Universidade de São Paulo
Instituto de Matemática e Estatística
Bachalerado em Ciência da Computação

Caio Lopes Demario
Erika Midori Akabane
Karyn Silva de Azevedo

# Desenvolvimento de um aplicativo Android baseado no estudo do comportamento de filas e com aplicação de crowdsourcing

São Paulo
2016

# Desenvolvimento de um aplicativo Android baseado no estudo do comportamento de filas e com aplicação de crowdsourcing

Monografia final da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof[a]. Dra. Ana Cristina Vieira de Melo

São Paulo
2016

# Agradecimentos

Gostaríamos de agradecer primeiramente a Deus, pela finalização de mais uma etapa tão almejada de nossas vidas. Também gostaríamos de agradecer a pessoas muito importantes no desenvolvimento deste projeto: Kathleen e Shodji pelo aconselhamento no design do produto, Paulo pelas opiniões sobre o cálculo do tempo de espera e nossos amigos da graduação que muito nos apoiaram, tanto nas matérias durante o curso como nessa reta final. Não podemos nos esquecer da nossa orientadora, Ana Cristina, que apoiou e acreditou na nossa ideia de projeto desde nossa primeira conversa. E também o professor Magalhães que se disponibilizou para sanarmos nossas dúvidas sobre essa matéria tão complexa chamada Estatística. Finalmente, obrigado pais e mães pelo apoio e acolhimento nos momentos de estudo e desenvolvimento. À todos nosso muito obrigado, nada disso seria possível sem vocês.

# Resumo

Este projeto possui como objetivo principal estudar e aplicar o processo de desenvolvimento de software de um aplicativo para a plataforma *Android* que fornece ao usuário a previsão de tempo de espera para restaurantes cadastrados no banco de dados.

Para auxiliar no desenvolvimento foram também estudadas modelagens para previsão de filas de restaurantes, para que o cálculo de previsão da espera fosse feito de forma mais precisa. O cálculo se utiliza dos seguintes dados: aqueles obtidos através de pesquisas nos próprios restaurantes e os extraídos de informações enviadas por usuários. De posse desses dados, o tempo de espera fornecido para aquele instante será cada vez mais preciso à medida que aumentam os *feedbacks* dos usuários.

Para o desenvolvimento deste aplicativo, algumas técnicas de gerenciamento de projeto foram usadas com o intuito de fornecer um produto final satisfatório para o usuário.

**Palavras-chave:** filas, restaurantes, desenvolvimento *Android*, previsão de espera

# Abstract

Our goal with this project is to study and apply the process of software development of an Android application that provides the user with the waiting time prediction for restaurants registered in our database.

To assist the development, some modeling methods to estimate restaurant queues were studied as well, so that the calculation were accomplished with more precision.

The waiting time prediction calculation is done using the following data: data obtained through research on the restaurants and data extracted from information sent by users. Thus, the waiting time provided for that moment will be increasingly more accurate as the feedback from the users gets more frequent.

In order to develop this application, some project management techniques were used with the interest of providing a satisfactory final product for the user.

**Keywords:** queues, restaurants, Android development, waiting time estimation.

# Sumário


**1 Introdução** **1**

1.1 Motivação . . . 1

1.2 Trabalho Proposto . . . 1

1.3 Estrutura da monografia . . . 2

**2 Estudos Preliminares** **3**

2.1 Desenvolvimento de Startups . . . 3

2.1.1 Desenvolvimento de Questionários . . . 4

2.2 Metodologia Ágil . . . 5

2.3 Jobs to be done . . . 6

2.4 Cálculo da previsão do tempo de espera na fila . . . 8

2.4.1 Descrição do problema . . . 8

2.4.2 Métodos de cálculo da espera . . . 9

**3 Tecnologias utilizadas** **15**

3.1 O sistema Android . . . 15

3.1.1 Organização de uma aplicação Android . . . 16

3.2 JSON . . . 17

3.3 Google APIs . . . 17

3.4 Localização do Usuário . . . 18

3.5 Controle de Versão . . . 19

3.6 Banco de dados . . . 19

3.6.1 Firebase . . . 20

3.7 Crowdsourcing . . . 21

**4 Desenvolvimento do Projeto** **23**

4.1 Pesquisa de Mercado . . . 23

4.1.1 Pesquisa com o usuário . . . 23

4.2 Projeto do Aplicativo . . . 32

4.3 Aplicação dos métodos de cálculo de espera . . . 32

4.3.1 Aplicando a média por período . . . 32

4.3.2 Aplicando teoria das filas . . . 34

4.3.3 Aplicando redes neurais . . . . . 34
4.3.4 Conclusão sobre os métodos estudados . . . . . 35
4.4 Definição da Interface do aplicativo . . . . . 35
4.5 Explicação das funcionalidades . . . . . 41
4.6 Implementação do banco de dados . . . . . 41

**5 Conclusão** **43**

**A Dados coletados** **45**

**Referências Bibliográficas** **47**

# Capítulo 1

# Introdução

Este capítulo introduz o problema abordado, a solução proposta, os objetivos do projeto e apresenta a estrutura da monografia.

## 1.1 Motivação

Enfrentamos filas para realizar diversas atividades ao longo do dia. Nosso tempo é gasto esperando desde alguns minutos para pedir o almoço até horas para assistir um determinado show. Ao todo, muitas horas de nossas vidas são desperdiçadas nessas esperas.

Muitas vezes somos surpreendidos por filas intermináveis ou então pela ausência delas, criando a impressão de que estas são totalmente imprevisíveis. Isso acaba gerando a dificuldade de programar, com precisão, outras atividades, pois não sabemos ao certo quanto tempo iremos gastar em esperas ao longo do dia. Deixamos muitos deveres ou até mesmo momentos de lazer de lado por causa disso e, se tivéssemos como prever esse tempo gasto, poderíamos otimizar nosso dia.

Esse e outros problemas podem ser minimizados à medida que o acesso à informação, seja histórica ou em tempo real, é facilitado. Uma forma de fornecer essa informação ao usuário é através da tecnologia. Um exemplo disso é o aplicativo de navegação *Waze* [24] que disponibiliza dados do trânsito em tempo real ajudando na locomoção diária. Isso é possível graças aos avanços tecnológicos e a maior acessibilidade aos dispositivos móveis conectados à internet, por exemplo.

## 1.2 Trabalho Proposto

Ao vislumbrarmos esse problema, tanto na vida dos outros como nas nossas, e motivados a utilizar nossos conhecimentos adquiridos ao longo do curso de Ciência da Computação, decidimos desenvolver um aplicativo para diminuir esse desconforto causado pelas filas. Essa solução é um sistema:

- desenvolvido na plataforma *Android*
- baseado em uma modelagem de sistema de filas

que tem como objetivo prever o tempo de espera dos usuários em filas e assim auxiliá-los nas suas decisões.

Para desenvolver uma solução para o problema e dada a limitação de tempo para o desenvolvimento, decidimos implementar a solução apenas para as filas de restaurantes. Esta escolha foi feita por serem locais muito frequentados e assim gerarem mais dados em menos tempo, e pela diversidade de tamanho das esperas gerando o interesse no estudo de sua modelagem.

Implementou-se um aplicativo no qual o usuário pode fazer uma consulta sobre os restaurantes cadastrados no banco de dados e ter acesso ao tempo de deslocamento somado ao tempo de espera até seu atendimento no estabelecimento. Além disso, ele tem a possibilidade de traçar uma rota até o local e enviar um *feedback* sobre quanto tempo foi gasto na fila para aprimorar o calculo do tempo de espera.

## 1.3 Estrutura da monografia

Para que fosse possível o desenvolvimento de um aplicativo para previsão de tempo de espera foi necessário estudar o processo de criação de *startups*, um processo de desenvolvimento de software, tecnologias para programação em *Android*, bem como algumas modelagens estatísticas de previsão de tempo de espera. A monografia foi estruturada de acordo com essas informações de modo que o leitor possa ter uma leitura progressiva, dividida em três principais partes:

- **Estudos preliminares:** Essa seção apresenta estudos sobre o que é necessário para o desenvolvimento de *Startups*, a definição da metodologia de desenvolvimento utilizada pelo grupo e o estudo da aplicação de três diferentes modelagens estatísticas feitas para analisar os dados iniciais, fornecendo um modo de calcular o tempo de espera em um restaurante em um dado instante.
- **Tecnologias utilizadas:** Onde são apresentadas as tecnologias e ferramentas utilizadas para auxiliar no desenvolvimento do aplicativo.
- **Desenvolvimento do projeto:** Essa parte apresenta a aplicação dos métodos de cálculo de espera vistos na seção anterior e o processo de desenvolvimento do aplicativo em si.

# Capítulo 2

# Estudos Preliminares

## 2.1 Desenvolvimento de Startups

O desenvolvimento do projeto foi baseado nos métodos utilizados em *Startups*. O grupo seguiu os princípios da *Startup Enxuta* [31], no qual um produto mínimo deve ser criado e então incrementado através do estudo das necessidades do usuário. A partir de uma pesquisa com o público alvo, a ideia inicial é lapidada e um produto com o mínimo de funcionalidades possível é desenvolvido para então retornar aos usuários que irão validá-lo. Este processo é repetido até que um produto final seja completamente desenvolvido, independente da ideia que se tinha no início. O processo tem como foco economizar recursos e suprir a necessidade do usuário de forma rápida e eficiente.

De acordo com o livro "A Startup Enxuta"[31], esta metodologia prioriza o desenvolvimento iterativo com a participação do cliente a cada etapa, ao invés de procurá-lo somente na entrega do produto final. Nessa metodologia temos alguns princípios que guiam o desenvolvimento de projetos:

1. Mapeamento do negócio através do diagrama conhecido como "*canvas* do modelo de negócios" que nos mostra resumidamente uma visão geral do projeto.
2. Pesquisas feitas com os usuários no mercado para saber suas opiniões sobre o negócio e posteriores mudanças ou ajustes no projeto com base nos resultados.
3. Desenvolvimento do produto utilizando algum método ágil.

Inicialmente, deve-se traçar os primeiros objetivos do projeto utilizando o diagrama *Canvas*[10] (figura 2.1). Esse diagrama é utilizado para auxiliar empresas a ter uma visão macro do projeto, fornecendo uma forma de visualizar suas dependências e permitindo planejar, traçar e gerenciar um plano de desenvolvimento. Ele possui perguntas que devem ser respondidas em seus quadros e, no decorrer desse processo, o projeto vai sendo adaptado de acordo com as limitações, impedimentos e novas ideias que vão surgindo.

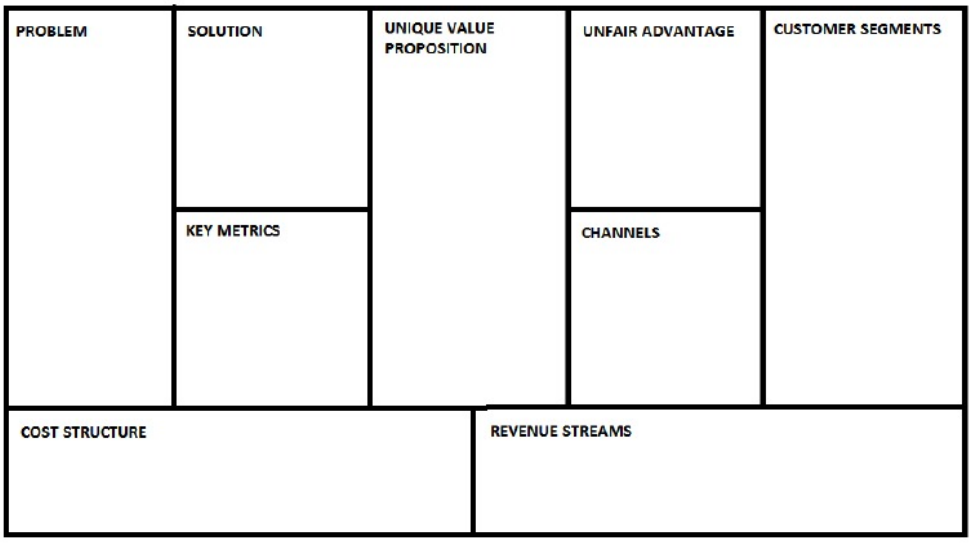



**Figura 2.1:** *Modelo do Diagrama Canvas*

Com o modelo de negócios definido, o próximo passo é a realização de uma pesquisa com o público para validá-lo. Essa fase do desenvolvimento pode ser refeita quantas vezes forem necessárias, reescrevendo o *Canvas* conforme a interpretação da necessidade do usuário.

### 2.1.1 Desenvolvimento de Questionários

De acordo com o manual *Designing a Survey* [4], um questionário é criado para obter informações para conduzir o desenvolvimento de um projeto de acordo com as preferências do público alvo. Ele deve ser curto e conciso e deve capturar somente informações que sejam relevantes para a pesquisa.

Os passos para o desenvolvimento de um bom questionário são:

- Definir os dados necessários.
- Desenvolver as questões.
- Remover qualquer questão que não captura os dados definidos previamente.

As questões podem ser moldadas de três maneiras:

- Estruturadas: Questões que possuem resposta fixa. Exemplo:

  *Você usa transporte público?*
  ( )*Sim*
  ( )*Não*

- Não Estruturadas: Questões de resposta aberta. Exemplo:

  *O que você costuma fazer para evitar filas?*

- Parcialmente Estruturadas: Questões mistas. Exemplo:

  *Qual meio de transporte você usa?*
  ( )*Carro*
  ( )*Moto*
  ( )*Outros:*_____

## 2.2 Metodologia Ágil

A metodologia ágil tem sido adotada pelas empresas hoje em dia por conta dos melhores resultados obtidos em relação às metodologias tradicionais[23]. Essa é caracterizada pela valorização de:

- "Indivíduos e interações mais que processos e ferramentas"
- "Software em funcionamento mais que documentação abrangente"
- "Colaboração com o cliente mais que negociação de contratos"
- "Responder a mudanças mais que seguir um plano" [12]

É uma metodologia que envolve trabalho em equipe, encontros regulares com o cliente, desenvolvimento iterativo e com menos foco em planejamento estrito e ideias definitivas, diferente de metodologias mais tradicionais que são mais focadas em projetos bem documentados com pouca possibilidade de mudança ao final do processo. Decidiu-se por utilizar a metodologia ágil neste projeto pois não haviam ideias completamente definidas inicialmente e, para o grupo, a opinião do público alvo sobre o produto era fundamental em cada iteração do processo, podendo haver mudanças inesperadas dependendo do *feedback* recebido.

Dentre as opções de metodologias ágeis, O método *Scrum*[16] é um que é definido como um gerenciador e um processo de controle que foca em construir um software que conhece as necessidades do negócio, devolvendo incrementalmente um software.

Com o *Scrum*, o software é desenvolvido em iterações de tamanho fixo chamadas de *Sprints*, que tem duração de tipicamente duas a quatro semanas. Em cada *Sprint* um número fixo de histórias (que se traduzem em tarefas), são selecionadas pelo grupo para serem utilizadas na *sprint*(figura 2.2). Há também um sistema de motivação para o time, como os *Milestones*, que são objetivos menores distribuídos por toda a *sprint* ajudando na medição de desemprenho da equipe, e gráficos *Burndown*, que mostram o progresso de cada iteração, para que o time sinta o progresso no desenvolvimento.

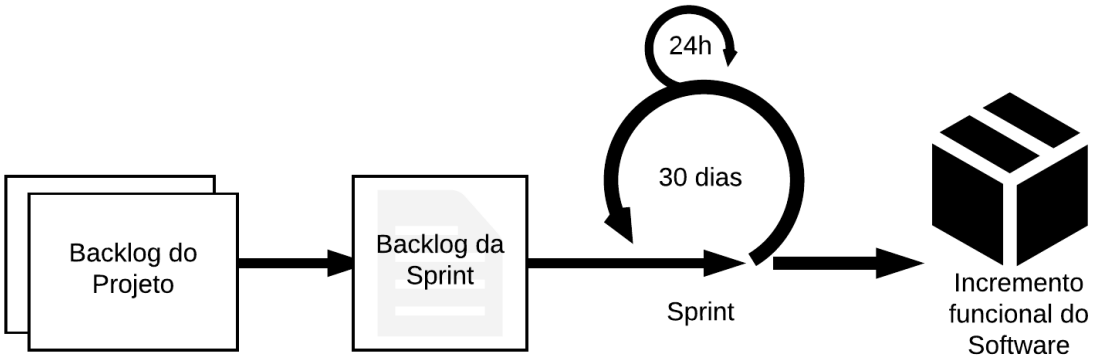



**Figura 2.2:** *Dinâmica do* Scrum

Para controle das iterações, foi utilizado o software *Trello* [21], que é um gerenciador de tarefas de projeto (figura 2.3).

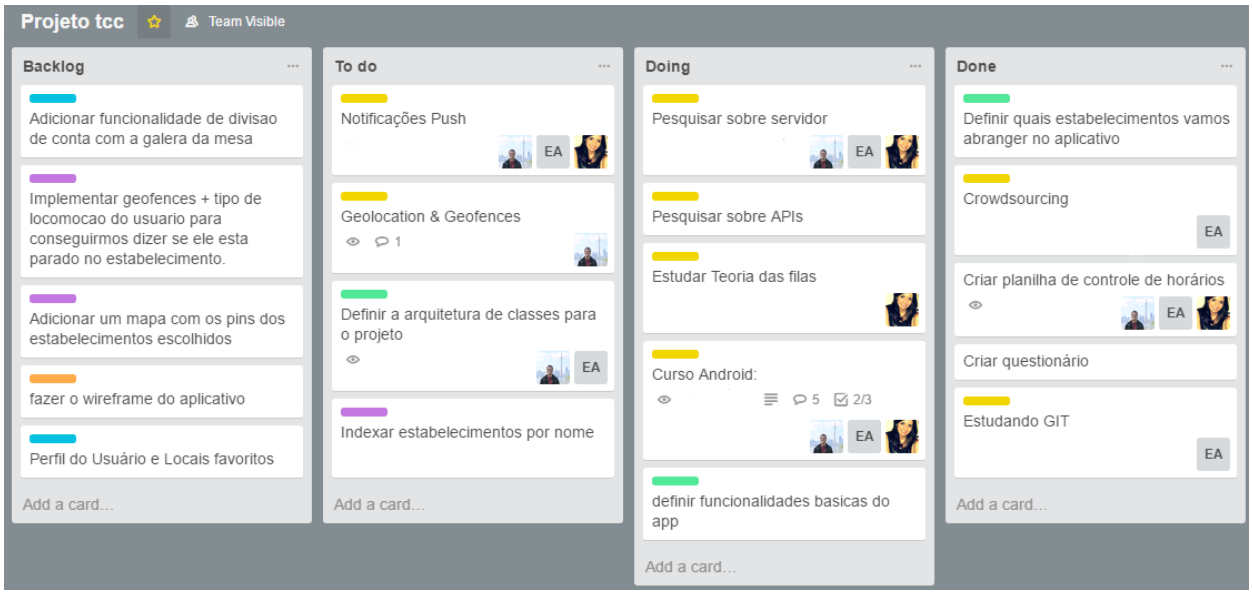



**Figura 2.3:** *Interface do Trello - durante o processo de desenvolvimento*

## 2.3 Jobs to be done

Para implementar o aplicativo, buscou-se uma metodologia para guiar esse processo desde sua criação até sua validação. Duas metodologias conhecidas pelo grupo foram analisadas e estudadas: *Jobs To Be Done* e *Personas and Use Cases*.

Segundo Alan Klement[29], *Personas* são clientes imaginários descritos pela idade, sexo, raça e hábitos semanais. O uso delas não é adequado para responder perguntas como "Porque o cliente quer saber o tamanho da fila?" pois, quando a resposta é "Ele está com pressa para comer" não fica claro quais são as reais necessidades do usuário. Essa descrição também não é útil no desenvolvimento de uma boa interface pois não ajuda na hora de responder se um certo botão realmente deixa claro sua funcionalidade.

Também, segundo Alan Klement, Use Cases possuem 3 grandes problemas: usam personas, casam implementações, motivações e resultado e ignoram contexto, situações e ansiedades.

Se uma funcionalidade definida por uma *Use Case* falhar será difícil identificar o problema pois, por casar informações diferentes, surgirá a dificuldade de saber o que está errado: a implementação, as suposições sobre as motivações ou os resultados.

Observando-se esses fatos optou-se pelo uso da metodologia *Jobs to Be Done*, pois há uma mudança na forma de fazer histórias a fim de evitar os problemas observados em Personas e Use Cases(Figura 2.4). Nesse modelo as Personas são pessoas reais e as histórias são escritas com a seguinte divisão: situação, motivação e resultados esperados.

Nessa metodologia, criamos histórias descrevendo tarefas para que os usuários se coloquem na situação descrita. Observando-os, é possível saber se a tarefa foi realizada de forma bem sucedida e, caso contrário, identificar onde e o que está causando a falha no instante da realização dela.

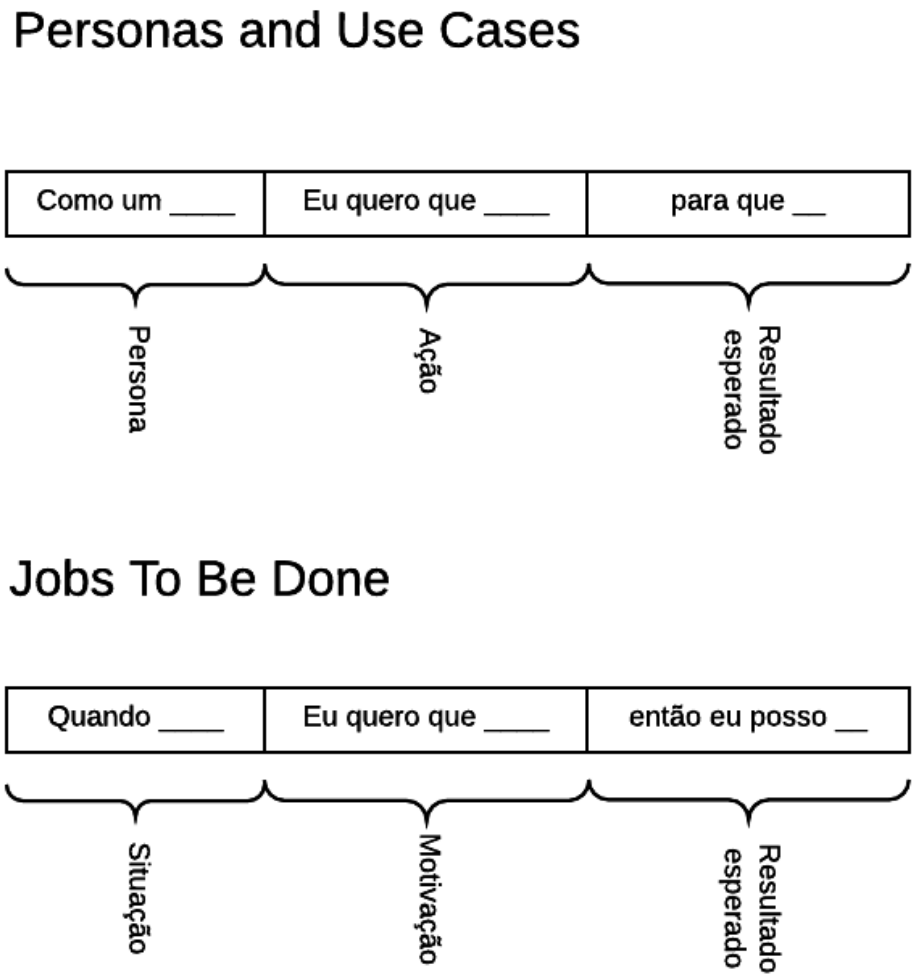



**Figura 2.4:** *Diferença entre estruturas de histórias*

Algumas das histórias criadas para os testes de interface do grupo foram:

- Quando eu estiver na tela de lista de restaurantes eu quero acessar a tela de detalhes do restaurante que eu desejo saber o tempo de espera, então eu posso ver com clareza as informações e posso traçar uma rota para o estabelecimento.
- Quando eu estiver na tela principal eu quero enviar meu *feedback* sobre o tempo de espera no estabelecimento que estou, então eu posso ajudar outros usuários por meio do aplicativo.

## 2.4 Cálculo da previsão do tempo de espera na fila

Através de pesquisas em busca de métodos estatísticos para se estudar o comportamento das filas, foram encontradas as três metodologias descritas nesta seção. Este capítulo contém a explicitação do problema estudado e suas possíveis soluções.

### 2.4.1 Descrição do problema

Para que se chegasse a uma solução apropriada, o problema de espera em fila precisou ser estudado. Esse problema pode ser descrito da seguinte forma: dado um restaurante, a entrada nesse é através de uma fila. Essa fila tem seu tamanho redefinido à medida que novos clientes chegam e saem dela. A saída dos clientes pode ser devido a desistência ou a acomodação deles nas mesas. A entrada é definida pela disponibilização de lugares livres à medida em que os grupos que estavam alocados terminam seu atendimento.

**Figura 2.5:** *Esquema simplificado do funcionamento do processo em um restaurante*

A fila de espera é influenciada por dois fatores: a taxa de chegada do cliente e a taxa de atendimento nas mesas.

- **chegada:** em restaurantes, a chegada de clientes é definida por uma taxa. O cliente, neste caso, representa um grupo com tamanho variável. A distribuição do tamanho do grupo nas chegadas é variável.
- **O atendimento:** a alocação do grupo é organizada pela recepção. O método utilizado por eles não é padronizado e muitas vezes é completamente subjetivo. Por exemplo, se acontecer de um grupo grande e vários casais chegarem ao mesmo tempo, e nesse

momento existir um espaço no salão que acomode o grupo (juntando as mesas) ou todos os casais (com as mesas separadas), dependendo de quem está responsável por alocá-los, cada um receberá uma prioridade diferente e essa decisão não é padronizada.

- **A disposição das mesas:** a disposição e quantidade de mesas não são padronizadas entre restaurantes de uma mesma rede, muito menos entre restaurantes diferentes. Além disso, a formatação do salão pode ser remodelada de acordo com a necessidade de alocação dos clientes, há gerentes que evitam essa movimentação ou até mesmo restaurantes que não têm essa possibilidade.

### 2.4.2 Métodos de cálculo da espera

Com o problema desenhado desta maneira, esboçou-se três possíveis métodos de resolução e um desses foi aplicado na solução final. Os métodos desenvolvidos são: média por período, aplicação da teoria das filas e utilização de redes neurais. Essas técnicas precisam de dados para serem desenvolvidas e aplicadas. Por conta disso, buscou-se fontes que suprissem essa necessidade.

#### Captação dos dados para a análise

Para que seja possível aplicar os métodos estudados é necessário obter dados para aplicar na análise. Estes foram obtidos de duas formas: a partir de uma parceria com a empresa *Get In* [7], desenvolvedora de um aplicativo de gerenciamento de filas de restaurantes, e através de pesquisas presenciais em restaurantes da rede *Outback Steakhouse*[14], selecionados para o estudo(Apêndice A). O aplicativo da *Get In* possui uma funcionalidade principal: realização de uma reserva em um determinado restaurante. O banco de dados mantém o registro de todos os usuários que entram na fila e esses dados são organizados em uma tabela com os seguintes campos:

- Origem: Se a reserva foi realizada de forma remota ou presencial. Assim,
  1. Reserva presencial (na entrada do restaurante incluída pela recepcionista)
  2. Reserva remota (realizada através do aplicativo)
- Entrada: este consiste na data e hora que o usuário entrou na fila.
- Qtd. Pessoas: tamanho do grupo para o qual a reserva foi feita.
- Prioritário: se este grupo requer atendimento prioritário
  0. Não
  1. Sim
- Notificado: A data e horário em que o grupo foi notificado que sua mesa estava pronta.
  - Quando NULL a recepcionista não notificou essa pessoa (possivelmente chamou diretamente pelo nome).
- Tempo na Espera: Tempo que o grupo ficou ativo na fila de espera (em segundos).
- Status: Qual a situação da reserva do grupo. Podendo assumir tais valores:
  0. Cancelado pela recepcionista.

2. Sentado
3. Cancelado pelo próprio Usuário

Para melhor entender estes parâmetros, o funcionamento do aplicativo deve ser explicado com maior detalhamento. O processo de reserva funciona da seguinte forma: o usuário reserva seu lugar através do aplicativo, ou vai ao restaurante e a recepcionista o adiciona à espera. No primeiro caso, o usuário tem um tempo de tolerância para chegar ao estabelecimento após receber a notificação, caso não consiga chegar a tempo, a reserva é cancelada. No segundo caso, quando o grupo a ser chamado está próximo da recepção, a funcionalidade de notificação não precisa ser usada, o responsável pode se direcionar diretamente aos clientes e chamá-los pelo nome. Mesmo nesse segundo caso, o tempo de espera na fila ainda é guardado pois a recepção precisa "fechar" a espera do grupo para chamar o próximo.

Porém, a utilização desses dados não foi liberada pela empresa *Get In* para serem usados como banco de dados do aplicativo, somente foram fornecidos dados do mês de agosto e setembro de 2016 para que fossem utilizados nas análises feitas com os modelos estatísticos. Para obter dados que poderiam ser utilizados no banco de dados do aplicativo, foi necessário uma pesquisa presencial onde o grupo visitou alguns restaurantes da rede *Outback Steakhouse* para obter informações sobre o tempo de espera nestes estabelecimentos.

Foram visitados 5 restaurantes: *Outback Steakhouse* Shopping Eldorado, *Outback Steakhouse* Shopping Villa Lobos, *Outback Steakhouse* Shopping Granja Vianna, *Outback Steakhouse* Shopping União de Osasco e *Outback Steakhouse* Shopping Market Place, onde questionamos sobre o tempo de espera por tamanho de grupos (em número de pessoas) e por horário e dia da semana.

Ao analisar os dados coletados pela pesquisa presencial, notamos que estes estavam suscetíveis a erros por terem sido obtidos por meio de diferentes observações humanas. Um método para obter dados mais precisos consistiria na realização de novas pesquisas no formato medição, onde cada integrante do grupo deveria coletar o tempo de espera de cada cliente através de um cronômetro e calcular o tempo de espera médio nesse horário. Como o grupo não conseguiria aplicar este método, optou-se por melhorar a precisão dos dados coletados usando como base o gráfico de horários de pico disponibilizado pela *Google* para cada restaurante. Essa melhora foi feita utilizando-se cálculos de proporção de valores.

### Primeiro método: Média por período

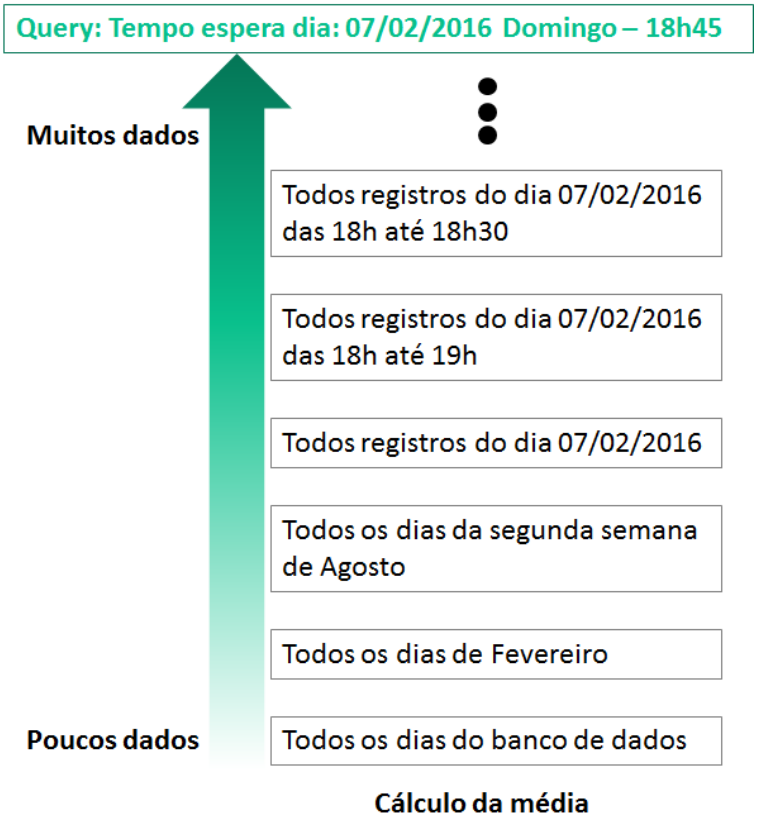



**Figura 2.6:** *Esquema do método média por período*

Para que uma previsão de espera possa ser fornecida ao usuário, essa deve basear-se, obrigatoriamente, em alguma informação prévia. Contudo, encontrar uma fonte para tais dados se mostrou uma tarefa muito complicada. Para conseguir, ainda assim, apresentar um tempo de espera em fila, métodos de análise foram definidos de acordo com a quantidade de dados disponível para cada estabelecimento.

De um modo geral, neste modelo, o cálculo do tempo de espera funcionará da seguinte forma: calcular-se-á a média de espera para o dia da semana e horário requeridos pelo usuário, sem diferenciar feriados de dias normais. Dependendo de quão longo é o histórico, níveis de especificidade podem ser adicionados, ou seja, numa busca "tempo de espera em 9 de Dezembro (segunda-feira) às 19h no restaurante X", a *query* selecionará todos os registros feitos no dia 9 de dezembro entre as 19h e às 20h dos últimos anos no restaurante X e realizará essa média, ou pode selecionar apenas linhas de todas as segundas-feiras do mês de Dezembro deste mesmo período.

- **Muitos dados:** neste caso, um maior nível de especificidade pode ser aplicado sem a utilização de outras fontes para complementar o banco de dados.
- **Alguns dados:** o nível de especificidade é bem menor, apenas filtrando por dia da semana e intervalo de hora, sem mês ou dia específico no ano. Em alguns casos, a quantidade de dados é tão pequena que deve-se recorrer a outras fontes. A *Google* tem disponibilizado gráficos com a distribuição dos clientes nos horários de funcionamento de estabelecimentos. Esses dados não estão disponíveis em sua API, ou seja, não

há uma forma automática para que o aproveitamento desses possa ser feito. Porém, mesmo omitindo-se os valores do gráfico, o perfil da distribuição e o horário de pico são aproveitados manualmente para nosso sistema.

- **Poucos dados:** para este caso, fizemos o balanceamento dos dados da *Google* com as informações retiradas das entrevistas. Além do aproveitamento das informações da *Google*, foram realizadas pesquisas com funcionários dos restaurantes.

**Segundo método: Teoria das filas**

Filas podem ser vistas como um sistema dinâmico no qual existe um conjunto de usuários que são atendidos por um conjunto de atendentes de acordo com a ordem de chegada dos usuários. A teoria de filas busca, utilizando análises matemáticas, estudar o comportamento destas filas para estimar o tempo de espera com base em um conjunto inicial de dados previamente fornecido.[30]

Um sistema de filas possui como características:

- **Modelo de chegada dos usuários:** Utiliza como base o tempo entre as chegadas dos usuários.
- **Modelo de serviço (atendimento aos usuários):** Utiliza como base o tempo gasto no atendimento ao usuário. No caso dos restaurantes, equivale ao tempo que um grupo passa na mesa até a sua saída.
- **Número de servidores:** É o número de atendentes disponíveis no sistema. Para o modelo do restaurante, este equivale às mesas disponíveis no salão.
- **Capacidade do sistema:** É o número de usuários que o sistema é capaz de atender. Inclui o número de usuários que estão sendo atendidos somado com os que esperam na fila. Se este parâmetro não for informado, o sistema é considerado com capacidade ilimitada.
- **Tamanho da população:** número potencial de clientes que podem chegar a um sistema. Pode ser finito ou infinito.
- **Disciplina da fila:** A disciplina da fila pode ser: FIFO (*first in, first out*) - o primeiro a chegar é o primeiro a ser atendido; LIFO (*last in, first out*) - o último a chegar é o primeiro a ser atendido; ALEATÓRIO - os atendimentos são feitos sem qualquer preocupação com a ordem de chegada; COM PRIORIDADE - os atendimentos são feitos de acordo com prioridades estabelecidas. No caso dos restaurantes, a disciplina adotada é a FIFO, porém, esta pode mudar de acordo com o atendente responsável.

Os tempos mencionados nos modelos de chegada e de serviço podem ser: constantes, quando os intervalos entre as chegadas dos usuários são iguais, ou variáveis, quando os valores desses intervalos variam. Nesse último caso, o intervalo de tempo segue uma distribuição de probabilidades. Nesse modelo também existe a taxa média de chegada dos usuários por unidade de tempo, ou seja, quantos usuários chegam em uma hora.

As filas podem ser moldadas em três situações básicas, como descrito por Azmat Nafees[30].

- Modelo de fila de canal único com fila única e atendimento único. (Figura 2.7)

- Modelo de fila de canal único com fila única e múltiplos atendimentos paralelos (Figura 2.8)
- Modelo de fila de canal único com múltiplas filas e múltiplos atendimentos paralelos (Figura 2.9)

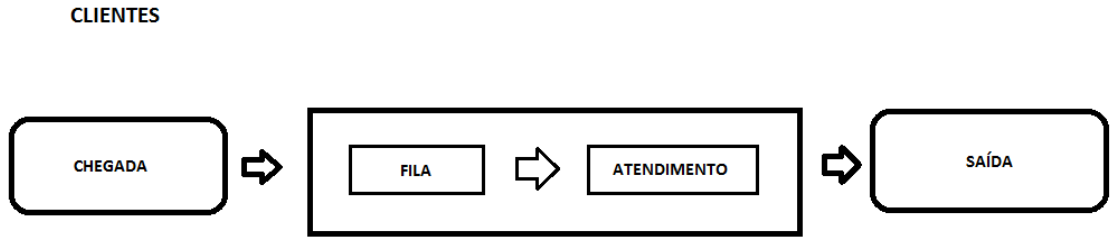



**Figura 2.7:** *Modelo de fila de canal único com fila única e atendimento único.*

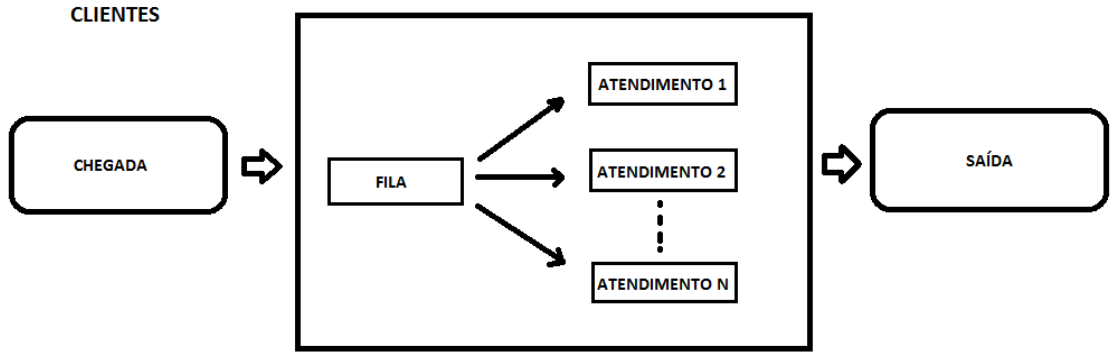



**Figura 2.8:** *Modelo de fila de canal único com fila única e múltiplos atendimentos paralelos*

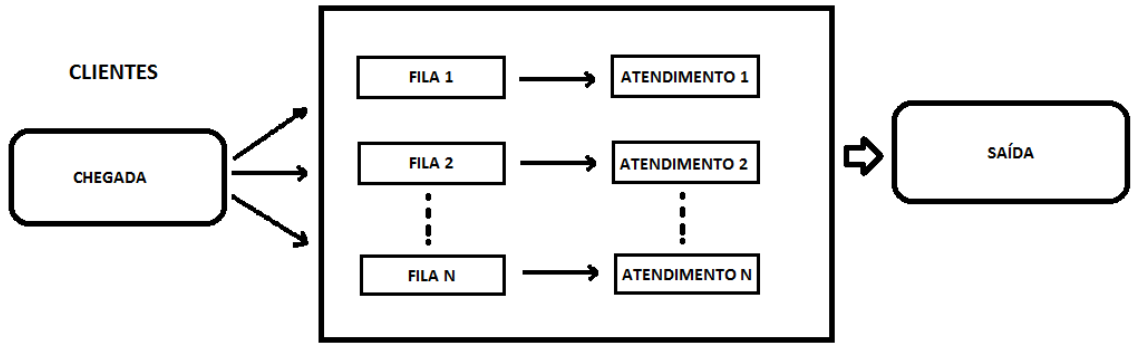



**Figura 2.9:** *Modelo de fila de canal único com múltiplas filas e múltiplos atendimentos paralelos*

Também por Azmat Nafees, nesses modelos, três subprocessos podem ser vistos:

- **Processo de chegada:** É um processo contínuo do evento "chegada de clientes", ou seja, quando o evento ocorre o processo aguarda a ocorrência do próximo evento. Esse evento coloca um cliente na fila quando ele chega.

- **Processo de espera:** É um processo que inclui o evento "comece a servir o próximo cliente da fila", que retira um cliente da fila, o coloca no processo de atendimento, e espera a conclusão do evento "cliente servido" para recomeçar. Nele se trabalha com tamanho e disciplina da fila.
- **Processo de atendimento:** É um processo que inclui o evento "cliente servido" e espera a conclusão do evento "comece a servir o próximo cliente da fila" para recomeçar. Nele se trabalha com tipo, taxa e tempo de atendimento.

Dentre esses 3 modelos, o que mais se encaixa na situação de um restaurante é o segundo modelo, canal único com fila única e múltiplos atendimentos paralelos. O processo de chegada de novos clientes na fila é tido como aleatório.

### Terceiro método: Rede Neural

Rede Neural é uma técnica computacional inspirada no sistema nervoso que, com o auxílio de um modelo matemático, apresenta respostas para problemas de decisão. Esses modelos funcionam da seguinte forma: primeiro um conjunto de dados (conjunto de treinamento) é apresentado ao sistema e este aprende com ele através de reconhecimento de padrão, então coloca-se o sistema para analisar os dados (conjunto de teste) para os quais deseja-se obter algum resultado. Este processo é chamado de aprendizagem de máquina. Quanto melhor for seu treinamento, mais precisas serão as respostas para o segundo conjunto.

Uma rede neural é formada por subestruturas chamadas neurônios. Um neurônio tem o seguinte funcionamento: ele recebe como entrada sinais, cada sinal recebe um peso de acordo com a influência dele na saída e uma soma é feita para produzir o nível de atividade, então, dependendo se o nível de atividade exceder o limite dado pela função de transferência, a resposta é produzida[27].

#### Software Multiple Back-Propagation

Para implementar a solução baseada em redes neurais, foi usado o *Software Multiple Back-Propagation* [1]. Este é um software de fácil utilização que fornece uma rede a qual aprende através do processo de propagação inversa. Esse método baseia-se na comparação dos dados esperados (conjunto de teste) com os dados obtidos a partir do treinamento, se necessário, adaptações são feitas a estrutura da rede para melhor aproximar aos dados do conjunto teste e isso é chamado de propagação inversa.

Por se tratar de um método de aprendizagem de máquinas, para se obter um resultado, deve-se treinar o algoritmo. Primeiramente, submete-se o conjunto de treinamento ao software: para isso foi escolhido o mês de Agosto. O critério de parada para o treino é o erro quadrático médio, o qual teve seu valor atribuído para 0,01.

# Capítulo 3

# Tecnologias utilizadas

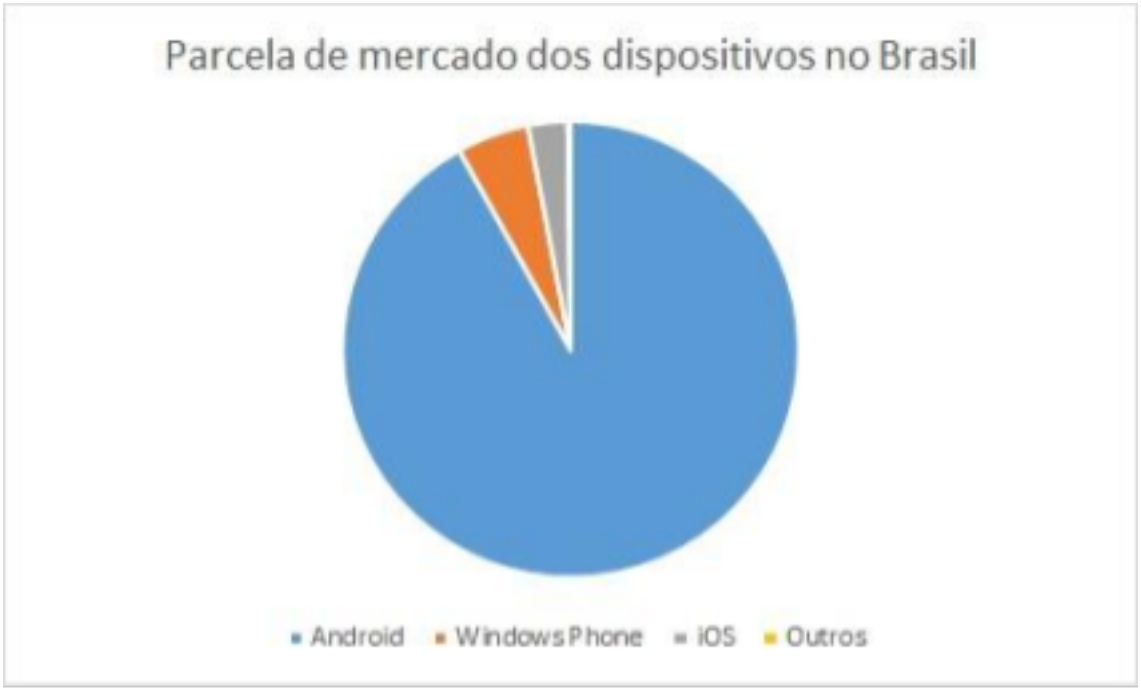
## 3.1 O sistema Android

*Android* é um Sistema operacional *open source* (software de código aberto, o qual as pessoas podem acessar e modificar, para contribuir com o projeto [25]) e para dispositivos móveis, desenvolvido pela *Google*, baseado no Kernel do Linux e utilizado principalmente em dispositivos *touchscreen*. Atualmente, o *Android* é o sistema operacional que detém a maior parcela do mercado[17], com 83.6% dos dispositivos, contra 13.4% do *iOS*, da *Apple*.

A *Google* também oferece um ambiente de desenvolvimento para a plataforma *Android* chamado *Android Studio*, no qual os aplicativos são criados e exportados para os dispositivos móveis. Aplicativos *Android* são programados na linguagem Java, com várias bibliotecas disponibilizadas pela *Google* para suporte ao sistema operacional.

A escolha da plataforma foi feita principalmente por conta de o sistema possuir a maior parte dos usuários de *Smartphones*, e também pela ampla comunidade ativa de desenvolvedores, à qual se pode recorrer em casos de dúvidas e problemas, tornando suas resoluções mais rápidas e com múltiplas alternativas. Outros motivos que contribuíram para a escolha do *Android* foram:

- A disponibilidade do ambiente de desenvolvimento *Android Studio*, que possui todos os recursos necessários para desenvolver um aplicativo funcional logo após a instalação, sem a necessidade de utilizar outras dependências, as quais são gerenciadas automaticamente pelo próprio software;
- O fato de, no mercado brasileiro, o sistema *Android* se encontrar em 91,8% dos dispositivos, segundo pesquisa da Kantar[18], e dado a pesquisa realizada para definição de público alvo onde foi possível perceber que 63.9% dos entrevistados utilizam *smartphones* com sistema *Android*;
- A utilização da linguagem de programação Java para desenvolvimento, que já era conhecida e fora utilizada anteriormente, ao contrário das linguagens Objective C e *Swift*, linguagens utilizadas para desenvolver aplicativos para o sistema *iOS*. Assim como a disponibilidade de dispositivos por parte do grupo, que em sua maioria era de celulares com o sistema *Android*.




Fonte: *Kantar*

**Figura 3.1:** *Parcela de mercado dos dispositivos no Brasil*

### 3.1.1 Organização de uma aplicação Android

Aplicativos *Android* são divididos em componentes. Há quatro tipos de componentes presentes: *Activities* (ou atividades), *Services* (ou serviços), *Content providers* (ou provedores de conteúdo), *Broadcast receivers* (ou receptores de transmissão).

- ***Activities***: São as telas e interfaces do usuário. Cada tela de uma aplicação é representada por uma *activity*, e cada *activity* é independente de outras, apesar de serem componentes de uma mesma aplicação. Isso se dá pelo fato de haver a possibilidade de utilizar *activities* de aplicativos durante a execução de outros. Atividades são iniciadas e se comunicam entre si através de objetos chamados *Intent*, que transferem um pacote de dados que pode ser carregado no início da nova atividade.

- ***Services***: Sempre são executados em segundo plano sem interferir na experiência do usuário, geralmente para realizar operações longas, ou realizar operações de troca de dados enquanto o usuário está utilizando outra aplicação em primeiro plano.

- ***Content Providers***: São gerenciadores de dados de aplicações *Android*, utilizados para ler e salvar dados compartilhados ou secretos dos aplicativos e do sistema.

- ***Broadcast Receivers***: São componentes responsáveis por obter eventos que aconteceram em outras aplicações ou no sistema e direcionar dados sobre esses eventos para eventuais aplicações que necessitem destes dados, alguns exemplos de eventos obtidos pelos receptores de transmissão são tela desligada e bateria fraca.

Além dos componentes, uma parte importante de uma aplicação *Android* é o Arquivo *Manifest*, que é uma declaração de permissões, serviços, e exigências de hardware necessários

para o funcionamento do aplicativo. Outra função é possuir declarações de todos os componentes utilizados no projeto, nomes, ícones, entre outros.

As interfaces de usuário do *Android* são definidas em arquivos XML (*eXtensible Markup Language*)[26], nos quais estão declarados todos os *layouts* e configurações das telas. Esses arquivos de *layout* são depois inflados e são gerados os componentes visuais da aplicação. Essa estrutura tem como objetivo facilitar o design da interface do usuário, já que arquivos xml são de fácil compreensão do desenvolvedor.

## 3.2 JSON

JSON, ou *Javascript Object Notation* é um formato de dados amplamente utilizado para troca de informações entre programas. Possui uma notação simples para grande parte dos programadores, pois se assemelha a convenções utilizadas em linguagens de programação muito conhecidas, como C, C++, Python e Java. O objetivo é que ele seja de fácil entendimento tanto para pessoas, como para máquinas.[11]

Um arquivo JSON é organizado como um objeto, contendo pares "nome:valor"separados por vírgulas e dentro de chaves(‘‘, ’’), ou como um array, que é uma lista de valores separados por vírgulas e entre colchetes(‘[‘, ’]’). Um valor, por sua vez, pode ser um objeto ou um array, mas também um número inteiro ou uma String, entre outros.

Um exemplo de arquivo JSON é:

```
1 {
2    "Carros": [
3       {
4           "Nome":  "Scenic",
5           "Marca": "Renault",
6           "Ano":   2012
7       },
8       {
9           "Nome":  "Sentra",
10          "Marca": "Nissan",
11          "Ano":   2009
12      }
13   ]
14 }
```

## 3.3 Google APIs

APIs (*Application Program Interfaces*) são interfaces para facilitar a troca de informações entre aplicações utilizando comandos específicos[15].

Ao longo do desenvolvimento, algumas APIs fornecidas pela *Google* foram utilizadas como fonte de informações adicionais para o aplicativo:

- ***Google Maps Distance Matrix API***: Esta API é utilizada para obter o tempo em segundos de trajeto entre dois pontos (lat/long) no mapa, podem ser consultados os tempos de trajeto de carro, bicicleta, a pé e utilizando transporte público.[28] Esta é

uma API web, logo, utiliza-se um request *http* para realizar uma consulta.

Por exemplo, o *request*:

https://maps.googleapis.com/maps/api/distancematrix/json?origins=Seattle&destinations=Victoria+BC&key=[user_key]

retorna o JSON

```
{
  "destination_addresses" : [ "Victoria, BC, Canada" ],
    "origin_addresses" : [ "Seattle, WA, EUA" ],
    "rows" : [
        {
           "elements" : [
              {
                 "distance" : {
                    "text" : "172 km",
                    "value" : 171668
                 },
                 "duration" : {
                    "text" : "4 horas 40 minutos",
                    "value" : 16809
                 },
                 "status" : "OK"
              }
           ]
        }
      ],
      "status" : "OK"
}
```

no qual é possível obter o tempo de trajeto em segundos.

- ***Google Maps Android API***: Esta API é utilizada para habilitar o uso de mapas da *Google* integrados à aplicação, já disponível com todas as funcionalidades de zoom e arrastar presentes no aplicativo nativo, permitindo também a personalização como a adição de pontos definidos pelo usuário ao mapa

## 3.4 Localização do Usuário

Outro serviço que foi utilizado com o intuito de obter o tempo de trajeto e a lista ordenada de forma personalizada de restaurantes foi a localização do usuário, que é obtida através do GPS, presente nos *smartphones*. Um objeto de localização é criado, e realiza a comunicação com o serviço de localização do aparelho. Ao obter a localização, ele lança um evento que pode ser acessado por outras partes da aplicação, obtendo a localização mais atualizada do dispositivo.

## 3.5 Controle de Versão

Controle de versão é uma prática usada para manter a organização do código e do seu desenvolvimento quando muitos desenvolvedores colaboram conjuntamente. Existem softwares que auxiliam essa organização, registrando todas as mudanças feitas ao longo do processo, e que possui recursos que permitem que vários usuários trabalhem nos mesmos arquivos sem perda de informações.

No caso do *Git* [8] , o projeto fica organizado em uma estrutura de árvore, onde cada *commit* (*checkpoints* criados pelos membros do grupo durante o processo) é marcado por um nó, para onde se pode retornar quando necessário. Existem também estruturas chamadas *branches*, que são bifurcações na estrutura para que partes do projeto sejam criadas separadamente, sem interferir com a linha principal. As *branches*, depois, são unidas à linha principal através do comando *merge*.

Neste projeto, o serviço de controle de versão utilizado foi o *Bitbucket*[2] (que é um serviço *web-based* de controle de versão que utiliza *Git*) com auxílio do software *SourceTree*[19], que é uma interface gráfica para os serviços do *Bitbucket* onde é possível visualizar com mais facilidade as fases do desenvolvimento do projeto e realizar os comandos mais intuitivamente.

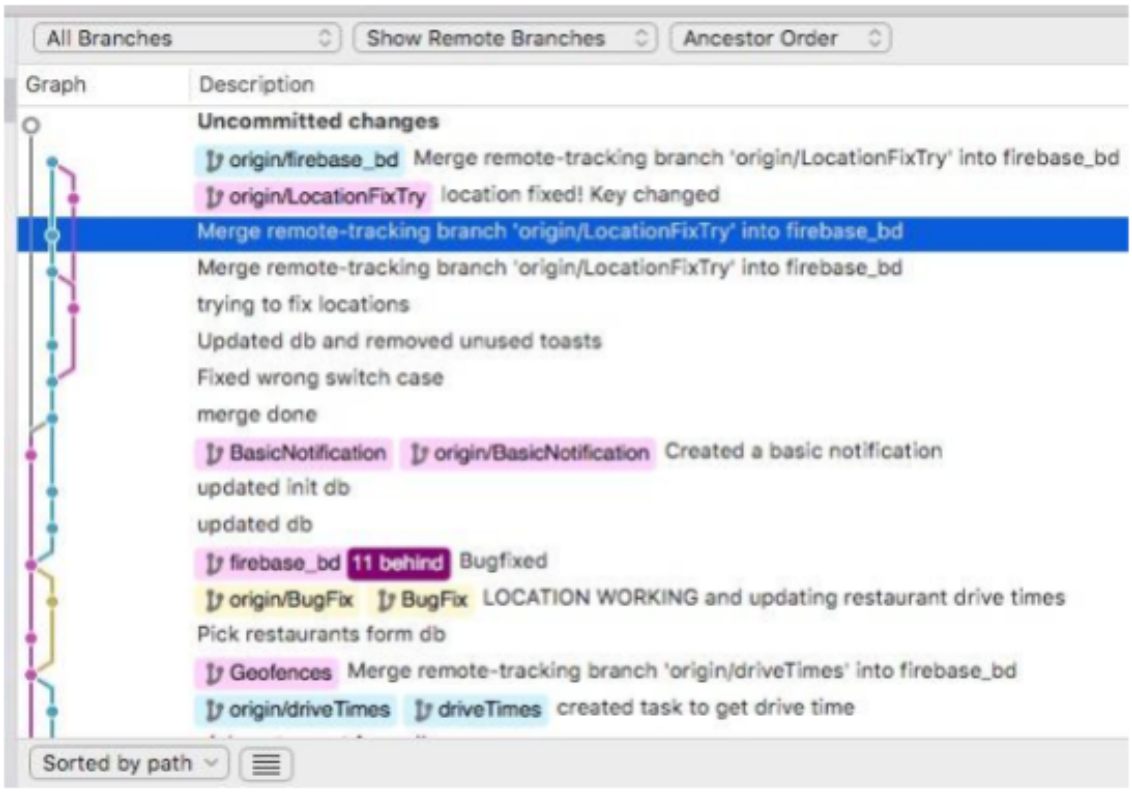



**Figura 3.2:** *Interface do SourceTree que mostra o controle de versionamento do projeto*

## 3.6 Banco de dados

Com o intuito de deixar o aplicativo mais completo, guardando informações dos usuários e dos estabelecimentos, optou-se por utilizar-se um banco de dados. Alguns exemplos de

banco de dados aprendidos no curso Laboratório de Banco de Dados foram:

- ***SQL*** (*Structured Query Language* - Linguagem de Consulta Estruturada)[20], uma linguagem para atualizar, deletar e requisitar informações através de consultas em tabelas que possuem relação entre si;
- ***noSQL***[13], um banco de dados que armazena e recupera informações modeladas de uma forma onde não são usadas tabelas relacionais.

Como este último não foi o foco da disciplina por falta de tempo, decidiu-se por utilizá-lo para que o grupo pudesse aprender melhor esse modelo.

Outra decisão tomada foi a hospedagem dos dados na nuvem, ou seja, colocá-los em um servidor que não seja local, e conseguir acessá-los através de uma conexão via internet. Desse modo os dados sempre estarão disponíveis para serem consultados.

### 3.6.1 Firebase

Buscando-se atender as decisões tomadas sobre o banco de dados, procuramos um serviço de banco de dados da *Google*, por conter ampla integração com a plataforma de desenvolvimento *Android*, por ser conhecido, bastante utilizado e por possuir uma ampla comunidade de desenvolvedores dispostos a oferecer auxílio em fóruns de discussão de códigos.

Para atender esses requisitos escolheu-se o *Firebase Realtime Database*[6], desenvolvido e disponibilizado pela *Google*, que consiste em um banco de dados *noSQL* com armazenamento na nuvem. Esse serviço armazena os dados do usuário no servidor da *Google* em formato *JSON* e, a cada alteração, sincroniza em tempo real com todos os clientes conectados.

O grupo pode observar as seguintes vantagens de se trabalhar com a ferramenta: existe a sincronização em tempo real dos dados de todos os clientes e sua disponibilidade mesmo com o aplicativo *offline*, ou seja, quando o aplicativo não estiver conectado a internet, os dados do dispositivo móvel serão armazenados e sincronizados com o banco de dados assim que a conexão estiver disponível novamente. Além disso, os desenvolvedores possuem acesso a um console online (figura 3.3) onde o *JSON* é mostrado numa interface interativa de uma forma mais amigável e no qual é permitido editá-lo facilmente.

**Figura 3.3:** *Console online do Firebase, onde os dados são mostrados de forma mais amigável e intuitiva.*

Outra observação feita foi que o *Firebase Realtime Database* funciona como o esperado se ele guardar uma cópia local dos dados em um arquivo *JSON* no celular. Porém, isso não foi visto como um problema pois, como essa cópia pode ser salva em um arquivo texto, o espaço utilizado diminui consideravelmente.

Na primeira vez que os dados são criados e todas as vezes em que os dados forem atualizados por algum dispositivo móvel é enviada uma atualização que será recebida através de *listeners* por todos os aparelhos que utilizam o aplicativo para que eles atualizem seus respectivos *JSONs*. *Listeners* são interfaces de comunicação que executam uma série de ações assim que um outro evento esperado acontece.

## 3.7 Crowdsourcing

O termo *Crowdsourcing* tem origem nas palavras da língua inglesa *Crowd* (multidão) e *Outsourcing* (terceirização), e é definido pelo dicionário Merriam Webster como "A prática de obtenção de serviços, ideias, ou conteúdo solicitando contribuições de grandes grupos de pessoas e especialmente da comunidade online "[3]. Esta prática é utilizada para que seja possível à comunidade contribuir com dados para algum programa ou serviço, como no aplicativo *Waze*, do qual a principal fonte de dados de trânsito, acidentes e radares é obtida por meio dos próprios usuários.

# Capítulo 4

# Desenvolvimento do Projeto

Este capítulo aborda o processo de desenvolvimento do projeto, iniciado com o preenchimento do diagrama *Canvas* e uma pesquisa de mercado direcionada ao usuário. A partir das informações coletadas foi possível projetar um aplicativo que atendesse as necessidades expostas pela pesquisa e selecionar as tecnologias necessárias. Também é abordada a aplicação dos diferentes métodos de cálculo da previsão de tempo de espera que foram estudados e explicados no Capítulo 2. Por fim são expostas as diferentes versões do aplicativo durante o processo e seus detalhes.

## 4.1 Pesquisa de Mercado

Antes de iniciar-se o desenvolvimento prático, algumas questões foram levantadas: se um aplicativo com as funcionalidades escolhidas seria de alguma forma útil para o usuário e, se sim, qual metodologia deveria ser usada para que este produto fosse entregue o mais completo possível até o final do ano. Para responder a primeira pergunta, desenvolveu-se um questionário.

### 4.1.1 Pesquisa com o usuário

O questionário foi desenvolvido para avaliar a viabilidade do aplicativo e suas funcionalidades, e foi criado utilizando-se o *Google Forms* [9]. Essa é uma ferramenta desenvolvida pelo *Google* que possibilita a criação de formulários contendo diversos formatos de perguntas, tanto dissertativas quanto de múltipla escolha.

Seguindo os princípios de desenvolvimento de questionários [4], o conjunto de perguntas foi mantido curto, simples e de fácil entendimento. Também variou-se o tipo de questões entre estruturadas (questões com respostas em forma de alternativas) e desestruturadas (questões com respostas em formato texto), seguindo esses mesmos princípios, para captar melhor as informações dos entrevistados e evitar erros de entendimento.

Com as respostas do questionário, tinha-se o intuito de traçar um possível público alvo, conhecer seu modo de uso de dispositivos móveis e validar a aplicabilidade de diferentes funcionalidades pretendidas inicialmente, sendo essas: engajamento do usuário através de notificações do sistema, aceitação de sugestões de locais alternativos, avaliação da existência do problema que queremos resolver (se este aplicativo supre uma necessidade real do usuário) e aceitação do aplicativo.

Para a divulgação, redes sociais como *Facebook* [5] e sistemas de mensagem instantâneas como *WhatsApp* foram usados, e o envio de respostas foi restrito até três semanas após sua primeira postagem. Neste período, foram obtidas 210 respostas, as quais ajudaram a definir os aspectos do sistema a ser desenvolvido.

O questionário foi dividido em duas subseções: perfil do usuário e funcionalidade. Com elas, obteve-se os seguintes resultados:

# Questão 1

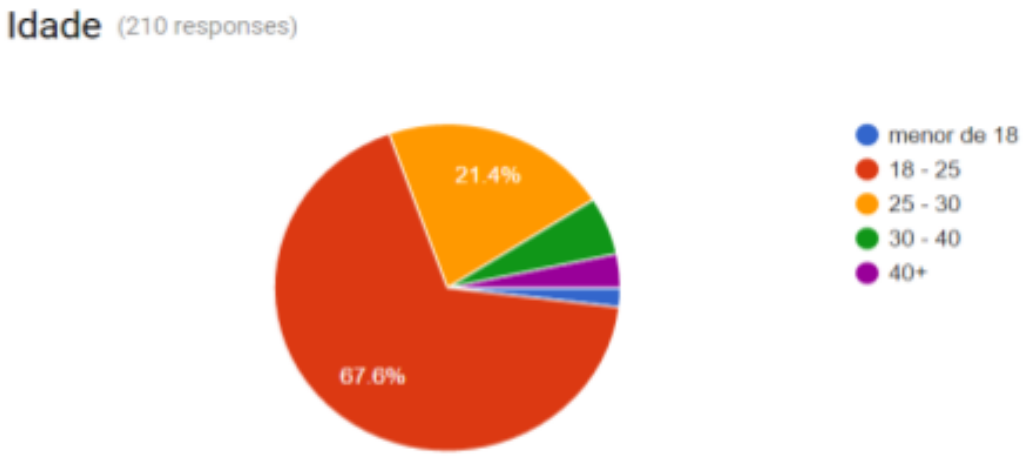



**Figura 4.1:** *Gráfico pizza da distribuição das respostas em relação a idade dos pesquisados*

As faixas de idade apresentadas estão desbalanceadas por não terem a mesma quantidade de anos em cada uma, porém, essa divisão foi pensada levando-se em conta a fluência com tecnologia que cada idade possui. Com o resultado verificou-se que, do público alcançado com a pesquisa, a maioria está na faixa de 18 a 25 anos de idade, assumida como parcela com maior conhecimento tecnológico.

# Questão 2

**Figura 4.2:** *Gráfico pizza da distribuição das respostas em relação ao sistema operacional*

Esta questão foi pensada para medir a porcentagem de cada sistema operacional presente nos celulares do público. Com ela descobriu-se que 63.9% das pessoas dentro desse

grupo possuem o sistema *Android*, enquanto 23.4% possuem *iOS* e 3.8% *Windows phone*. Essa pergunta foi incluída para um melhor direcionamento do desenvolvimento do aplicativo.

Uma questão levantada ao aplicar esse questionário foi a dificuldade que algumas pessoas encontraram para identificar a versão do próprio aparelho. Porém, para manter a consistência das respostas, decidiu-se não mudar nada nesta questão para remediar isso.

## Questão 3

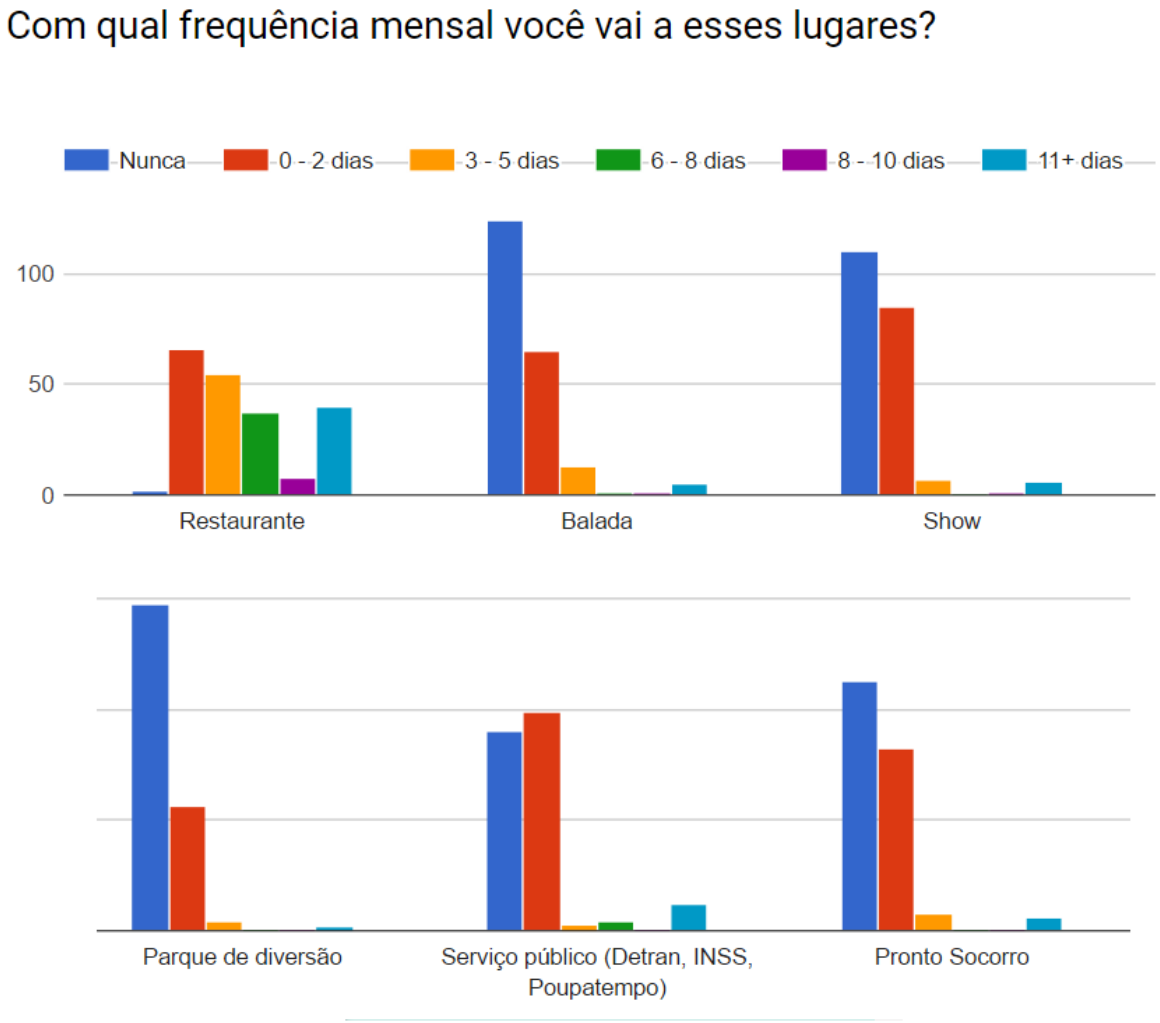



**Figura 4.3:** *Gráfico em barras da distribuição de frequências mensais de cada local analisado*

# Questão 4

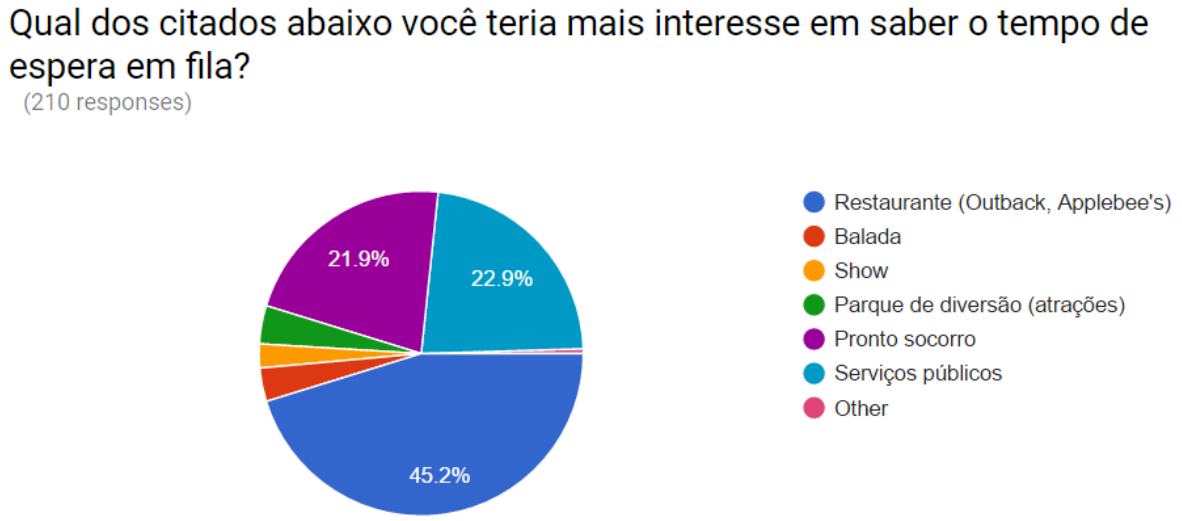



**Figura 4.4:** *Gráfico em pizza da distribuição de interesse do usuário no tempo de espera de cada local*

A princípio viu-se muitas possíveis aplicações para a previsão de tempo de espera. Para entender qual seria a prioridade do público, as questões acima foram incluídas. A primeira verificou a frequência de visitas em alguns locais e a segunda serviu para analisar qual estabelecimento despertava mais interesse nos entrevistados quanto ao quesito fila. Nos dois casos, restaurantes se destacaram com o maior interesse.

# Questão 5

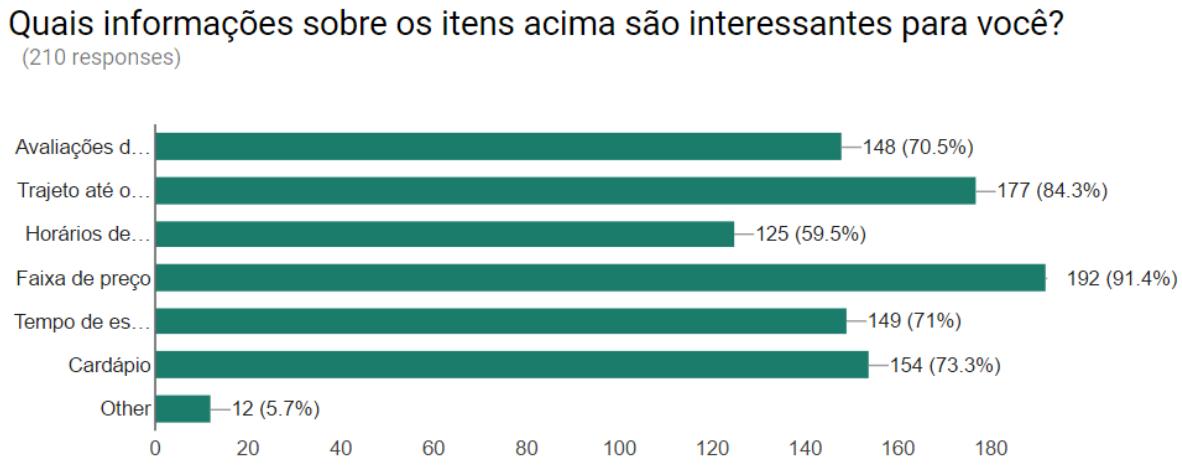



**Figura 4.5:** *Gráfico em barras da distribuição de interesse do usuário por cada item avaliado*

Para acompanhar o dado de espera e fornecer mais informações relevantes ao usuários em potencial, perguntou-se quais outras informações eles acham interessante saber sobre um determinado estabelecimento. Com isso, descobriu-se que a faixa de preço, o trajeto até o local e o cardápio seriam dados importantes de se obter juntamente com o tempo de espera.

# Questão 6

**O que você costuma fazer para evitar filas?**
Neste ponto, testou-se se a solução proposta através do aplicativo resolveria o problema do usuário de uma forma melhor do que a solução atualmente utilizada. Para isso, foi perguntado o que as pessoas fazem para evitar filas e as principais respostas foram:

- Evitar horários de pico
- Chegar cedo
- Ir em outro lugar caso uma fila seja encontrada
- Tentar realizar reservas antecipadamente
- Se distrair com livros/celular

Notou-se que as soluções utilizadas atualmente não suprem as necessidades dos entrevistados, pois mesmo em horários não convencionais, pode-se encontrar filas ou, quando decide-se esperar na fila, muito tempo é perdido. Concluiu-se que a solução proposta a partir do aplicativo resolve de forma mais abrangente o problema, pois ajuda o usuário a evitar horários de pico inesperados.

# Questão 7

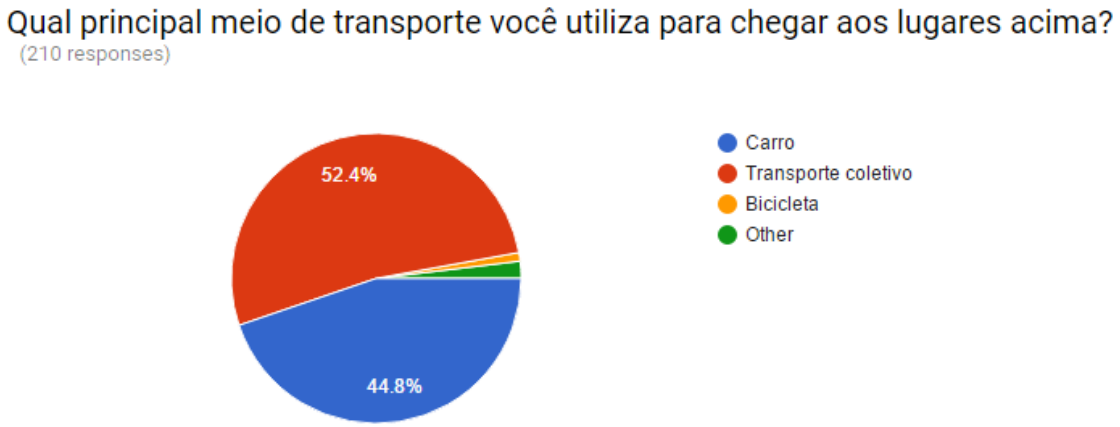



**Figura 4.6:** *Gráfico em pizza da distribuição do meio de transporte utilizado pelos usuários*

Essa questão foi formulada com o objetivo de verificar quais meios de transporte seriam incluídos com previsão de tempo de trajeto no aplicativo. Foi verificado que a maioria dos entrevistados utilizam transporte público, seguido por carro.

## Questão 8

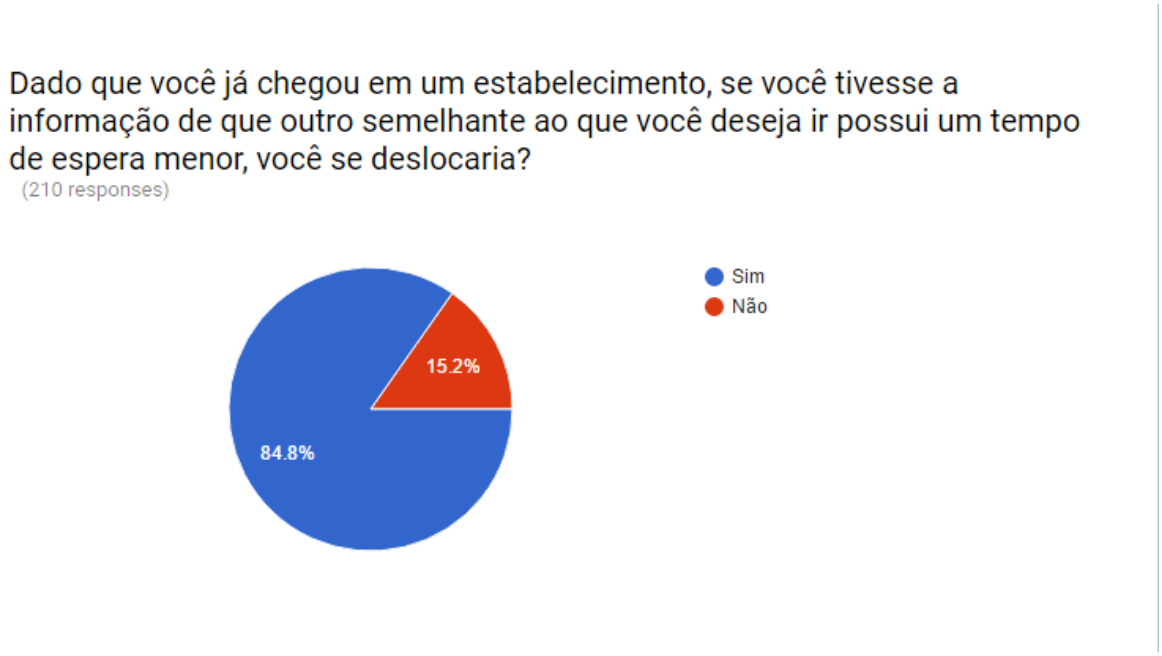



**Figura 4.7:** *Gráfico em pizza da distribuição do interesse do usuário pela informação fornecida*

## Questão 9

**Figura 4.8:** *Gráfico em pizza da distribuição de quantidade de quilômetros que o usuário se deslocaria para ir a um local com menos fila*

Com essas questões, foi verificado que algo importante para o usuário é saber de outros estabelecimentos que possam oferecer tempos de espera menores nas redondezas de onde o usuário se encontra, que é uma das funcionalidades que desejávamos utilizar no sistema.

# Questão 10

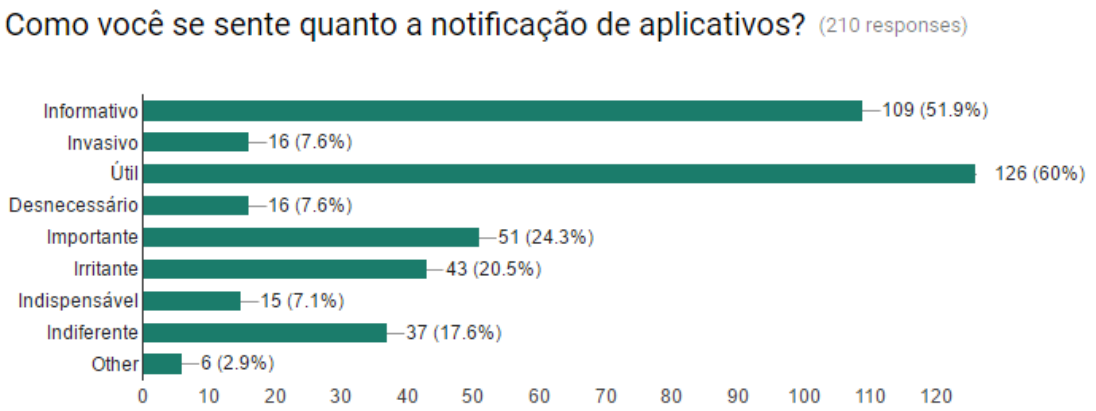



**Figura 4.9:** *Gráfico em barras da distribuição de sentimento do usuário em relação a notificação de aplicativos*

# Questão 11

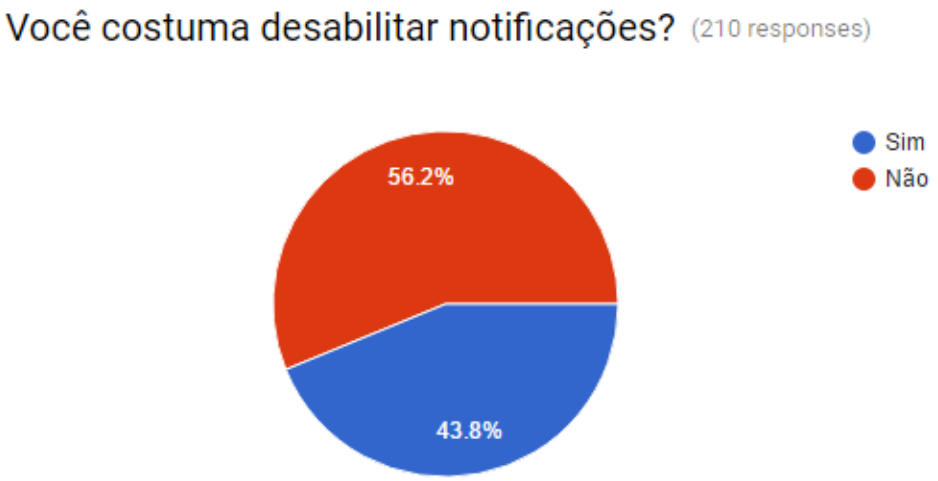



**Figura 4.10:** *Gráfico em pizza da distribuição de opinião do usuário sobre desabilitar notificações em celular.*

# Questão 12

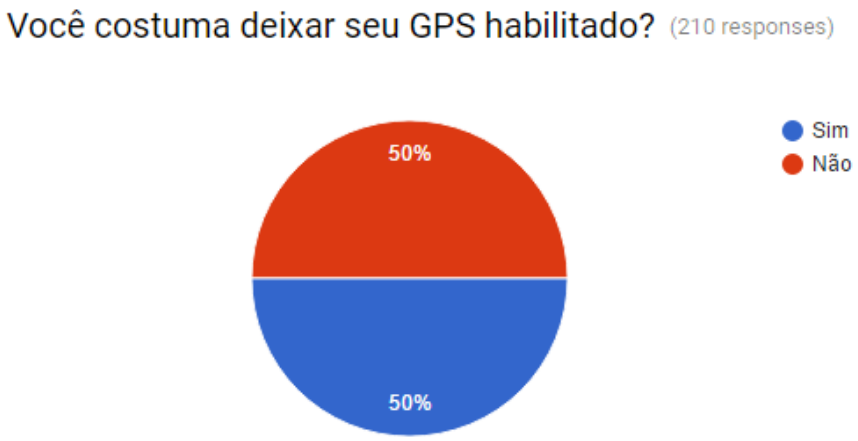



**Figura 4.11:** *Gráfico em pizza da distribuição de opinião do usuário sobre habilitar GPS em celular*

A ideia inicial era poder usar o GPS para verificar a localização do usuário e enviar notificações quando este estivesse parado nas proximidades de algum dos estabelecimentos cadastrados. Para que essa funcionalidade possa existir, é necessário que o aplicativo tenha acesso à localização precisa da pessoa através do GPS ou internet até mesmo quando não estiver aberto. Além disso, também é necessário que as notificações estejam habilitadas para finalizar esta interação. Então, as perguntas 10, 11 e 12 foram incluídas no questionário. Observando as respostas, foi possível verificar que 29.5% das pessoas que responderam deixam ambas as funcionalidades habilitadas. (GPS e notificações). Com isso, pudemos prosseguir com essa ideia e também pensar em possíveis alternativas de interação e engajamento com o usuário.

Por conta dessa pesquisa ter sido compartilhada através das contas das redes sociais e comunicadores dos componentes deste grupo, sabe-se que este pode apresentar um resultado viesado mas, que mesmo assim, não deve ser desconsiderado e que muito ajudou a sanar as dúvidas iniciais.
Como resultado, traçamos o perfil do entrevistado:

- Faixa etária predominante entre 18 e 30 anos.
- Possui sistema operacional *Android* (versões 4, 5 e 6).
- Vai a restaurantes com certa frequência mensal resultando em interesses sobre informações relacionadas a esse estabelecimento.
- Costuma chegar cedo ou evitar os horários de pico para evitar filas nos locais.
- Carro e transporte coletivo são os principais meios de transporte.
- Se deslocaria para um estabelecimento semelhante que possui menos fila, contanto que este fique num raio de 2km de distância.

## 4.2 Projeto do Aplicativo

O grupo começou o projeto com o preenchimento do diagrama *Canvas*(figura 4.12) apresentado na Seção 2.1. Desse modo foi adquirida uma ideia mais consistente e uma base para começar a implementação.

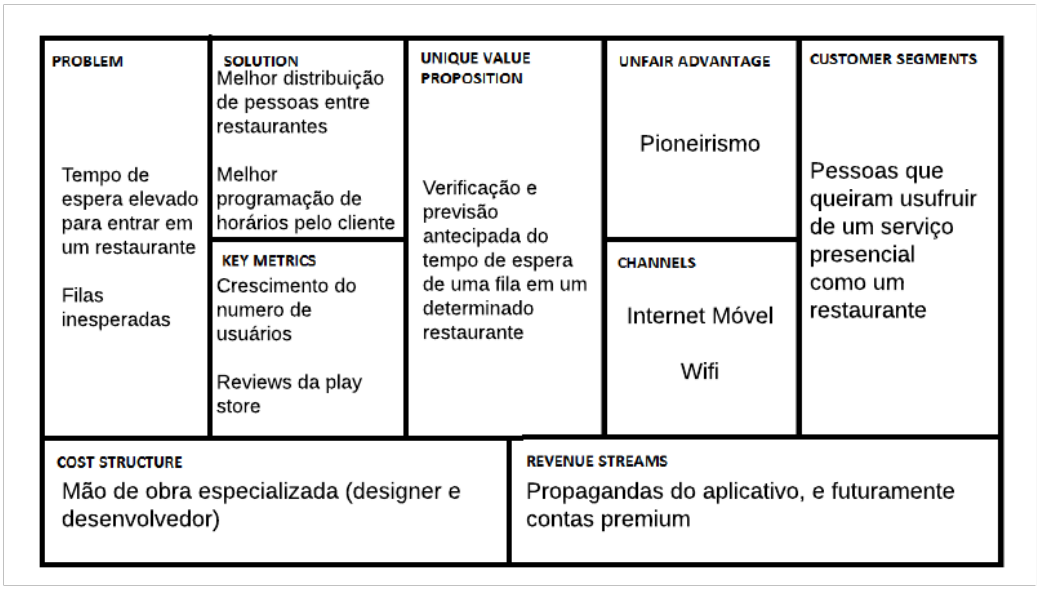



**Figura 4.12:** *Diagrama Canvas preenchido pelo grupo*

Com o modelo de negócios definido e com base no resultado da pesquisa com o usuário, apresentada na seção seguinte, foram aplicadas pequenas mudanças no projeto para melhor adequar-se às necessidades e ao perfil do público alvo.

Para o desenvolvimento, foi utilizado um método ágil baseado no *scrum* com *sprints* de 3 a 4 semanas, devido à disponibilidade dos membros do grupo. Os encontros não eram diários como recomendado para o método *scrum*, também em razão da disponibilidade. Haviam *stand ups*, (discussões rápidas no começo de cada encontro) e retrospectivas ao final de cada *sprint*. A cada *sprint* eram selecionadas histórias e tarefas para serem executadas, e caso não fossem concluídas ao final da iteração, eram inseridas ao *backlog* e feitas no *sprint* seguinte.

## 4.3 Aplicação dos métodos de cálculo de espera

### 4.3.1 Aplicando a média por período

Aplicando-se no trecho abaixo a técnica desenvolvida na Seção 2.4.2 para uma *query* realizada no dia 19/08/2016 no intervalo de 20:40 - 20:55 , analisa-se a semana anterior que possua tais intervalos. Assim, o sábado dia 06/08/2016 será selecionado.

| Entrada | Qtd. Pessoas | Notificado | Espera |
|---|---|---|---|
| 6/8/16 20:43 | 2 | 8/6/2016 20:53 | 1606 |
| 6/8/16 20:44 | 2 | 8/6/2016 20:55 | 674 |
| 6/8/16 20:48 | 2 | 8/6/2016 20:56 | 528 |
| 6/8/16 20:48 | 2 | 8/6/2016 20:57 | 520 |
| 6/8/16 20:51 | 2 | 8/6/2016 21:07 | 1118 |
| 6/8/16 20:51 | 3 | 8/6/2016 21:10 | 1305 |
| 6/8/16 20:52 | 2 | 8/6/2016 21:10 | 1273 |
| 6/8/16 20:52 | 2 | 8/6/2016 21:11 | 1179 |
| 6/8/16 20:53 | 2 | 8/6/2016 21:12 | 1217 |
| 6/8/16 20:53 | 2 | NULL | 59 |
| 6/8/16 20:54 | 3 | 8/6/2016 21:19 | 1541 |
| 6/8/16 20:55 | 2 | 8/6/2016 21:17 | 1444 |
| 6/8/16 20:55 | 2 | 8/6/2016 21:20 | 1570 |
| 6/8/16 20:55 | 2 | 8/6/2016 21:22 | 1613 |

**Tabela 4.1:** *Trecho do dia 06/08/2016, entre os horários das 20:43 às 20:55, um intervalo de 12 minutos, retirado dos dados da empresa Get In.*

Com esses dados, obteve-se a média de espera de 1080 segundos (18 minutos), a qual seria sugerida para o usuário com a busca, e desvio padrão de 465.144719 segundos (7,75 minutos). Dentro desse intervalo, tem-se como máximo tempo de espera 1613 segundos (26.88 minutos) e como mínimo 59 segundos (1 minuto). Analisando-se os dados do dia 19/08/2016 no mesmo período temos:

| Entrada | Pessoas | Notificado | Espera | Erro previsão | Erro p/ 2 pessoas |
|---|---|---|---|---|---|
| 19/8/16 20:43 | 2 | 8/19/2016 20:51 | 544 | -563 | 522 |
| 19/8/16 20:43 | 2 | 8/19/2016 20:52 | 605 | -502 | 461 |
| 19/8/16 20:44 | 2 | 8/19/2016 20:56 | 723 | -384 | 343 |
| 19/8/16 20:46 | 4 | 8/19/2016 21:19 | 1988 | 881 | X |
| 19/8/16 20:48 | 2 | 8/19/2016 21:02 | 919 | 188 | 147 |
| 19/8/16 20:51 | 2 | 8/19/2016 21:03 | 821 | 286 | 245 |
| 19/8/16 20:51 | 3 | 8/19/2016 21:08 | 1773 | 666 | X |
| 19/8/16 20:53 | 2 | 8/19/2016 21:10 | 1083 | 24 | -17 |
| 19/8/16 20:53 | 2 | 8/19/2016 21:11 | 1105 | 2 | -39 |
| 19/8/16 20:53 | 2 | 8/19/2016 21:12 | 1169 | 62 | -103 |
| 19/8/16 20:55 | 2 | 8/19/2016 21:14 | 1447 | 340 | -381 |

**Tabela 4.2:** *Trecho do dia 19/08/2016, entre os horários das 20:43 às 20:55,um intervalo de 12 minutos, retirado dos dados da empresa Get In.*

Ou seja, se um grupo chegasse dentro desse período e a previsão dada fosse a média obtida anteriormente, o erro, calculado pela diferença entre o tempo sugerido e o real tempo de espera, estaria no intervalo de 15 minutos para menos ou 8 minutos para mais. Sendo esse um valor aceitável.

Um próximo passo é a restrição dos registros a grupos com 2 integrantes. Com tal restrição a previsão de espera é de 1066 segundos (17.76 minutos). Aplicando essa nova média aos registros com 2 integrantes do dia 19/08/2016, obtém-se um erro (Coluna 6 - erro previsão para 2 pessoas) menor do que o anterior. O intervalo do erro calculado neste caso seria de 522 segundos (8.7 minutos) para mais e 381 segundos (6.35 minutos) para menos. Este resultado é ainda mais favorável, mostrando que a inclusão de mais filtros para o cálculo da previsão melhorou o tempo previsto.

### 4.3.2 Aplicando teoria das filas

A teoria das filas (Seção 2.4.2) é uma ferramenta muito completa que calcula muitos dados úteis para a previsão de espera. Contudo, por conta da necessidade de informações sobre o tempo de atendimento, a qual não está contida no banco utilizado, esta teoria não pode ser aplicada neste caso. Porém, esse dado pode ter um valor assumido para que a utilização dessa teoria fosse possível, entretanto a sua precisão ficaria comprometida e imprevisível.

### 4.3.3 Aplicando redes neurais

Para aplicar o que foi estudado na Seção 2.4.2, foi executado o software com o mês de Agosto como teste, o mês de Setembro como treino e o limite do erro quadrático como 0.01, e obteve-se a tabela presente no apêndice B. Na tabela podemos ver o tempo de fila esperado, ou seja, o tempo contido na tabela do mês de Setembro, o tempo que o algoritmo deu como resposta baseado no mês de Agosto e o erro entre eles, respectivamente.

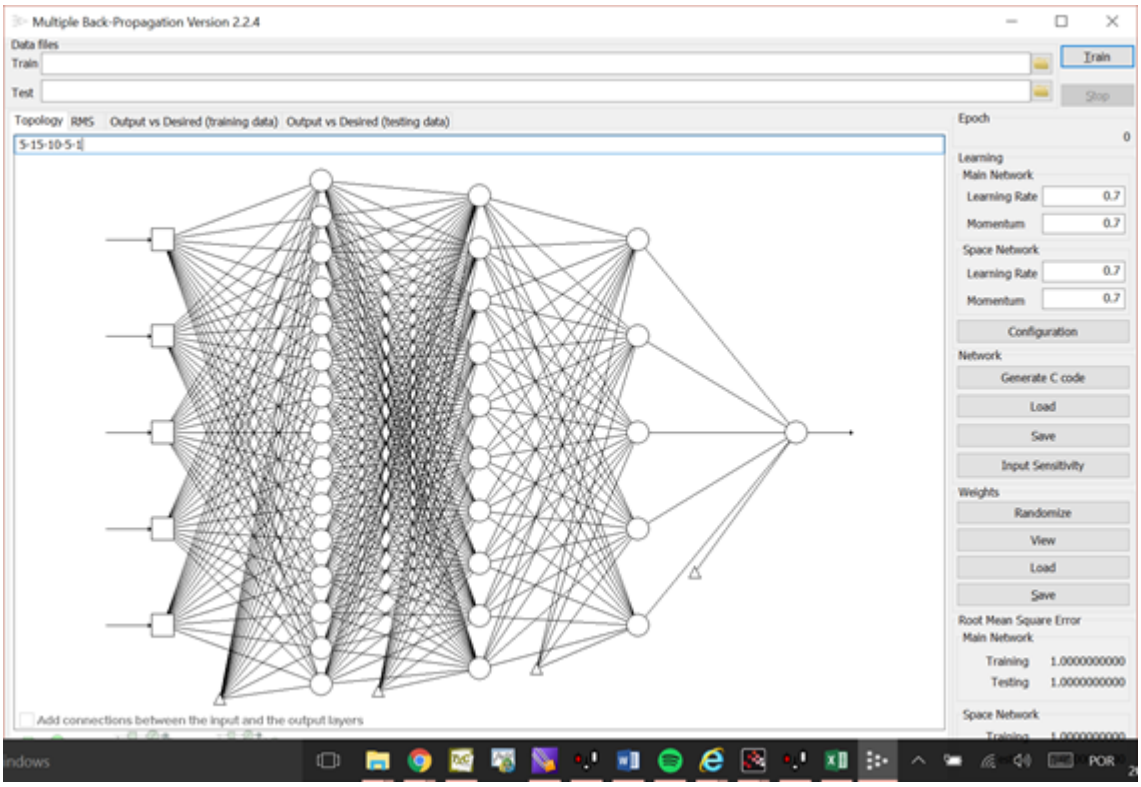



**Figura 4.13:** *Software Multiple Back-Propagation com a configuração escolhida para os dados disponíveis.*

Mesmo o erro se mantendo em 0.018, os dados obtidos possuem um erro padrão de 15 minutos. Se mais dados forem adicionados, o critério de parada pode ser atingido e assim, uma previsão melhor pode ser alcançada.

### 4.3.4 Conclusão sobre os métodos estudados

Dentre todas as possibilidades estudadas, visualizamos na média por período a melhor opção assumindo os dados disponíveis para teste. Se mais parâmetros fossem fornecidos, como o tempo de atendimento, a teoria das filas seria uma boa opção para fornecer a previsão de espera. E se mais dados estivessem disponíveis, as redes neurais conseguiriam uma aproximação melhor e assim, seria uma boa opção também.

## 4.4 Definição da Interface do aplicativo

Como um primeiro protótipo, foi desenvolvida uma interface simples que tinha como tela de início um mapa com um campo para buscas na parte superior.

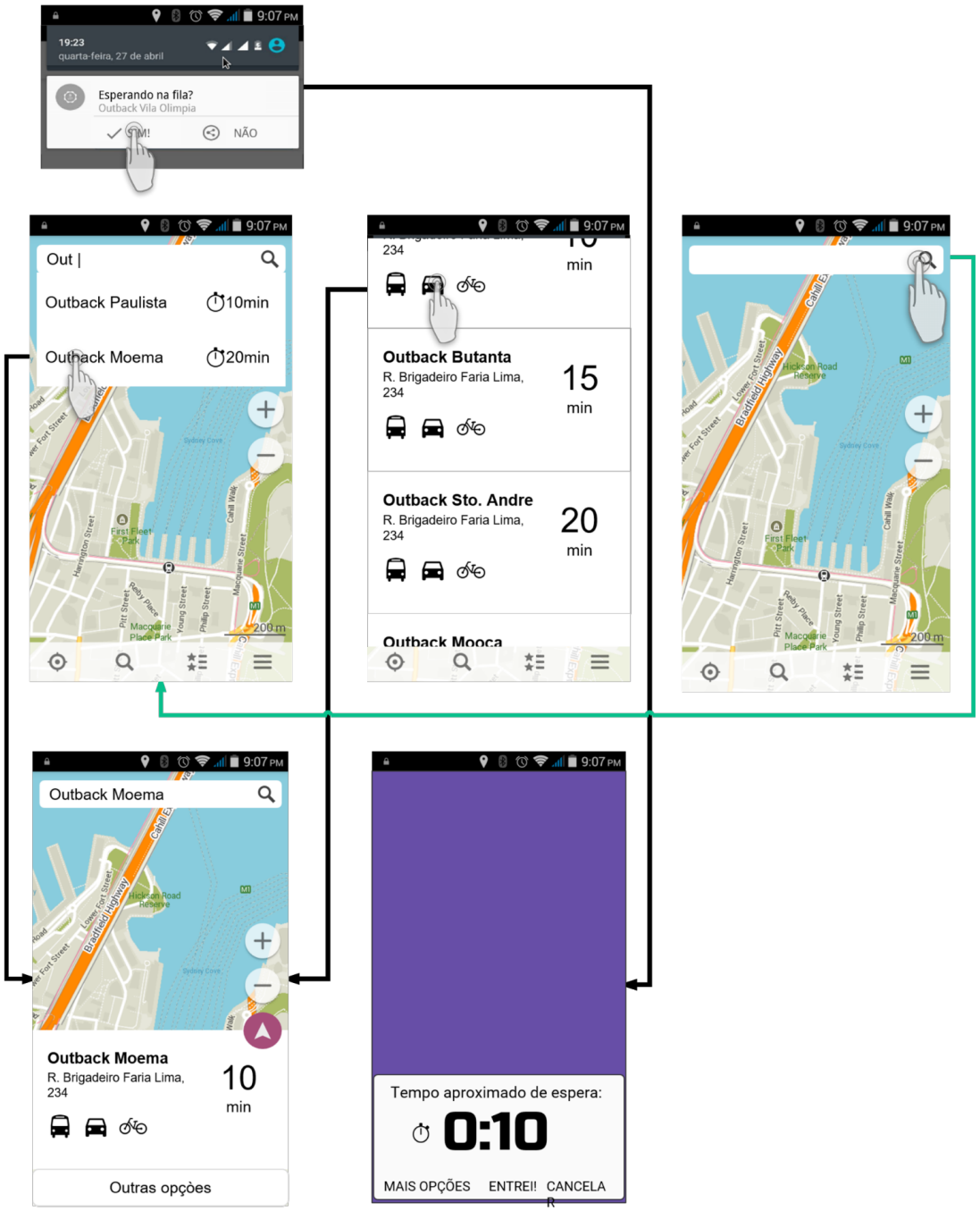



**Figura 4.14:** *Fluxo de telas do primeiro protótipo.*

Ao efetuar uma busca por algum estabelecimento, uma tela mostrando o tempo de espera aproximado da fila no local era exibida junto com botões que permitiam selecionar o meio de transporte, visualizar outras opções ou dar início a rota até o local desejado. Também havia uma notificação para quando a pessoa estivesse parada próxima a um restaurante cadastrado, que direcionaria o usuário a uma tela com um cronômetro do tempo de espera na fila.

Para testar essa visualização, escolhemos algumas pessoas e realizamos um teste de usabilidade[22], este teste é feito da seguinte forma:

Uma contextualização é feita para o voluntário e então uma tarefa é passada para ser completada. Os integrantes do grupo observam dificuldades enfrentadas pelo entrevistado e tomam notas. Esse teste é repetido algumas vezes e seu resultado é coletado para avaliação e modificação das funcionalidades.

**Figura 4.15:** *Tela de detalhes de um restaurante (segunda versão)*

Notou-se que algumas funcionalidades pareceram confusas e não intuitivas, por exemplo, não ficou claro se o tempo mostrado era de trajeto ou de espera na fila, nem qual era o meio de transporte utilizado.

Logo, esse protótipo foi reformulado com o intuito de melhorar a experiência do usuário com o nosso aplicativo.

Foram colocados, além do tempo de espera da fila, o tempo que o usuário levaria para chegar até o local selecionado (conforme o tipo de transporte escolhido) e a quantidade estimada de pessoas na fila. O tempo total (espera somado ao tempo de deslocamento) ganhou destaque entre as informações do estabelecimento e o botão que direciona o usuário ao trajeto ficou mais específico.

Também cogitou-se a possibilidade de adicionar login e um perfil de usuário, junto com uma lista de restaurantes favoritos.

O fluxo das telas deste novo protótipo ficou como mostra o esquema a seguir:

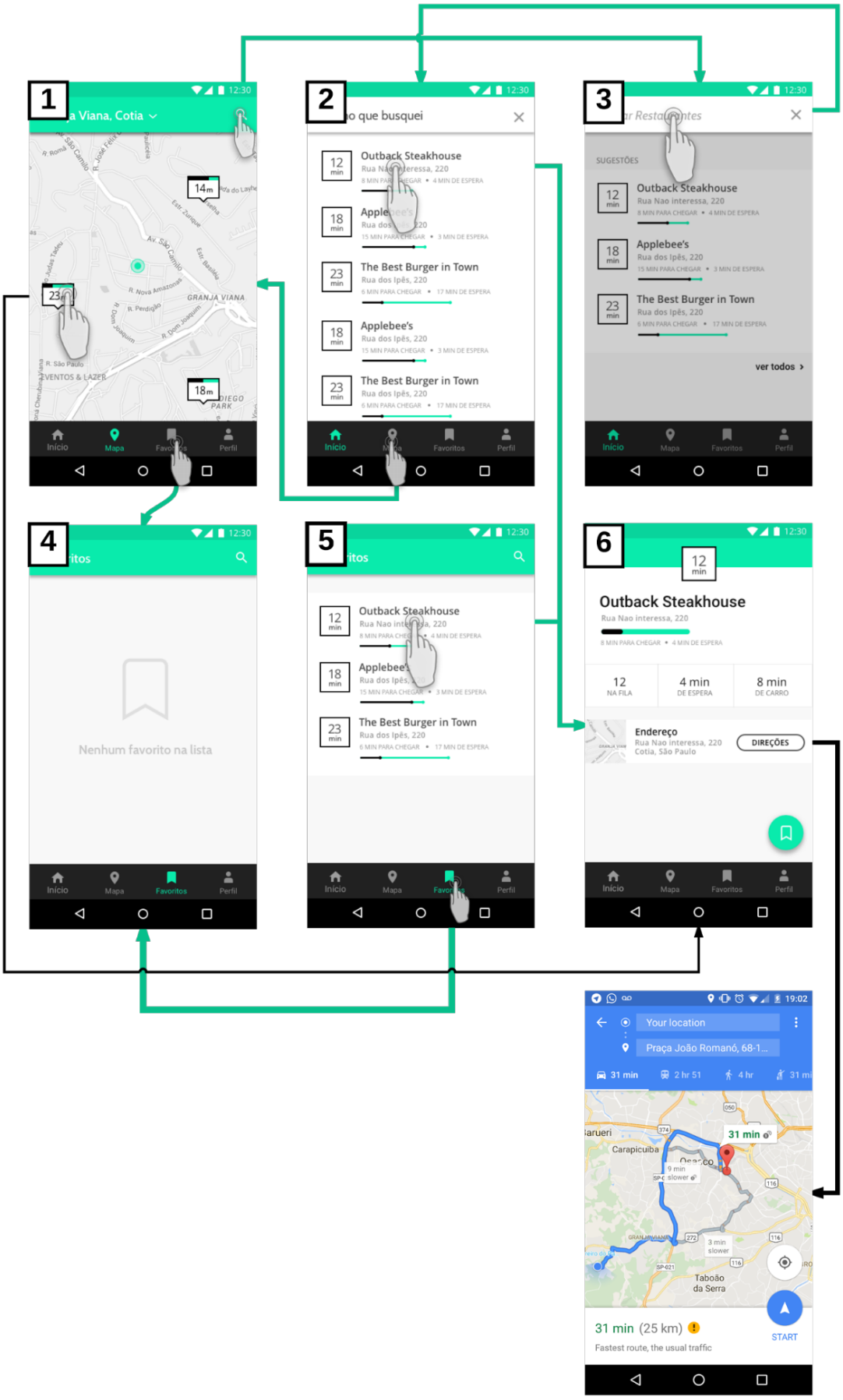



**Figura 4.16:** *Fluxo de telas do segundo protótipo.*

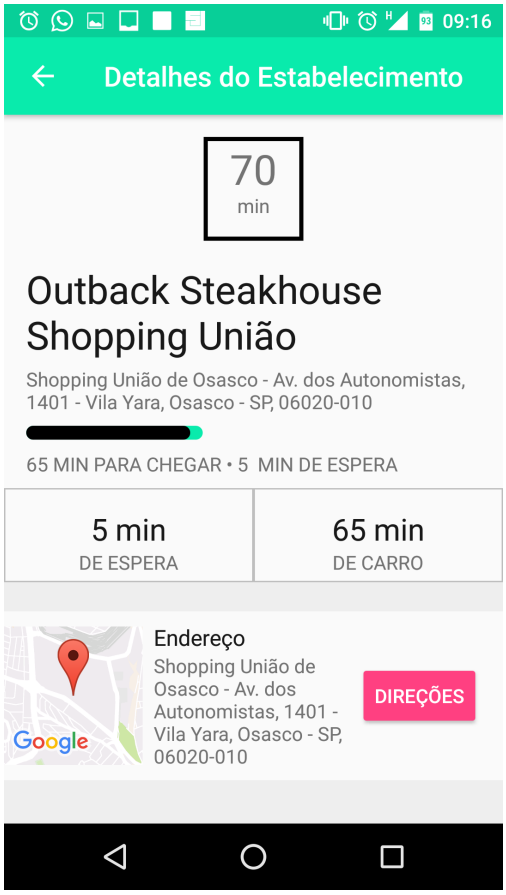



**Figura 4.17:** *Tela de detalhes de um restaurante (terceira versão)*

Após outras avaliações, mais algumas pequenas alterações foram realizadas na interface. Foi decidido que seria inviável estimar o número de pessoas na fila, então essa funcionalidade foi removida.

Notificações foram adicionadas para quando o usuário estiver parado na área do restaurante, para que ele possa contribuir com informações sobre a fila que pegou, além de um botão de contribuição ter sido inserido na tela inicial.

O fluxo de telas do aplicativo atual ficou da seguinte forma:

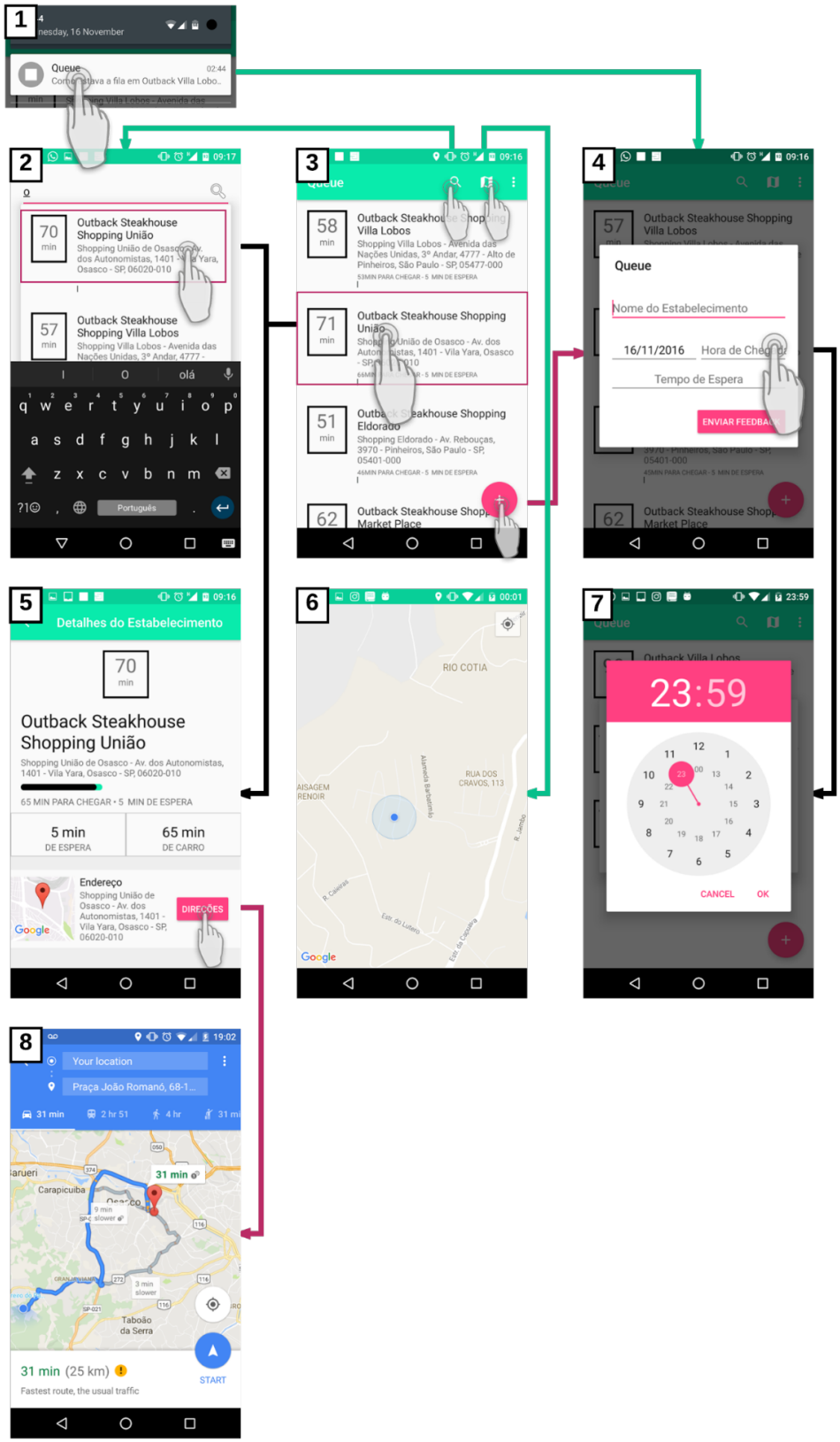



**Figura 4.18:** *Fluxo de telas do terceiro protótipo.*

## 4.5 Explicação das funcionalidades

Esta seção explica as funcionalidades do produto utilizando a figura 4.18.

A tela inicial (3) exibe uma lista de restaurantes incluindo seus respectivos tempos de trajeto e tempo de espera na fila. Ao clicar no botão "+" uma caixa de diálogo é aberta (4) onde é possível adicionar o tempo que o usuário esperou na fila. Primeiro, é escolhido o restaurante, depois, a data e hora que a pessoa chegou na fila e por fim o tempo de espera em minutos. Ao tocar em "Enviar *Feedback*", os dados são incluídos no banco de dados e atualizados para os outros usuários. Assim é apicado o conceito de *Crowdsourcing* no aplicativo

Ao tocar em um item da lista de restaurantes, será aberta a tela de detalhes de um restaurante (5), na qual é possível ver com mais clareza os tempos de trajeto e espera, o tempo total, o endereço do estabelecimento e um pequeno mapa centrado no local. Pressionando o botão "Direções", o usuário será redirecionado ao aplicativo de mapas de sua preferência, no caso do exemplo (8), o *Google Maps*, já traçando a rota da localização do usuário para o estabelecimento em questão.

Novamente na tela inicial, tocando no ícone de lupa, é possível realizar uma busca digitando o nome do estabelecimento na caixa de texto, a qual exibe sugestões dependendo do que foi digitado. Ao clicar em um dos resultados da busca o usuário é novamente direcionado para a tela de detalhes do restaurante (5).

Também é possível tocar no ícone de mapa na tela inicial, onde será mostrado um mapa centrado na localização do usuário.

Ao ficar parado por um tempo em um mesmo local, próximo a um estabelecimento cadastrado, o usuário é perguntado em uma notificação quanto tempo ele permaneceu na fila. Ao tocar na notificação, a caixa de diálogo para *feedback* do usuário será aberta (4), já com o restaurante preenchido, e após o *feedback*, a janela será fechada e os dados enviados ao banco de dados.

## 4.6 Implementação do banco de dados

O banco de dados do aplicativo foi modelado e populado em forma de árvore (Figura 4.19), de forma a facilitar o acesso aos dados necessários ao aplicativo. No primeiro nível estão os ids dos restaurantes. Cada id de restaurante guarda a localização (latitude, longitude) do restaurante, o nome, o endereço e um objeto fila. O objeto fila guarda todos os dias da semana (domingo a sábado), e cada dia da semana possui 24 campos, um para cada hora do dia. Cada hora possui o tempo médio de espera da fila naquela hora e um peso para esse tempo. O peso é um valor que aumenta de acordo com o número de *feedbacks* dados, para que o novo valor do tempo médio seja calculado de forma consistente.

Modelamos o banco de dados de uma forma que respeitasse uma hierarquia de dados.

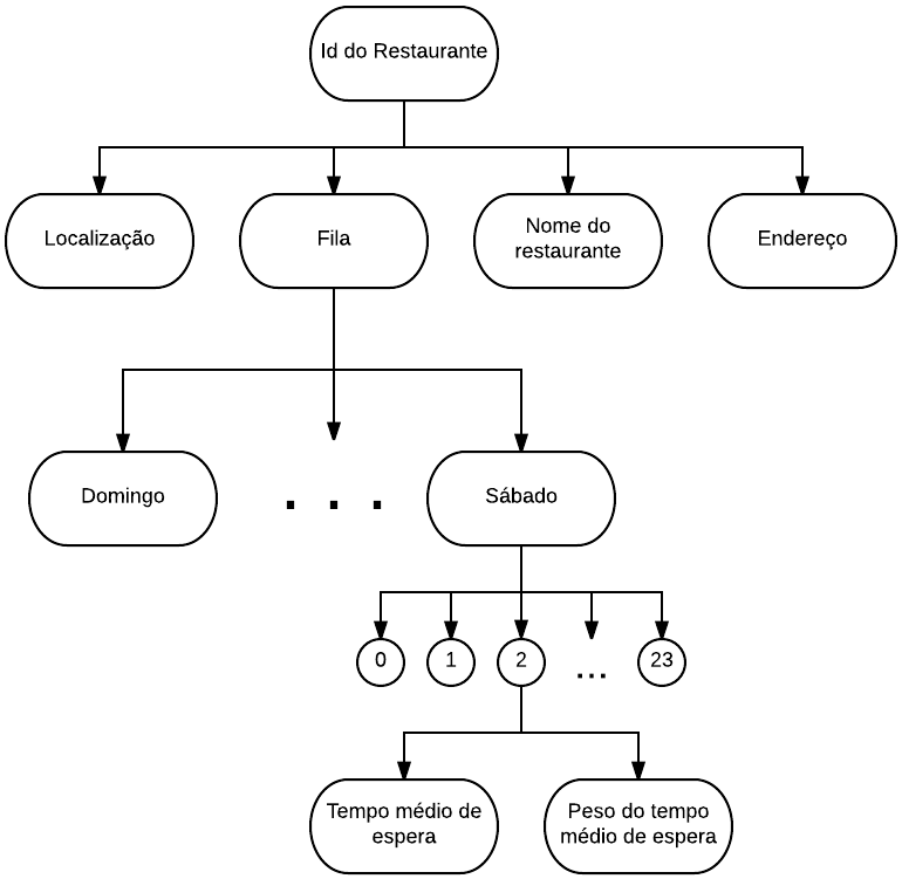



**Figura 4.19:** *Árvore que ilustra a organização de um restaurante no banco de dados*

Com o banco de dados já populado, o aplicativo faz consultas e atualizações de dados. Consultas são feitas ao criar a tela de listagem de restaurantes e de detalhes de um restaurante consultado. As atualizações são feitas quando o usuário manda o *feedback* de tempo de espera em um estabelecimento em determinado horário: o aplicativo calcula a nova média e atualiza o campo do respectivo horário.

# Capítulo 5

# Conclusão

Ao longo dos capítulos deste trabalho pôde-se acompanhar todo o processo de criação e desenvolvimento do aplicativo que, após finalizado, foi batizado com o nome *Queue*.

Começando com um problema para o qual o grupo almejava oferecer uma solução, acabou-se por adentrar-se um pouco no campo da Estatística e estudar diversas metodologias que poderiam ser úteis para fornecer um cálculo de previsão de espera: média por período, aplicação da teoria das filas e utilização de redes neurais.

Através de uma parceria com a empresa *GetIn*, obteve-se acesso a uma grande quantidade de dados e foi possível utilizá-los em análises feitas nas metodologias escolhidas, conseguindo assim resultados relevantes. Por meio destes foi feita a escolha de um dos métodos para o cálculo da previsão de filas: a média por período.

Além desses estudos, ao longo do ano o grupo aprendeu a desenvolver para a plataforma *Android*, sobre a qual nenhum dos integrantes tinha conhecimento prévio. Também foi possível aprender e utilizar a metodologia ágil com *Scrum* para guiar o processo de desenvolvimento.

Ao fim do ciclo de desenvolvimento, conseguiu-se entregar um aplicativo em funcionamento que fornece a previsão de espera, no instante da consulta, de estabelecimentos cadastrados no banco de dados armazenado na nuvem.

Na versão atual do aplicativo não foi implementado um sistema para o cadastro de novos estabelecimentos. Isso só é possível através da inserção direta no banco de dados através da interface fornecida pelo console do *Firebase*. Atualmente estão cadastrados 5 restaurantes, os quais foram visitados durante o desenvolvimento do projeto. Para que o cadastro de estabelecimentos escale, seria necessário a implementação de uma funcionalidade que permita ao usuário adicionar novos estabelecimentos a partir de locais disponíveis na API de locais da *Google*.

# Apêndice A

# Dados coletados

Outback Shopping União

| | | 9/5 MONDAY | 9/6 TUESDAY | 9/7 WEDNESDAY | 9/8 THURSDAY | 9/9 FRIDAY | 9/10 SATURDAY | 9/11 SUNDAY |
|---|---|---|---|---|---|---|---|---|
| 11:30 AM | 2 pessoas<br>3-6 pessoas<br>7+ pessoas | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 20 min<br>50 min<br>1h + | 20 min<br>50 min<br>1h + | 20 min<br>50 min<br>1h + |
| 12:00 PM | 2 pessoas<br>3-6 pessoas<br>7+ pessoas | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 20 min<br>50 min<br>1h + | 20 min<br>50 min<br>1h + | 20 min<br>50 min<br>1h + |
| 1:00 PM | 2 pessoas<br>3-6 pessoas<br>7+ pessoas | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 20 min<br>50 min<br>1h + | 20 min<br>50 min<br>1h + | 20 min<br>50 min<br>1h + |
| 2:00 PM | 2 pessoas<br>3-6 pessoas<br>7+ pessoas | 5 min | 5 min | 5 min | 5 min | 5 min | 5 min | 5 min |
| 3:00 PM | 2 pessoas<br>3-6 pessoas<br>7+ pessoas | 5 min | 5 min | 5 min | 5 min | 5 min | 5 min | 5 min |
| 4:00 PM | 2 pessoas<br>3-6 pessoas<br>7+ pessoas | 5 min | 5 min | 5 min | 5 min | 5 min | 5 min | 5 min |
| 5:00 PM | 2 pessoas<br>3-6 pessoas<br>7+ pessoas | 5 min | 5 min | 5 min | 5 min | 5 min | 5 min | 5 min |
| 6:00 PM | 2 pessoas<br>3-6 pessoas<br>7+ pessoas | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 5-10min<br>30 min<br>35 min | 20 min<br>50 min<br>1h + | 20 min<br>50 min<br>1h + | 20 min<br>50 min<br>1h + |

**Figura A.1:** *Dados de tempo de espera para grupos de tamanhos variados, para cada horário e em cada dia da semana, obtidos em entrevista em restaurantes Outback Steakhouse Shopping União no mês de outubro.*

# Referências Bibliográficas


[1] Backpropagation. http://mbp.sourceforge.net/. Accesso em Janeiro, 2017. 14

[2] Bitbucket. https://bitbucket.org/product. Accesso em Dezembro, 2016. 19

[3] Crowdsourcing. https://www.merriam-webster.com/dictionary/crowdsourcing. Accesso em Novembro, 2016. 21

[4] Designing a survey. http://www.sciencebuddies.org/science-fair-projects/project_ideas/Soc_survey.shtml. Accesso em Outubro, 2016. 4, 23

[5] Facebook. https://www.facebook.com/. Accesso em Novembro, 2016. 24

[6] Firebase. https://firebase.google.com/. Accesso em Agosto, 2016. 20

[7] Getin app. https://www.getinapp.com.br/#!/home/sao-paulo. Accesso em Novembro, 2016. 9

[8] Git, 1.1 primeiros passos - sobre controle de versão. https://git-scm.com/book/pt-br/v1/Primeiros-passos-Sobre-Controle-de-Vers. Accesso em Novembro, 2016. 19

[9] Google forms. https://docs.google.com/forms/. Accesso em Novembro, 2016. 23

[10] *Guia definitivo do project model canvas.* Project Builder. 3

[11] Json.org, introducing json. http://www.json.org/. Accesso em Novembro, 2016. 17

[12] Manifesto para desenvolvimento Ágil de software. http://agilemanifesto.org/iso/ptbr/manifesto.html. Accesso em Novembro, 2016. 5

[13] Nosql. https://en.wikipedia.org/wiki/NoSQL. Accesso em Novembro, 2016. 20

[14] Outback steakhouse. https://www.outback.com.br/home. Accesso em Novembro, 2016. 9

[15] Restful api development with lumen microframework. https://belitsoft.com/laravel-development-services/api-development. Accesso em Novembro, 2016. 17

[16] Scrum. https://www.atlassian.com/agile/scrum. Accesso em Novembro, 2016. 5

[17] Smartphone os market share, 2016 q2. http://www.idc.com/prodserv/smartphone-os-market-share.jsp. Accesso em Novembro, 2016. 15

[18] Smartphone os sales market share kantar worldpanel comtech. http://www.kantarworldpanel.com/smartphone-os-market-share/. Accesso em Novembro, 2016. 15

[19] Sourcetree. https://www.sourcetreeapp.com/. Accesso em Dezembro, 2016. 19

[20] Sql. https://pt.khanacademy.org/computing/computer-programming/sql. Accesso em Dezembro, 2016. 20

[21] Trello. https://trello.com. Accesso em Novembro, 2016. 6

[22] Usability test. https://en.wikipedia.org/wiki/Usability_testing. Accesso em Novembro, 2016. 36

[23] Versionone 2012 agile dev survey. https://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf. Accesso em Novembro, 2016. 5

[24] Waze. https://www.waze.com/. Accesso em Novembro, 2016. 1

[25] What is an open source. https://opensource.com/resources/what-open-source. Accesso em Novembro, 2016. 15

[26] Xml tutorial. http://www.w3schools.com/xml/. Accesso em Novembro, 2016. 17

[27] Paulo F. Cabral dos Anjos Jr. Aron Ifanger Maciel, Christopher Silva de Pádua. *Composição de carteiras de ações utilizando modelo de redes neurais artificiais e modelo de risco e retorno de Markowitz*. 2012. 14

[28] Google. Distance matrix api documentation. https://developers.google.com/maps/documentation/distance-matrix/. Accesso em Novembro, 2016. 17

[29] Alan Klement. Designing features using job stories. https://blog.intercom.com/using-job-stories-design-features-ui-ux/. Accesso em Novembro, 2016. 6

[30] Azmat Nafees. Queuing theory and its application: analysis of the sales checkout operation in ica supermarket. http://www.statistics.du.se/essays/D07E.Nafees.pdf. Accesso em Novembro, 2016. 12

[31] Eric Ries. *A Startup Enxuta*. Lua de papel, primeira edição edition, 2012. 3