

Análise e Desenvolvimento de Sistemas



Apresentação da equipe

- Nome dos participantes:
 - o Ana Karolyne da Silva
 - o João Vitor Araújo
 - o Luana Figueiredo de Andrade Caboz
 - o Stephanie De Souza Viana

Papéis e responsabilidades de cada participante

Luana – Gerente de projeto e desenvolvedora back end

Stephanie – desenvolvedora back end

Ana Karolyne – desenvolvedora front end

Apresentação do projeto

Descrição do problema

- Justificativa / Contexto de negócio:

Salão de beleza sem sistema de gestão, onde tudo é feito de forma manual. Queremos automatizar alguns processos e centralizar a gestão, para facilitar a administração.
- Descrição das personas:

João, Cabeleireiro do Salão

Idade: 32 anos

Perfil: Profissional autônomo que trabalha no salão e atende vários clientes por dia.

Necessidades: Quer ver sua agenda organizada e acompanhar os serviços realizados.

Dores: Agendamentos confusos e clientes que desmarcam em cima da hora.

Comportamento digital: Usa WhatsApp e Instagram para divulgar o trabalho.

Patrícia, Gerente do Salão

Idade: 40 anos

Perfil: Responsável pela administração do salão, controla caixa, equipe.

Necessidades: Acompanhar fluxo de caixa, organizar equipe e evitar sobrecarga de horários.

Dores: Falta de controle financeiro e dificuldade na gestão de funcionários.

Comportamento digital: Usa planilhas, mas precisa de algo mais automatizado.

Macro descrição da proposta de solução

- Principais funcionalidades previstas
 - Agendamento de Serviços
Permite que os funcionários do salão criem e gerenciem seus horários de forma online.
 - Gestão Financeira (Caixa)
Controle de entradas e saídas do caixa, registro de pagamentos realizados pelos clientes e acompanhamento de recebimentos.
 - Cadastro e Gerenciamento de Clientes
Registro de informações dos clientes, incluindo histórico de agendamentos.
 - Gestão da Equipe
Controle dos profissionais do salão, com atribuição de serviços, acompanhamento de agendas.
 - Gerenciamento de Serviços
Cadastro e personalização dos serviços oferecidos e profissionais responsáveis.
- Escopo do projeto

Plataforma web para acesso via internet.

Painel administrativo para gerenciar clientes, equipe, serviços e finanças.

Funcionalidade de agendamento.

Relatórios financeiros e gestão de caixa.

Macro cronograma

Fase 1 – Planejamento e Levantamento de Requisitos (1 semana)

- Definição detalhada das funcionalidades e do escopo do sistema.
- Escolha das tecnologias para front-end e back-end.

- Definição da estrutura do banco de dados.

Fase 2 – Desenvolvimento Inicial (6 semanas)

- **Semana 1 e 2 – Configuração do ambiente e primeiros módulos**

Back-end: Criação da estrutura inicial do projeto e Banco de Dados.

Front-end: Configuração do projeto e implementação das telas principais.

- **Semana 3 e 4 – Implementação das funcionalidades principais**

Back-end: CRUD de usuários, equipe, clientes e serviços, além da lógica de agendamento.

Front-end: Desenvolvimento das telas de login, cadastro e agendamento.

- **Semana 5 e 6 – Integração e ajustes**

Back-end: Implementação da gestão financeira.

Front-end: Integração com o back-end.

Fase 3 – Testes e Validação (3 semanas)

Testes unitários e de integração.

Revisão da interface e experiência do usuário.

Correção de bugs e otimizações.

Fase 4 – Implantação e Avaliação Final (3 semanas)

Testes finais com o professor.

Ajustes e refinamentos com base no feedback.

Documentação final do projeto.

Detalhamento da Proposta de solução

Regras de negócio

Agendamentos

- Um profissional não pode ter dois atendimentos agendados no mesmo horário.

Gestão de Equipe

- Apenas usuários com permissão de administrador ou gerente podem cadastrar e gerenciar a equipe.

Gestão Financeira

- O sistema deve calcular o total de faturamento com base nas entradas e saídas feitas no caixa.

Requisitos Funcionais

Autenticação e Controle de Acesso

- O sistema deve permitir o cadastro e login de funcionários.
- Deve haver diferentes níveis de acesso: profissional e gerente/administrador.

Agendamento de Serviços

- O profissional pode realizar agendamentos.
- O profissional pode cancelar ou editar um agendamento.

Gestão de Clientes

- O sistema deve armazenar informações básicas dos clientes (nome, telefone, histórico de agendamentos).
- Todos podem visualizar e gerenciar a lista de clientes.

Gestão da Equipe

- O sistema deve permitir o cadastro de profissionais com suas respectivas funções.

Gestão Financeira

- O gerente pode registrar pagamentos e movimentações no caixa.

Requisitos Não Funcionais

Desempenho e Escalabilidade

- O sistema deve suportar múltiplos acessos simultâneos sem perda de desempenho.

Segurança

- Apenas usuários autenticados podem acessar funcionalidades protegidas.
- Deve haver restrição de acesso conforme o nível do usuário.

Usabilidade e Interface

- O sistema deve ser responsivo, funcionando bem em desktop, tablets e celulares.

Manutenção e Evolução

- O código deve ser bem estruturado para facilitar futuras manutenções.
- A documentação do projeto deve estar atualizada para facilitar melhorias futuras.

Diagramas

Diagrama de classes

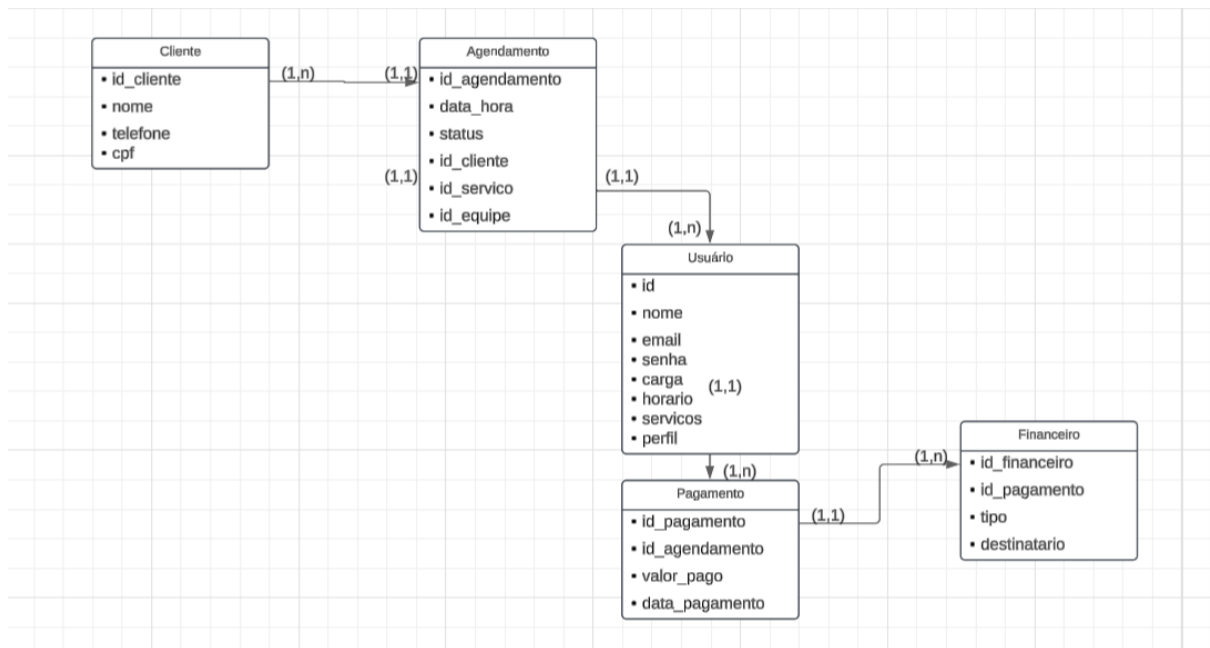
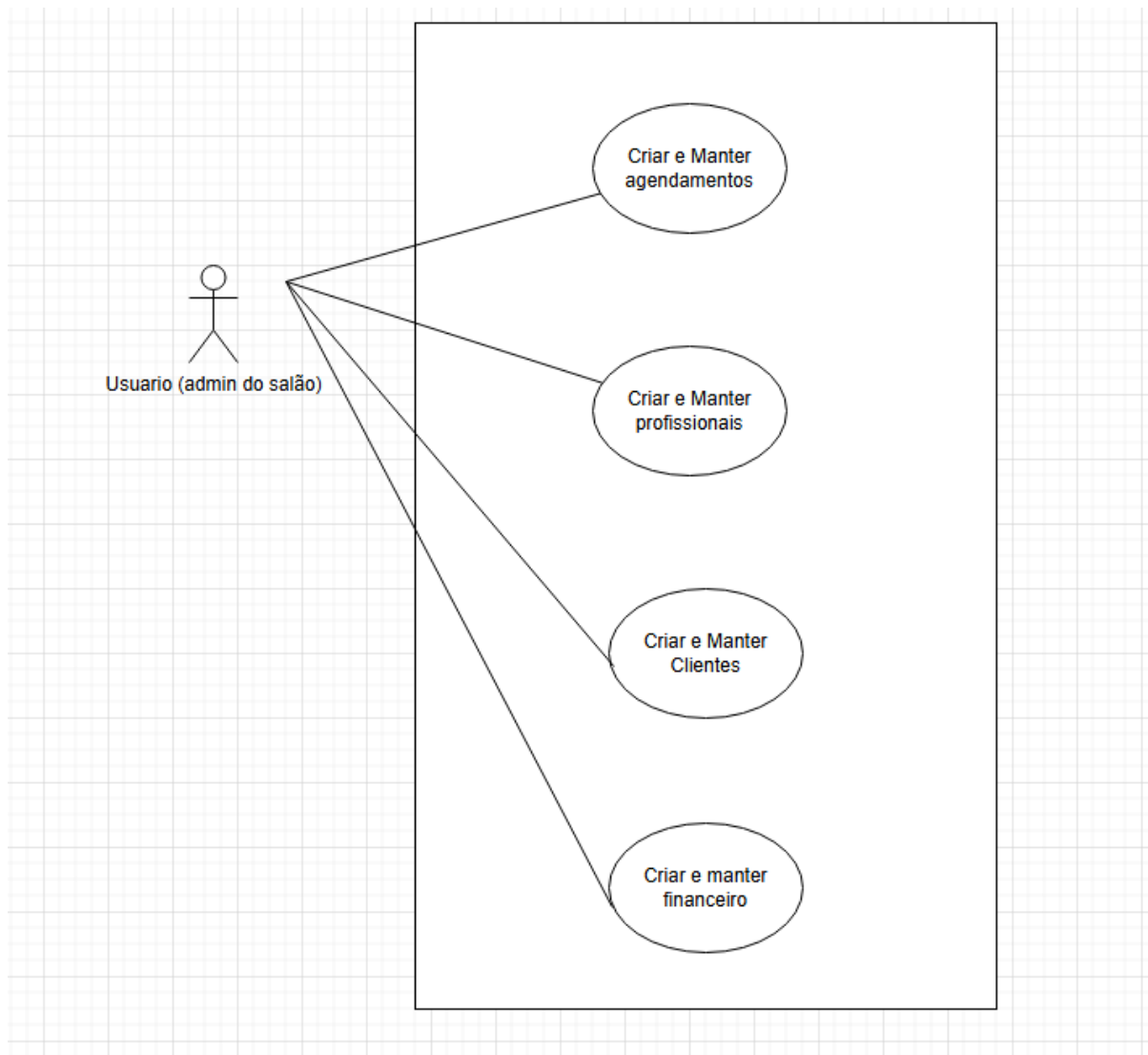


Diagrama de caso de uso



Modelos

- Banco de dados
 - o MER/DER - enviado separadamente

- o **Dicionário de Dados:**

Tabela Cliente:

Nome do campo	Tipo de Dado	Descrição	Observação
id_cliente	integer	Identificador único do cliente	PK
nome	VARCHAR(100)	Nome completo do cliente	Not null
telefone	VARCHAR(15)	Número de telefone do cliente	Not null
cpf	VARCHAR(11)	CPF do cliente	Not Null

Tabela Agendamento:

Nome do campo	Tipo de Dado	Descrição	Observação
id_agendamento	integer	Identificador único do agendamento	PK
data_hora	TIMESTAMP	Data e horário do agendamento	NOT NULL
status	ENUM ('pendente', 'concluído', 'cancelado')	Status do agendamento	DEFAULT 'pendente'
id_cliente	integer	Cliente que fez o agendamento	FK → Cliente(id_cliente)
id_profissional	integer	Profissional que atenderá o cliente	FK → profissional(id_profissional)

Tabela Usuario:

Nome do campo	Tipo de Dado	Descrição	Observação
id_usuario	integer	Identificador único do usuario	PK
nome	VARCHAR(100)	Nome do usuario	Not null
email	VARCHAR(100)	Email do usuario	Not null
senha	VARCHAR(50)	Senha do usuario	Not null
cargo	VARCHAR(50)	Cargo do usuario	Not null
horario	VARCHAR(50)	Horario de trabalho do usuario	Not null
servicos	VARCHAR(50)	Servicos prestados pelo usuario	Not null

perfil	VARCHAR(20)	Admin ou profissional	Not null
--------	-------------	-----------------------	----------

Tabela Financeiro:

Nome do campo	Tipo de Dado	Descrição	Observação
id_financeiro	UUID	Identificador único do registro	PK
descricao	VARCHAR(255)	descrição da movimentação financeira	Not null
tipo_pagamento	VARCHAR(255)	forma de pagamento da movimentação financeira	Not null
preco	VARCHAR(255)	preço da movimentação financeira	Not null
status	VARCHAR(255)	status da movimentação financeira	Not null
categoria	VARCHAR(255)	tipo de movimentação financeira	Not null

Especificação da arquitetura de software

Arquitetura Modular com Controllers.

- **Divisão por módulos** → Cada funcionalidade do sistema tem seu próprio módulo (financeiro, agendamento, etc.).
- **Camadas organizadas** → Controllers lidam com as requisições, e processam a lógica de negócio.

Ecossistema da solução

- *Refere-se ao conjunto de ferramentas, tecnologias, plataformas e serviços que são utilizados para desenvolver, implementar e operar a solução de software. Isso inclui tanto os componentes internos do sistema quanto as integrações com sistemas externos e serviços de terceiros. O ecossistema da solução abrange todo o ambiente em que o software opera. Neste sentido devemos representar o usuário no ecossistema da solução. O usuário é uma parte fundamental do sistema, e sua interação com a solução deve ser considerando design e na implementação. No ecossistema da solução, o usuário é representado como uma entidade que interage com o sistema através de diferentes interfaces e dispositivos.*

Tipos de Usuários e Acessos

Usuários → Controlam agendamentos, profissionais e financeiro.

Interfaces Disponíveis

Web App (React.js) → Interface responsiva acessível por desktop, tablet e mobile.

API REST (Node.js) → *Back end, armazena a lógica e é a camada intermediária entre o front end e o banco de dados*

Definição das tecnologias a serem empregadas na solução

Linguagem

- Front-end: NextJs (React.js)
- Back-end: Nest.js (Node.js)

Banco de dados

- Postgres

Segurança

- JWT

Repositórios

- Controle de versão: Git
- Plataforma: GitHub (para versionamento e colaboração da equipe)

Provedores

- Docker

Recursos utilizados no projeto

- VS Code: Editor de código principal
- Postman: Testes de API durante o desenvolvimento
- Jest: Framework de testes para validação do back-end

Frameworks de desenvolvimento

- Back-end: Nest.js
- Front-end: Next.JS
- Testes: Jest

Softwares de apoio

- Postman: Para testar requisições à API
- Testes unitários: Jest