Fatec São José do Rio Preto

Curso Superior de Tecnologia em Informática para Negócios

Projeto de Extensão Comunitária

(Curricularização)

Desenvolvimento de um MPV (Mínimo Produto Viável) de software para gestão de uma organização

Integrantes do Projeto:

Álvaro Reis Soares
Carlos Daniel Xavier Moreira da Silva
Fernando Euzébio
Gabriel Henrique Pagani Silva
Santiago Andres de Souza

São José do Rio Preto

Semestre 2/2024

Sumário

1. Características do Projeto de Extensão

Título	Desenvolvimento de um MPV (Mínimo Produto Viável) de software para gestão de uma organização (3º Semestre).
Temática	☐Programas ☒Projetos ☐Cursos e Oficinas ☐Eventos ☐Prestação de Serviços
Descrição	O projeto poderá se basear nos resultados obtidos em estudos de projetos de extensão anteriores. A partir de levantamento de regras de negócio, sistemas de informação demandados e tecnologias emergentes, deverá consistir do desenvolvimento de um MPV de um software/sistema para gestão de uma parte delimitada das regras de negócio existentes na organização estudada.
Objetivos	▪ Analisar as regras de negócio e relatórios diagnósticos para produzir as documentações de Engenharia de Software; ▪ Projetar, desenvolver, testar e entregar um MPV funcional; ▪ Projetar, modelar e implementar a base de dados; ▪ Aplicar técnicas de gerenciamento de equipes a partir de análises das habilidades técnicas e humanas dos integrantes.
Carga horária	▪ Total: **130** horas/aula.
Público-alvo	Empresas, ONGs, órgãos públicos e entidades sociais.
Ações/Etapas de execução	1. Desenvolver documentação de Engenharia de Software; 2. Programar, testar e efetuar implantação em ambiente de testes para um MPV de sistema/software; 3. Modelar, implementar e testar uma base persistente de dados para armazenamento dos dados da solução MPV desenvolvida; 4. Documentar e relatar as ações e funções desempenhadas por cada integrante da equipe executora do projeto.
Entregas	▪ 01 Projeto de banco de dados do MPV; ▪ 01 Documentações de software do MPV; ▪ 01 Código-fonte do MPV em repositório online na internet (*Github* ou similar); ▪ 01 Relatório de composição de equipe, funções e suas descrições de responsabilidades, relato de execução de tarefas para o desenvolvimento do MPV, descrição das

	personas envolvidas e suas habilidades pessoais e técnicas utilizadas.
Instrumentos e procedimentos de avaliação	**Aluno** – trabalho em grupo, eficácia na realização das tarefas, entrega digital do resultado das tarefas em relatório padronizado de atividade de extensão. **Projeto** – resultados obtidos, publicação dos resultados em repositório online, envio de resultados para a entidade beneficiada na comunidade ou execução de evento para apresentação dos resultados e integração com representantes da comunidade externa beneficiada.
Componente(s)curricular(es) envolvidos	▪ Banco de Dados (30 horas/aula); ▪ Linguagens de Programação I (40 horas/aula); ▪ Engenharia de Software (40 horas/aula); ▪ Gestão de Equipes (20 horas/aula).
Formas de evidência	▪ Entrega de relatório de atividade de extensão com os resultados em cada disciplina.

2. Informações da Equipe de Trabalho

Nome do Grupo: Grupo 01

Nr.	**Nome Completo do(s) Aluno(s)**	**Contatos**	
01	Álvaro Reis Soares	Email:	alvoreis@gmail.com
		Whatsapp:	+55 17 99227-3582
02	Carlos Daniel Xavier Moreira da Silva	Email:	cs520589@gmail.com
		Whatsapp:	+55 17 99671-0451
03	Fernando Euzébio	Email:	fernandinhoreis100@gmail.com
		Whatsapp:	+55 17 99275-1468
04	Gabriel Henrique Pagani Silva	Email:	s.paganigabriel@gmail.com
		Whatsapp:	+55 17 99632-1558
05	Santiago Andres de Souza	Email:	supersantisouza@gmail.com
		Whatsapp:	+55 17 99724-7745

3. Etapa 1 – Desenvolvimento de documentação de Engenharia de Software

Objetivo: Analisar as regras de negócio e relatórios diagnósticos para produzir as documentações de Engenharia de Software.

3.1 Paradigma (Ciclo de Vida)

A problemática se refere à complexidade existente no controle manual dos agendamentos de serviços de barbearia e controle dos mesmos. Visando sanar essa questão, pensamos em um sistema que seria responsável por gerir os agendamentos. A partir disso, surge a oportunidade de expandir o escopo do mesmo, já que, tendo os agendamentos em mãos, torna-se possível análises avançadas do negócio como: Serviços e profissionais mais solicitados, qtd de clientes atendidos por profissional (Ajudaria no cálculo do salário do mesmo), sistema de agendamento dinâmico, facilitando as escalas e controle de horários, etc.

3.2 Descrição do Sistema

Um salão de beleza gerencia seus agendamentos, clientes, profissionais e serviços através de um sistema.

Clientes interessados em agendar um horário primeiramente precisam ser cadastrados no sistema, caso ainda não o sejam. Uma vez cadastrados, os clientes podem ter seus dados (nome, telefone, email, senha, status) atualizados ou, em casos específicos, serem excluídos. O sistema permite listar todos os clientes para consulta.

Para a prestação de serviços, o salão conta com profissionais, que também são cadastrados no sistema com suas informações (nome, telefone, email, senha, salário fixo, comissão e status). Seus dados podem ser atualizados e, assim como os clientes, podem ser excluídos ou listados para visualização.
Os serviços oferecidos pelo salão (corte, coloração, manicure, etc.) são igualmente cadastrados no sistema, com sua descrição, valor e tempo médio de execução. Esses serviços podem ser atualizados, excluídos ou listados.

O sistema mantém um catálogo que associa quais profissionais realizam quais serviços. Um item é adicionado ao catálogo selecionando um profissional e um serviço. O catálogo pode ser listado para verificar as especialidades de cada profissional, e os itens podem ser atualizados (alterando o status) ou removidos.

A funcionalidade central do sistema são as agendas. Um cliente pode criar uma nova agenda, especificando a data e a hora desejadas. Todas as agendas criadas podem ser listadas, mostrando quais serviços já foram agendados para cada uma. Agendas podem ser excluídas ou ter sua data, hora e status atualizados.

Ao agendar um serviço em uma agenda existente, o cliente (ou o operador do sistema) escolhe um serviço e o sistema verifica a disponibilidade dos profissionais para aquele serviço no horário da agenda, considerando a duração do serviço e outros agendamentos existentes. A grade de disponibilidade exibe os horários livres e ocupados de todos os profissionais para um dia específico, enquanto a disponibilidade do profissional mostra os horários livres de um profissional em particular para um

determinado dia. Após a escolha do profissional disponível, o serviço é então agendado para aquela agenda. Os serviços agendados em uma agenda podem ser editados para alterar seu status (ativo ou cancelado).

3.3 Problema a ser Resolvido

O sistema de agendamento para barbearia foi pensado para ser simples e eficiente, proporcionando uma experiência rápida e sem complicação. Os clientes poderão escolher os horários disponíveis e confirmar seus serviços de forma prática, com uma interface fácil de usar. A plataforma será acessível via web, sem precisar de instalação, tornando o processo de agendamento muito mais ágil. Além disso, será possível consultar os agendamentos e cadastros diretamente pelo site, ou até mesmo na própria barbearia, no programa, de forma simples e rápida.

Os barbeiros poderão acessar sua agenda para gerenciar seus compromissos de forma organizada, enquanto os administradores terão o controle total para ajustar horários e os serviços oferecidos.

Com foco no desempenho, o sistema será leve e rápido, garantindo respostas rápidas, seja para agendar ou cancelar um horário. Para manter tudo seguro e funcionando bem, backups automáticos serão feitos todos os dias, e as confirmações de agendamento serão enviadas por e-mail ou SMS, conforme preferir o cliente. Além disso, o sistema contará com recursos de segurança, como a prevenção de agendamentos duplicados e uma forma fácil de recuperação de senha.

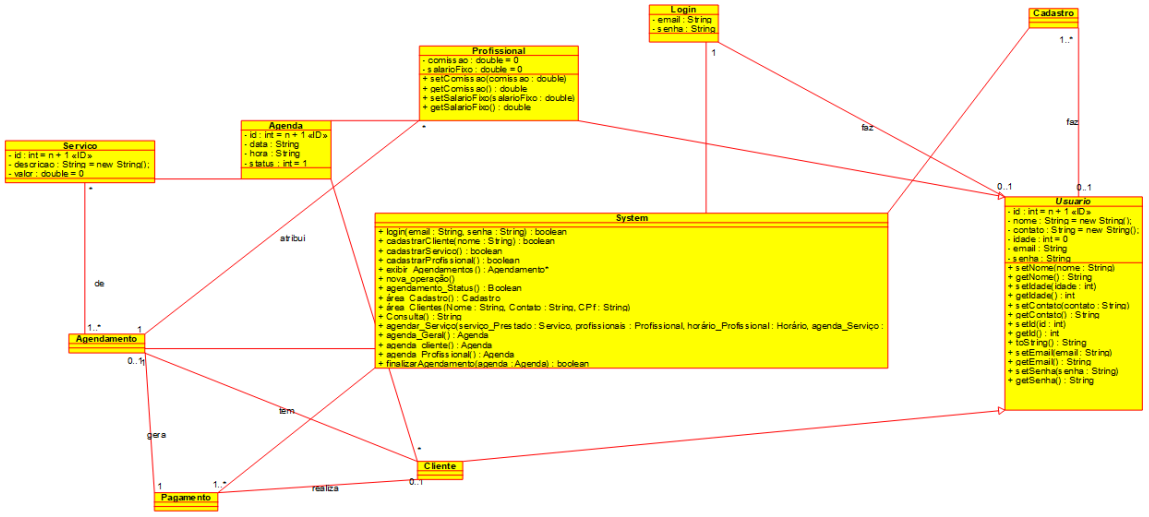
Embora a plataforma seja bem simples, algumas limitações iniciais vão existir, como a falta de integração com pagamentos online e o suporte apenas para uma unidade da barbearia. Mesmo assim, o fluxo será eficiente: o cliente se cadastra com as informações básicas, escolhe o serviço desejado e recebe a confirmação do agendamento, tudo de forma tranquila. Assim, o sistema proporciona uma experiência prática, confiável e organizada tanto para os clientes quanto para os barbeiros.

3.4 Relação das Funcionalidades (funções)

Nr.	Nome	Breve Descrição
01	Cadastrar Cliente	Funcionalidade necessária para efetuar o cadastro (inclusão, alteração e alteração) de clientes.
02	Cadastrar Profissional	Funcionalidade necessária para efetuar o cadastro (inclusão, alteração e alteração) de profissionais.
03	Cadastrar Serviço	Funcionalidade necessária para efetuar o cadastro (inclusão, alteração e alteração) de serviços.
04	Cadastrar Catálogo	Funcionalidade necessária para efetuar o cadastro (inclusão, alteração e alteração) de catálogos, ou seja, serviços prestados por profissional.
05	Criar Agenda	Funcionalidade necessária para efetuar o agendamentos de horários (inclusão, alteração

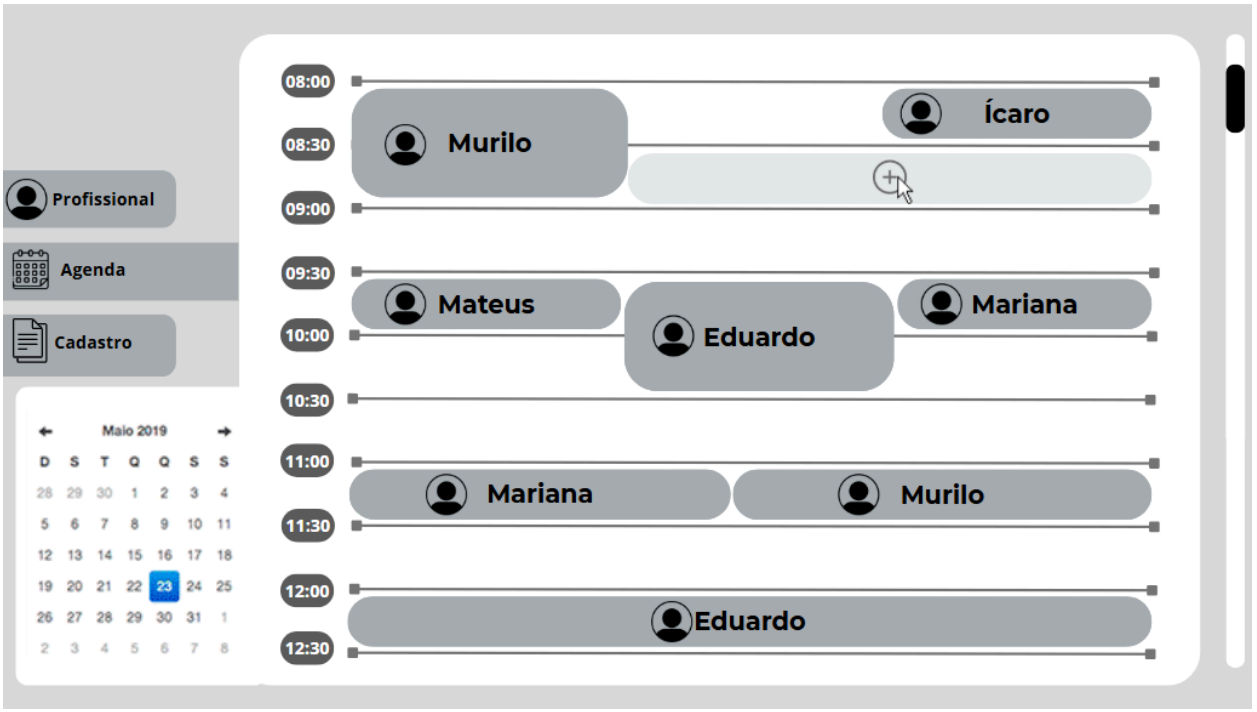
		e alteração) a fim de adicionar Serviços Agendados.
06	Adicionar Serviço Agendado	Funcionalidade que permite registrar pedidos de produtos solicitados pelos clientes. Associado Serviço e Profissional (Desde que o profissional preste o determinado serviço (Catalogo)).

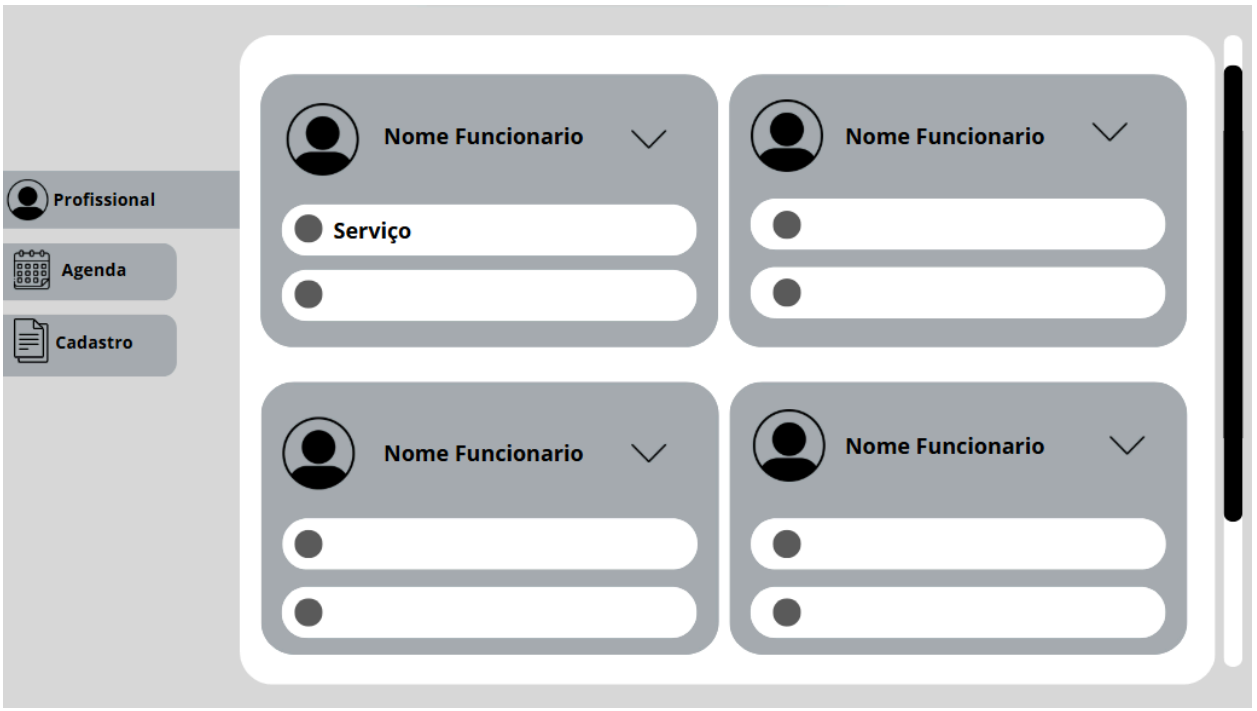
3.5 Diagrama de Classe

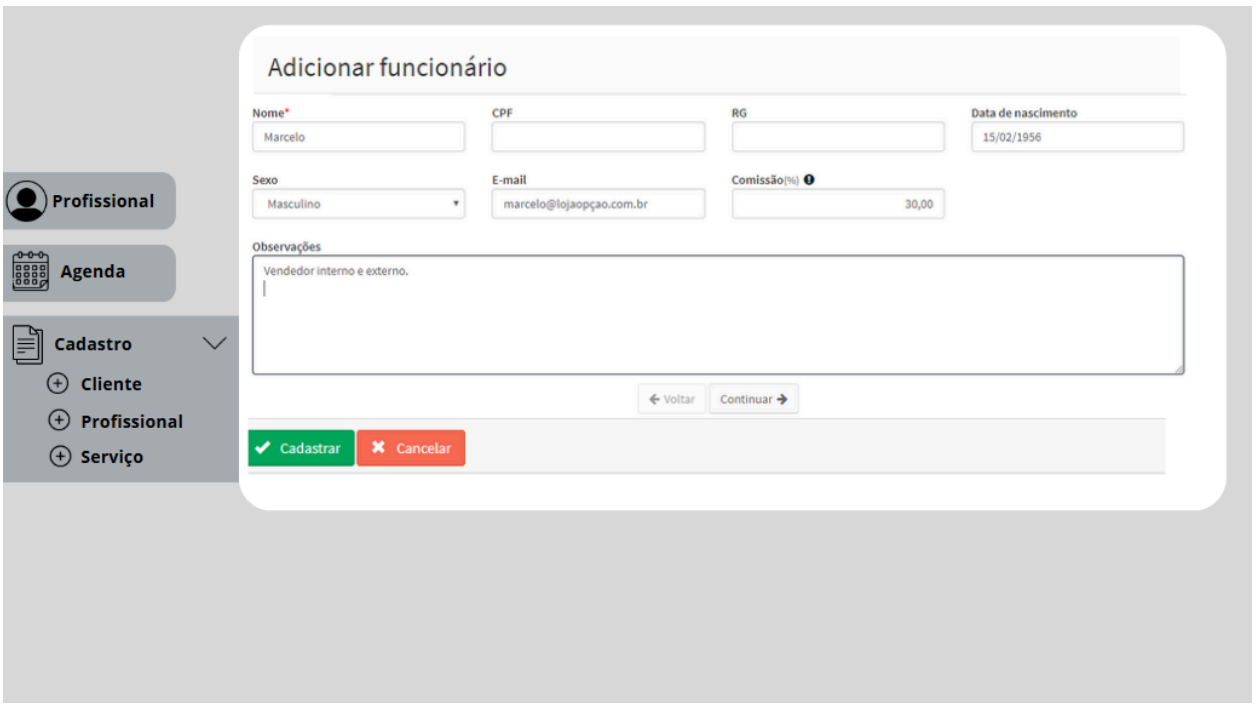


3.6 Protótipo

Imagem das páginas da aplicação que vocês estão desenvolvendo em LP1.




Profissional
Agenda
Cadastro
Nome Funcionario
Serviço
Nome Funcionario
Nome Funcionario
Nome Funcionario

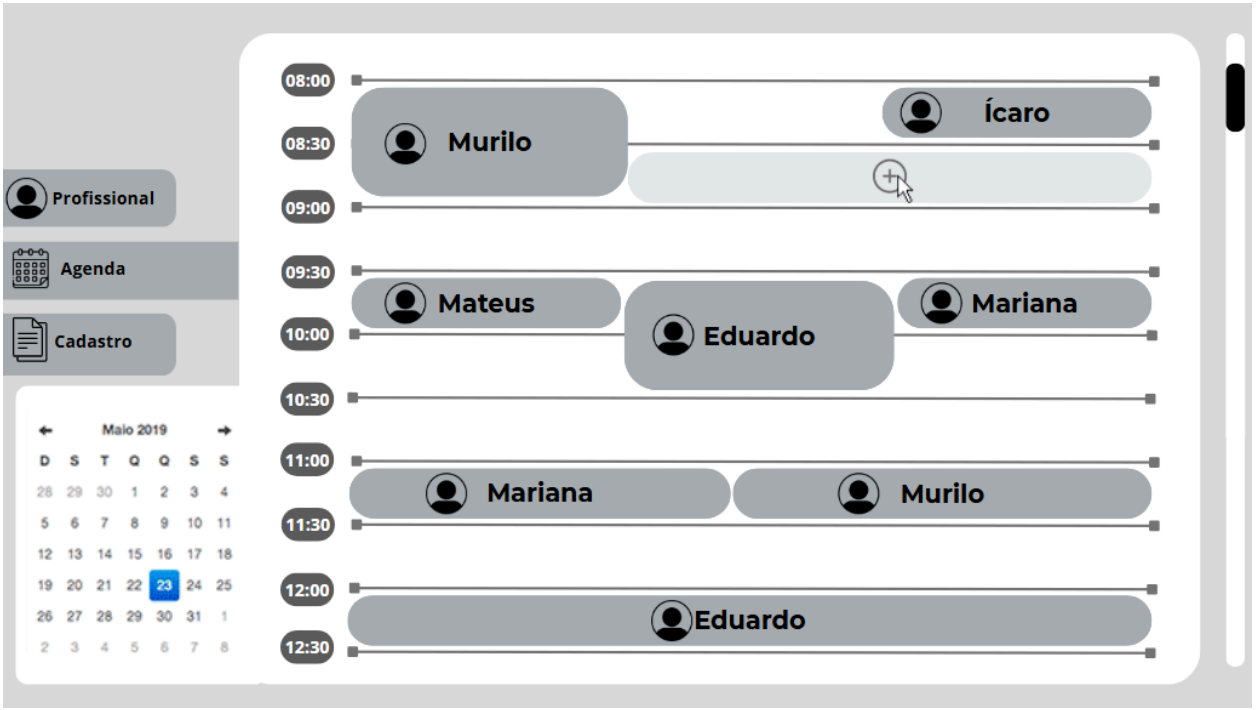

Profissional
Agenda
Cadastro
Cliente
Profissional
Serviço
Adicionar funcionário
Nome*
Marcelo
CPF
RG
Data de nascimento
15/02/1956
Sexo
Masculino
E-mail
marcelo@lojaopçao.com.br
Comissão(%)
30,00
Observações
Vendedor interno e externo.
Voltar
Continuar
Cadastrar
Cancelar

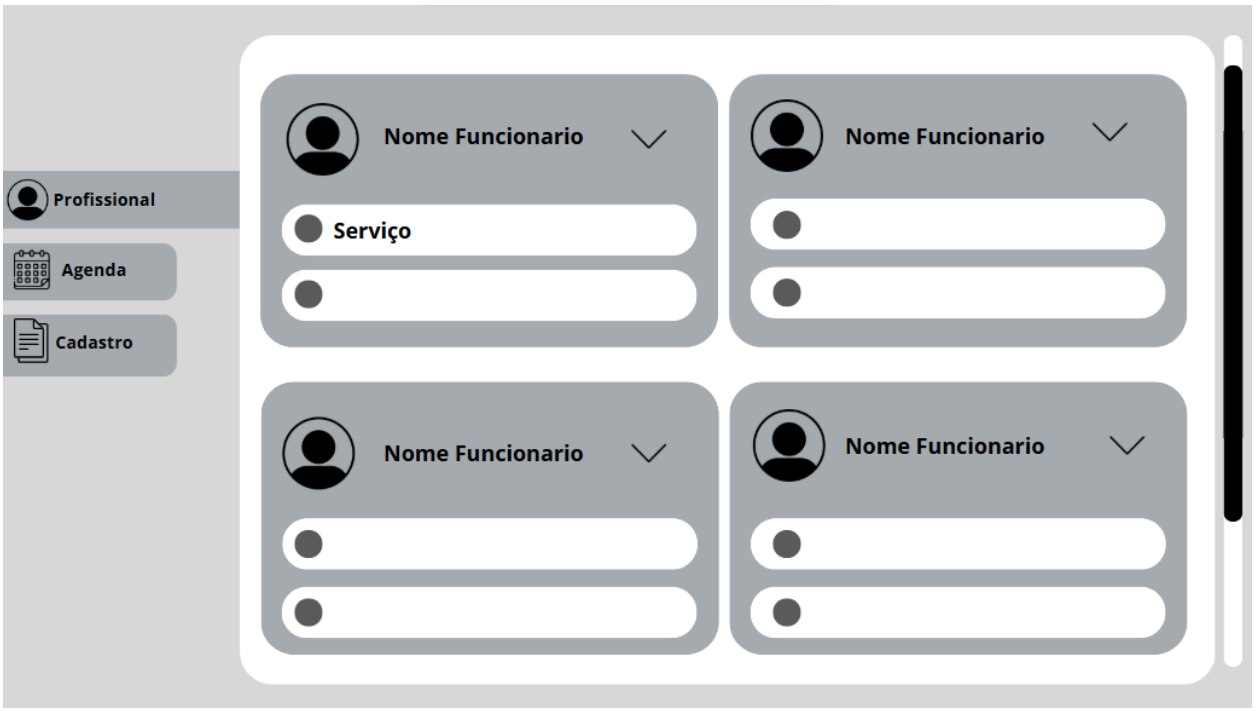
4. Etapa 2 – Programação, teste e implantação em ambiente de testes para um MPV de sistema/software

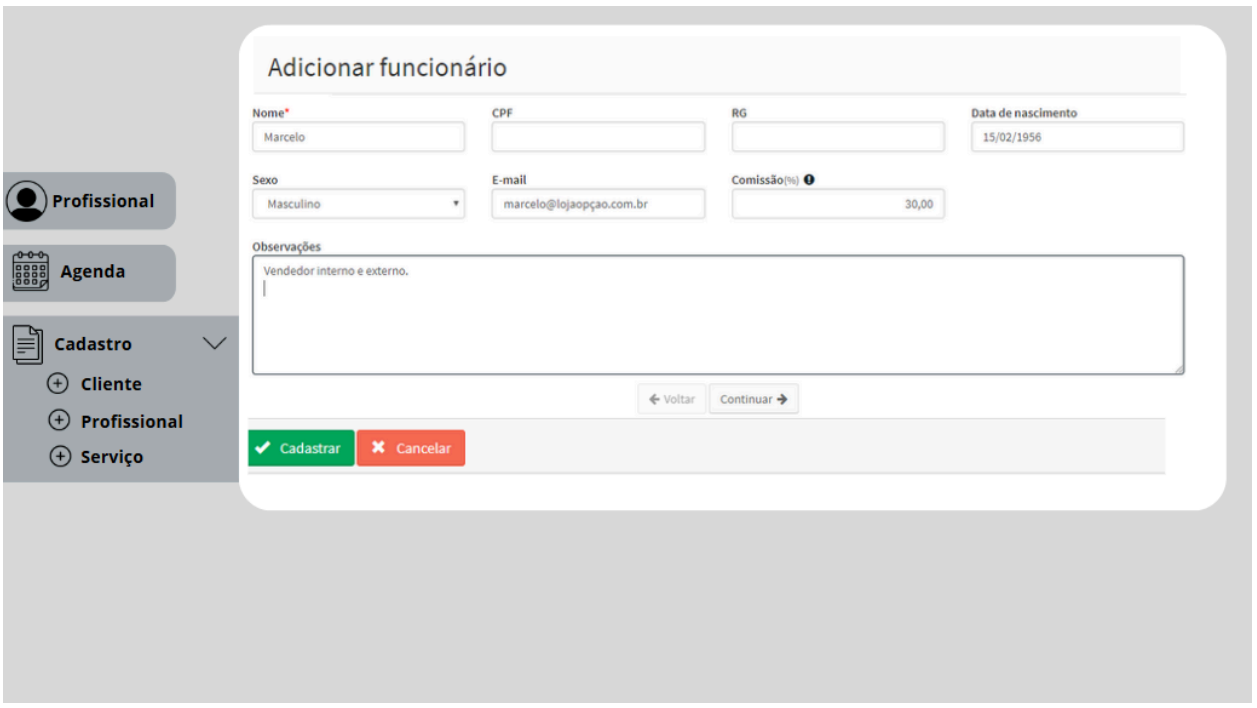
Objetivo: Projetar, desenvolver, testar e entregar um MPV funcional.

4.1 Interfaces gráficas desenvolvidas

Até o momento não foi desenvolvido interfaces. Há apenas “protótipos”







4.2 Classes, atributos e métodos (pacote model)

FATECInter3P/Linguagem de Programação 1/src/entities at main · CarlosDanielXMS/FATECInter3P

4.3 Link do projeto no github (atualizar o READ.me)

FATECInter3P/Linguagem de Programação 1 at main · CarlosDanielXMS/FATECInter3P

4.4 Injeção de dados (Script do banco de dados)

FATECInter3P/Linguagem de Programação 1/src/banco at main · CarlosDanielXMS/FATECInter3P

FATECInter3P/Linguagem de Programação 1/src/repository at main · CarlosDanielXMS/FATECInter3P

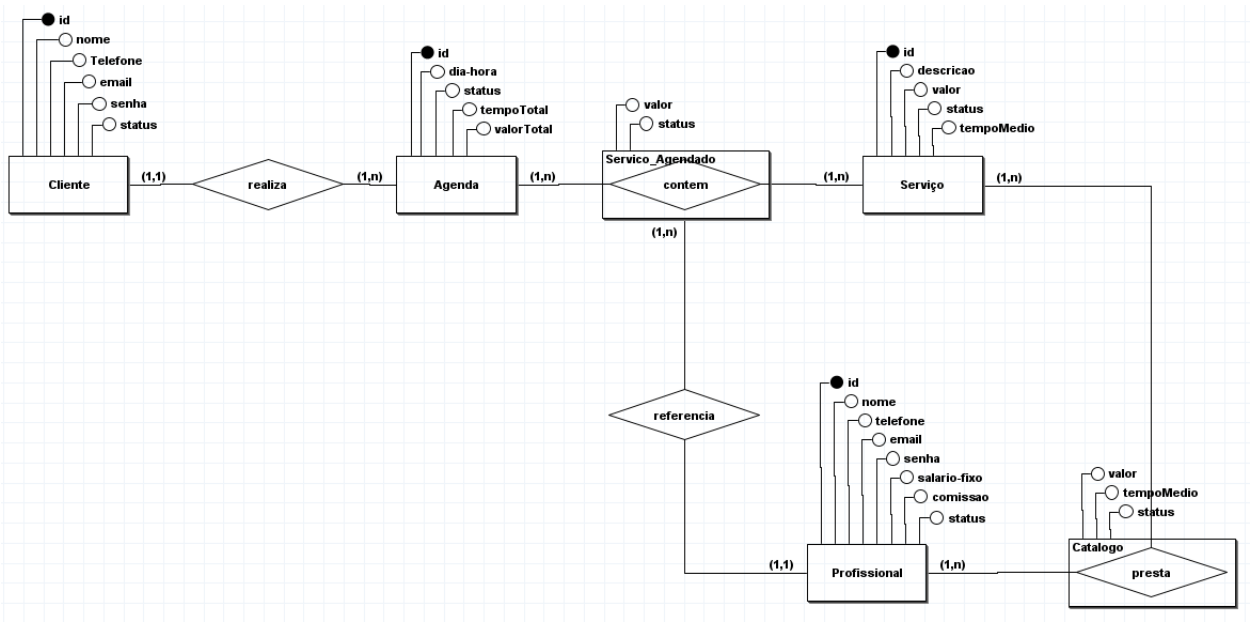
contendo no mínimo 10 inserts em cada tabela

"Não fizemos os 10 inserts. Porém fizemos toda a lógica do CRUD e fluxo do sistema."

5. Etapa 3 – Modelagem, implementação e teste de uma base persistente de dados para armazenamento dos dados da solução MPV desenvolvida

Objetivo: Projetar, modelar e implementar a base de dados.

5.1 Diagrama do Modelo Conceitual



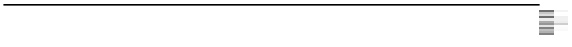
5.2 Mapeamento para Modelo Lógico

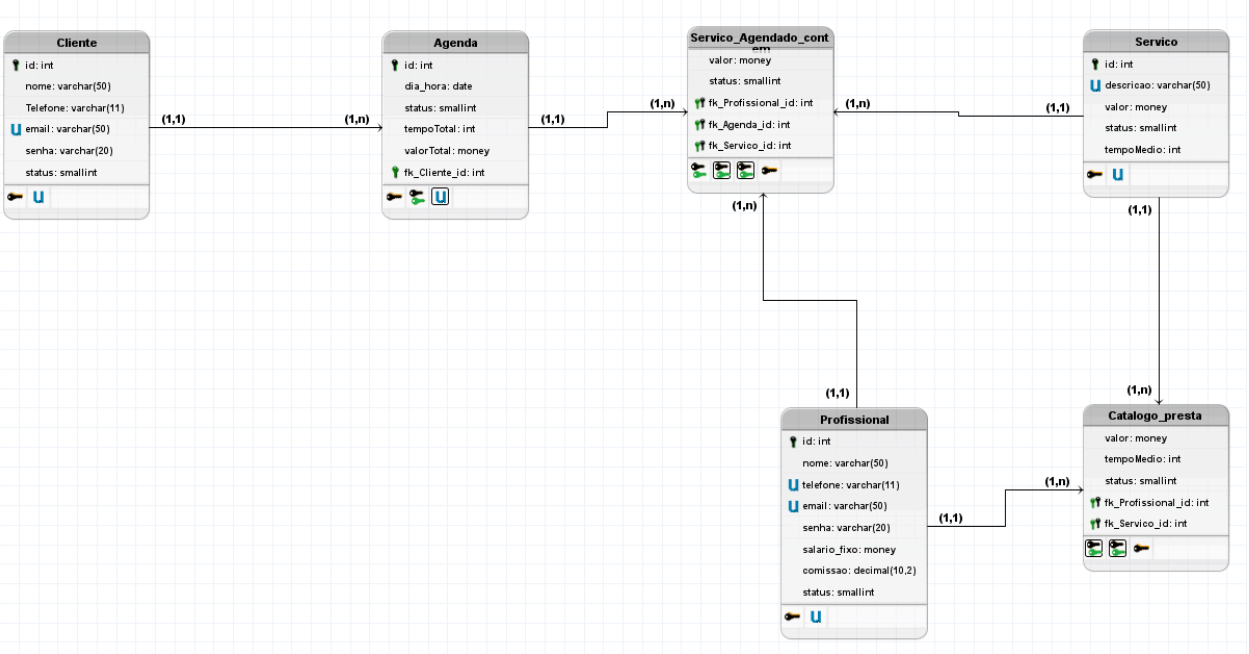
1. Cliente
   - Nome da Tabela: CLIENTE
   - Atributos:
     - id_cliente (INT, PK, Auto-incremento): Identificador único do cliente.
     - nome (VARCHAR): Nome completo do cliente.
     - telefone (VARCHAR): Número de telefone do cliente.
     - email (VARCHAR, UNIQUE): Endereço de e-mail do cliente (deve ser único).
     - senha (VARCHAR): Senha do cliente (idealmente armazenada de forma criptografada).
     - status (INT): Status do cliente (ex: 1 para ativo, 0 ou 2 para inativo/cancelado).
2. Profissional
   - Nome da Tabela: PROFISSIONAL
   - Atributos:
     - id_profissional (INT, PK, Auto-incremento): Identificador único do profissional.
     - nome (VARCHAR): Nome completo do profissional.
     - telefone (VARCHAR): Número de telefone do profissional.
     - email (VARCHAR, UNIQUE): Endereço de e-mail do profissional (deve ser único).
     - senha (VARCHAR): Senha do profissional (idealmente armazenada de forma criptografada).
     - status (INT): Status do profissional (ex: 1 para ativo).

    - salario_fixo (DECIMAL): Salário fixo do profissional.
    - comissao (DECIMAL): Percentual de comissão do profissional.

3. Serviço
   - Nome da Tabela: SERVICO
   - Atributos:
     - id_servico (INT, PK, Auto-incremento): Identificador único do serviço.
     - descricao (VARCHAR): Descrição do serviço (ex: "Corte de Cabelo", "Manicure").
     - valor (DECIMAL): Valor do serviço.
     - tempo_medio (INT): Tempo médio de duração do serviço em minutos.
     - status (INT): Status do serviço (ex: 1 para ativo).
4. Catálogo
   - Nome da Tabela: CATALOGO
   - Atributos:
     - id_profissional (INT, PK, FK): Referencia id_profissional da tabela PROFISSIONAL.
     - id_servico (INT, PK, FK): Referencia id_servico da tabela SERVICO.
     - status (INT): Status da associação (ex: 1 para ativo, indicando que o profissional oferece aquele serviço).
     - Chave Primária Composta: (id_profissional, id_servico)
5. Agenda
   - Nome da Tabela: AGENDA
   - Atributos:
     - id_agenda (INT, PK, Auto-incremento): Identificador único do agendamento principal.
     - id_cliente (INT, FK): Referencia id_cliente da tabela CLIENTE.
     - data_hora (DATETIME): Data e hora do agendamento.
     - status (INT): Status do agendamento (ex: 1 para agendado, 2 para cancelado, 3 para concluído).
6. ServicoAgendado
   - Nome da Tabela: SERVICO_AGENDADO
   - Atributos:
     - id_servico_agendado (INT, PK, Auto-incremento): Identificador único para cada serviço dentro de uma agenda.
     - id_agenda (INT, FK): Referencia id_agenda da tabela AGENDA.
     - id_servico (INT, FK): Referencia id_servico da tabela SERVICO.
     - id_profissional (INT, FK): Referencia id_profissional da tabela PROFISSIONAL.
     - status (INT): Status do serviço agendado (ex: 1 para ativo, 2 para cancelado).

5.3 Modelo Lógico de Banco de Dados com Tipos de Dados SQL Server





5.4 Dicionário de Dados

Cliente										
Coluna	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**UK**	**FK**	**Referência**	**Check**
id	int	4	10	0	Não	sim	sim	não	-	-
nome	varchar	50	-	-	Não	não	não	não	-	-
telefone	varchar	11	-	-	Não	não	não	não	-	-
email	varchar	30	-	-	sim	não	sim	não	-	-
senha	varchar	20	-	-	sim	não	não	não	-	> 8 caracteres
status	smallint	4	10	0	não	não	não	não	-	(1, 2)

Servico										
Coluna	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**UK**	**FK**	**Referência**	**Check**
id	int	4	10	0	não	sim	sim	não	-	-
descricao	varchar	50	-	-	não	não	sim	não	-	-
valor	money	5	20	3	não	não	não	não	-	>0
tempoMedio	int	4	10	0	não	não	não	não	-	>0
status	smallint	4	10	0	não	não	não	não	-	(1, 2)

Profissional										
Coluna	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**UK**	**FK**	**Referência**	**Check**
id	int	4	10	0	não	sim	sim	não	-	-
nome	varchar	50	-	-	não	não	não	não	-	-
telefone	varchar	11	-	-	não	não	não	não	-	-
email	varchar	30	-	-	sim	não	sim	não	-	-
senha	varchar	20	-	-	sim	não	não	não	-	> 8 caracteres
salarioFixo	money	5	20	3	não	não	não	não	-	>0
comissao	decimal	5	10	2	não	não	não	não	-	>=0
status	smallint	4	10	0	não	não	não	não	-	(1, 2)

Agenda										
Coluna	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**UK**	**FK**	**Referência**	**Check**
id	int	4	10	0	não	sim	sim	não	-	-
idCliente	int	4	10	0	não	não	não	sim	Cliente	-
dataHora	date	-	-	-	não	não	não	não	-	-
status	smallint	4	10	0	não	não	não	não	-	(1, 2, 3)
tempoTotal	int	4	10	0	não	não	não	não	-	>= 0
valorTotal	money	5	20	3	não	não	não	não	-	>=0

Servico_Agendado										
Coluna	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**UK**	**FK**	**Referência**	**Check**
idAgenda	int	4	10	0	não	sim	sim	sim	Agenda	-
idServico	int	4	10	0	não	sim	sim	sim	Servico	-
idProfissional	int	4	10	0	não	sim	sim	sim	Profissional	-
valor	money	5	20	3	não	não	não	não	-	>=0
status	smallint	4	10	0	não	não	não	não	-	(1, 2)

Catalogo										
Coluna	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**UK**	**FK**	**Referência**	**Check**
idProfissional	int	4	10	0	não	sim	sim	sim	Profissional	-
idServico	int	4	10	0	não	sim	sim	sim	Servico	-
valor	money	5	20	3	sim	não	não	não	-	>0
tempoMedio	int	4	10	0	sim	não	não	não	-	>0
status	smallint	4	10	0	não	não	não	não	-	(1, 2)

5.5 Script do Modelo Físico de Banco de Dados para SQL Server

```
CREATE DATABASE ProjetoInter

CREATE TABLE Cliente (
    id          INT             NOT NULL IDENTITY,
```

```
    nome        VARCHAR(50)     NOT NULL,
    telefone    VARCHAR(11)     NOT NULL,
    email       VARCHAR(30),
    senha       VARCHAR(20),
    status      SMALLINT        NOT NULL,

    PRIMARY KEY (id),
    UNIQUE(telefone, email),
    CHECK (LEN(senha) > 8),
    CHECK (status IN (1, 2))
)

CREATE TABLE Profissional (
    id              INT             NOT NULL IDENTITY,
    nome            VARCHAR(50)     NOT NULL,
    telefone        VARCHAR(11)     NOT NULL,
    email           VARCHAR(30),
    senha           VARCHAR(20),
    salarioFixo     SMALLMONEY      NOT NULL,
    comissao        DECIMAL(10,2)   NOT NULL,
    status          SMALLINT        NOT NULL,

    PRIMARY KEY (id),
    UNIQUE(telefone, email),
    CHECK (salarioFixo > 0),
    CHECK (comissao >= 0),
    CHECK (LEN(senha) > 8),
    CHECK (status IN (1, 2))
)

CREATE TABLE Servico (
    id          INT             NOT NULL IDENTITY,
    descricao   VARCHAR(50)     NOT NULL,
    valor       MONEY           NOT NULL,
    tempoMedio  INT             NOT NULL,
    status      SMALLINT        NOT NULL,

    PRIMARY KEY (id),
    UNIQUE(descricao),
    CHECK (valor > 0),
    CHECK (tempoMedio > 0),
    CHECK (status IN (1, 2))
)

CREATE TABLE Agenda (
    id          INT         NOT NULL IDENTITY,
    idCliente   INT         NOT NULL,
    dataHora    DATETIME2   NOT NULL,
    tempoTotal  INT         NOT NULL,
    valorTotal  MONEY       NOT NULL,
    status      SMALLINT    NOT NULL,
```

```
    PRIMARY KEY (id),
    FOREIGN KEY (idCliente) REFERENCES Cliente,
    CHECK (tempoTotal >= 0),
    CHECK (valorTotal >= 0),
    CHECK (status IN (1, 2, 3))
)

CREATE TABLE Servico_Agendado (
    idAgenda            INT             NOT NULL,
    idServico           INT             NOT NULL,
    idProfissional      INT             NOT NULL,
    valor               MONEY           NOT NULL,
    status              INT             NOT NULL,

    PRIMARY KEY (idAgenda, idServico, idProfissional),
    FOREIGN KEY (idAgenda) REFERENCES Agenda,
    FOREIGN KEY (idServico) REFERENCES Servico,
    FOREIGN KEY (idProfissional) REFERENCES Profissional,
    CHECK (valor >= 0),
    CHECK (status in (1, 2))
)

CREATE TABLE Catalogo (
    idProfissional      INT     NOT NULL,
    idServico           INT     NOT NULL,
    valor               MONEY,
    tempoMedio          INT,
    status              INT     NOT NULL,

    PRIMARY KEY(idProfissional, idServico),
    FOREIGN KEY (idServico) REFERENCES Servico,
    FOREIGN KEY (idProfissional) REFERENCES Profissional,
    CHECK (valor > 0),
    CHECK (tempoMedio > 0),
    CHECK (status IN (1, 2))
)

DROP TABLE Agenda;
DROP TABLE Cliente;
DROP TABLE Profissional;
DROP TABLE Servico;
DROP TABLE Servico_Agendado;
DROP TABLE Catalogo;
```

5.6 Script de Comandos SQL de Inserção de Dados no BD para SQL Server

```
insert into Cliente (nome,telefone,email,senha,status)values
('Carlos',17996710451,'Carlinho@gmail.com','23or2o3nd33d3',1),
('Santiago',17996789873,'Santi@gmail.com','34f3g1d3fc31c',1),
('Alvaro',17238402749,'Alvo@gmail.com','23r2323c31c3rv',1),
('Gabriel',17235739036,'Gabizinho@.hotmail.com','34523d313v13vv',1),
('Fernando',17992751489,'fernandinhoreis@gmail.com','23d23dv31vr1v',1)
```

```sql
insert into Profissional(nome,telefone,email,salarioFixo,comissao,senha,status)values
('Mario da silva',1723453223,'mario@hotmail.com',3000,0.05,'23d321d123r1v31',1),
('Cleber fernando',1791283182,'cleber@gmail.com',4000,0.07,'32d23d23dq13rvv',1),
('Daniel Xavier',1726127676,'Daniel@outlook.com',3500,0.09,'23ed22d3rv31v',1),
('Carmem queiroz',1734738282,'Carmem@hotmail.com',2500,0.04,'32d2d23v13v',1),
('Felipe anderson de
souza',1743524352,'FelipinZN@gmail.com',1500,0.09,'32d2d2d3vr314',1)

insert into Agenda(idCliente,dataHora,tempoTotal,valorTotal,status)values
(1,'2025-05-20 13:30:00.000',0.5,20,1),
(1,'2025-05-20 14:00:00.000',0.5,20,1),
(2,'2025-05-20 14:30:00.000',0.5,20,1),
(4,'2025-05-20 15:00:00.000',0.5,20,1),
(3,'2025-05-20 15:30:00.000',0.5,20,1)

insert into Servico(descricao,valor,tempoMedio,status)values
('Corte Cabelo',40,25,1),
('Corte Barba',20,30,1),
('Pintura Cabelo',50,20,1),
('Lavagem Cabelo',20,10,1),
('Escova Cabelo',40,60,1)

insert into Servico_Agendado(idAgenda,idServico,idProfissional,valor,status)values
(1,1,1,23,1),
(2,2,1,25,1),
(3,3,2,43,1),
(4,4,3,32,1),
(5,2,4,36,1)

insert into Catalogo(idProfissional,idServico,status)values
(1,1,1),
(2,2,1),
(3,3,1),
(4,4,1),
(5,5,1)


UPDATE Profissional
set email = 'felipin@gmail.com'
where id = 5;

UPDATE Cliente
set senha = 'senhanova'
where id = 3;

UPDATE Agenda
set status = 1
where id = 2;

UPDATE Catalogo
set status = 2
```

```
where idProfissional IN (1,2);

UPDATE Servico
set valor = valor*1.3
where id =3;


DELETE from Servico_Agendado where idAgenda = 1;
DELETE from Agenda where id = 1;
DELETE from Catalogo where idProfissional = 3;

DELETE FROM Servico_Agendado
DELETE FROM Agenda
DELETE FROM Catalogo
DELETE FROM Cliente
DELETE FROM Profissional
DELETE FROM Servico

select * from Cliente
select * from Profissional
select * from Agenda
select * from Servico
select * from Servico_Agendado
select * from Catalogo
```

5.7 Script de Comandos SQL de Seleção de Dados para BD para SQL Server

```
-- Listar Agendamentos
SELECT *
FROM Agenda

-- Serviço mais Caro
SELECT TOP 1 * FROM Servico
ORDER BY valor

-- Clientes atendidos por profissional
SELECT Profissional.id as idProfissional, Profissional.nome as Nome, COUNT(DISTINCT
Servico_Agendado.idAgenda) as 'Clientes Atendidos'
FROM Servico_Agendado, Profissional
WHERE Profissional.id = Servico_Agendado.idProfissional
GROUP BY Profissional.id, Profissional.nome

-- Cliente mais frequente.
SELECT TOP 1 Cliente.id as 'idCliente', Cliente.nome as 'Cliente',
COUNT(Agenda.idCliente) as 'Agendamentos'
FROM Cliente, Agenda
WHERE Cliente.id = Agenda.idCliente
GROUP BY Cliente.id, Cliente.nome
ORDER BY 'Agendamentos' DESC
```

```sql
-- Top 3 serviços mais solicitados
SELECT TOP 3 Servico.id as idServico, Servico.descricao as 'Serviçoo',
COUNT(Servico_Agendado.idServico) as 'Solicitações'
FROM Servico, Servico_Agendado
WHERE Servico.id = Servico_Agendado.idServico
GROUP BY Servico.id, Servico.descricao
ORDER BY 'Solicitações' DESC

-- Serviço mais solicitado por Profissional
SELECT Profissional.id, Profissional.nome, Servico.descricao,
COUNT(Servico_Agendado.idServico) as 'Solicitações'
FROM Servico, Servico_Agendado, Profissional
WHERE Servico.id = Servico_Agendado.idServico and Profissional.id =
Servico_Agendado.idProfissional
GROUP BY Profissional.id, Profissional.nome, Servico.descricao
ORDER BY 'Solicitações' DESC

-- Horário de pico
SELECT Agenda.dataHora, COUNT(*) Servico_Agendado
FROM Servico_Agendado, Agenda
WHERE Servico_Agendado.idAgenda = Agenda.id
GROUP BY Agenda.dataHora

-- Tempo médio das Agendas
SELECT SUM(tempoTotal)/COUNT(id) as TempoMedio
FROM Agenda

-- Profissional com a maior comissão
SELECT TOP 1 id, nome, comissao
FROM Profissional
ORDER BY comissao DESC

-- Número de serviços agendados por agenda
SELECT Agenda.id, Agenda.dataHora, COUNT(Servico_Agendado.idAgenda) as
NumeroAgendamentos
FROM Servico_Agendado, Agenda
GROUP BY Agenda.id, Agenda.dataHora


select * from Agenda
```

6. Etapa 4 – Documentação e relato das ações e funções desempenhadas por cada integrante da equipe executora do projeto

Objetivo: Aplicar técnicas de gerenciamento de equipes a partir de análises das habilidades técnicas e humanas dos integrantes.

6.1 Identificação de cada elemento do grupo e seu cargo ou função (perfil)

1-FERNANDO EUZÉBIO SOARES ===> Backend Developer e Banco de Dados
2-CARLOS DANIEL XAVIER ===> Backend Developer e Banco de Dados
3-GABRIEL PAGANI ===> Frontend Developer e Engenheiro de Software
4-ALVARO REIS ===> Frontend Developer e Engenheiro de Software
5-SANTIAGO ANDRES ===> Gestor de Equipe (Líder de Projeto)

6.2 Quais qualificações que cada um deveria ter para efetivamente desenvolver o cargo

Nome Qualificações necessárias

Fernando Euzébio Soares: Conhecimento avançado em Java, APIs RESTful, controle de versões (Git), modelagem de dados, SQL, noções de segurança de backend.

Carlos Daniel Xavier: Experiência com Java, Spring Boot, banco de dados relacionais (MySQL/PostgreSQL), versionamento de código, testes de integração.

Gabriel Pagani: Domínio de HTML, CSS, JavaScript, frameworks como React ou JavaFX (se app desktop), design responsivo, boas práticas de UI/UX.

Álvaro Reis: Conhecimento em arquitetura de software, versionamento com Git, integração entre frontend e backend, testes unitários.

Santiago Andres: Habilidades em liderança, gestão de projetos (SCRUM, Kanban), comunicação eficaz, tomada de decisão, ferramentas como Trello ou Jira.

6.3 Formas de gestão, recrutamento e seleção, descrição de cargos, avaliação de desempenho.etc

Gestão:

Utilização de metodologia ágil (SCRUM), com sprints semanais e reuniões de alinhamento.

Uso de ferramentas como Trello, Slack e GitHub para organização e comunicação.

Reuniões regulares de retrospectiva para ajustar processos.

Recrutamento e Seleção:

Análise de competências técnicas (testes de programação ou portfólio).

Entrevistas individuais avaliando soft skills (comunicação, trabalho em equipe).

Definição de perfis necessários com base nas áreas do projeto (frontend, backend, gestão, etc).

Descrição de cargos:

Backend Developer: Responsável por construir e manter a lógica do servidor, APIs e integração com banco de dados.

Frontend Developer: Responsável pela interface visual do app, garantindo boa experiência do usuário.

Engenheiro de Software: Responsável por decisões técnicas, arquitetura do app e integração entre as camadas.

Gestor de Equipe: Coordena a equipe, define prioridades, resolve conflitos e garante o cumprimento dos prazos.

Avaliação de desempenho: Entregas dentro dos prazos.

Qualidade do código (avaliada por code review).

Participação nas reuniões e comunicação com o grupo.

Colaboração e comprometimento.

6.4 Planejamento estratégico de gestão de pessoas

Objetivo: Garantir produtividade, boa comunicação, engajamento e desenvolvimento profissional dentro da equipe.

Ações estratégicas:

Definição clara de responsabilidades: Cada membro tem seu papel bem definido no projeto, evitando sobrecarga ou tarefas mal distribuídas.

Capacitação contínua: Incentivo ao aprendizado com tutoriais, cursos ou troca de conhecimento entre os membros.

Feedbacks frequentes: Reuniões de feedback a cada sprint para melhorias individuais e coletivas.

Motivação e reconhecimento: Destaque para boas práticas, entrega de funcionalidades e inovação.

Ambiente colaborativo: Incentivo à ajuda mútua e suporte entre as áreas técnicas e de gestão.

7. Conclusão

Para estabelecimentos que trabalham no formato de agendamento, e que ainda utilizam de métodos físicos para controle, este sistema representa um avanço significativo na gestão do negócio.

Entre os benefícios claros, destacam-se a otimização do tempo, tanto para clientes quanto para profissionais, através da facilidade de agendamento e consulta. A centralização das informações de clientes, profissionais e serviços em repositórios dedicados proporciona uma visão organizada e acessível, facilitando o controle e a tomada de decisões. A funcionalidade de grade de disponibilidade e verificação de conflitos de horário é crucial para evitar sobreposições e maximizar a utilização dos recursos humanos do salão. Além disso, a possibilidade de gerenciar o catálogo de serviços por profissional melhora a precisão na oferta de serviços e a atribuição de tarefas.

Os benefícios ao estabelecimento são substanciais. Para o salão, o sistema oferece um controle mais eficiente das operações, reduzindo erros manuais e retrabalho.