Diagrama de Caso de Uso

Segundo Booch (2005) o Diagrama de Casos de Uso é uma ferramenta fundamental em engenharia de software para descrever as funcionalidades de um sistema e as interações entre os usuários (atores) e o próprio sistema. Ele serve para identificar os casos de uso, ou seja, as ações que os usuários podem realizar no sistema, bem como os atores envolvidos nesses processos.

O objetivo principal de um Diagrama de Casos de Uso é representar o comportamento do sistema do ponto de vista do usuário. Ele ajuda a documentar de forma visual as funcionalidades que o sistema oferece e como os usuários interagem com essas funcionalidades. Esse tipo de diagrama é útil tanto para os desenvolvedores quanto para os stakeholders, pois facilita a comunicação entre as equipes técnicas e os clientes.

Os diagramas de casos de uso são uma maneira simples de capturar e representar as funcionalidades de um sistema, permitindo que os desenvolvedores e clientes entendam as expectativas do sistema de maneira clara e objetiva." (Booch, 2005).



**Fonte: MAROSTEGA, 2022**

### Cadastrar

Ator: Usuário / Prestador de Serviço

Caso de Uso: Cadastrar-se

Descrição: O usuário preenche um formulário com suas informações (nome, cpf, senha, etc) para se registrar no sistema.

Cenário Principal:

1. O usuário acessa a página de cadastro.

2. Preenche os campos obrigatórios.

3. O sistema valida as informações e cria uma conta de usuário.

4. O sistema notifica o sucesso do cadastro.

Cenário Alternativo (erro de validação):

1. Se o sistema detectar erros no formulário (por exemplo, cpf já cadastrado), ele exibe uma mensagem de erro e solicita ao cliente que corrija os dados digitados.

### Logar

Ator: Usuário / Prestador de serviço

Caso de Uso: Login

Descrição: O usuário entra no sistema fornecendo suas credenciais (cpf e senha).

Cenário Principal:

1. O usuário acessa a página de login.

2. Insere cpf e senha.

3. O sistema valida as credenciais.

4. O sistema autentica o usuário e permite o acesso à área restrita.

Cenário Alternativo (senha incorreta):

1. Se a senha inserida estiver incorreta, o sistema exibe uma mensagem de erro e oferece a opção de redefinir a senha.

### Cadastro de funcionário/profissional

Ator: Usuário / Prestador de serviço

Caso de Uso: Cadastrar-se

Descrição: O usuário preenche um formulário com suas informações (nome, cpf, serviço prestado e colaboradores) para se registrar no sistema.

Cenário Principal:

1. O usuário acessa a página de cadastro.

2. Preenche os campos obrigatórios.

3. O sistema valida as informações e cria o serviço disponível.

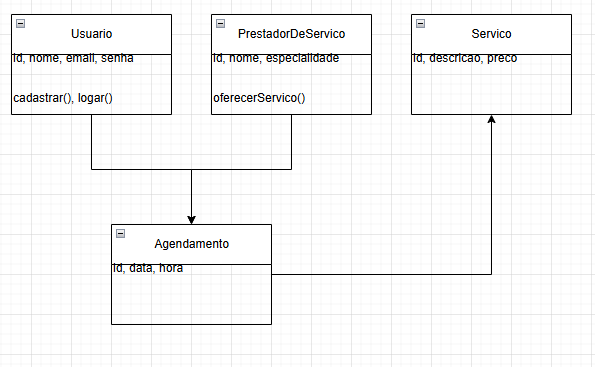
4. O sistema notifica o sucesso do cadastro.

Cenário Alternativo (erro de validação):

1. Se o sistema detectar erros no formulário (por exemplo, cpf já cadastrado), ele exibe uma mensagem de erro e solicita ao cliente que corrija os dados digitados.

## Diagrama de Classe

Um diagrama de classe é um modelo da UML que mostra a estrutura de um sistema orientado a objetos. Segundo RODRIGUES (2016), ele representa as classes, seus atributos, métodos e os relacionamentos entre elas, como herança, associação e dependência. É usado para visualizar como os objetos interagem no sistema.

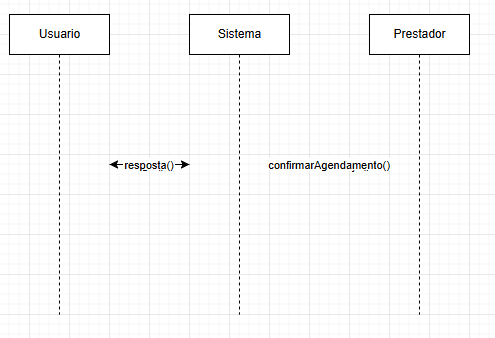


**Fonte: MAROSTEGA, 2022**

## Diagrama de Sequência

Um diagrama de sequência é uma espécie de diagrama de interação, pois descreve como, e em qual ordem, um grupo de objetos trabalha em conjunto. Estes diagramas são usados por 31 desenvolvedores de software e profissionais de negócios para entender as necessidades de um novo sistema ou para documentar um processo existente. (GROW, 2024).

Um diagrama de sequência é um modelo da UML que mostra a interação entre os objetos de um sistema ao longo do tempo. Ele descreve a ordem em que as mensagens (ou chamadas de métodos) são trocadas entre os objetos para realizar uma função específica, representando o fluxo de controle no sistema.



**Fonte: MAROSTEGA, 2022**

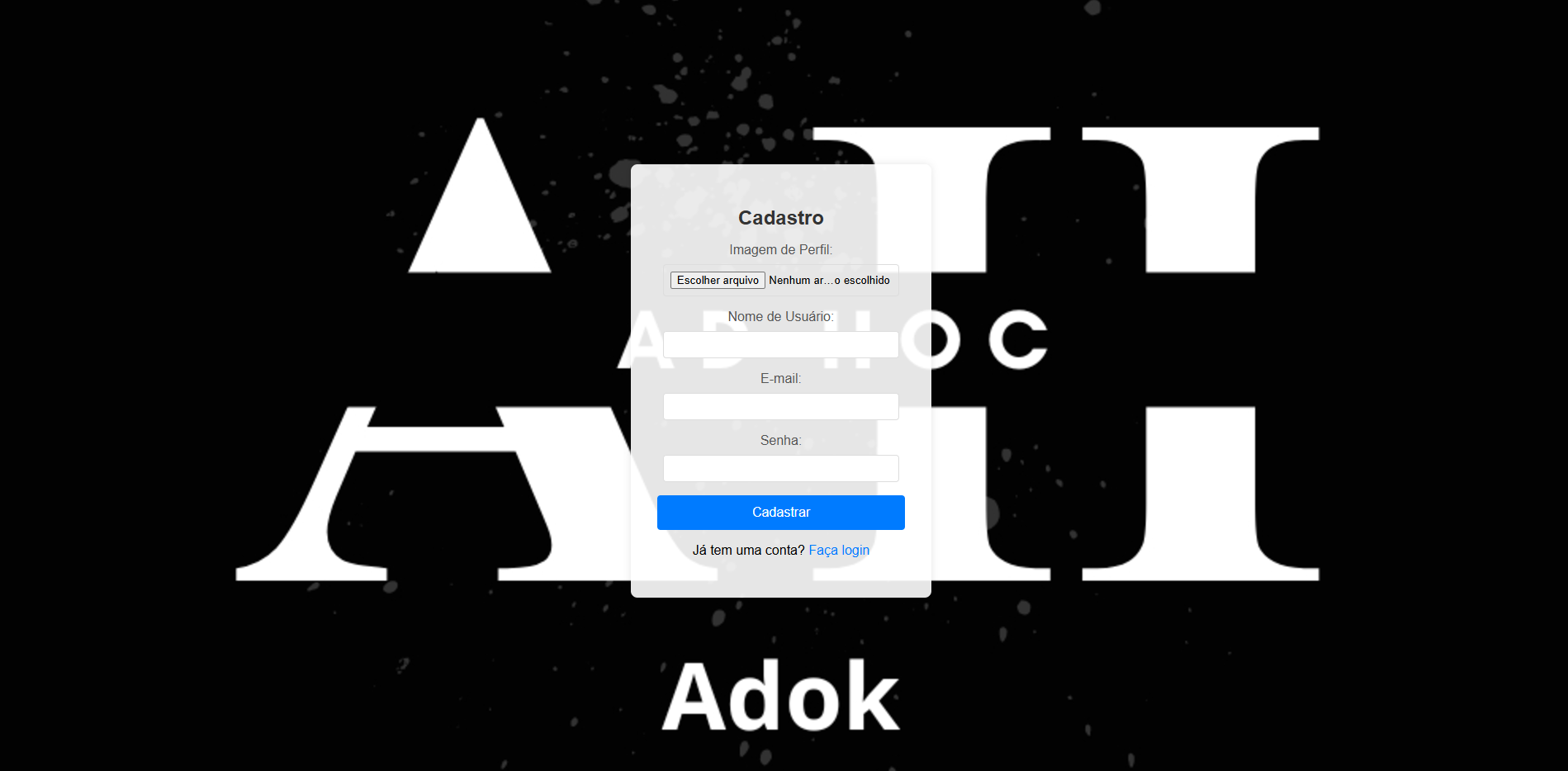
## Diagrama de Atividade

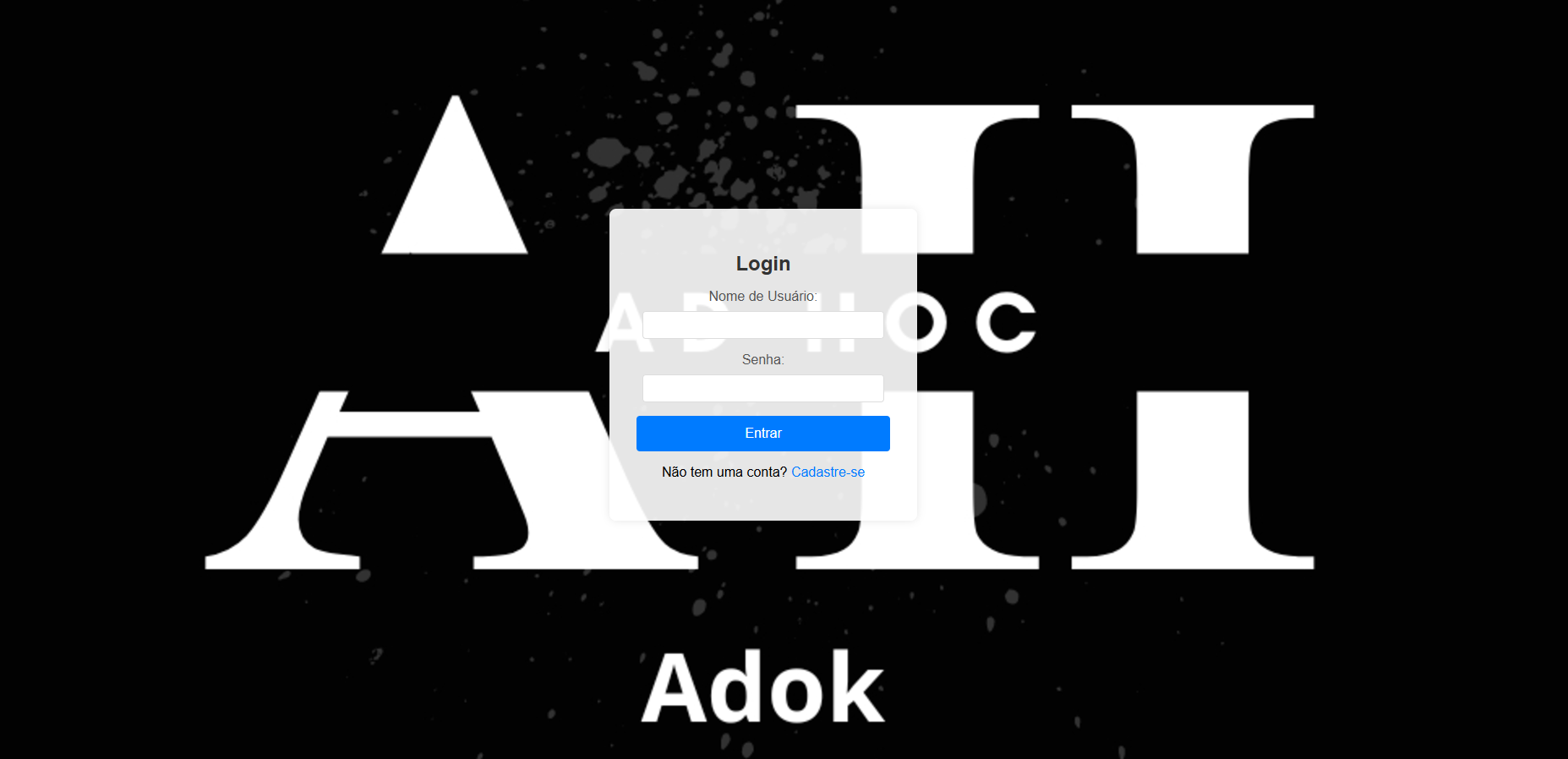
Um diagrama de atividade, conforme GROW (2024), representa o fluxo de trabalho ou processo de um sistema. Ele mostra as atividades, decisões e ramificações envolvidas na execução de uma função ou processo, ajudando a 32 visualizar o comportamento dinâmico e o fluxo de controle entre atividades.

**Fonte: MAROSTEGA, 2022**

# Telas









# Conclusão

Adok (Ad Hoc) foi desenvolvido para ser uma ferramenta completa, pensada para garantir a praticidade e eficiência no agendamento de serviços de beleza, sendo a escolha perfeita para quem busca qualidade, conforto e rapidez no dia a dia. Em um futuro próximo, a plataforma poderá expandir suas funcionalidades, trazendo novas ferramentas para tornar ainda mais simples o processo de cuidar de si mesmo.

O Adok é uma plataforma inovadora criada para simplificar o agendamento de serviços de beleza, incluindo barbeiros, cabeleireiros, maquiadores e manicures. O objetivo do site é proporcionar aos usuários uma maneira prática, rápida e eficiente de agendar seus serviços de beleza, garantindo que tenham acesso aos melhores profissionais, sempre no horário mais conveniente.

Através da plataforma, os clientes podem facilmente consultar a disponibilidade dos profissionais, agendar seus atendimentos e receber confirmações instantâneas, otimizando o processo de marcação e evitando possíveis contratempos. Além disso, a ferramenta permite que os profissionais de beleza acompanhem seus compromissos de forma organizada, facilitando a gestão de seus horários e aumentando a satisfação dos clientes.

Com Adok, a experiência de agendamento se torna simples e transparente, contribuindo para uma gestão eficiente e melhorando a experiência tanto dos usuários quanto dos profissionais. O site foi projetado para tornar a escolha de serviços de beleza mais acessível, oferecendo uma interface intuitiva que pode ser utilizada por qualquer pessoa, sem a necessidade de um grande esforço ou conhecimento técnico.

O sistema também oferece recursos para os profissionais atualizarem suas ofertas, disponibilizando informações sobre serviços, preços e horários disponíveis. Ao automatizar a gestão de agendamentos, Adok não apenas facilita o dia a dia de quem oferece os serviços, mas também garante um atendimento de alta qualidade para quem busca cuidados com a aparência.

# REFERÊNCIAS

Cirne, R. (2024). Banco de Dados Relacional: MySQL na Prática. DataTech Press.

Cristo, M. (2024). Desenvolvimento Web com PHP: Do Básico ao Avançado. TechMaster Editions.

Eich, B. (2024). Fundamentos da Web: HTML, CSS e JavaScript. TechBooks Publishing.

CHATGPT (comp.). **Get answers. Find inspiration. Be more productive**. 2021. Disponível em: https://openai.com/chatgpt/. Acesso em: 08 ago. 2024.

Mesquita, P. (2024). Gestão de Projetos: Documentação e Eficiência Organizacional. ProjectMaster Publications.

Pressman, R. S. (2014). Software Engineering: A Practitioner’s Approach. McGraw-Hill.

Pahl, C., & Booch, G. (2012). Cloud Computing: Principles and Paradigms. Wiley.

Myers, G. J., & Sandler, C. (2004). The Art of Software Testing. Wiley.

McConnell, S. (2004). Code Complete. Microsoft Press.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modeling Language User Guide. Addison-Wesley.

Schwalbe, K. (2018). Information Technology Project Management. Cengage Learning..

Pressman, R. S. (2014). Software Engineering: A Practitioner’s Approach. McGraw-Hill.

Kendall, K. E., & Kendall, J. E. (2014). Sistemas de Informação. São Paulo: Pearson.

Pressman, R. S. (2016). Engenharia de Software. Porto Alegre: AMGH Editora.

Sommerville, I. (2011). Engenharia de Software. São Paulo: Pearson.

Elmasri, R., & Navathe, S. B. (2011). Sistemas de Banco de Dados. São Paulo: Pearson.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Sistemas de Banco de Dados. São Paulo: Bookman Editora.

Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson Education.

Booch, G. (2005). *Object-Oriented Analysis and Design with Applications* (3rd ed.). Addison-Wesley.

RODRIGUES, Douglas. Orientações básicas na elaboração de um diagrama de classes. 2016. DEVMEDIA. Disponível em: https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama de-classes/37224. Acesso em: 19 set. 2024.

GROW, Dave. O que é um diagrama de sequência UML? 2024. Disponível em: https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml. Acesso em: 20 set. 2024.

GROW, Dave. O que é diagrama de atividades UML? 2024. Disponível em: https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-atividades-uml. Acesso em: 25 set. 2024.

1. Especialista em Educação Permanente: Saúde e educação pela FioCruz – Fundação Osvaldo Cruz. Especialista em tecnologias da Informação pela UNIVEL – União Educacional de Cascavel. Pedagoga formada pela UNIPAR – Universidade Paranaense. Professora do núcleo técnico do Estado do Paraná – Ensino médio técnico.

   2Especialização em Educação Profissional Tecnológica. Faculdade Origenes Lessa, FACOL, Brasil. Especialização em Tecnologias e Educação a Distância. Faculdade Origenes Lessa, FACOL, Brasil.Especialização em MBA em Data Warehouse e Business Inteligence.UNYLEYA EDITORA E CURSOS S/A, Unyleya, Brasil. Graduação em Programa Especial de Formação Pedagógica - Docência em Informática. Faculdade de Administração, Ciências, Educação e Letras, FACEL, Brasil. Graduação em Gestão da Tecnologia da Informação.Universidade Estácio de Sá, UNESA, Brasil.

   3Graduação em Sistemas Distribuidos para Internet JAVA.Universidade Federal do Paraná, UTFPR, Brasil. Graduação em Tecnologo em Processamento de Dados. União Educacional de Cascavel, UNIVEL, Brasil. [↑](#footnote-ref-1)