

## **INFORMAÇÕES SOBRE O PROJETO**

**Teste prático DSIN**

**Desenvolvedor:** Renan Henrique Medeiros F.

**Projeto:** Cabeleleila Leila Salão de Beleiza

## **SUMÁRIO**

<b>INFORMAÇÕES SOBRE O PROJETO</b>	<b>1</b>
<b>SUMÁRIO</b>	<b>2</b>
<b>Tecnologias utilizadas</b>	<b>3</b>
<b>Print das telas</b>	<b>4</b>
<b>Instruções de como rodar o projeto</b>	<b>7</b>
<b>Observações sobre o projeto</b>	<b>10</b>
<b>Scripts</b>	<b>14</b>

## **Tecnologias utilizadas**

Para o desenvolvimento desse projeto foram selecionadas as seguintes tecnologias: Python e o framework Flask para o backend, ReactJS para o frontend e banco de dados MySQL.

O ambiente de desenvolvimento utilizado foi o Visual Studio Code, o banco de dados foi instanciado localmente utilizando o software Xampp, a aplicação foi testada nos navegadores (Web Browser) Opera GX e Google Chrome e as API's foram testadas utilizando o software Insomnia.

# Print das telas

## 1. Tela inicial do sistema.

AGENDAMENTO DE HORÁRIOS DA CABELELEIRA LEILA

DATA E HORA

SERVIÇO

AGENDAR

VISUALIZAR HISTÓRICO

## 2. Seleção de data.

AGENDAMENTO DE HORÁRIOS DA CABELELEIRA LEILA

DATA E HORA

SERVIÇO

AGENDAR

July 2022

S

M

T

W

T

F

S

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

## 3. Seleção de hora.

AGENDAMENTO DE HORÁRIOS DA CABELELEIRA LEILA

DATA E HORA

SERVIÇO

AGENDAR

00

05

10

15

20

25

30

35

40

45

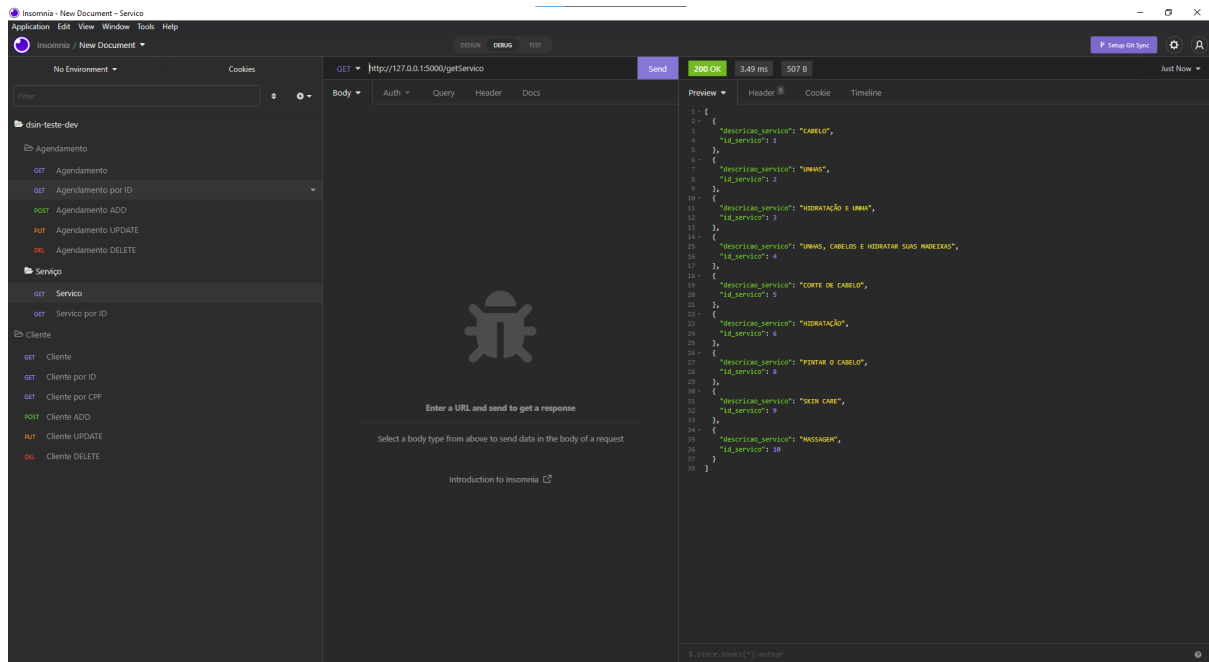
50

55

AM

PM

## 4. Listagem de serviços utilizando o Insomnia



## 5. Opções de serviços.

AGENDAMENTO DE HORÁRIOS DA CABELELEILA LEILA

Form for scheduling appointments at CABELELEILA LEILA. The form includes fields for NOME \*, CPF \*, and DATA E HORA. A dropdown menu for SERVIÇO is open, showing a list of services: CABELO, UNHAS, HIDRATAÇÃO E UNHA, UNHAS, CABELOS E HIDRATAR SUAS MADEIXAS, CORTE DE CABELO, HIDRATAÇÃO, PINTAR O CABELO, SKIN CARE, and MASSAGEM. Below the form are buttons for AGENDAR and VISUALIZAR HISTÓRICO.

## 6. Seleção de múltiplos serviços.

AGENDAMENTO DE HORÁRIOS DA CABELELEILA LEILA

Form for scheduling appointments at CABELELEILA LEILA. The form includes fields for NOME \*, CPF \*, and DATA E HORA. A dropdown menu for SERVIÇO is open, showing a list of services: CABELO, UNHAS, CABELOS E HIDRATAR SUAS MADEIXAS, and UNHAS. Below the form are buttons for AGENDAR and VISUALIZAR HISTÓRICO.

## 7. Seleção de serviço único.

AGENDAMENTO DE HORÁRIOS DA CABELELEILA LEILA

Form for scheduling appointments at CABELELEILA LEILA. The form includes fields for NOME \*, CPF \*, and DATA E HORA. A dropdown menu for SERVIÇO is open, showing a list of services: SKIN CARE. Below the form are buttons for AGENDAR and VISUALIZAR HISTÓRICO.

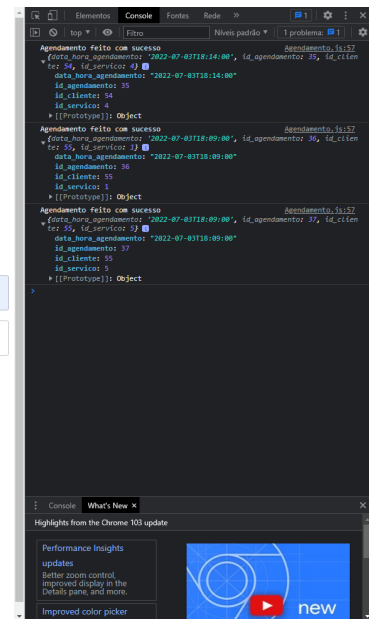
## 8. Confirmação de agendamento visualizada pelo console de desenvolvimento do navegador.

AGENDAMENTO DE HORÁRIOS DA CABELELEIRA LEILA

NOME:  CPF:

DATA E HORA:

SERVIÇO: UNHAS HIDRATAÇÃO



## 9. Mockup tela de Histórico de Agendamento.

HISTÓRICO AGENDAMENTOS X

NOME:  CPF:

NOME	CPF	DATA E HORA	SERVIÇO AGENDADO
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO

## 10. Comportamento inesperado de um componente ao abrir tela de Histórico de Agendamento.

AGENDAMENTO DE HORÁRIOS DA CABELELEIRA LEILA

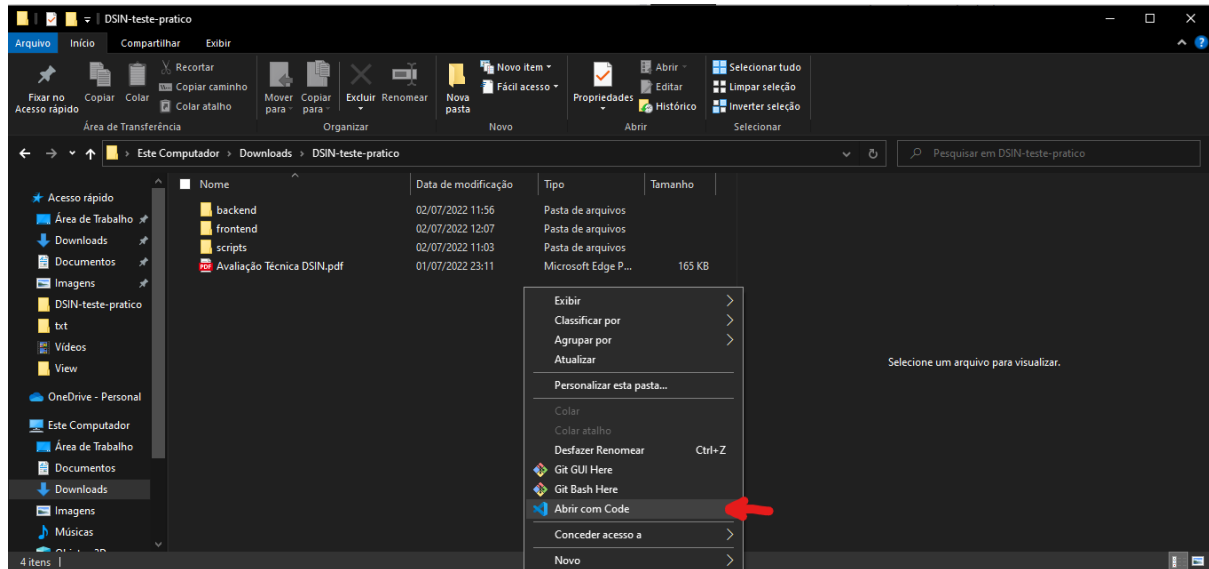
NOME:  CPF:

DATA E HORA:  SERVIÇO: UNHAS HIDRATAÇÃO

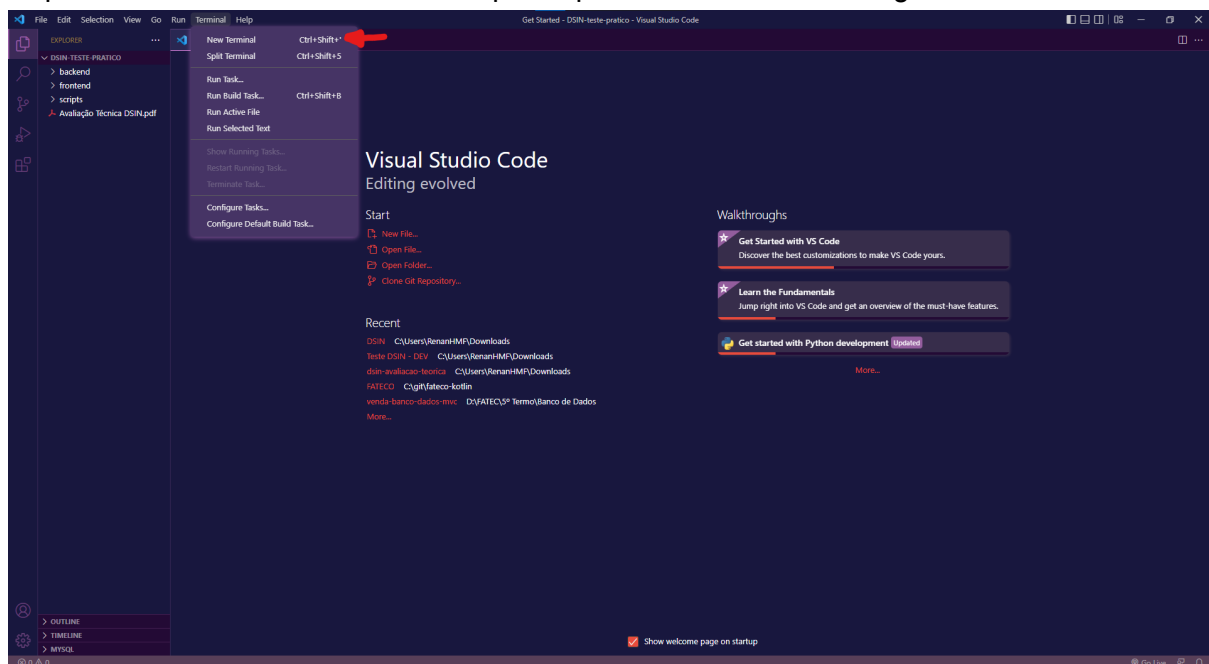
Histórico Agendamento

## Instruções de como rodar o projeto

Abra o projeto utilizando o Visual Studio Code, isso pode ser feito utilizando o Explorador de Arquivos do Windows. Abra o diretório no qual o projeto está armazenado, clique com o botão direito do mouse e selecione a opção “Abrir com Code”.



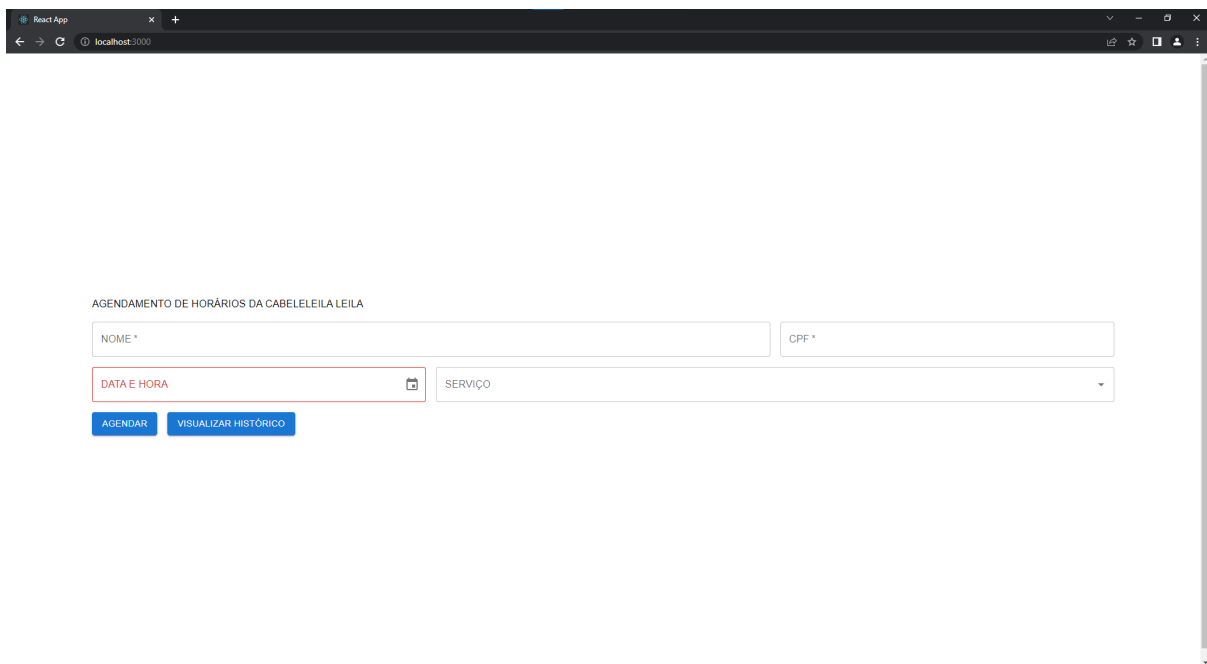
Com o VS Code aberto, utilize o comando `Ctrl + Shift + `` ou selecione Terminal e em sequência `New Terminal` no menu superior para abrir o terminal integrado do VS Code.



Para iniciar o frontend, utilize o comando **`cd frontend/cabeleleila-leila/`** para acessar o diretório no qual os arquivos do frontend estão armazenados. Em seguida, utilize o comando **`npm start`**. O terminal exibirá uma mensagem informando que o projeto foi iniciado.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Renan@MINGW64 ~/Downloads/DSIN-teste-pratico
$ cd frontend/cabeleleila-leila/
Renan@MINGW64 ~/Downloads/DSIN-teste-pratico/frontend/cabeleleila-leila (master)
$ npm start
> cabeleleila-leila@0.1.0 start
> react-scripts start
```

Após alguns instantes, uma aba será aberta no seu navegador padrão exibindo o frontend.



Ao verificar o terminal novamente, ele exibirá uma mensagem informando que o projeto foi compilado com sucesso e qual url pode ser utilizada para acessar o projeto no navegador.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Compiled successfully!

You can now view cabeleleila-leila in the browser.

  Local:    http://localhost:3000
  On Your Network: http://192.168.1.13:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Para iniciar o backend, abra um novo terminal, acesse o diretório backend com o comando **cd .\backend\**. Caso necessário, utilize o comando **py -3 -m venv .venv** para criar um ambiente isolado no qual serão instaladas as dependências necessárias para



execução do projeto. Acesse o *virtual enviroment* (venv) com o comando **.venv\scripts\activate**.

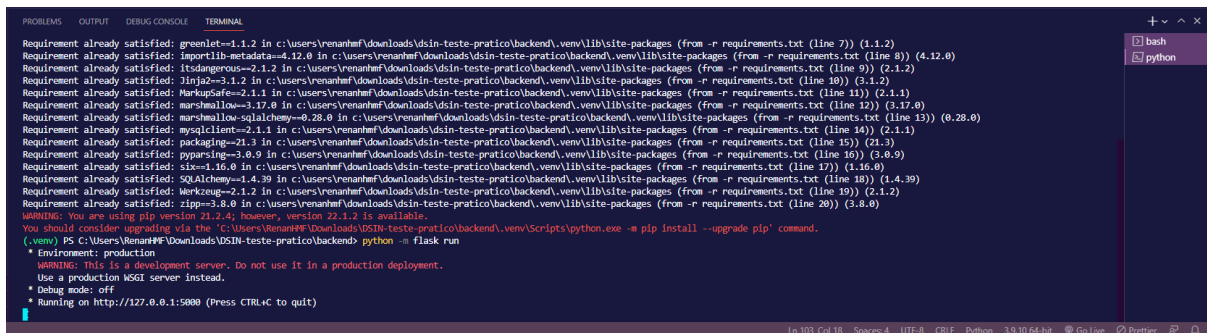


```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/powershell

PS C:\Users\RenanHF\Downloads\DSIN-teste-pratico> cd .\backend\
PS C:\Users\RenanHF\Downloads\DSIN-teste-pratico\backend> py -3 -m venv .venv
PS C:\Users\RenanHF\Downloads\DSIN-teste-pratico\backend> .venv\scripts\activate
(.venv) PS C:\Users\RenanHF\Downloads\DSIN-teste-pratico\backend>
```

As dependências do projeto podem ser instaladas por meio do comando ***pip install -r requirements.txt***. Em seguida, utilize o comando ***python -m flask run***. O terminal exibirá uma mensagem informando que o projeto foi iniciado.



```
Requirement already satisfied: greenlet==1.1.2 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 7)) (1.1.2)
Requirement already satisfied: importlib-metadata==4.12.0 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 8)) (4.12.0)
Requirement already satisfied: itsdangerous==2.1.2 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 9)) (2.1.2)
Requirement already satisfied: Jinja2==3.1.2 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 10)) (3.1.2)
Requirement already satisfied: MarkupSafe==2.1.1 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 11)) (2.1.1)
Requirement already satisfied: marshmallow==3.17.0 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 12)) (3.17.0)
Requirement already satisfied: marshmallow-sqlalchemy==0.28.0 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 13)) (0.28.0)
Requirement already satisfied: mysqlclient==2.1.1 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 14)) (2.1.1)
Requirement already satisfied: packaging==21.3 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 15)) (21.3)
Requirement already satisfied: pyParsing==3.0.9 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 16)) (3.0.9)
Requirement already satisfied: six==1.16.0 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 17)) (1.16.0)
Requirement already satisfied: SQLAlchemy==1.4.39 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 18)) (1.4.39)
Requirement already satisfied: Werkzeug==2.1.2 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 19)) (2.1.2)
Requirement already satisfied: zipp==3.8.0 in c:\users\renanhf\downloads\dsin-teste-pratico\backend\.venv\lib\site-packages (from -r requirements.txt (line 20)) (3.8.0)
WARNING: You are using pip version 21.2.4; however, version 22.1.2 is available.
You should consider upgrading via the 'C:\Users\RenanHF\Downloads\DSIN-teste-pratico\backend\.venv\Scripts\python.exe -m pip install --upgrade pip' command.
(.venv) PS C:\Users\RenanHF\Downloads\DSIN-teste-pratico\backend> python -m flask run

WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
```

Acessando a url informada pelo terminal, é exibida a mensagem que o backend foi iniciado.



## Observações sobre o projeto

Ao receber o projeto foram elencados os pontos essenciais para o desenvolvimento do sistema para Cabeleleila Leila Salão de Beleiza. Os pontos opcionais foram avaliados, porém, não foram priorizados. Dessa forma, realcei no texto do teste os principais pontos para que eles pudessem ser reescritos como critérios de aceite.

Ao consultar seus clientes sobre o que eles gostariam que existisse nesse programa, os mesmos responderam que seria a possibilidade de fazer o agendamento de um ou mais serviços e caso necessário realizar a alteração do mesmo (Permitindo a alteração pelo sistema até 2 dias antes do agendado, caso a data agendada seja menor que 2 dias, a alteração só poderá ser feita por telefone), e uma opção de histórico dos agendamentos realizados em determinado período, com a possibilidade de visualização dos detalhes desses agendamentos.

A Leila ainda gostaria que o sistema, ao identificar que existe um agendamento da mesma cliente para a mesma semana, sugira que os serviços sejam agendados na mesma data (data do primeiro agendamento).

Imagem retirada do texto do teste prático.

Sendo assim, utilizei a ferramenta Notion para criar uma *to-do list* com todos os pontos identificados no texto.

# Avaliação DSIN - Cabeleleila Leila Salão de Beleiza

- + :: ☐ Fazer agendamento de um ou mais serviços;
- ☐ Realizar alteração do serviço agendado;
- ☐ As alterações podem ser feitas somente até 2 dias antes do dia agendado;
- ☐ Alterações dentro de 2 dois dias podem ser feitas somente por telefone;
- ☐ Visualizar histórico de agendamentos realizados em determinado período;
- ☐ O histórico deve exibir detalhes dos agendamentos;
- ☐ Identificar agendamentos para uma mesma cliente no período de 7 dias (uma semana);
- ☐ Sugerir reagendar horário caso existam agendamentos para uma mesma cliente dentro de uma semana;
- ☐ Ao reagendar horário, sugira que os serviços sejam agendados na data do primeiro agendamento;

Imagem retirada do Notion.

Após fazer o levantamento de requisitos, utilizei a ferramenta draw.io para elaborar a arquitetura do projeto e o relacionamento entre as tabelas necessárias para o sistema.

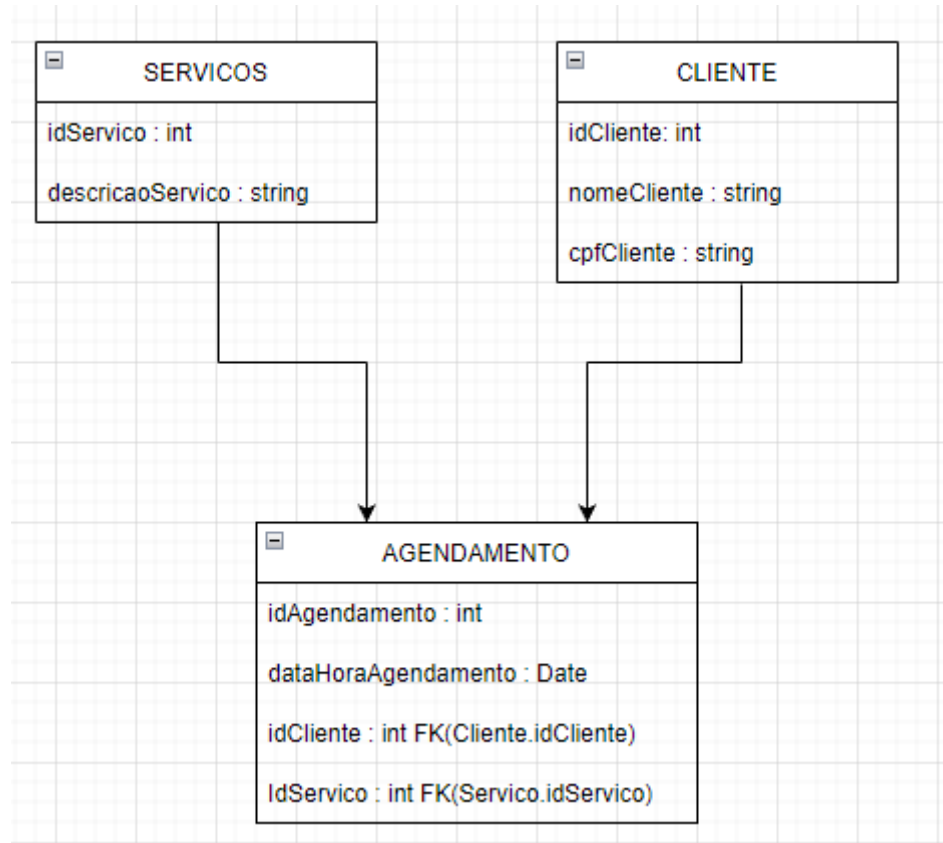


Imagem retirada do draw.io.

O projeto foi armazenado no diretório DSIN-TESTE-PRÁTICO, que por sua vez dividido em quatro diretórios, backend, frontend, prints e scripts. Destinados, respectivamente, para os arquivos do backend, para os arquivos do frontend, para os prints de telas e para os prints das API's e para os scripts sql de criação de banco de dados, criação de tabelas e inserção de registros.

Para o backend, foram criadas API's de listagem geral, retornando todos registros da tabela no formato JSON, API's de listagem específica utilizando determinados campos como identificador. API's de inserção, na qual é necessário informar um corpo de texto no formato JSON contendo todos os campos que devem ser registrados na tabela. API's de edição, na qual também é necessário informar um corpo de texto no formato JSON, mas somente com os campos que podem sofrer edição. Por fim, API's de exclusão, que utilizam um determinado identificador para removerem um registro da tabela. Todas API's retornavam um corpo JSON preenchidos com a informações referentes a elas em caso de execução com sucesso, em caso negativo, o corpo JSON é retornado vazio.

Para o frontend, foi pensado uma tela de agendamento com 4 campos (Nome, CPF, Data e Hora e Serviços) e 2 botões (Agendar e Visualizar histórico).

O Nome é destinado para o nome do cliente, o CPF para o número do documento homônimo do cliente, o campo Data e Hora para o horário do agendamento e o campo Serviços para os serviços que o cliente deseja agendar. Dado que não foi mapeada a necessidade de inserção, alteração e remoção dos registros armazenados na tabela de serviços, foram implementadas API's somente para listagem de serviços.

O botão Agendar é destinado para adicionar o registro do agendamento no banco de dados. Contudo, durante minha modelagem, deixei passar o mapeamento do cadastro de clientes. Por essa razão, ao acionar o botão Agendar o cadastro do cliente é realizado e em seguida o agendamento é feito. Embora realizar o cadastro do cliente antes do agendamento contorne o problema gerado pela falha no mapeamento, seria necessário implementar uma validação utilizando o CPF do cliente a fim de garantir que não existam registros duplicados para um mesmo cliente.

Para tal validação seria necessário a execução de uma consulta no banco de dados, caso essa consulta retornasse algum registro, o cadastro do cliente não seria necessário, permitindo que o agendamento fosse feito. Caso contrário, se a consulta não retornasse registros, o cadastro do cliente deveria ser feito para que então fosse possível realizar o agendamento. Infelizmente, devido problemas com a implementação da api para realizar essa consulta, não foi possível adicionar essa validação no projeto.

✓ A mostrar registros de 0 - 0 (1 total, A consulta demorou 0,0003 segundos.)

```
SELECT * FROM `cliente` WHERE `CPF_CLIENTE` like '12345678900';
```

☐ Perfil [ Editar em linha ] [ Editar ] [ Explicar SQL ] [ Criar código PHP ] [ Atualizar ]

☐ Mostrar tudo | Número de registros: 25 | Filtrar registros:

Opções extra

	ID_CLIENTE	NOME_CLIENTE	CPF_CLIENTE
<input type="checkbox"/> Editar <input type="text" value="Copiar"/> <input type="text" value="Apagar"/>	2	Shaq	12345678900

Exemplo de cliente já registrado na tabela de Clientes.

O botão Visualizar histórico seria destinado para abertura de um componente contendo os campos Nome, CPF e o botão Consultar, o qual ao ser acionado deveria retornar uma tabela contendo nome e CPF do cliente, data e hora do agendamento e descrição do agendamento. Em frente de cada registro estariam disponíveis os botões de edição e de exclusão de um determinado registro.

HISTÓRICO AGENDAMENTOS

X

NOME:

CPF:



CONSULTAR

NOME	CPF	DATA E HORA	SERVIÇO AGENDADO	<input type="text" value=""/>	<input type="text" value=""/>
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO	<input type="text" value=""/>	<input type="text" value=""/>
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO	<input type="text" value=""/>	<input type="text" value=""/>
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO	<input type="text" value=""/>	<input type="text" value=""/>
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO	<input type="text" value=""/>	<input type="text" value=""/>
NOME	CPF	DATA E HORA	SERVIÇO AGENDADO	<input type="text" value=""/>	<input type="text" value=""/>

Mockup desenvolvido para tela de Histórico de Agendamentos.

Entretanto, ao tentar implementar o componente Dialog, seu funcionamento não ocorreu como esperado e descrito em sua documentação, disponível em <https://www.primefaces.org/primereact/dialog/>. De modo coloquial, o componente deveria “agir como um pop-up” sendo exibido na frente da tela principal e permitindo que o usuário realizasse as ações de consulta, edição e remoção conforme o mockup. O resultado obtido está exibido abaixo, o componente comportou-se de modo anômalo, ficando na parte inferior da página, impossibilitando a visualização dos campos esperados na tela.

AGENDAMENTO DE HORÁRIOS DA CABELELEIRA LEILA

NOME *	CPF *
DATA E HORA 	SERVIÇO 
<button>AGENDAR</button>	<button>VISUALIZAR HISTÓRICO</button>

Histórico Agendamento

 CONSULTAR  VOLTAR

Comportamento do componente Dialog ao acionar o botão para visualizar o Histórico de Agendamento.

## Scripts

### 1. Criação de database para o projeto Cabeleireira Leila

```
CREATE DATABASE cabeleleilaleila;
```

### 2. Criação da tabela de Clientes.

```
CREATE TABLE IF NOT EXISTS CLIENTE (  
    ID_CLIENTE INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    NOME_CLIENTE VARCHAR(200) NOT NULL,  
    CPF_CLIENTE VARCHAR(11) NOT NULL  
);
```

### 3. Criação da tabela de Serviços.

```
CREATE TABLE IF NOT EXISTS SERVICO (  
    ID_SERVICO INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    DESCRICAO_SERVICO VARCHAR(100) NOT NULL  
);
```

### 4. Criação da tabela de Agendamentos.

```
CREATE TABLE IF NOT EXISTS AGENDAMENTO(  
    ID_AGENDAMENTO INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    DATA_HORA_AGENDAMENTO DATETIME NOT NULL,  
    ID_CLIENTE INT NOT NULL,  
    ID_SERVICO INT NOT NULL,  
    FOREIGN KEY (ID_CLIENTE) REFERENCES CLIENTE(ID_CLIENTE),  
    FOREIGN KEY (ID_SERVICO) REFERENCES SERVICO(ID_SERVICO)  
);
```

### 5. Inserção de registros na tabela de Serviços.

```
INSERT INTO SERVICO(ID_SERVICO, DESCRICAO_SERVICO)  
VALUES (NULL, 'CABELO'),  
       (NULL, 'UNHAS'),  
       (NULL, 'HIDRATAÇÃO E UNHA'),  
       (NULL, 'UNHAS, CABELOS E HIDRATAR SUAS MADEIXAS'),  
       (NULL, 'CORTE DE CABELO'),  
       (NULL, 'HIDRATAÇÃO'),  
       (NULL, 'PINTAR O CABELO'),  
       (NULL, 'SKIN CARE'),  
       (NULL, 'MASSAGEM');
```

### 6. Consulta para validação de cliente existente no banco de dados.

[não implementado]

```
SELECT * FROM CLIENTE WHERE CPF_CLIENTE like {cpf_cliente}
```

7. Consulta para listagem de agendamentos da tabela Histórico de Agendamentos.  
[não implementado]

```
SELECT
    CLIENTE.NOME_CLIENTE,
    CLIENTE.CPF_CLIENTE,
    AGENDAMENTO.DATA_HORA_AGENDAMENTO,
    SERVICO.DESCRICAO_SERVICO
FROM
    AGENDAMENTO
INNER JOIN CLIENTE ON
    AGENDAMENTO.ID_CLIENTE = CLIENTE.ID_CLIENTE
INNER JOIN SERVICO ON
    AGENDAMENTO.ID_SERVICO = SERVICO.ID_SERVICO
WHERE
    CLIENTE.CPF_CLIENTE LIKE '12345678900';
```