**Especificação de Software Mercado Fácil**

**Produzido por Alan Ricardo, Bruno Neves,**

**Felipe Ramos Vaz**

# Introdução

## Propósito

O projeto apresentado neste documento é chamado de Mercado Fácil, sua proposta tem como base reduzir o tempo de um cliente de supermercado, oferecendo a ele uma maneira simples de saber onde e quais supermercados oferecem o menor preço de seu produtos, tornado um trabalho longo em algo mais simples de ser planejado.

## Público alvo

O nosso público alvo está diretamente relacionado a consumidores, cliente de supermercados, que fazem compras grandes ou pequenas, pessoas idosas ou jovens, mas normalmente as pessoas que estão mais relacionados tem a idade acima dos 30 anos.

## Escopo do Projeto

A principal proposta é oferecer ao usuário uma interface acessível, onde ele possa identificar e criar uma lista de produtos para evitar longas viagens e ver qual supermercado oferece a ele o melhor preço nos determinados produtos, mas também lhe dá a opção de escolher um único produto, e também oferecer a ele o que tem a menor distância da posição atual do usuário. Sendo assim o mercado fácil proporciona ao cliente uma experiência de praticidade e comodidade aos usuário de supermercados.

Uma das propostas é oferecer ao usuário respostas em tempo real, já que a principal tarefa é tornar sua experiência com compras algo rápido, tornar os dados de pesquisa rápidos deve ser uma tarefa fundamental no projeto.

# Descrição Geral

## Perspectiva do produto

Ao longo dos anos o ser humano tem o âmbito de transformar as tarefas da vivência diária em algo mais simples e mais práticos tornando-o sua vida mais dinâmica. Sabendo disso, a ideia de criar um aplicativo onde facilita e agiliza as compras do dia a dia do consumidor torna-se algo relevante para a sociedade, sendo assim, o mercado fácil tem como objetivo aplicar o seus recurso e transformar e cobrir essa dor que existe na sociedade.

## Funções do produto

Nosso aplicativo deve ser capaz de oferecer aos usuario produtos de supermercados, identificando quais supermercados e os preços oferecidos por cada um deles. O usuário é capaz de criar uma lista de produtos onde ele pode ver qual supermercado oferece a ele o menor preço pelos produtos selecionados. Uma função que ele deve oferecer está relacionado a distância que o supermercados estão da sua localização atual, tornando-o assim mais viável uma ida para realização de suas compras.

Com a quantidade de compras que o usuário fará em nosso aplicativo, será feita uma coleta de dados para uma análise de quais produtos mais comprados por seus usuários, e qual a frequência, assim podemos oferecer ao cliente promoções em destaque e atualizações de produtos que ele já realizou uma compra, oferecendo a ele a informação de que o produto sofreu alteração no preço, tornado ainda mais efetivo a prática da compra e sua economia.

Para identificação de cada usuário a plataforma deverá ser capaz a de cadastrar cada um banco de dados, para obtermos informações individuais e oferecer a ele as devidas promoções, e sugestões baseadas em compras anteriores ou compras de outros usuário que possuem a mesma faixa etária ou outro tipo de relação.

A velocidade de entrega de informações ao usuário dever alta, oferecendo a ele a melhor experiência possível, já que o objetivo e ser pratico e rapido.

## Classe de Usuário

Definir a classe de usuário em nosso projeto não é uma tarefa simples, pelo fato de qualquer pessoa poder realizar uma compra, no entanto, como iremos obter informações do usuário pela sua faixa etária ao realizar o seu cadastro, podemos analisar quais são tipo de usuário que teremos, assim podendo saber exatamente como será trabalhado o seu perfil.

## Plataforma

A prioridade do aplicativo é ser operado em smartphones, já que o objetivo é ter o máximo de praticidade e frequência, o que pelos smartphones é muito maior que sites. Ele vai oferecer acesso às duas plataformas, IOS e Android.

Será oferecido a plataforma Web, porém ela será feita como menos prioridade.

## Sistema

Para facilitar o desenvolvimento da plataforma, vamos utilizar recursos que oferecem mais agilidade para os programadores. Usaremos Ubuntu como sistema operacional para melhor instalação de recurso e melhor contato como github, além de ser OpenSource.

Para a plataforma Web, será usado handlebars. Handlebars é uma framework em javascript, possui sintaxe similar com HTML. E para efeitos visual usaremos Materialize, que uma framework que oferece praticidade na hora de deixar o site visualmente melhor, ela é aplicada diretamente nas páginas de hbs (handlebars) o que facilita no desenvolvimento.

E por último será usado Node js para back-end com e react-native para a criação do app.

Para o banco de dados nos utilizaremos Mongodb, um banco não relacional, que é muito viável nesse projeto pelo fato de receber muitos dados simultâneos e armazenamentos de imagens.

## Problemas na implementação

Como sabemos, Implementar um sistema não é uma tarefa simples, esse projeto não foge disso. Um dos problemas que encontramos foi a ligação do banco de dados externo com o nosso, na recepção e no tratamento de dados, pois temos que armazenar cada dado recebido pelas empresas de mercado ( Caso elas aceitem a proposta), e provavelmente elas trabalham com tipo diferente de dados, ou seja, o manuseio dos dados é uma tarefa bem complicada. Uma forma de atacar esse problema e a criação do banco local, armazenado os dados e convertendo localmente, assim a cada modelo que chega podemos documentá lo, em caso de futuros dados sejam do mesmo jeito, já temos um modelo automatizado.

Outro problema encontrado foi a velocidade do app, pelo fato do número de informações existentes podem atrasar e deixar o site lento. Uma forma de atacar esse outro problema é usar algoritmos de ordenação para agilizar as buscas.

# Interface de Requerimento externo

## Interface do Usuário

O mercado fácil (app como web) deve possuir uma interface ao usuário bem explicativa e de simples entendimento visual, o que significa que o termo usabilidade será de extrema importância, pelo fato de não podermos padronizar a faixa etária dos nossos usuario.

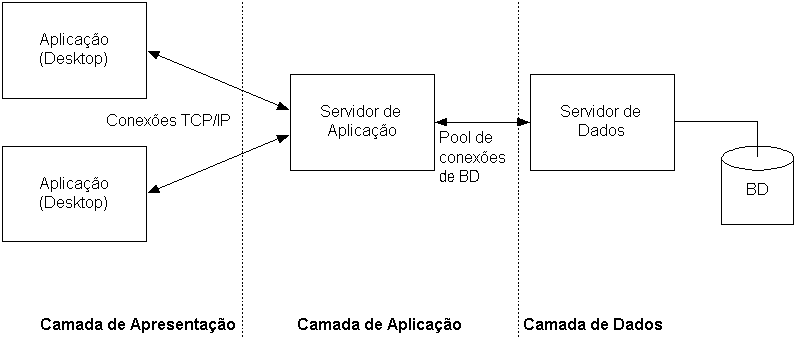
Para isso usaremos modelos básicos e bem comuns para qualquer pessoa que frequenta um site ou um app, um exemplo é o botão de cadastro no canto superior da tela, a barra de busca do itens explícita e com preenchimento automático.

Nas funções principais do site, como criar um lista de compras, vamos deixar a visibilidade das tarefas e dos itens selecionados centralizados na tela do usuário, tornando assim um meio prático de ver e utilizar a plataforma.

A plataforma vai oferecer ao usuário a distância entre a posição do usuário com o supermercado que ela escolheu para realizar sua compra, para isso vamos usar o modelo de mapa mais comum entre os sites, como vemos em apps como Uber, pop 99 e mais. Tornando assim mais agradável a manuseio do app.

## Comunicação Entre Interfaces

Como nosso aplicativo vai oferecer duas plataformas aos usuários haverá comunicação entre as duas, portanto deverá ser implementado um modelo de acesso aos dados de forma que não interfira no desempenho das duas. Um método usado para esse tipo de caso é a arquitetura em camadas. A arquitetura em camadas ou também chamada de arquitetura multicamadas é um sistema Cliente-Servidor onde as camadas de apresentação, processamento de aplicativos e gerenciamento de dados são separados.



Essa arquitetura nos fornece uma melhor controle entre as duas aplicações.

# Segurança

Como nosso aplicativo não oferece serviço de compras, nós não precisamos nos preocupar a segurança dos usuários e seus cartões. E como os dados enviados a nós pelas empresas de supermercados serão publicados para nossos clientes, a única cautela que devemos ter é em relação ao dados pessoas de nossos usuários, evitando a exposição deles.

# Desenvolvimento do Projeto

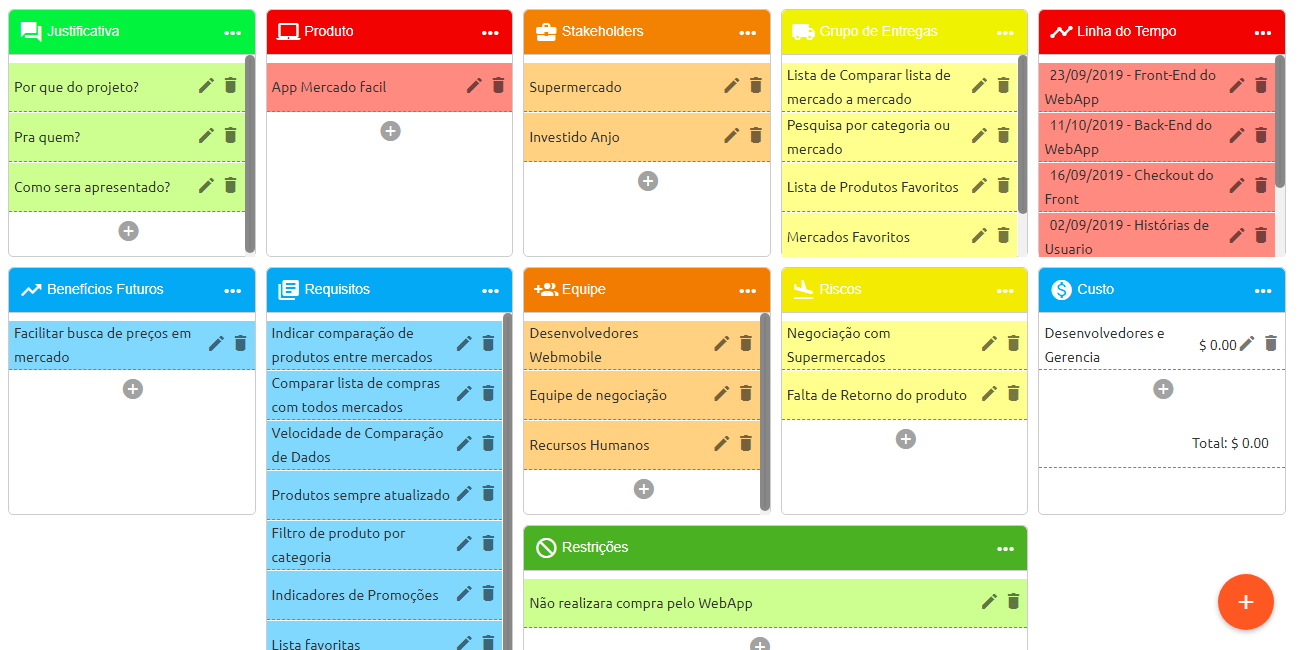
## Modelo de Desenvolvimento

O modelo de desenvolvimento talvez seja uma das partes mais importantes do projeto por estar relacionado diretamente como a plataforma será feita ao longo do tempo. Como nosso projeto é um projeto a longo prazo, nós vamos utilizar o modelo em espiral.

O modelo em espiral tem como objetivo prover um metamodelo que pode acomodar diversos processos específicos. Isto significa que podemos encaixar nele as principais características dos outros modelos de desenvolvimento de software, adaptando-os a necessidades específicas de desenvolvedores ou às particularidades do software a ser desenvolvido. Em um esforço para combinar as vantagens dos conceitos de [top-down](https://pt.wikipedia.org/wiki/Abordagem_top-down_e_bottom-up) e [bottom-up](https://pt.wikipedia.org/w/index.php?title=Bottom-up&action=edit&redlink=1), acrescentando um novo elemento, a análise de riscos que falta a essas conservas, sendo assim, torna-se viável utilizar ele na aplicação presente.

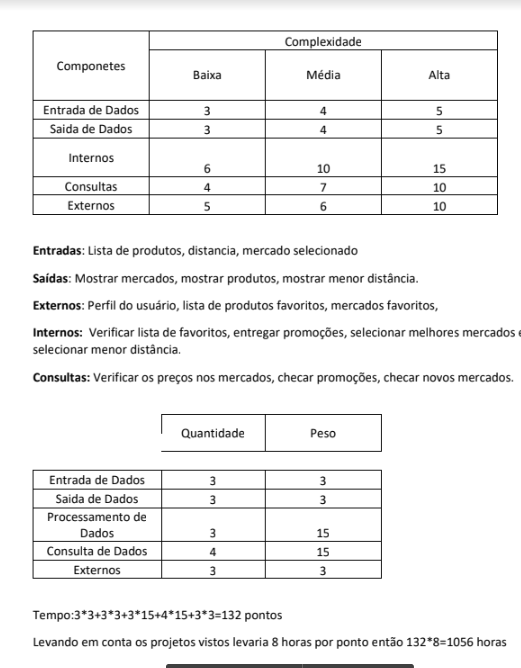
## Canvas

Com a necessidade de desenvolvimento contínuo do projeto, foi criado um canvas determinando os fatores relevantes e pontos específicos de criação de cada elemento que contém no projeto, sendo assim apresentamos o seguinte canvas:



## Estimativa

Como em todo projeto, é necessário atender uma demanda determinada por quem financia o projeto, sendo assim, foi realizado uma estimativa baseada nos determinadas requisições apontada na imagem a seguir:



## Manutenção

Manter a plataforma em condições estáveis não é uma tarefa simples, pela fato dela estar dívida como mostramos acima na definição da arquitetura, ou seja, deve ter um responsável por cada parte do projeto, um para cuidar dos dados recebidos pelos fornecedores, ou seja, alguém que entenda bem de mongodb, outro para os controles da api, e alguém para cuidar do contato direto com interface dos usuários.