EscapeFilas

**Proposta Comercial**

* Descrição do problema

*Atualmente, num mundo no qual o tempo é algo muito precioso, as pessoas acabam perdendo um tempo desnecessários em filas pelo simples fato de desconhecer que aquele local teria uma fila grande, nosso aplicativo disponibiliza essa informação para o usuário para que ele possa tomar sua decisão de forma que mais lhe agrade, monitorando em tempo real a fila do local desejado, tudo isso graças a colaboração de todos os usuários do EscapeFilas.*

* Soluções do mercado para o problema

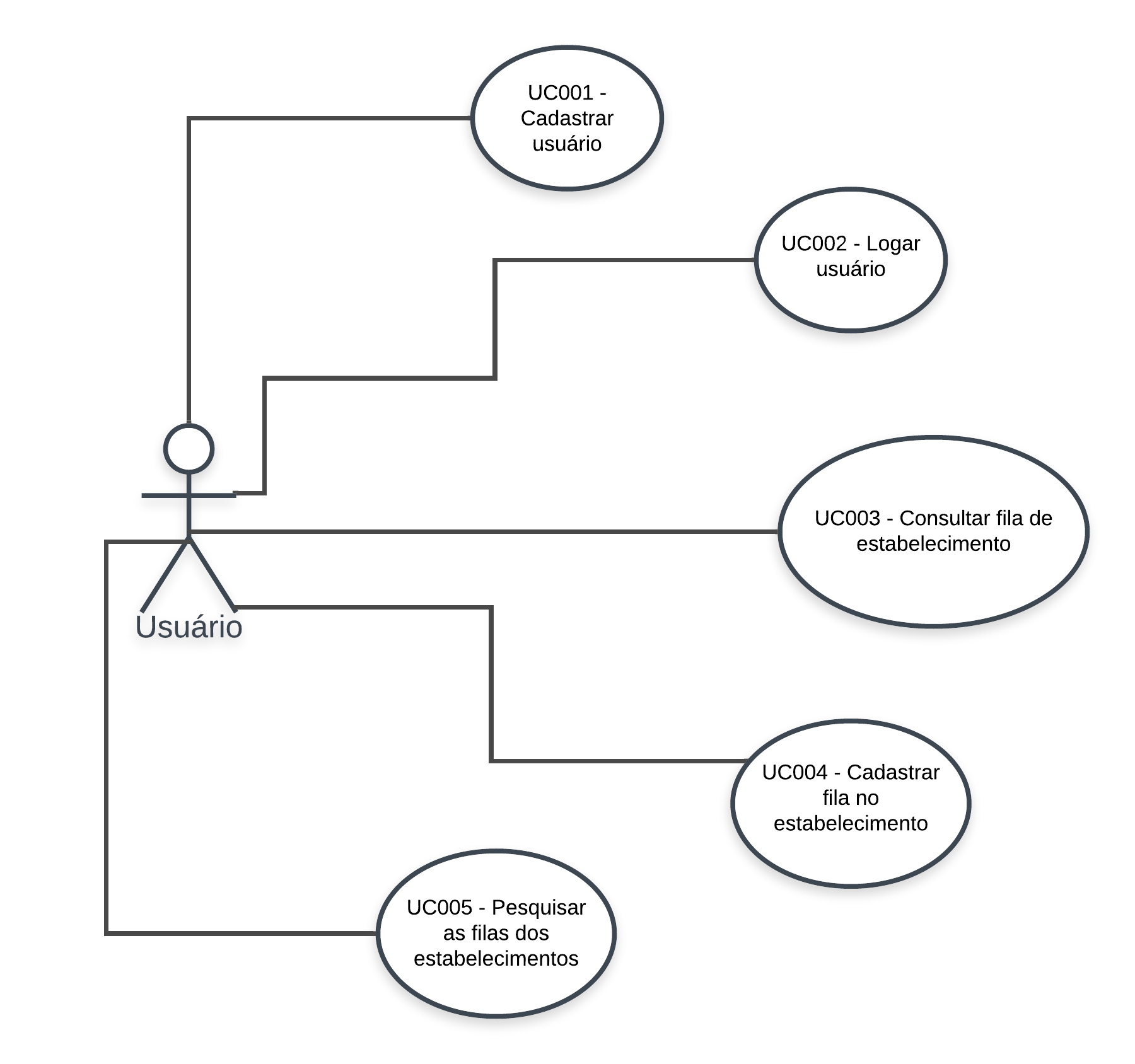
*Foi encontrado 1 trabalho correlato*

[*http://www.temfila.com.br/*](http://www.temfila.com.br/) *- o nosso diferencial é que terá mais opções de filtro/não vai ser web e sim um aplicativo, o que facilita muito a interação com o usuário por ser mais fácil de usar e de ter na sua mão.*

* Descrição da solução concebida

o EscapeFilas disponibiliza uma interface amigável nos dispositivos móveis que permite ao usuário informar e consultar estabelecimentos com filas.

* Caso de uso



* Indicativo da tecnologia

Para o desenvolvimento do sistema será utilizado a linguagem de programação Node.js, o express será a tecnologia utilizada para comunicação com o banco de dados, banco de dados GraphQL para armazenamento de dados, React Native para o desenvolvimento do aplicativo mobile.

* Indicativo para escolha da tecnologia

Escolhemos o Node.js pela flexibilidade e pela fácil curva de aprendizado, diferente do Java que requer um tempo considerável para que todos da equipe consigam aprender, também, Node.js tem acesso a biblioteca NPM, que contém milhares de conteúdo open source para nos ajudar tanto com a parte de login/abstração da logica de banco de dados, além de ser uma das linguagens mais utilizada para backend nos dias atuais.

Para o aplicativo móvel, decidimos usar React Native pois é também uma tecnologia que esta ganhando muita fama nos últimos meses pela ideia de que você programa em React Native, e a engine do React se preocupa em transformar o código para Android/iOS nativo.

Escolhemos GraphQL para armazenamento de dados por ser a alternativa mais rápida e porque estamos interessados em aprender mais sobre ela.

* Requisitos funcionais

**RF01 –** O aplicativo deve permitir o usuário efetuar um cadastro na aplicação;

**RF02 –** O aplicativo deve permitir o usuário efetuar login na aplicação;

**RF03 –** O aplicativo deve permitir ao usuário reportar filas no local que estiver;

**RF04 –** O aplicativo deve permitir ao usuário uma interface com o nível de filas dos locais próximos;

**RF05 –** O aplicativo deve permitir o usuário favoritar lugares favoritos;

**RF06 –** O aplicativo deve permitir o usuário filtrar por estabelecimentos;

**RNF01 –** O aplicativo utilizará linguagem para desenvolvimento (React-Native) compatível com SO Android e IOS;

**RNF02 –** O sistema será autenticado via usuário e senha ou conta Google;

**RNF03 –** Os dados dos usuários devem ser armazenados de forma criptografada;

**RNF04 –** A IDE para desenvolvimento do aplicativo será o VisualStudioCode;

**RNF05 –** As funcionalidades relacionadas à interação do usuário com mapas serão feitas através da API do Google Maps, disponibilizada pela empresa Google;

**RNF06 –** A interface do aplicativo deverá ser simples e intuitiva para fácil compreensão do usuário;

**RNF07 –** O sistema deve estar disponível 24 horas por dia, já que é um aplicativo mobile;

**RNF08 –** O sistema deve ser responsivo;

**RNF09 –** O sistema deve ser ágil;

**RNF10 –** O sistema deve apresentar uma interface amigável e intuitiva;

* Projeto das telas/interfaces do Sistema (front end/back end)

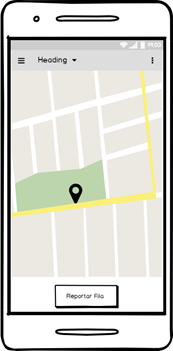
1. **Tela de Login**



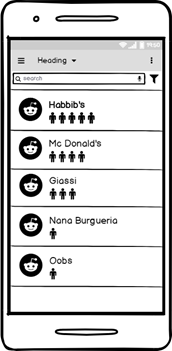
1. **Tela de cadastro**



1. **Reportar uma fila**



1. **Pesquisa de estabelecimentos por categoria**



1. **Modal de Filtros**



* Plano de testes

Pretendemos testar cada funcionalidade de forma especifica, sempre após a implementação e sempre uma pessoa diferente que fez a implementação deve testar a mesma, para que possamos pensar nos mais diversos casos possíveis.

* Cronograma para execução do projeto

**RF01** – (28/03) – O aplicativo deve permitir o usuário efetuar um cadastro na aplicação;

**RF02** – (04/04) – O aplicativo deve permitir o usuário efetuar login na aplicação;

**RF03** – (25/04) – O aplicativo deve permitir ao usuário reportar filas no local que estiver;

**RF04** – (16/05) – O aplicativo deve permitir ao usuário uma interface com o nível de filas dos locais próximos;

**RF05** – (23/05) – O aplicativo deve permitir o usuário favoritar lugares favoritos;

**RF06** – (13/06) – O aplicativo deve permitir o usuário filtrar por estabelecimentos;

* Perfil técnico da equipe construtora

Exemplo: A equipe será formada por 3 programadores, conforme detalhes abaixo:

Programador 1

* Formação: superior incompleto em Bacharelado em Sistemas de Informação
* Experiência: Trabalha na Senior Sistemas como desenvolvedor de sistemas.
* Conhecimento: Node, Java, Angular, Javascript

**Wallace Reetz**

****

**EscapeFilas**

Programador 1

* Formação: superior incompleto em Bacharelado em Sistemas de Informação
* Experiência: Trabalha na Senior Sistemas como desenvolvedor de sistemas.
* Conhecimento: Node, Java, Angular, Typescript, Javascript

**Gustavo Merini Seibt**

****

**EscapeFilas**

Programador 1

* Formação: superior incompleto em Bacharelado em Sistemas de Informação
* Experiência: Trabalha na Teclogica como desenvolvedor de sistemas.
* Conhecimento: Java, Javascript, React.js, Angular

**Marcelo Wippel**

****

**EscapeFilas**

* Custo
* **Versão gratuita:** A versão gratuita será disponibilizada na internet.