TQS: Especificação do produto

Conteúdo

[1 Introdução 1](#_Toc40959476)

[1.1 Contexto do projeto 1](#_Toc40959477)

[1.2 Limitações 2](#_Toc40959478)

[2 Conceito do produto 2](#_Toc40959479)

[2.1 Visão 2](#_Toc40959480)

[*2.2* *Personas* 2](#_Toc40959481)

[2.3 Cenários principais 4](#_Toc40959482)

[2.4 Prioridades e *epics* do projeto 4](#_Toc40959483)

[3 Modelo de domínio 4](#_Toc40959484)

[4 Arquitetura 4](#_Toc40959485)

[4.1 Requisitos principais e restrições 4](#_Toc40959486)

[4.2 Modelo da arquitetura 4](#_Toc40959487)

[4.3 Arquitetura de deployment 5](#_Toc40959488)

[4.4 Arquitetura do sistema 5](#_Toc40959489)

[5 API para desenvolvedores 5](#_Toc40959490)

[6 Referências e recursos 6](#_Toc40959491)

# Introdução

## Contexto do projeto

Cada vez mais, as tecnologias têm vindo a facilitar a nossa vida, dando-nos oportunidade de realizar certas tarefas muito mais rapidamente e com menos esforço. Para além disso, com o surgimento da pandemia do COVID-19, surgem também vários obstáculos e limitações, que têm começado a afetar o nosso quotidiano.

Neste contexto, decidimos criar uma plataforma de reserva de serviços de cabeleireiros e barbeiros em Portugal. O nosso sistema irá auxiliar tanto a vida dos gestores destes estabelecimentos, que terão a sua agenda mais facilmente organizada e estruturada, como também a vida dos seus clientes, que não terão de se deslocar ao local para efetuar a reserva, ou apenas para questionar-se acerca da disponibilidade, podendo fazer tudo isso à “distância de um clique”, bem como verificar as marcações efetuadas, em qualquer momento, e sem o receio de perder essa informação.

## Limitações

Não foi implementado um calendário e horário dinâmico e totalmente verdadeiro. Assim, é possível efetuar reservas em qualquer dia e escolher qualquer hora, mesmo havendo nesse mesmo dia e hora uma marcação já efetuada. Num ambiente real, isto seria algo muito irelevante, já que é dos pontos mais importantes de uma plataforma de reservas; no entanto, por falta de tempo não conseguimos implementar, já que o mesmo iria também dependeria de vários fatores como:

- Cada salão teria de escolher os dias da semana disponíveis para cada serviço, e esses dias teriam de ser mapeados para um calendário annual;

- Um salão poderia ter reservas para o mesmo serviço, hora e dia, já que poderia ter vários funcionários que pudessem atender;

Portanto, esta seria uma feature que teríamos de implementar futuramente.

Para além disso, não conseguimos implementar o sistema de busca e geolocalização da Google, também por falta de tempo.

# Conceito do produto

## Visão

|  |  |
| --- | --- |
| **Para:** | Pessoas interessadas em serviços de cabeleireiro ou barbeiro |
| **Que apresentam:** | Dificuldade em marcar uma hora para serem atendidas |
| **O produto:** | Cabeleireiro/barbearia |
| **Que:** | Permite de forma cómoda e simples ver horários disponíveis de um certo cabeleireiro na sua cidade e reservar uma hora de atendimento |
| **Ao contrário de:** | Marcação presencial ou por voz |
| **O nosso produto:** | Visa a facilitar as marcações nos cabeleireiros e barbeiros, sem a necessidade de se dirigir ao local, e tendo a hipótese de escolher mais flexivelmente os horários |

## *Personas*

|  |
| --- |
| A Laura Pereira tem 33 anos e é empresária, dona de um cabeleireiro em Lisboa. Mãe de 2 gémeos, é uma pessoa bastante atarefada, mas ao mesmo tempo muito organizada e perfeccionista. Atualmente, possui uma equipa de 8 pessoas a trabalhar para ela, sendo que é Laura que realiza todo o trabalho de gestão do cabeleireiro. Nos últimos meses, com o surgimento da pandemia do COVID-19, o seu cabeleireiro teve de fechar durante algum tempo; e ao abrir novamente, Laura deparou-se com um autêntico caos, já que o telefone do estabelecimento não pára de tocar, com pessoas a tentar realizar marcações, o livro de reservas encontra-se completamente cheio, e é impossível manter o cabeleireiro limpo e praticamente vazio, já que muitas pessoas lá se dirigem para pedir informações publicamente. Laura tem seguido a evolução das novas tecnologias, no entanto, por falta de tempo, a única informação presente, do seu cabeleireiro, na Internet, é o contacto telefónico.  **Motivação:** A Laura gostaria de ter uma forma simples de controlar a agenda do cabeleireiro, permitindo às pessoas saber, facilmente e rapidamente, os dias e horas disponíveis para marcações, sem necessidade de lá se dirigir ou telefonar. |

|  |
| --- |
| Cátia Moreira, tem 25 anos e é enfermeira. É uma pessoa bastante atenciosa e confiante. Devido à pandemia do COVID-19, esteve durante 3 meses a trabalhar arduamente, sem dar atenção ao seu visual e cuidados pessoais. Ao fim desse tempo, deparou-se com o facto de o seu cabelo ter crescido muito e não apresentar uma forma que lhe agrade, o que lhe retirou confiança e tornou-se, também, bastante incomodativo no seu local de trabalho. Cátia já se dirigiu a vários cabeleireiros, para tentar marcar um atendimento, no entanto, estes apenas possuíam disponibilidade para os próximos meses. Muitas amigas da Cátia, têm sofrido do mesmo problema, tendo de estar sempre a contactar vários estabelecimentos e a perder tempo a tentar arranjar uma alternativa.  **Motivação:** Cátia gostava de poder visualizar a disponibilidade dos estabelecimentos que lhe interessam, sem ter a necessidade de lá se dirigir ou telefonar, e decidir, sem pressão e com flexibilidade, a melhor opção para si. Para além disso, gostava de ter uma forma segura, de conseguir consultar a qualquer hora, as reservas que efetuou. |

## Cenários principais

* ***Laura quer adicionar o seu cabeleireiro***

Laura decide aderir à plataforma e registar lá o seu salão. Para isso, regista-se na plataforma e na secção de gestão, adiciona a informação do seu cabeleireiro, verificando que este foi publicado.

* ***Laura quer editar a informação do seu cabeleireiro***

Laura encontra-se a navegar no Web Site quando repara que o número de telefone do seu salão está incorreto. Por isso, dirige-se à secção de gestão e edita a informação do seu cabeleireiro, colocando o correto.

* ***Laura quer remover o seu cabeleireiro***

O negócio de Laura, depois do Covid, começa a trazer mais prejuízos do que lucro e, por isso, Laura decide fechá-lo por algum tempo, querendo removê-lo do Web Site. Por isso, dirige-se à secção de gestão e remove-o.

* ***Laura quer ver as reservas efetuadas para o seu salão***

Laura está em casa a descansar e decide ir verificar as reservas que foram efetuadas para o dia se amanhã. Para isso, entra no Web Site e na secção de reservas do seu salão, repara que existem bastantes marcações para amanhã, estando tudo cheio.

* ***Cátia quer efetuar uma reserva***

Cátia descobre que terá um dia livre amanhã, e, por isso, decide quer quer cortar, finalmente, o seu cabelo. Para verificar se o serviço está disponível, regista-se no Web Site; efetua login com as credenciais; pesquisa por salões que existem na sua cidade; entra na página do primeiro; escolhe o serviço de Corte, o dia de amanhã e verifica que existem horas disponíves. Escolhe também a hora, e efetua a reserva.

* ***Cátia quer confirmar a hora para o serviço que marcou***

Cátia acorda de manhã, e repara que se esqueceu a que horas reservou o serviço de Corte. Por isso, entra na plataforma, através do seu telemóvel, e navega até à secção de suas reservas. Lá, verifica que a hora para o qual marcou é de 15h.

## Prioridades e *epics* do projeto

***EPIC 1:***

* Construir a REST API e integrá-la com a base de dados;

***EPIC 2:***

* Construir a plataforma Web e conectá-la à REST API, permitindo a realização de tarefas de pesquisa, reserva e gestão.

***EPIC 3:***

* Construir a plataforma Mobile e integrá-la com a REST API, estando completamente funcional.

***EPIC 4:***

* Todas as funcionalidades deverão ser testadas.

Como maiores prioridades, definimos a construção do Web Site e sua integração com a REST API, e o desenvolvimento de testes.

# Modelo de domínio

<which information concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes>

# Arquitetura

## Requisitos principais e restrições

O presente projeto não será adaptado de nenhum projeto existente. Não é preciso grandes preocupações no que se refere a hardware e condições usuais, pois trata-se de uma plataforma Web simples.

O projeto será integrado com o sistema de busca e geolocalização da Google, a fim de mostrar aos usuários estabelecimentos próximos. Caso isso se torne muito complicado, e não haver tempo suficiente, a ideia do projeto é manter uma base de dados dos estabelecimentos, em Portugal, com os respectivos endereços.

O projeto terá uma plataforma Web e uma aplicação móvel para Android e iOS, construída no Flutter. A aplicação móvel, apenas poderá ser utilizada por clientes, que queiram reservas serviços e ver as reservas efetuadas. A plataforma Web será utilizada tanto por clientes, como por donos de salões, que queiram publicar e gerir os seus estabelecimentos, bem como ver as reservas que os clientes efetuaram.

Para navegar em qualquer plataforma, é necessário primeiro registar-se e autenticar-se. A parte de gestão é restrita somente aos donos dos salões.

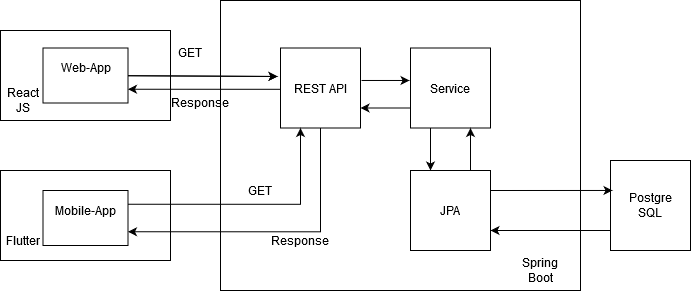
Requerimentos principais:

* Sistema de autenticação;
* Gerir salões;
* Procurar por salões, por nome ou cidade;
* Consular reservas efetuadas;
* Reserver/agendar um serviço;
* Cancelar uma reserva;

Requerimentos secundários:

* Procurar por estabelecimentos próximos;

## Modelo da arquitetura



## Arquitetura de deployment

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Modelar num diagrama de *deployment*]

## Arquitetura do sistema

<briefly present the software architecture. Include diagrams.>

<explain the supporting data models/data structures, i.e., the entities of your problem>

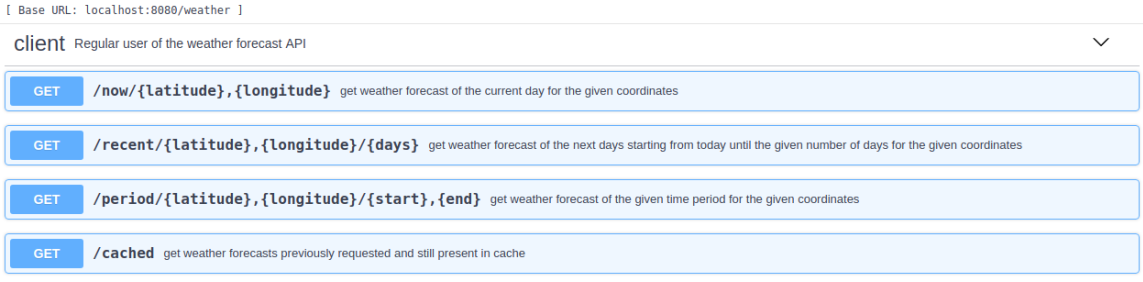
<detail the specific technologies/frameworks that were used>

# API para desenvolvedores

[Explicar a organização da API. Os detalhes detalhes/documentação dos métodos devem ficar numa solução *hosted* de documentação de APIs, como o [Swagger](https://swagger.io/), ou <https://apiary.io/> ]

<what services/resources can a developer obtain from your REST-API?>

<document the support endpoints>



# Referências e recursos

Sobre testes em Flutter:

[**https://flutter.dev/docs/testing**](https://flutter.dev/docs/testing)

Sobre integração Spring+React JS:

**https://www.springboottutorial.com/spring-boot-react-full-stack-crud-maven-application**

Sobre pedidos HTTP em Flutter à REST API:

**https://www.tutorialspoint.com/flutter/flutter\_accessing\_rest\_api.htm**