

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DO RIO GRANDE DO SUL  
CAMPUS BENTO GONÇALVES

MARCELO MILANI

UM ESTUDO SOBRE O DESENVOLVIMENTO DE UM  
APLICATIVO PARA FABRICAÇÃO DE CERVEJA  
ARTESANAL USANDO O FRAMEWORK IONIC

Bento Gonçalves, Novembro de 2019.

MARCELO MILANI

UM ESTUDO SOBRE O DESENVOLVIMENTO DE UM  
APLICATIVO PARA FABRICAÇÃO DE CERVEJA  
ARTESANAL USANDO O FRAMEWORK IONIC

Trabalho de Conclusão de Curso apresentado junto ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Bento Gonçalves, como requisito parcial à obtenção de título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Maurício Covolan Rosito.

Bento Gonçalves, Novembro de 2019.

MARCELO MILANI

## UM ESTUDO SOBRE O DESENVOLVIMENTO DE UM APLICATIVO PARA FABRICAÇÃO DE CERVEJA ARTESANAL USANDO O FRAMEWORK IONIC

Trabalho de Conclusão de Curso apresentado junto ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Bento Gonçalves, como requisito parcial à obtenção de título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Maurício Covolan Rosito.

Aprovado em mês, ano.

---

Prof. Dr. Maurício Covolan Rosito – Orientador

---

Prof. Ivan Prá – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Bento Gonçalves

---

Prof. Dr. Jean da Rolt Joaquim – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Bento Gonçalves

## RESUMO

Com o crescimento exponencial do uso de dispositivos móveis como o *smartphone*, houve um aumento no mercado de desenvolvimento de aplicativos. Afinal, para todo tipo de necessidade, pode-se encontrar ou desenvolver algum tipo de aplicativo que possa auxiliar tarefas diárias, desde uma simples agenda ou calculadora, até sistemas que ajudam a cuidar da saúde do corpo. Atualmente, os sistemas operacionais mais utilizados nos *smartphones* são o *Android*, *IOs* e *Windows Phone*. Cada sistema operacional tem uma linguagem de programação nativa, ou seja, necessita que o *software* seja desenvolvido para aquela plataforma exclusiva. Isso acaba encarecendo muito o desenvolvimento quando pretende-se alcançar todas estas plataformas. A necessidade de atender a usuários de diferentes sistemas operacionais fez com que o desenvolvimento de softwares para dispositivos móveis passasse por uma revolução. Afinal, para se ter um aplicativo nativo que rode tanto em *Android* quanto *IOs*, seria preciso programá-lo duas vezes, em linguagens distintas. Os aplicativos híbridos, então, surgiram como uma solução extremamente eficiente para esse problema. Por meio de uma ferramenta de desenvolvimento específica, o programador pode criar um modelo único que seja compatível com ambas às plataformas. O *Framework Ionic* é uma poderosa ferramenta para desenvolvimento híbrido, lançado em 2013, e que é baseada em *AngularJS* e *Apache Cordova*. Outro mercado que está em plena expansão é o de cervejas artesanais caseiras. Este mercado foca na diferenciação da qualidade, do aroma e do sabor das cervejas, com produções em menor escala, quando comparado a indústria cervejeira. Pensando em aprimorar e padronizar as receitas de cerveja, o presente trabalho de conclusão pretende apresentar um estudo sobre o desenvolvimento um aplicativo híbrido para dispositivos móveis usando o *Framework Ionic*. Para tanto, uma revisão de trabalhos correlatos foi conduzida e, em seguida, um questionário eletrônico foi aplicado a equipes distintas de cervejeiros. Com base nas respostas dos entrevistados e conhecimentos de programação adquiridos ao longo do curso, foi desenvolvido um protótipo de aplicativo híbrido para dispositivos móveis para auxiliar na produção de cervejas artesanais caseiras. Este protótipo consiste em um aplicativo para cadastro de ingredientes e formulação de receitas, de forma a facilitar a vida do cervejeiro na hora de consultar seu catálogo de receitas e insumos. Após o protótipo desenvolvido ele foi apresentado a quatro equipes de cervejeiros distintas e aplicado novo questionário eletrônico. Este questionário identificou a faixa etária e experiência dos cervejeiros. Também foi possível identificar o sistema qual o sistema operacional mais utilizado por eles. Com base nos *feedbacks*, pode-se concluir que *Framework Ionic* é de fato uma poderosa e robusta ferramenta para desenvolvimento híbrido, gerando um aplicativo que atende às necessidades dos usuários. Com uma menor curva de aprendizado para o desenvolvimento, pôde-se desenvolver um aplicativo para auxiliar na produção de cerveja artesanal, que roda em diferentes plataformas de forma similar à nativa, sem a necessidade de nova programação.

**Palavras-chaves:** Aplicativo Híbrido, *Framework Ionic*, *Apache Cordova*, *AngularJS*, *Json-Server*, Cerveja Artesanal.

## **ABSTRACT**

With the exponential growth in the use of mobile devices such as the smartphone, there has been an increase in the app development market. After all, for every need, you can find or develop some kind of application that can help you with daily tasks, from a simple diary or calculator, to systems that help take care of your body health. Currently, the operating systems most used in smartphones are Android, IOs and Windows Phone. Each operating system has a native programming language, meaning that the software must be developed for that unique platform. This ends up making development very costly when it is intended to reach all of these platforms. The need to cater to users of different operating systems has made mobile software development a revolution. After all, to have a native application that runs on both Android and IOs, you would have to program it twice, in different languages. Hybrid applications then emerged as an extremely efficient solution to this problem. Through a specific development tool, the programmer can create a unique model that is compatible with both platforms. The Ionic Framework is a powerful hybrid development tool released in 2013 based on AngularJS and Apache Cordova. Another booming market is homemade craft beer. This market focuses on differentiating the quality, aroma and flavor of beers, with smaller scale productions compared to the brewing industry. Thinking about improving and standardizing beer recipes, this concluding paper aims to present a study on developing a hybrid mobile app using the Ionic Framework. To this end, a review of related works was conducted and then an electronic questionnaire was applied to different brewer teams. Based on respondents' responses and programming skills gained throughout the course, a prototype hybrid mobile app was developed to assist in the production of homemade craft beers. This prototype consists of an application for ingredient registration and recipe formulation, in order to make the brewer's life easier when consulting his catalog of recipes and inputs. After the prototype developed it was presented to four teams of brewers and applied a new electronic questionnaire. This questionnaire identified the brewers' age and experience. It was also possible to identify the system which operating system most used by them. Based on the feedback, it can be concluded that Ionic Framework is indeed a powerful and robust tool for hybrid development, generating an application that meets the needs of users. With a shorter learning curve for development, an application could be developed to assist in the production of craft beer, which runs on different platforms in a similar way to the native one, without the need for new programming.

**Keywords:** Hybrid Application, Ionic Framework, Apache Cordova, AngularJS, Json-Server, Craft Beer.

## LISTA DE FIGURAS

Figura 1: Imagem gráfico uso Android entre 2012 e 2017 .....	12
Figura 2: Imagem Arquitetura Aplicativo mobile desenvolvido com Ionic .....	20
Figura 3: Imagem Arquitetura Aplicação Cordova .....	21
Figura 4: Diagrama de Casos de Uso.....	33
Figura 5: Diagrama de Classes .....	36
Figura 6: Exemplo de função para criar item no banco de dados JSON acessado pela API REST .....	37
Figura 7: Banco de dados JSON acessado pela API REST .....	38
Figura 8: Tela Inicial do Aplicativo .....	39
Figura 9: Tela Receitas do Aplicativo .....	39
Figura 10: Tela Cadastrar Lúpulo .....	40
Figura 11: Tela Cadastrar Lúpulo .....	40
Figura 12: Tela Listar Lúpulo, detalhe Botão Apagar Item.....	41
Figura 13: Tela Listar Lúpulo, detalhe Botão Editar Item.....	41
Figura 14: Tela Detalhes do Lúpulo.....	42
Figura 15: Tela Editar Lúpulo.....	42
Figura 16: Tela Cadastrar Malte. ....	43
Figura 17: Tela Listar Malte. ....	43
Figura 18: Tela Detalhe Malte.....	44
Figura 19: Tela Editar Malte.....	44
Figura 20: Cadastrar Levedura. ....	45
Figura 21: Tela Listar Levedura.....	45
Figura 22: Tela Editar Levedura. ....	45
Figura 23: Cadastrar Água.....	47
Figura 24: Tela Listar Água.....	47
Figura 25: Tela Detalhes Água. ....	48
Figura 26: Tela Editar Água.....	48
Figura 27: Tela Cadastrar tipo de Cerveja.....	49
Figura 28: Tela Listar Tipos de Cerveja .....	49
Figura 29: Tela Detalhes Tipo de Cerveja. ....	50

Figura 30: Tela Editar Tipos de Cerveja. ....	50
Figura 31: Tela Cadastro Receita, aba Características. ....	51
Figura 32: Tela Cadastro Receita, aba Característica, <i>popup</i> Tipo de Cerveja. .....	51
Figura 33: Tela Cadastro Receita, aba Data Início. ....	52
Figura 34: Tela Cadastro Receita, aba Características. ....	52
Figura 35: Tela Cadastro Receita, aba Ingredientes. ....	53
Figura 36: Tela Cadastro Receita, aba Ingredientes, <i>popup</i> origem da água. ....	53
Figura 37: Tela Cadastro Receita, aba Água. ....	54
Figura 38: Tela Cadastro Receita, aba Ingredientes, Adicionar Água. ....	54
Figura 39: Tela Cadastro Receita, aba Ingredientes, Malte. ....	55
Figura 40: Tela Cadastro Receita, aba Ingredientes, Lúpulo. ....	55
Figura 41: Tela Cadastro Receita, aba Ingredientes, Uso do Lúpulo. ....	55
Figura 42: Tela Listar Receitas. ....	55
Figura 43: Tela Detalhe Receita ....	56
Figura 44: Gráfico Sistema operacional mais utilizado pelos cervejeiros. ....	58
Figura 45: Índice de aprovação do aplicativo. ....	59

## **LISTA DE QUADROS**

Quadro 1: Cronograma do Trabalho de Conclusão de Curso.....	17
---	----



## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BJCP	Beer Judge Certification Program
CO2	Dióxido de Carbono
CRUD	Create, Read, Update, Delete
CSS3	Cascading Style Sheets
EBC	European Brewing Convention
HTML5	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBOPE	Instituto Brasileiro de Opinião Pública e Estatística
JSON	JavaScript Object Notation
MIT	Massachusetts Institute of Technology
MVC	Model-View-Controller
OSI	Open Source Initiative
PC	Personal Computer
pH	Potencial Hidrogeniônico
POO	Programação Orientada a Objetos
PPM	Partes Por Milhão
REST	REpresentational State Transfer
SEBRAE	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
SICOBEBE	Sindicato de Controle de Produção de Bebidas da Receita Federal
UML	Unified Modeling Language
URL	Uniform Resource Locator

## SUMÁRIO

1. CONTEXTUALIZAÇÃO .....	11
1.1 PROBLEMA .....	14
1.2 HIPÓTESE .....	15
1.3 OBJETIVOS .....	15
1.3.1 Geral .....	15
1.3.2 Específicos .....	15
1.4 JUSTIFICATIVA .....	16
1.5 CRONOGRAMA DO TRABALHO DE CONCLUSÃO.....	17
1.6 ESTRUTURA DO TEXTO .....	17
2. REVISÃO BIBLIOGRÁFICA .....	19
2.1 FRAMEWORK IONIC.....	19
2.2 FABRICAÇÃO DE CERVEJA ARTESANAL .....	22
2.2.1 História da Cerveja .....	22
2.2.2 O Processo de Produção da Cerveja Artesanal .....	23
2.2.2.1 Malte .....	23
2.2.2.2 Leveduras .....	24
2.2.2.3 Lúpulo .....	25
3. METODOLOGIA .....	28
4. DESENVOLVIMENTO DA PESQUISA.....	29
4.1 COLETA DE DADOS .....	29
4.2 ANÁLISE DAS RESPOSTAS .....	30
5. DESENVOLVIMENTO DO PROTÓTIPO.....	32
5.1 ESPECIFICAÇÃO .....	32
5.1.1 Manter Lúpulo.....	34

5.1.2 Manter Malte.....	34
5.1.3 Manter Levedura.....	34
5.1.4 Manter Água .....	34
5.1.5 Manter Tipos de Cerveja .....	34
5.1.6 Manter Receita .....	35
5.2 DESENVOLVIMENTO .....	38
5.2.1 Tela Inicial.....	38
5.2.2 Lúpulo.....	39
5.2.3 Malte.....	42
5.2.4 Levedura.....	44
5.2.5 Água .....	46
5.2.6 Tipos de Cerveja.....	48
5.2.7 Receitas.....	50
5.2.7.1 Aba Características .....	50
5.2.7.2 Ingredientes .....	52
6. ANÁLISE DO PROTÓTIPO .....	57
7. CONSIDERAÇÕES FINAIS.....	60
8. REFERÊNCIAS BIBLIOGRÁFICAS .....	61
9. APÊNDICE A.....	65
10. APÊNDICE B.....	67
11. APÊNDICE C.....	69
12. APÊNDICE D.....	72

## 1. CONTEXTUALIZAÇÃO

O mercado para o desenvolvimento de aplicativos para dispositivos móveis tem permanecido bastante aquecido nos últimos tempos. Segundo Morimoto (2009), dispositivo móvel é todo aquele equipamento, que pode ser levado a qualquer lugar. Talvez esta seja um dos principais atrativos deste equipamento, sua capacidade de ser utilizado em diferentes lugares e situações. Atualmente, existem diversas plataformas para dispositivos móveis, tais como: *Android*, *iOS*, *Windows Phone*, *Firefox OS*, *BlackBerry*, *Ubuntu Touch*, *Fire OS*, entre outros. Entretanto, deve-se salientar que dois sistemas operacionais para dispositivos móveis tem obtido maior destaque no mercado, a saber: *Android* e *iOS*. O *Android* é um sistema operacional baseado no núcleo *Linux* e atualmente desenvolvido pela empresa de tecnologia *Google*. O *iOS*, por sua vez, é um sistema operacional móvel da *Apple*, desenvolvido originalmente para o *iPhone*. Segundo pesquisa realizada em 2017 pelo *IDC* (2017) o sistema operacional *Android* domina cerca de 85% do mercado de dispositivos móveis. O *iOS*, por sua vez, atende quase 15%, sobrando aproximadamente 0,2% para outros sistemas operacionais.

Para se ter uma noção do crescimento deste mercado, segundo dados do *Statcounter* (2017), o *Android* se tornou em 2017 o sistema operacional mais usado do mundo. Assim, a plataforma do *Google* supera o *Windows* pela primeira vez na história, indicando o poder do *smartphone* em relação ao *PC* (*Personal Computer*). A vantagem é pequena: o *Android* tem 37,93% de usuários e a plataforma da *Microsoft* alcança 37,91%, como pode se conferir na imagem abaixo (ZURIARRAIN, 2017).

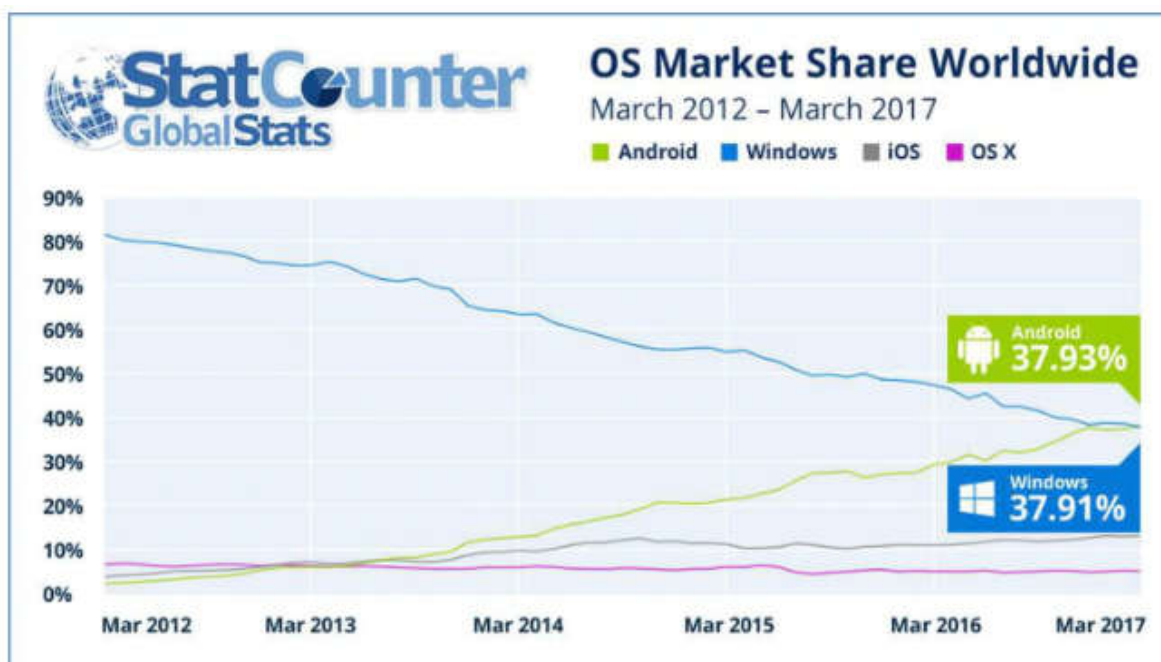


Figura 1: Imagem gráfico uso Android entre 2012 e 2017

Fonte: El País, 2018<sup>1</sup>.

Observa-se que, cada vez mais, não se pode mais viver sem o *smartphone* e que já não se pode dizer o mesmo de um computador. O crescimento das plataformas móveis em número de usuários está aumentando a cada ano.

Considerando este expressivo crescimento das tecnologias móveis (tais como *tablets* e *smartphones*) nestes últimos anos, observa-se que a diversidade de tais equipamentos aumentou muito. Cada plataforma possui diferentes formas e linguagens para desenvolvimento de aplicações nativas, como por exemplo, o *Android* utiliza *Java* ou *Kotlin* como linguagens de programação e o *iOS* utiliza *Objective C* ou *Swift*. As aplicações nativas são aquelas que são desenvolvidas especialmente para uma determinada plataforma, utilizando para isso as ferramentas e tecnologias especializadas, tais como: a linguagem de desenvolvimento, ambiente de desenvolvimento e o emulador. Assim, neste cenário, cabe à equipe de desenvolvimento optar pela plataforma e pela quantidade de sistemas diferentes que a aplicação deve dar suporte (PREZOTTO e BONIATTI, 2014).

<sup>1</sup> Disponível em: <<https://gs.statcounter.com/os-market-share#monthly-201810-201910>> Acessado em 01 nov. 2019.

Percebe-se, desta forma, que o desenvolvimento de uma aplicação nativa que venha a atender a todos os tipos de dispositivos móveis não é uma tarefa fácil, pois são necessários conhecimentos específicos de cada plataforma e sistema operacional. Uma alternativa seria o desenvolvimento de aplicações híbridas para dispositivos móveis. Por meio de uma ferramenta de desenvolvimento específica, um programador possa criar um aplicativo único e que seja compatível com diversas plataformas, (GONÇALVES, 2017). Assim partindo do princípio de utilizar uma mesma aplicação para diferentes plataformas, normalmente as aplicações híbridas são desenvolvidas em linguagens *Web HTML5 (Hypertext Markup Language)*, *CSS3 (Cascading Style Sheets)* que são interpretadas pelo *browser* nativo do sistema. Estas aplicações ficam instaladas no dispositivo móvel e, quando são acionadas, chamam o navegador, onde serão executadas independentemente de estarem ou não conectadas à *internet* (PREZOTTO e BONIATTI, 2014).

Neste trabalho, será considerado o desenvolvimento *mobile* híbrido com *Ionic*. *Ionic* é um *framework open source* criado no final de 2013 que visa à criação de aplicações híbridas para dispositivos móveis. O *Ionic* foi construído considerando as tecnologias *AngularJS* e *Apache Cordova*. Dessa forma, ele fornece ferramentas e serviços para desenvolver aplicativos móveis híbridos usando tecnologias da *Web* como *CSS*, *HTML5* e *JavaScript* (PORTALGSTI, 2018?). Ele tem como objetivo prover para o desenvolvedor uma gama de componentes pré-prontos de alta qualidade e desempenho. Sendo assim diversos segmentos do mercado poderiam se beneficiar do desenvolvimento de aplicativos *mobile* com este *framework*, tais como os cervejeiros artesanais caseiros.

O Brasil é um dos maiores consumidores de cerveja do mundo. Segundo o SEBRAE (Serviço Brasileiro de Apoio às Micro e Pequenas Empresas) (2015), as microcervejarias estão ganhando espaço no mercado nacional. Ainda conforme o SEBRAE, uma pesquisa realizada pelo Ibope (Instituto Brasileiro de Opinião Pública e Estatística) em 2013 revelou que a cerveja é a bebida preferida de dois terços dos brasileiros para comemorações. As cervejas artesanais, por sua vez, cada vez mais vêm conquistando o mercado. Atualmente, muitos são os bares que reservam espaços específicos para este tipo de cerveja. As cervejas

artesanais advêm de um processo artesanal mais cuidadoso em sua produção, enquanto as cervejas industrializadas tem por objetivo a produção em grande escala com grandes maquinários (LARA, 2018). Nota-se, recentemente, o crescimento de cervejeiros artesanais caseiros no Brasil. Muitas pessoas começaram a fabricar suas próprias cervejas em casa. Outras começaram a comprar as cervejas artesanais, comumente fabricadas por micro cervejarias.

Considerando o cenário apresentado, o presente trabalho de conclusão pretende realizar um estudo sobre o desenvolvimento *mobile* híbrido com *Ionic* considerando o cenário de fabricação de cervejas artesanais.

## 1.1 PROBLEMA

Com o expressivo crescimento dos dispositivos móveis, suas variedades de modelos, fabricantes e sistemas operacionais aumentaram muito. Havendo a necessidade de se disponibilizar um *software* que atenda todas (ou mais de uma plataforma), faz-se necessário desenvolver uma aplicação específica para cada uma delas. Tal característica eleva muito os custos e o tempo do desenvolvimento. Neste contexto surgiram as aplicações híbridas, que permitem que um mesmo aplicativo seja executado em diferentes plataformas. O *framework Ionic* tem se destacado no mercado pela sua facilidade de aprendizado e pelos seus componentes de desenvolvimento.

Este trabalho pretende demonstrar algumas das principais características da programação híbrida de aplicativos móveis com *Ionic*, destacando seus pontos fortes e fracos. Para melhor compreender o funcionamento deste *framework* será desenvolvido um estudo de caso com cervejeiros artesanais. Neste estudo, um aplicativo será desenvolvido que os auxilie na produção de cervejas artesanais.

Neste contexto, o problema de pesquisa é: De que forma um aplicativo móvel deve ser projetado e implementado utilizando o *framework Ionic* de forma a auxiliar no registro das informações necessárias para o desenvolvimento de uma cerveja artesanal?

## 1.2 HIPÓTESE

Para responder à pergunta "De que forma um aplicativo móvel deve ser projetado e implementado utilizando o *framework Ionic* de forma a auxiliar no registro das informações necessárias para o desenvolvimento de uma cerveja artesanal?", definiu-se a seguinte hipótese principal: "É possível que um aplicativo móvel desenvolvido com o *framework Ionic*, considerando as características do desenvolvimento *mobile* híbrido, auxilie no processo de produção de cerveja artesanal".

A verificação desta hipótese foi realizada em duas etapas:

- Análise sistemática: Consiste na revisão de artigos e trabalhos relacionados ao desenvolvimento *mobile* híbrido.
- Análise qualitativa: Compreende a coleta de dados junto a cervejeiros artesanais, realizada por meio de um questionário. Os dados obtidos foram analisados qualitativamente, de modo a fornecer informações relevantes para a especificação e desenvolvimento de um protótipo.

## 1.3 OBJETIVOS

### 1.3.1 Geral

Investigar como o desenvolvimento de um aplicativo móvel híbrido pode auxiliar cervejeiros caseiros no processo de produção de cervejas artesanais.

### 1.3.2 Específicos

- Pesquisar sobre os conceitos relacionados ao desenvolvimento de aplicativos híbridos com o *Ionic* por intermédio de revisão na literatura;
- Pesquisar sobre os conceitos relacionados à produção de cervejas artesanais por intermédio de revisão na literatura;



- Verificar através de um *Survey*<sup>2</sup> com cervejeiros caseiros como um aplicativo *mobile* pode auxiliar na produção de cervejas artesanais.
- Implementar um protótipo, conforme as características identificadas na pesquisa realizada com os cervejeiros caseiros;
- Analisar e avaliar a proposta de solução por meio do protótipo desenvolvido.

#### 1.4 JUSTIFICATIVA

Na última década, a produção de cerveja no Brasil cresceu cerca de 64%, saltando de 8,2 bilhões para 13,4 bilhões de litros anuais, segundo dados do Sindicato de Controle de Produção de Bebidas da Receita Federal (Sicobe). Segundo o SEBRAE (2015), as cervejarias ganham espaço no mercado nacional, de modo que as cervejas especiais representavam 8% do mercado nacional da bebida em 2012 e encerraram 2014 com uma participação de 11%. Ainda, conforme o SEBRAE (apud Sindicato Nacional da Indústria da Cerveja) existem cerca de 300 microcervejarias no país. A projeção é de que essa cota suba para 20% em 2020.

No entanto, as cervejas artesanais sofrem com a falta de escala para competir com as cervejas de massa. Além do volume menor, as bebidas especiais costumam trabalhar com matérias-primas mais caras, muitas vezes importadas, o que faz com que o preço final seja mais de cinco vezes o cobrado pelas tradicionais. Para inovar nesse segmento, os fabricantes de cervejas artesanais priorizam a qualidade dos ingredientes e investem em tecnologia, propaganda e insumos locais.

Ao tratar dos principais problemas e sugestões sobre o desenvolvimento de aplicativos móveis híbridos, esta pesquisa se qualifica como Trabalho de Conclusão de Curso, uma vez que aplica técnicas de Desenvolvimento e

---

<sup>2</sup> Pesquisa direcionada a um grupo do qual se deseja obter informações, tais como características ou opiniões.

Programação Orientada a Objetos, Banco de Dados e de Programação para Dispositivos Móveis, as quais foram adquiridas ao longo do curso de graduação.

### 1.5 CRONOGRAMA DO TRABALHO DE CONCLUSÃO

O cronograma do Trabalho de Conclusão de Curso foi definido de acordo com as atividades identificadas no Capítulo 3 METODOLOGIA. As etapas e sua respectiva realização são apresentadas no Quadro 1:

**Quadro 1: Cronograma do Trabalho de Conclusão de Curso.**

Descrição da etapa	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
1. Delimitação do Tema	X	X							
2. Definição do Problema		X	X						
3. Revisão de artigos e trabalhos correlatos		X	X	X					
4. Estudo do <i>framework Ionic</i>	X	X	X	X	X				
4. Aplicação do Questionário com os cervejeiros						X			
5. Análise das Respostas do Questionário						X			
6. Análise e Desenvolvimento do Protótipo						X	X	X	X
7. Avaliação do Protótipo									X
8. Escrita do TCC	X	X	X	X	X	X	X	X	X

Fonte: Autor, 2019.

## 1.6 ESTRUTURA DO TEXTO

Os tópicos deste Trabalho de Conclusão de Curso foram organizados de acordo com a cronologia das etapas realizadas. A seção "Contextualização" apresenta uma discussão inicial a respeito do problema de pesquisa, incluindo também um resumo da solução proposta, hipóteses verificadas, objetivos, justificativa e cronograma do trabalho.

Na sequência, a "Revisão Bibliográfica" aborda conceitos vinculados à realização da pesquisa sobre *Framework Ionic*, sistemas operacionais para dispositivos móveis, desenvolvimento de sistemas nativos e sistemas híbridos. A seção introduz ainda os estudos relacionados que foram identificados neste período.

A metodologia adotada é descrita na seção 3. Neste capítulo são apresentadas definições sobre o tipo da pesquisa e os processos que orientaram sua construção.

A seção 4 apresenta o modelo de levantamento das necessidades observadas pelos cervejeiros caseiros. Tal análise foi baseada no uso de um questionário eletrônico.

A seção 5 apresenta o processo de análise e especificação do protótipo com base nos requisitos identificados a partir do questionário aplicado aos cervejeiros assim como da revisão bibliográfica. A subseção "Especificação" traz detalhes referentes ao planejamento da ferramenta, enquanto a subseção "Desenvolvimento" apresenta os resultados obtidos após o processo de desenvolvimento.

A seção 6 traz a verificação dos pontos altos e baixos do desenvolvimento do protótipo utilizando o *Framework Ionic* e de como o protótipo desenvolvido pode afetar a maneira como os cervejeiros produzem suas cervejas artesanais. Tal análise foi baseada em uma entrevista com quatro grupos de cervejeiros.

A seção 7 traz as considerações finais e conclusões acerca do trabalho, além de perspectivas e possibilidades para novos estudos.

## 2. REVISÃO BIBLIOGRÁFICA

O desenvolvimento deste trabalho está vinculado a conceitos e termos relacionados ao desenvolvimento de aplicativos móveis e à produção de cervejas artesanais. A seção a seguir apresenta definições e discussões acerca desses tópicos.

### 2.1 FRAMEWORK IONIC

O *Ionic Framework* é um conjunto de ferramentas de interface de usuário de código aberto para criação e desenvolvimento de aplicativos móveis de alta qualidade e desempenho, usando tecnologias da *Web: HTML, CSS, e Javascript*. (IONICFRAMEWORK, 2019). Ele foi lançado sob a licença permitida do MIT (*Massachusetts Institute of Technology*) que é responsável pelo projeto *OSI (Open Source Initiative)*.

A *OSI* é uma corporação sem fins lucrativos com escopo global formado para educar e defender os benefícios do código aberto e para construir pontes entre os diferentes grupos constituintes da comunidade de código aberto.

O código aberto permite um método de desenvolvimento de *software* que aproveita o poder da revisão por pares distribuída e a transparência do processo. A promessa do código aberto é de ser de melhor qualidade, melhor confiabilidade, maior flexibilidade, menor custo e o fim do aprisionamento predatório do fornecedor. Uma das principais atividades é manter a definição de Código Aberto para o bem da comunidade. A marca registrada e o programa de licença aprovada da *OSI* criam umnexo de confiança em torno do qual, desenvolvedores, usuários, empresas e governos podem organizar a cooperação de código aberto.

O Ionic é um *framework* para desenvolvimento de aplicativos móveis multiplataforma (DEV MEDIA, 2018). Desta forma, ele possibilita a implementação do aplicativo utilizando tecnologias comumente empregadas na construção do *Front-end* de soluções *web*: *HTML*, *CSS* e *JavaScript*. No entanto, como diferencial em relação ao *framework* que adota como base, o *Apache Cordova*, traz recursos que simplificam ainda mais o desenvolvimento e dão ao aplicativo um aspecto mais profissional. Além do *Cordova* ele traz consigo outra linguagem e *framework*, *TypeScript* e *Angular*, conforme observado na Figura 2.

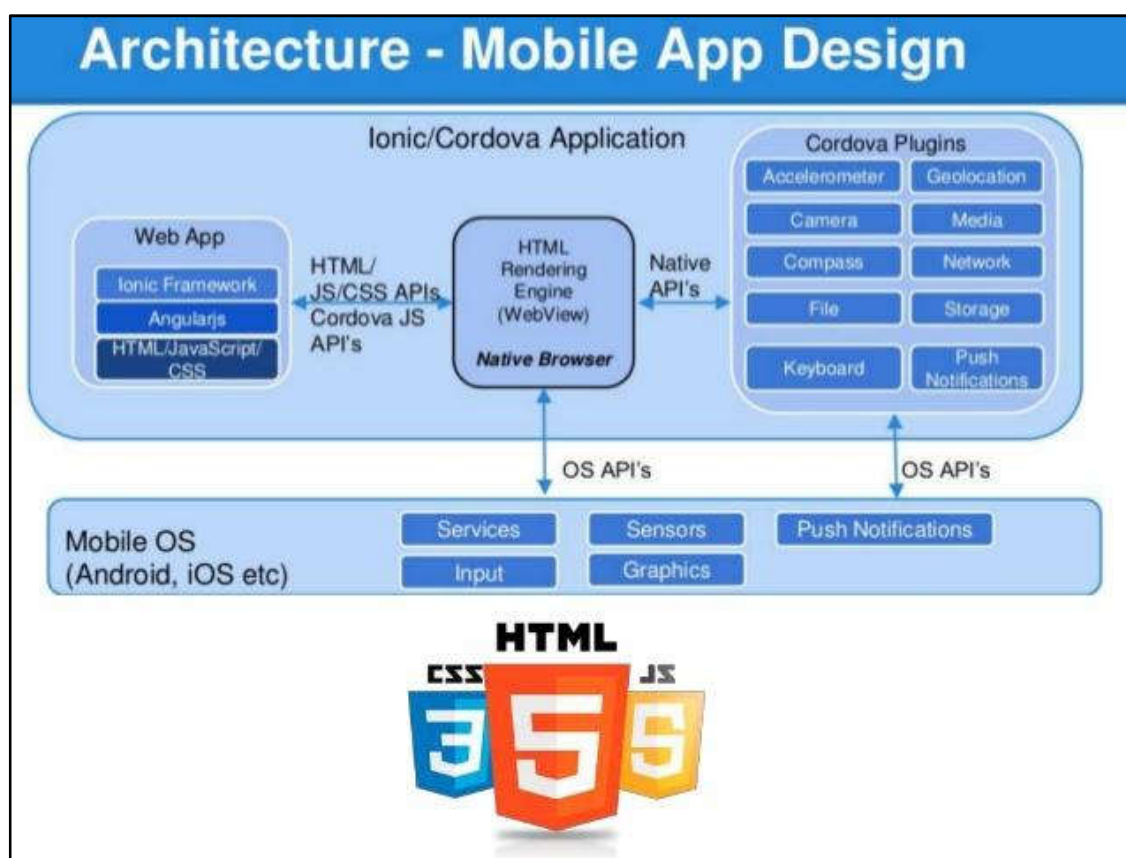


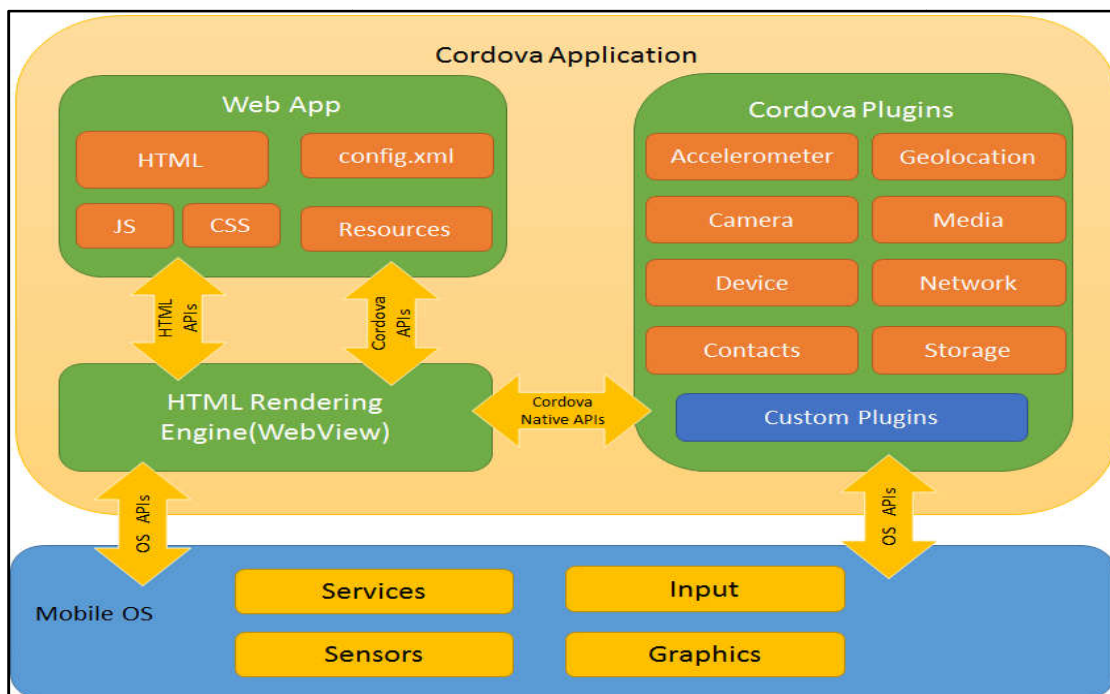
Figura 2: Imagem Arquitetura Aplicativo mobile desenvolvido com Ionic  
Fonte: Site slideshare.net, 2019<sup>3</sup>

O *Apache Cordova* é uma estrutura de desenvolvimento móvel de código aberto. Da mesma forma que o *Ionic*, ele permite que você use tecnologias padrão da *Web* - *HTML5*, *CSS3* e *JavaScript* para o desenvolvimento de plataforma cruzada. Os aplicativos são executados em invólucros direcionados a cada plataforma e dependem de ligações de *API* (*Application Programming Interface*)

<sup>3</sup> Disponível em: <<https://pt.slideshare.net/AlexMontanha/ionic-labs>> acessado 15 out. 2019.

compatíveis com os padrões para acessar os recursos de cada dispositivo, como sensores, dados, status da rede e etc. Cordova (2019). Assim, o *Apache Cordova* é responsável por misturar componentes de aplicativos nativos com um *WebView* (janela especial do navegador) que possa acessar *APIs* no nível do dispositivo ou se desejar desenvolver uma interface de *plug-in* entre componentes nativos e *WebView*, por exemplo, usando recursos do dispositivo, como câmera, acelerômetro ou sensor biométrico.

Quando se cria uma aplicação com o *Ionic*, por padrão não são adicionados *plugins* do *Cordova* para que o aplicativo não fique sobrecarregado com itens que não serão utilizados. O desenvolvedor pode adicionar *plugins* em sua aplicação de acordo com a necessidade através de alguns comandos simples e uma conexão com a *internet*. Em síntese, pode-se dizer que o *Ionic Framework* trabalha sobre o *Cordova*, ou seja, o *Cordova* é uma camada inferior, responsável pelo acesso aos recursos nativos dos dispositivos e pelo empacotamento da aplicação para as diferentes plataformas. O *Ionic* é a camada superior, focada na interface gráfica de usuário e regras de negócios da aplicação, conforme ilustrado da Figura 2.



**Figura 3: Imagem Arquitetura Aplicação Cordova**

Fonte :Site Cordova apache, 2019<sup>4</sup>

<sup>4</sup> Disponível em: <<https://cordova.apache.org/docs/en/latest/guide/overview/>>. Acessado em 20 out. 2019.

O *TypeScript* é uma linguagem para *JavaScript* em escala de aplicativo. O *TypeScript* adiciona tipos opcionais ao *JavaScript* que suportam ferramentas para aplicativos *JavaScript* de grande escala para qualquer navegador, para qualquer *host*, em qualquer sistema operacional. O *TypeScript* compila para *JavaScript* legível e baseado em padrões (NPMJS, 2018?).

O *Angular* é uma plataforma que facilita a criação de aplicativos com a *web*. *Angular* combina modelos declarativos, injeção de dependência, ferramentas de ponta a ponta e práticas recomendadas integradas para resolver desafios de desenvolvimento. *Angular* permite que os desenvolvedores criem aplicativos que vivem na *Web*, em dispositivos móveis ou na área de trabalho, (ANGULAR 2018?).

## 2.2 FABRICAÇÃO DE CERVEJA ARTESANAL

### 2.2.1 História da Cerveja

Conforme descrito por Young (2019), a cerveja é uma bebida alcoólica produzida pela extração de matérias-primas com água, fervura (geralmente com lúpulo) e fermentação. Em alguns países, a cerveja é definida por lei - como na Alemanha, onde os ingredientes padrão, além da água, são malte (cevada germinada em estufa), lúpulo e fermento. Lei esta, que segundo Comofazercerveja (2019?), foi promulgada em 23 de Abril de 1516 pelo Duque Wilhelm IV (Guilherme IV) da Baviera. Esta regulamentava que para a produção de cerveja somente poderiam ser utilizados os três ingredientes: malte, lúpulo e água. Um dos primeiros códigos de defesa do consumidor que se tem conhecimento na história.

Segundo Ortiz (2014, p. 01, apud Brigido, 2006: Netto, 2006) com a Revolução Industrial o modo de produção e distribuição da cerveja sofreu mudanças decisivas, estabelecendo-se fábricas cada vez maiores, com processos cada vez mais automatizados. Nos últimos 80 a 100 anos, formaram-se as bases de uma tecnologia cervejeira, mediante análise dos processos biológicos e químicos, permitindo assim o uso mais racional das matérias primas. Cabe ainda destacar a importância dos conhecimentos obtidos sobre os aspectos químicos e biológicos da fermentação e maturação da cerveja.

Algumas bibliografias estimam que a cerveja seja a quinta bebida mais popular do mundo, em contrapartida outras bibliografias apontam para o terceiro lugar, ficando atrás somente da água e do café.

Cervejas industrializadas utilizam em média 60% de malte e os 40% restante corresponde a outros tipos de cereais para que o custo de produção seja reduzido. Logo, a cerveja artesanal é produzida com maior quantidade de malte, e também leva em conta, particularidades de sabor e aroma que são preferência dos consumidores.

Conforme Lara (2018) a cerveja artesanal advém de um processo artesanal mais cuidadoso em sua produção mais focado em qualidade e sabores diferenciados, enquanto as cervejas industrializadas tem por objetivo a produção em grande escala com grandes maquinários.

## **2.2.2 O Processo de Produção da Cerveja Artesanal**

A cerveja necessita de alguns ingredientes para sua produção: são eles o malte, lúpulo, fermentadores e água.

### **2.2.2.1 Malte**

Conforme consta em Rosenthal (2018), malte é qualquer cereal que tenha passado pelo processo de malteamento ou malteação, que consiste em induzir a germinação dos grãos em ambiente controlado até a formação da raiz (radícula) e caule (caulículo), então é chegada a hora de interromper o desenvolvimento da planta, secando e torrando os grãos. Durante o processo de germinação, o grão gera açúcares, amidos solúveis e amidos que podem ser transformados em açúcar durante uma das etapas do processo de produção de cerveja, que se chama Mostura. Durante este processo o amido será transformado em açúcares simples que servirão de alimento para as leveduras, que por sua vez, transformaram esses açúcares em álcool e gás carbônico. Leveduras não conseguem se alimentar de amido diretamente, necessita-se então esta transformação. Durante o processo de germinação as sementes geram enzimas



que quebram o amido para poderem se alimentar em um primeiro momento. As enzimas são de extrema importância para produção de cerveja, pois serão usadas para quebrar o amido dos grãos liberando açúcares simples para alimentar as leveduras.

### **2.2.2.2 Leveduras**

As leveduras por sua vez, ou também conhecidas como levedo, são um tipo de fungo, minúsculos organismos unicelulares capazes de se dividir formando célula mãe e célula filha, multiplicando-se sem depender de ninguém.

O nome genérico que as leveduras cervejeiras recebem é *Saccharomyces cerevisiae*. Após muitos anos de estudo e seleção de leveduras, atualmente existem centenas de espécies que se enquadram nos padrões para fabricação cervejeira, cada uma com uma peculiaridade diferente.

As leveduras se alimentam de glicose, que é um açúcar simples presente no malte. Contudo elas podem se alimentar desta glicose de duas formas, na presença ou na ausência de oxigênio. A presença do oxigênio é essencial no início da fermentação, pois as leveduras têm melhor desempenho para reprodução na presença deste gás. Quando as leveduras consomem glicose sem oxigênio, esse processo denomina-se fermentação, e é somente nessas condições que as leveduras produzem gás carbônico e álcool.

Antes de a fermentação acontecer, o preparo da cerveja se chama mosto, que é um caldo açucarado, a base de água e malte, que o cervejeiro prepara para alimentar as leveduras, e com isso, fazê-las produzir o álcool e o CO<sub>2</sub> (Dióxido de Carbono), ou, como é mais conhecido, gás carbônico. Após este processo de fermentação, as leveduras transformaram o mosto em cerveja. O álcool e o gás carbônico são em maior quantidade na fermentação, porém diversas outras substâncias que também são produzidas em menor quantidade e que proporcionarão o aroma da cerveja Rosenthal (2018).

### 2.2.2.3 Lúpulo

Lúpulo é a flor de uma planta da espécie *Humulus lupulus*, que pertence à família *Cannabaceae*. Esta planta se desenvolve nas regiões temperadas da América do Norte, Europa e Ásia. Essa espécie é uma rosácea liana, ou seja, brota no solo, mas necessita de um suporte para se sustentar verticalmente, podendo chegar a 20 metros de altura. Para a produção do lúpulo, são coletadas as flores femininas, que são verde-claras e apresentam formato semelhante a pinhas miniaturas. Na base de suas pétalas, elas possuem diversas glândulas amareladas, conhecidas como lupulinas, que são fundamentais por produzirem óleos e resinas essenciais utilizados na cerveja. Esses óleos essenciais são conhecidos como alfa-ácido, também chamados de humulonas, e, beta-ácido também chamado de lupulonas. Estes ácidos têm atuação durante o processo de fervura do mosto. Ainda segundo o *Cursoservejeiroartesanal* (2019), o lúpulo é o principal tempero da cerveja. Ele é responsável primariamente pelo amargor característico da cerveja, bem como ajuda a incrementar o aroma e sabor.

Atualmente o lúpulo pode ser comercializado de três maneiras diferentes, na forma natural, em pellets ou como extrato. De forma natural, as folhas e cones são embalados em volumes prensado e removido o oxigênio, a fim de reduzir o volume e evitar oxidação. Pellets, toda a flor do lúpulo é prensada em uma massa e cortada em pequenas porções cilíndricas. Embora seja um formato ótimo para armazenamento e transporte, tem-se como desvantagem o fato de toda a estrutura da flor estar contida no produto. A única parte que interessa é a lupulina, e nos pellets é compactada toda a massa verde que forma a flor (CRAFT, 2018).

A técnica de obtenção de extratos de lúpulo é comum desde o ano de 1960, quando inicialmente os compostos da flor eram extraídos por meio de destilação a vapor, devido à percepção de características negativas em relação ao sabor, com o passar do tempo um novo método, a partir de extração por dióxido de carbono líquido, passou a ser o mais empregado. O extraído é concentrado por evaporação e posteriormente destinado à indústria cervejeira (KUNZE, 2006).

Para se produzir uma cerveja artesanal, inicialmente deve ser definida qual a receita da cerveja que busca ser produzida com relação ao teor alcoólico, cor e

amargor por exemplo. Ainda segundo o site, processo de brassagem, são quatro etapas:

- A mosturação - A mosturação pode durar de 60 a 90 minutos, o malte moído é misturado com água aquecida com diferentes temperaturas controladas, pois nesta etapa o malte é hidratado enquanto que as enzimas são ativadas, favorecendo a conversão de amido em açúcares fermentáveis.
- A filtração - Na filtração é separada a parte da solução açucarada do bagaço e também as impurezas do mosto, por meio de um processo de clarificação ou recirculação. Esta fase, apesar de lenta é de extrema importância para a qualidade do produto final. Quando o mosto se encontrar mais límpido ocorre a etapa de lavagem para que os açúcares residuais do malte sejam extraídos através de altas temperaturas, e em seguida o mosto é colocado na tina de fervura.
- A fervura - A intensa fervura tem por objetivo esterilizar o mosto, isomerizar o lúpulo, que é quando o lúpulo libera suas enzimas e óleos essenciais que dão sabores diferenciados as cervejas, também é importante para acertar a densidade inicial da cerveja e evaporar compostos indesejáveis da mistura. Há os lúpulos que conferem amargor, e os que conferem aroma, sua ordem de adição à mistura parte da opção de cada cervejeiro, normalmente os lúpulos de amargor são adicionados 60 minutos antes do término da fervura enquanto que os lúpulos aromáticos são acrescentados nos 15 minutos finais à fervura.
- O resfriamento - O resfriamento é uma etapa importante para que o mosto atinja a temperatura ideal para adição de levedura e evitar a contaminação do mosto. A temperatura escolhida dependerá do tipo de fermentação desejada.
- Processo de fermentação - Podendo durar de 5 a 10 dias, nesta etapa inicia a inoculação de leveduras, o consumo de açúcares fermentáveis e a produção do gás carbônico, álcool e alguns ácidos orgânicos. Neste procedimento é necessário que haja um controle da temperatura para que não afete no resultado final da cerveja.

- Processo de maturação - Etapa essencial para que a cerveja atinja seu equilíbrio, a maturação resulta no amadurecimento dos compostos responsáveis pelo aroma, sabor e formação de ésteres, pode durar em média 10 dias.
- Processo de envase - Este é o momento do engarrafamento da cerveja, deve-se evitar contato da cerveja com oxigênio do ar, pois isso pode provocar alterações no paladar, turbidez e coloração da bebida.

### 3. METODOLOGIA

O presente Trabalho de Conclusão de Curso tem como objetivo identificar como um aplicativo *mobile* híbrido com o *Ionic* pode auxiliar na produção de cervejas artesanais caseiras. Portanto, este trabalho classifica-se como sendo uma pesquisa aplicada, visto que se busca gerar conhecimentos para uma aplicação prática, dirigida à solução de problemas específicos (SILVEIRA, 2009).

Para o processo de coleta de dados, ocorreu em duas perspectivas distintas: consulta a produções bibliográficas relacionadas ao desenvolvimento com o *framework Ionic* e à produção de cerveja artesanal; e aplicação de um questionário (Apêndice A) voltado a cervejeiros caseiros. A amostra intencional foi composta por 2 (dois) grupos de cervejeiros artesanais, caracterizando uma pesquisa com *Survey* (SILVEIRA; CÓRDOVA, 2009).

A coleta de dados nesta etapa foi realizada exclusivamente por correio eletrônico. Tal escolha se justifica pela velocidade conferida pelo *e-mail* ao envio/recebimento dos questionários, somada à flexibilidade de tempo para resposta do entrevistado. Os dados obtidos foram analisados qualitativamente, de modo que pudessem servir de entrada para a etapa subsequente.

Os dados obtidos foram analisados qualitativamente, de modo que pudessem servir de entrada para a etapa subsequente. Nesta próxima etapa, um protótipo de *software* foi especificado e posteriormente desenvolvido, empregando técnicas de programação e conhecimentos de desenvolvimento *mobile*. A etapa seguinte consistiu em uma avaliação de como o protótipo pode auxiliar na produção de cervejas artesanais. Para isto, foi adotada uma nova entrevista com os cervejeiros caseiros.

## 4. DESENVOLVIMENTO DA PESQUISA

Esta seção descreve a etapa inicial do Trabalho de Conclusão de Curso, realizada no mês de agosto de 2019. A etapa incluiu a realização de um questionário aplicado a 2 (dois) grupos de cervejeiros artesanais. Este questionário é composto de vinte perguntas, cinco questões para identificar o perfil do cervejeiro, às demais questões sobre como o uso e como um software pode contribuir na produção de cerveja artesanal. A análise das respostas ao questionário auxiliou na identificação da visão cervejeiro artesanal sobre quais informações são necessárias para a produção de cervejas, destacando-se suas dificuldades e sugestões de melhorias neste processo.

As seções a seguir descrevem o processo de coleta de dados e a análise das respostas obtidas.

### 4.1 COLETA DE DADOS

Segundo Aaker et al. (2007) a coleta de dados utilizando o e-mail pode oferecer algumas vantagens como:

- a) os questionários podem ser enviados quantas vezes forem necessárias com maior velocidade;
- b) maior velocidade também no recebimento das respostas;
- c) Os questionários podem ser respondidos de acordo com a conveniência e tempo do entrevistado.

Tendo em vista a disponibilidade de tempo limitada do público-alvo, optou-se por uma abordagem via preenchimento de formulário online enviado por *e-mail*. O questionário, desta forma, aborda sobre o conhecimento de produção de cerveja artesanal, bem como, se o cervejeiro em questão utiliza algum tipo de aplicativo para

auxiliar nas brassagens. Também é levantado algumas hipóteses sobre possíveis ferramentas para auxiliar na montagem de receitas.

## 4.2 ANÁLISE DAS RESPOSTAS

De acordo com o questionário, os entrevistados têm entre 33 e 45 anos. Sobre sua experiência na produção de cerveja, estes estão atuando entre 3 a 5 anos, já tem certa experiência. Verificou-se que os sistemas operacionais dos *smartphones* mais comum entre eles é o *Android* e o *IOs* respectivamente.

De acordo com os entrevistados, o aplicativo mais utilizado por eles é o *BeerSmith*, ferramenta esta que é paga, e bastante técnica. Eles informaram que é necessário a utilização de outros aplicativos para certos cálculos além do *BeerSmith*. Um dos entrevistados utiliza uma calculadora de outro aplicativo para garantir os parâmetros da receita. Mesmo utilizando um *software* para ajudar fazer os cálculos, os cervejeiros acabam imprimindo às receitas em papel para poder consultar durante a produção, pois no caso deles, a aplicação estava instalada em um computador, como teriam que comprar nova licença para o celular, não o fizeram. Geralmente os cervejeiros fazem pequenos ajustes nas quantidades dos insumos para aprimorar o aroma e o sabor da cerveja, tais ajustes são anotados junto às receitas impressas e podem acabar se perdendo. Quando questionados sobre cadastrar a origem da água utilizada na receita com seu respectivo pH (potencial Hidrogeniônico), eles disseram que seria bastante interessante, pois o pH da água influencia diretamente no sabor da cerveja. Desta forma, dependendo da origem da água é necessário um correção no pH para brassagem. Os entrevistados falaram também sobre a importância de registrar as datas de início e fim da fermentação, para garantir uma melhor qualidade da cerveja.

Analisando as respostas do questionário pensou-se em elaborar um aplicativo com interface mais simples, com funcionalidades de fácil compreensão para que pessoas iniciantes ou até mesmo pessoas leigas entusiastas na arte da cerveja artesanal possam ter um primeiro contato com este mundo fantástico, sem ter a necessidade de comprar um aplicativo pago, com a vantagem de poder rodar tanto nas plataformas *Android* como *IOs* e *Windows Phone*.

Este aplicativo tem a possibilidade de armazenar todos os insumos utilizados para confecção da receita de cerveja artesanal com todos os dados seus relevantes, ele ainda permite montar e armazenar a receita. Com o auxílio de um banco de dados *Json*, possibilita manter todos os dados no aparelho onde o aplicativo está instalado, facilitando assim o acesso, no caso de precisar utilizar o aplicativo em local sem acesso à *internet*, eliminando a necessidade de impressão de receitas em papel. Outra vantagem é que o aplicativo permite fazer ajustes nas receitas, permite também inserir notas que possam ser relevantes durante o processo de produção, aprimorando a receita a cada brassagem.



## 5. DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo é apresentado o protótipo de software desenvolvido com base nos requisitos identificados a partir do questionário aplicado aos cervejeiros, assim como da revisão bibliográfica. A subseção “Especificação” traz detalhes referentes ao planejamento da ferramenta, enquanto a subseção “Desenvolvimento” apresenta os resultados obtidos após o processo de desenvolvimento.

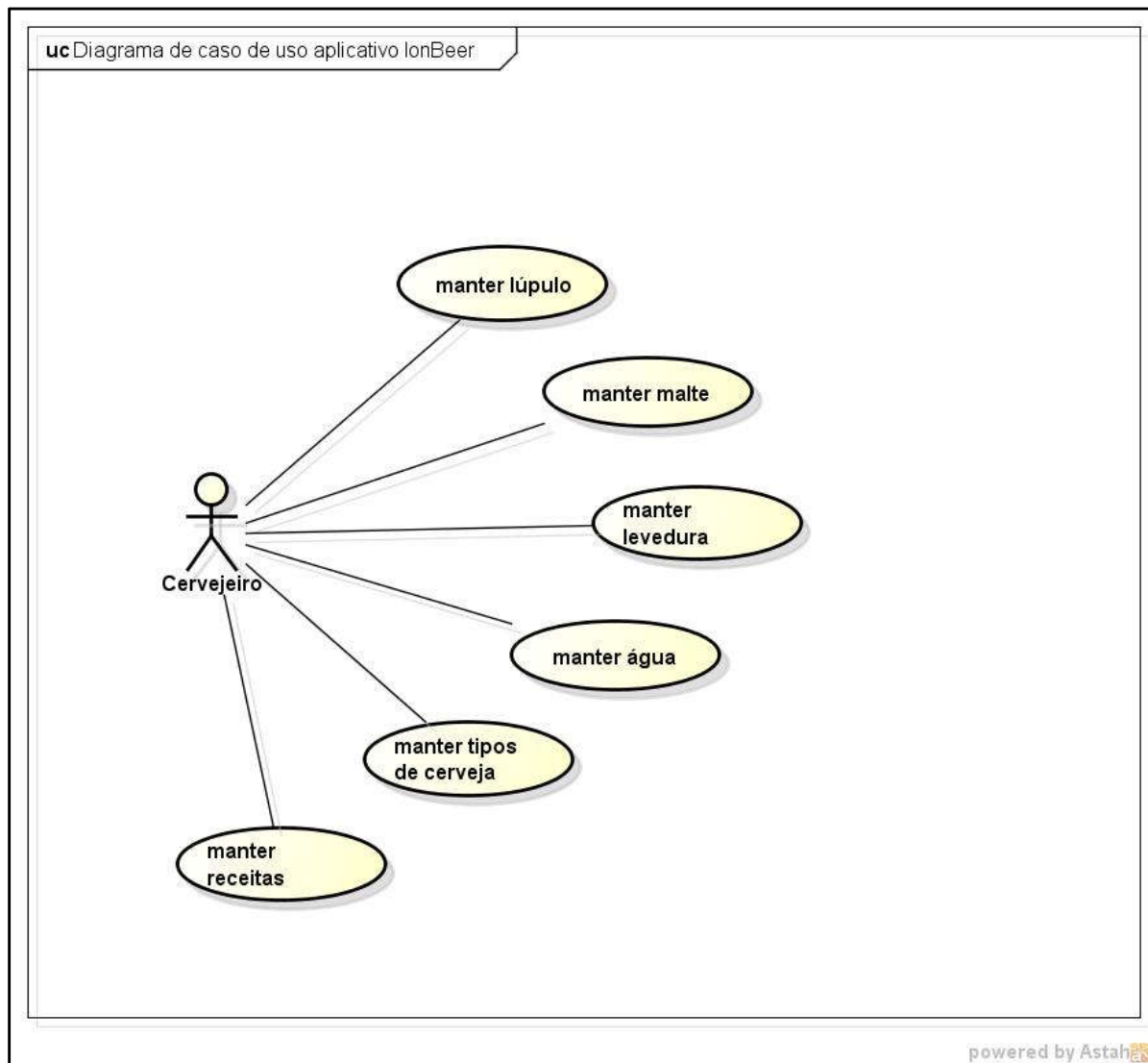
### 5.1 ESPECIFICAÇÃO

Nessa subseção serão apresentados os diagramas relacionados à análise do protótipo desenvolvido no presente trabalho.

Segundo Booch (2000, p.XIII), *UML (Unified Modeling Language)*, ou Linguagem Unificada de Modelagem é uma linguagem gráfica simplificada para visualização, especificação, construção e documentação de objetos de sistemas complexos de *software*. A *UML*, como o próprio nome diz, proporciona uma forma padrão e unificada para a representação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais como processos de negócio e funções do sistema, assim como itens concretos, classes escritas em determinadas linguagens de programação, esquemas de banco de dados e outros componentes de *software* reutilizáveis.

Através da *UML* serão apresentados o diagrama de casos de uso e diagrama de classes, projetados para o desenvolvimento do protótipo do aplicativo. Diagrama de casos de uso, conforme Booch (2000, p.217), é a representação gráfica de um conjunto de sequências de ações, inclusive variantes, que um sistema executa para produzir um resultado de valor observável por um ator. Na *UML* o diagrama de casos de uso é representado por ator ou atores, que seriam os usuários que interagem com o *software*. As ações que o *software* executa são escritas no infinitivo

e dentro de elipses, esta ação é o resultado da interação do usuário com o sistema. Conforme se pode constatar abaixo na Figura 4, o diagrama de caso de uso do protótipo de aplicativo desenvolvido. Vide Casos de Uso em APÊNDICE B.



**Figura 4: Diagrama de Casos de Uso**

Fonte: Autor, 2019.

No caso de uso, acima citado, especifica-se a interação entre o usuário, no caso, o cervejeiro, e o aplicativo para dispositivo móvel. O cervejeiro acessa o aplicativo para e pode realizar os seguintes casos de uso:

### **5.1.1 Manter Lúpulo**

O cervejeiro realiza o cadastro do Lúpulo no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar, editar e excluir o lúpulo cadastrado.

### **5.1.2 Manter Malte**

O cervejeiro realiza o cadastro do Malte no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar e editar o Malte cadastrado.

### **5.1.3 Manter Levedura**

O cervejeiro realiza o cadastro da Levedura no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar, editar e excluir a Levedura cadastrada.

### **5.1.4 Manter Água**

Da mesma forma que os demais ingredientes, a água tem suas particularidades, conforme a região de onde a água é extraída, ela tem quantidades de sais diferentes. Para uma produção de qualidade, é necessário verificar as quantidades de sais e pH da água. Então o cervejeiro realiza o cadastro da Água no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar, editar e excluir a Água cadastrada.

### **5.1.5 Manter Tipos de Cerveja**

Conforme Bonaccorsi (2016), os estilos de cerveja estão catalogados em um guia de cervejas, conhecido no mundo dos cervejeiros como *BJCP (Beer Judge Certification Program)*. Neste guia estão catalogados os estilos de cervejas, com suas características e particularidades. Manter tipo de cerveja permite o cervejeiro cadastrar tipo de cerveja com os valores especificados no *BJCP*, o qual é salvo no banco de dados, dados ele precisará buscar para posterior montagem da receita. Permite listar todos os tipos de cerveja previamente cadastrados. O usuário pode

editar alguma informação cadastrada, e por fim, pode-se excluir o tipo de cerveja cadastrado se assim o desejar.

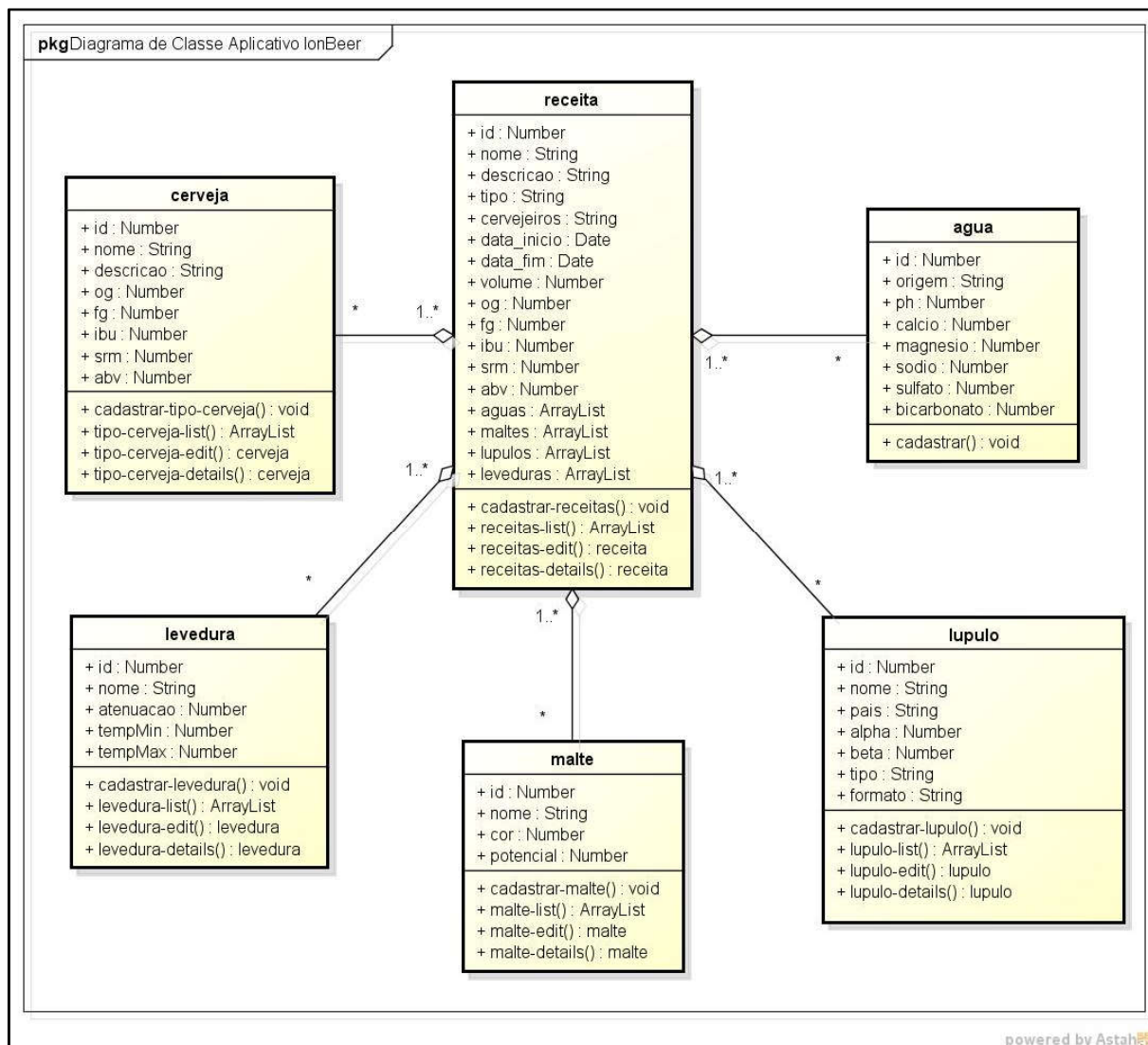
#### **5.1.6 Manter Receita**

O cervejeiro pode cadastrar receitas, neste cadastro o cervejeiro insere informações importantes para produção da cerveja e salva no banco de dados. Permite listar todas as receitas previamente cadastradas. O usuário pode editar informações cadastradas, e por fim, pode-se excluir a receita se assim o desejar.

Diagramas de classes, ainda segundo Booch (2000, p.104), são diagramas encontrados com mais frequência na modelagem de sistemas orientados a objetos. Um diagrama de classes mostra um conjunto de classes, interfaces, colaborações e seus relacionamentos. Utilizados para fazer a modelagem da visão estática do projeto de um sistema. No diagrama de classes a linguagem começa a convergir para o vocabulário de programação. Os diagramas de classe são muito importantes não só para visualização, especificação e documentação de modelos estruturais, mas também para construção de sistemas executáveis por intermédio de engenharia de produção reversa. Na UML uma classe é representada por um retângulo com três divisões, são elas: O nome da classe, seus atributos e por fim os métodos.

Para Ricarte (2001) uma classe é um gabarito para a definição de objetos, ou seja, uma descrição que abstrai um conjunto de objetos com características similares. Através da definição de uma classe, descrevem-se as propriedades ou atributos o objeto terá. Além da especificação de atributos, a definição de uma classe descreve também qual o comportamento de objetos da classe, ou seja, que funcionalidades podem ser aplicadas a objetos da classe. Essas funcionalidades são descritas através de métodos. Um método nada mais é que o equivalente a um procedimento ou função, com a restrição que ele manipula apenas suas variáveis locais e os atributos que foram definidos para a classe. Objeto em POO (Programação Orientada a Objetos) se refere a uma instância de uma classe. E método, em POO, se refere a uma ação associada a um objeto, por exemplo, método “salvar” quando for chamado, irá executar a ação de salvamento previamente programada no sistema.

Pode-se verificar abaixo (Figura 5), com Diagrama de classes desenvolvido para programação do código do aplicativo, fruto deste estudo.



**Figura 5: Diagrama de Classes**

Fonte: Autor, 2019.

A persistência dos dados inseridos nos formulários se dá através de *json-server*. Segundo Gonçalves (2012), *JSON (JavaScript Object Notation)* é um modelo para armazenamento e transmissão de informações no formato texto. Apesar de muito simples, tem sido bastante utilizado por aplicações *Web* devido a sua capacidade de estruturar informações de uma forma bem mais compacta do que a conseguida pelo modelo *XML (Extensible Markup Language)*, tornando mais rápido o *parsing* dessas informações. Por este motivo *Google* adotou este formato, para poder transmitir grandes volumes de dados. Conforme Agasthyan (2017), *REST*

(*REpresentational State Transfer*) é um estilo de arquitetura para projetar aplicações conectadas, usando os simples verbos *HTTP* (*Hypertext Transfer Protocol*) para permitir a comunicação entre as máquinas. Assim, ao invés de usar uma *URL* (*Uniform Resource Locator*) para manipular informação do usuário, *REST* envia uma requisição *HTTP*, como *GET*, *POST*, *DELETE*, *UPDATE*, para uma *URL* manipular os dados. *APIs REST* formam o *backend* de aplicações *web* e móveis. Para se ver um aplicativo *web* ou móvel em ação precisa-se então de um servidor que retorne dados *JSON* falsos, é neste momento que a *API REST* falsa entra em ação, o *json-server* provê a funcionalidade para configurar um servidor de *API REST* falso. Diz-se que a *API REST* é falsa pois os dados *JSON* estão salvos na aplicação, e a requisição destes dados é feita através de *URLs*, simulando uma requisição na rede. Através de um serviço disponível como uma classe injetável, com métodos para executar solicitações *HTTP*. Cada método de solicitação possui várias assinaturas e o tipo de retorno varia de acordo com a assinatura que é chamada. O código fonte para acessar o banco de dados *JSON* fica da seguinte forma: “base\_path = '<http://localhost:3000/lupulos>';“. O *localhost* se refere ao servidor da máquina local, apontando para porta 3000 no item “lupulos”. A partir de funções definidas dentro deste serviço, é executada a manipulação dos dados conforme função chamada. Figura 6, exemplo do código fonte para criar um item “lupulo”.

```
41
42 // Create a new item
43 createItem(item): Observable<Lupulo> {
44     return this.http.post<Lupulo>(this.base_path, JSON.stringify(item), this.httpOptions)
45         .pipe(
46             retry(2),
47             catchError(this.handleError)
48         )
49     }
50
```

**Figura 6: Exemplo de função para criar item no banco de dados JSON acessado pela API REST**  
Fonte: Autor, 2019.

Na Figura 7 segue exemplo de uma estrutura *JSON* com a qual a *API REST fake* se conecta para manter “lupulos” e “maltes”.

```
1  {
2  |   "lupulos": [
3  |     {
4  |       "nome": "Brewer's Gold (UK) N",
5  |       "pais": "Reino Unido",
6  |       "alpha": "6",
7  |       "beta": "4",
8  |       "tipo": "Aroma",
9  |       "formato": "Pellet",
10 |       "id": 3
11 |     }
12 |   ],
13 |   "maltes": [
14 |     {
15 |       "nome": "Château Pilsen",
16 |       "cor": 2.5,
17 |       "potencial": 100,
18 |       "id": 13
19 |     }
20 |   ],
```

**Figura 7: Banco de dados JSON acessado pela API REST**

Fonte: Autor, 2019.

## 5.2 DESENVOLVIMENTO

O desenvolvimento deste protótipo foi realizado utilizando o *Framework Ionic*. Conforme descrito anteriormente neste documento, o *Framework Ionic* gera toda a estrutura base da aplicação, fazendo todas as associações necessárias para um bom funcionamento do aplicativo. Ele é responsável por gerar classes de teste para evitar possíveis erros ou conflitos, testando o código simultaneamente a cada salvamento feito. A seguir, são apresentados os detalhes das telas desenvolvidas.

### 5.2.1 Tela Inicial

Na tela inicial existe uma breve apresentação do aplicativo, no *header*, ou, cabeçalho, encontra-se o nome do aplicativo e imagens de vários tipos de cerveja. No conteúdo da tela inicial verifica-se uma imagem com ingredientes para produção de cerveja artesanal, logo abaixo uma breve apresentação do aplicativo. Foram

inseridos dois botões, um deles direcionam para rede social para divulgação do aplicativo, e outro botão para acessar as funcionalidades do aplicativo. Ao clicar no botão “Receitas”, conforme Figura 8, o aplicativo direciona para a tela de “Receitas”, (Figura 9), esta tela será abordada nos próximos tópicos, devido necessidade de cadastro prévio de ingredientes.



**Figura 8: Tela Inicial do Aplicativo**  
Fonte: Autor, 2019.



**Figura 9: Tela Receitas do Aplicativo**  
Fonte: Autor, 2019.

### 5.2.2 Lúpulo

Na tela de cadastro de lúpulo (Figura 10), são inseridos os dados pertinentes às características do lúpulo que se deseja cadastrar. Será salvo o nome do lúpulo, o país de origem, geralmente cervejeiros caseiros utilizam lúpulos importados, conforme as condições climáticas, solo e uma série de fatores contribuem para a diferenciação do lúpulo, isso faz com que cada espécie ou tipo seja mais indicado para produção de determinado tipo de cerveja. Serão inseridos os valores de alpha e beta ácido, estes campos aceitam somente números, conforme o *placeholder* sugere. O atributo placeholder é uma pequena dica, uma pequena frase, uma



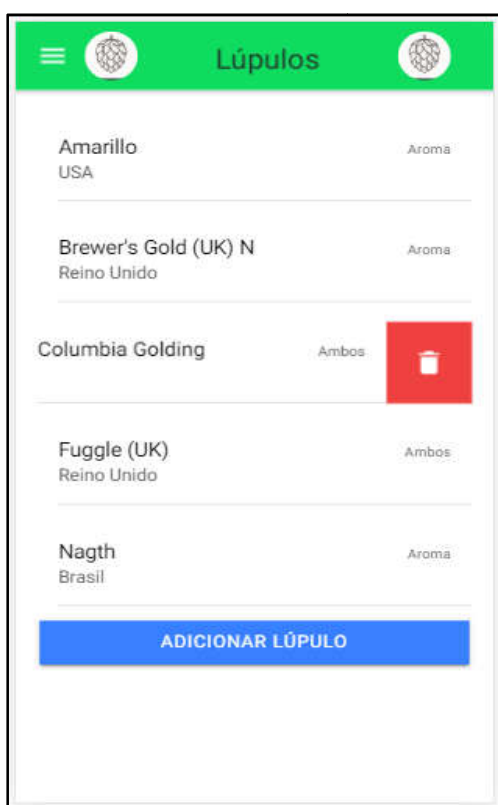
palavra, que tem o intuito de ajudar o usuário a entender como ele deve preencher aquele formulário. Também será selecionado o tipo de lúpulo, geralmente encontra-se no mercado três tipos de lúpulo com aplicações diferentes na receita, para amargor, para aroma e para ambos (aroma e amargor), vide Figura 11.

**Figura 10: Tela Cadastrar Lúpulo**  
Fonte: Autor, 2019.

**Figura 11: Tela Cadastrar Lúpulo**  
Fonte: Autor, 2019.

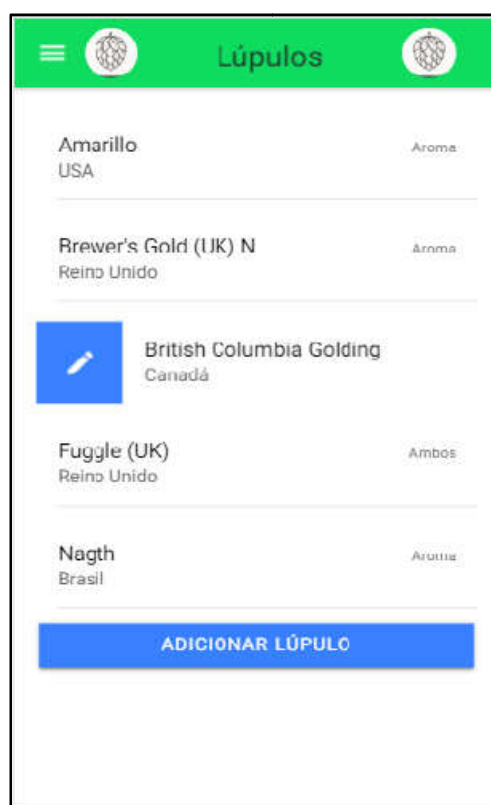
Por fim seleciona-se o formato do lúpulo. No mercado encontra-se lúpulo em três formatos diferentes, em *pelet* que é o formato mais comum, onde o ingrediente é compactado e forma pedaços cilíndricos de dez a quinze milímetros de comprimento. Em flor desidratada que é a forma natural do ingrediente e extrato, onde o lupulona encontra-se concentrada. Depois de feitos estes passos, clicar no botão de cadastrar, uma mensagem de sucesso é mostrada e todos estes dados serão inseridos em um banco de dados *JSON*, e o aplicativo redireciona para a tela com a lista dos lúpulos cadastrados, conforme a Figura 12. O sistema permite também, ao invés de cadastrar um lúpulo, clicar no botão “Listar” ir direto para a tela listar lúpulo, onde serão apresentados todos os lúpulos cadastrados.

Nesta tela da lista, tem a opção de ver todos os detalhes do item cadastrado clicando sobre ele. Permite também, apagar o item, caso o mesmo não se faça mais necessário, para isso, é só deslizar a linha que contenha o item desejado da direita para esquerda, então surgirá um botão vermelho com símbolo de uma lixeira (Figura 12), clicando neste botão o item será permanentemente excluído do sistema, ou pode-se também editar algum dado que possa ter sido cadastrado equivocadamente, para isso existem duas formas, deslizando a linha do item desejado da esquerda para direita, então surgirá um botão azul com símbolo de um lápis (Figura 13), clicando neste botão, o aplicativo redireciona para a tela de edição, onde todos os dados são carregados e podem ser alterados (Figura 15).



**Figura 12: Tela Listar Lúpulo, detalhe Botão Apagar Item.**

Fonte: Autor, 2019



**Figura 13: Tela Listar Lúpulo, detalhe Botão Editar Item.**

Fonte: Autor, 2019.

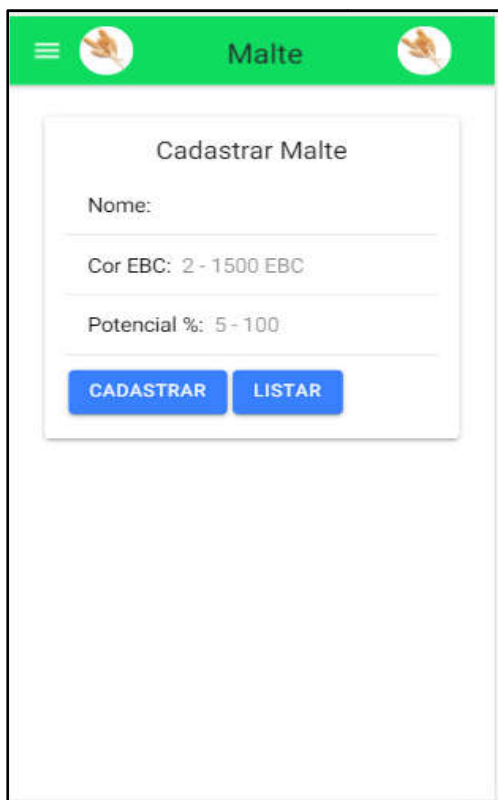
Outra maneira de direcionar para tela de edição de dados é clicando sobre o item, o qual será direcionado para a tela de detalhes (Figura 14), onde serão apresentados todos os dados cadastrados referentes àquele item, nesta tela tem um botão de cancelar, que volta para a tela anterior e um botão de editar, que direciona para tela de edição.

**Figura 14: Tela Detalhes do Lúpulo.**  
Fonte: Autor, 2019.

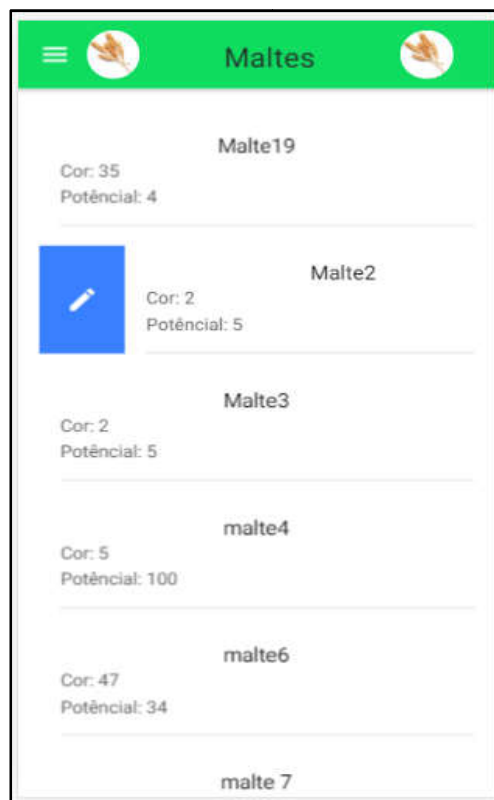
**Figura 15: Tela Editar Lúpulo.**  
Fonte: Autor, 2019.

### 5.2.3 Malte

Na tela de cadastro de malte são inseridos os seguintes dados: nome do malte, cor *EBC (European Brewing Convention)* escala europeia aplicada para medir a cor da cerveja. O potencial de cada malte é calculado pelo fornecedor (Figura 16). Após a inserção dos dados pode-se salvar no banco de dados clicando no botão “Cadastrar”, uma mensagem de sucesso será exibida, os dados serão salvos e o aplicativo direciona para tela de lista de maltes previamente cadastrada (Figura 17). Da mesma forma que acontece na tela listar lúpulo, apresentada no tópico anterior, existe a possibilidade de editarmos ou excluir os dados salvos.



**Figura 16: Tela Cadastrar Malte.**  
Fonte: Autor, 2019.



**Figura 17: Tela Listar Malte.**  
Fonte: Autor, 2019.

Pode-se também acessar a tela de detalhes do malte, clicando sobre o item desejado (Figura 17). Para acessar a tela de edição de malte basta clicar no botão “Editar”, vide Figura 19. Na tela de edição podem-se ajustar os dados e atualizar o cadastro do malte.

**Figura 18: Tela Detalhe Malte.**  
Fonte: Autor, 2019.

**Figura 19: Tela Editar Malte.**  
Fonte: Autor, 2019.

#### 5.2.4 Levedura

Na tela de cadastro de levedura cadastra-se o nome, a atenuação, a temperatura máxima e mínima de trabalho, conforme Figura 20.

Atenuação da levedura é informada pelo fornecedor, é a capacidade de atenuar os açúcares de fermentação que se encontram no mosto. Assim como a temperatura máxima e mínima, são informações importantes, pois um dos para cada tipo de levedura existe uma faixa ideal de temperatura de trabalho. Inseridos estes dados pode-se finalizar o cadastro clicando no botão cadastrar. Os dados são inseridos no banco de dados e uma mensagem de sucesso é apresentada, o aplicativo é redirecionado para tela de lista de leveduras previamente cadastradas (Figura 21). Seguindo a lógica das outras telas de listagem, é possível visualizar todos os detalhes do item cadastrado assim como editar ou excluir um item da lista.

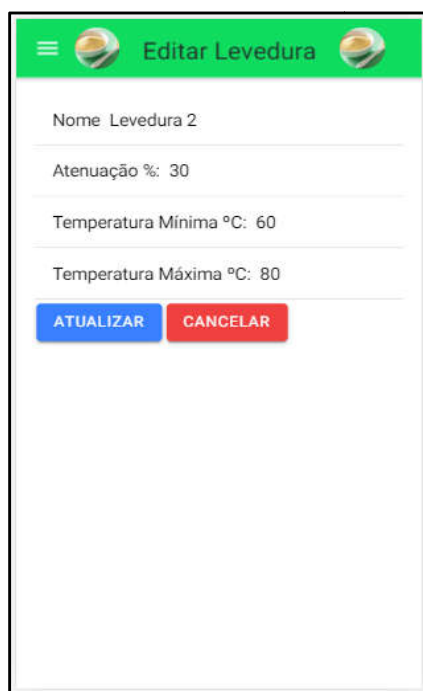


**Figura 20: Cadastrar Levedura.**  
Fonte: Autor, 2019.



**Figura 21: Tela Listar Levedura.**  
Fonte: Autor, 2019.

Na tela de edição de Levedura, pode-se alterar qualquer dado ali apresentado, de forma simples e fácil, o item selecionado carrega todos seus dados nesta tela e a partir disto permite editar conforme a necessidade (Figura 22).



**Figura 22: Tela Editar Levedura.**  
Fonte: Autor, 2019.

### 5.2.5 Água

Por que é importante cadastrar água para fabricação de cerveja?

Conforme consta em Comofazercerveja (2019?), a água corresponde a 90% da matéria prima utilizada na fabricação de cerveja, logo, a qualidade da cerveja está diretamente associada à qualidade da água. Atualmente existem processos laboratoriais para corrigir e adequar às características da água, geralmente utilizadas na indústria da cerveja, para que se tenha uma padronização deste ingrediente tão importante. No caso dos cervejeiros caseiros, a água é oriunda da torneira, e pode ter diferentes composições dependendo da origem, isto pode ocorrer na mesma cidade, pois a água pode ser oriunda de poços diferentes. A principal característica que o cervejeiro precisa saber é o pH da água que vai de uma escala de 0 (ácido) a 14(alcalino). O pH ideal para produção de cerveja fica entre 6.5 e 7.0, mas podendo variar conforme o tipo de cerveja que se deseja produzir. Para fazer a medição do pH da água existem equipamentos eletrônicos chamados de peagômetros, ou também é possível medir através das fita de teste de pH, facilmente encontradas nos mercados. Os demais sais encontrados na água podem ser quantificados em testes de laboratório.

Portanto é essencial cadastrar a água que se utiliza na brassagem, para isso o aplicativo tem os seguintes campos a serem preenchidos, conforme Figura 23:

- Origem - neste campo inserimos de onde esta água é oriunda.
- pH - neste campo se insere o valor do pH, valor ideal entre 6.5 e 7.0.
- Cálcio(Ca<sup>+</sup>) - neste campo inserimos a quantidades de cálcio na água, quantidade ideal fica entre 50 e 100 ppm (partes por milhão).
- Magnésio(Mg<sup>+</sup>) - neste campo inserimos a quantidades de magnésio na água, ideal fica entre 10 e 30 ppm.
- Sódio(Na<sup>+</sup>) - neste campo inserimos a quantidades de sódio na água, não podendo exceder 100 ppm.
- Sulfato(SO<sub>4</sub><sup>2-</sup>): neste campo inserimos a quantidades de sulfato na água, podendo variar entre 10 e 250 ppm.
- Bicarbonato de Sódio (HCO<sub>3</sub><sup>-</sup>) - neste campo inserimos a quantidades de bicarbonato.

**Figura 23: Cadastrar Água.**  
Fonte: Autor, 2019.

**Figura 24: Tela Listar Água.**  
Fonte: Autor, 2019.

Após a inserção dos dados climas no botão cadastrar e todos os dados serão salvos no banco de dados, uma mensagem de sucesso aparece e o aplicativo redireciona para página de lista de águas cadastradas (Figura 24). O Aplicativo também pode direcionar diretamente para a tela de lista de água clicando no botão “Listar”. Nesta tela confere-se tudo que já foi cadastrado, caso necessite cadastrar novo cadastro de água, pode-se clicar no botão “Adicionar Água”.

Para verificar todos os detalhes de determinado item basta clicar sobre ele e será direcionado para tela de detalhes da água, (Figura 25). Pode-se também, excluir ou editar um item diretamente na tela listar de água, (Figura 24). Clicando e arrastando da direita para esquerda surgirá um botão de exclusão, ao clicar sobre este botão o item é removido da memória imediatamente. Clicando e arrastando da esquerda para direita aparece o botão de edição que ao ser acionado direciona para tela de edição do item selecionado, conforme Figura 27.



**Figura 25: Tela Detalhes Água.**  
Fonte: Autor, 2019

**Figura 26: Tela Editar Água.**  
Fonte: Autor, 2019

### 5.2.6 Tipos de Cerveja

Na tela de cadastro de tipos de cerveja, insere-se algumas características pré-definidas que podem ser encontradas no Brauakademie (2015). Conforme Figura 27, cadastra-se o nome da cerveja. Pode cadastrar uma breve descrição sobre a cerveja ou sobre algum processo específico que se tenha necessidade de uma anotação. Os demais valores a serem cadastrados podem ser encontrados no Guia de Estilos de Cerveja, famoso entre os cervejeiros como *BJCP*, neste guia está sendo especificados os tipos de cerveja com todas suas características. A combinação dos valores armazenados nos campos de “Densidade Original (OG)”, “Densidade Final”, “Amargor”, “Cor” e “Graduação alcoólica (ABV)” é que definem o tipo de cerveja que será produzida. Após alimentar os dados pode-se armazenar no banco de dados clicando no botão “Cadastrar”, uma mensagem de sucesso será exibida e será redirecionado para tela de listagem de cervejas cadastradas conforme Figura 28.

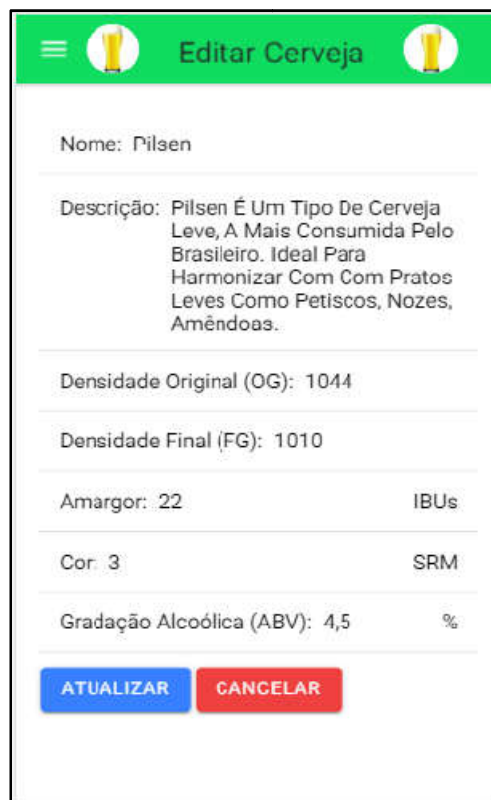
**Figura 27: Tela Cadastrar tipo de Cerveja.**  
Fonte: Autor, 2019.

**Figura 28: Tela Listar Tipos de Cerveja**  
Fonte: Autor, 2019.

De acordo com a lógica das demais telas apresentadas até o momento, a partir da tela de listagem de cervejas cadastradas pode-se selecionar um item para verificar detalhes, da mesma forma, editar e excluir este item, apenas arrastando para direita ou para esquerda e clicando no botão que aparece, sendo direcionado respectivamente para às telas representadas nas Figuras 29 e 30.



**Figura 29: Tela Detalhes Tipo de Cerveja.**  
Fonte: Autor, 2019.



**Figura 30: Tela Editar Tipos de Cerveja.**  
Fonte: Autor, 2019.

## 5.2.7 Receitas

Para poder cadastrar uma receita, primeiramente precisam-se cadastrar todos os ingredientes necessários. Após este primeiro passo, acessar a tela de cadastro de receita, nela encontram-se duas abas no rodapé da tela.

### 5.2.7.1 Aba Características

Aba Características, representado por um círculo com a letra “i” no centro, fica posicionada no lado esquerdo do rodapé, esta aba é selecionada por padrão ao acessar esta tela, pode-se identificar visualizando a cor do botão está azul, enquanto o botão da outra aba está cinza. A aba “Ingredientes” fica posicionada no rodapé no lado direito da tela. Para acessar o conteúdo da tela desejada, basta clicar sobre a descrição (Ingredientes ou Características), para que o aplicativo direcione ao requerido cadastro.

Na aba características insere-se o nome da cerveja. Uma breve descrição pertinente ao processo de produção, ou anotação desejada conforme Figura 31.

**Receitas**

Cadastrar Receita

Nome: Pilsen

Descrição: Aquecer a água até 68°. Inserir o malte previamente moído. Manter na primeira rampa de temperatura por 60 minutos. Após elevar para segunda rampa de temperatura a 75° e manter.

Tipo: Seleccione ... ▾

Cervejeiros:

Data Inicio: Dd/Mm/Aaaa

Data Fim: Dd/Mm/Aaaa

Volume: L

Densidade Original (OG):

Características Ingredientes

**Figura 31: Tela Cadastro Receita, aba Características.**

Fonte: Autor, 2019.

**Receitas**

Cadastrar Receita

Nome: Pilsen

Des: ...

Tipo: Seleccione ... ▾

Cervejeiros:

Data Inicio: Dd/Mm/Aaaa

Data Fim: Dd/Mm/Aaaa

Volume: L

Densidade Original (OG):

Características Ingredientes

Tipo:

Cream Ale

American Double IPA

Cream Ale

Pilsen

CANCEL OK

**Figura 32: Tela Cadastro Receita, aba Característica, *pop up* Tipo de Cerveja.**

Fonte: Autor, 2019.

No campo “Tipo” abre um *pop up* com os tipos de cerveja pré-cadastrados no aplicativo, conforme Figura 32, ali é selecionado o tipo de cerveja que será produzido.

No campo “Cervejeiros” insere-se o nome das pessoas que estão participando da brassagem, caso futuramente se queira saber quem produziu determinada receita pode-se consultar no cadastro da própria receita. “Data Início e Data Fim” são selecionadas às datas que se iniciaram a produção da receita (Figura 33), e respectivamente a data de conclusão da mesma, pois conforme já mencionado, a cerveja precisa de uns dias para maturação antes do envase.

Nome: Pilsen

novembro de 2019

dom	seg	ter	qua	qui	sex	sáb
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Data Fim: Dd/Mm/Aaaa

Volume: L

Densidade Original (OG):

Características | Ingredientes

**Figura 33: Tela Cadastro Receita, aba Data Início.**

Fonte: Autor, 2019.

temperatura a 75° e mantas

Tipo: Pilsen

Cervejeiros: Marcelo Milani

Data Início: 14/11/2019

Data Fim: 21/11/2019

Volume: 50 L

Densidade Original (OG): 1044

Densidade Final (FG): 1010

Amargor: 22 IBUs

Cor: 5 SRM

Gradação Alcoólica (ABV): 4,5 %

Características | Ingredientes

**Figura 34: Tela Cadastro Receita, aba Características.**

Fonte: Autor, 2019.

### 5.2.7.2 Ingredientes

Na aba “Ingredientes” da tela de cadastro de receitas quantificam-se os ingredientes que compõem a receita (Figura 35). A quantidade de água é definida da seguinte forma, considerando o volume total da receita, deve ser dividida em dois momentos, brassagem e lavagem, ou, água primária e água secundária. Exemplo hipotético, segundo Lara (2018) se o recipiente, ou panela onde a cerveja será produzida comporta 30 litros de água, e se pretende usar a capacidade máxima, divide-se este número por 2,5 resultando um total de 12 litros de água primária. Para calcular a quantidade de água da lavagem deve-se considerar a mesma quantidade de água da brassagem multiplicada por 1,5, ou seja, 12 vezes 1,5 resultado 18 litros.

Fórmula:

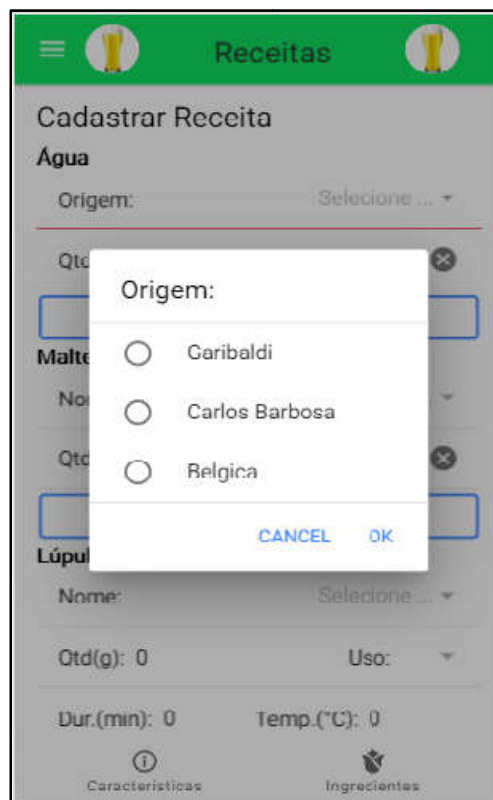
$$\text{Água Brassagem} = \text{Volume Recipiente(L)} / 2,5$$

$$\text{Água da Lavagem} = \text{Água da Brassagem} \times 1,5.$$



**Figura 35: Tela Cadastro Receita, aba Ingredientes.**

Fonte: Autor, 2019.



**Figura 36: Tela Cadastro Receita, aba Ingredientes, popup origem da água.**

Fonte: Autor, 2019.

No campo de “Água” ao clicar em selecionar aparece um *popup* com a lista de águas previamente cadastradas (Figura 37), seleciona-se então a água que será utilizada, logo após especifica-se a quantidade em litros e qual o uso desta quantidade de água, se é para brassagem ou para processo de lavagem, conforme Figura 38. Seguindo a ordem cronológica de execução da produção de cerveja por primeiro se insere a água da brassagem. Para adicionar a água da lavagem, existe o botão “Adicionar Água”, este insere no formulário novamente campos referentes à água da receita, conforme destacado com retângulo preto na Figura 38. Este princípio de adição de mais de um mesmo tipo de ingrediente em uma mesma receita é válido para todos os componentes, malte, lúpulo e levedura, pois existem receitas que necessitam dois ou mais ingrediente do mesmo tipo para se chegar ao objetivo do aroma e sabor.

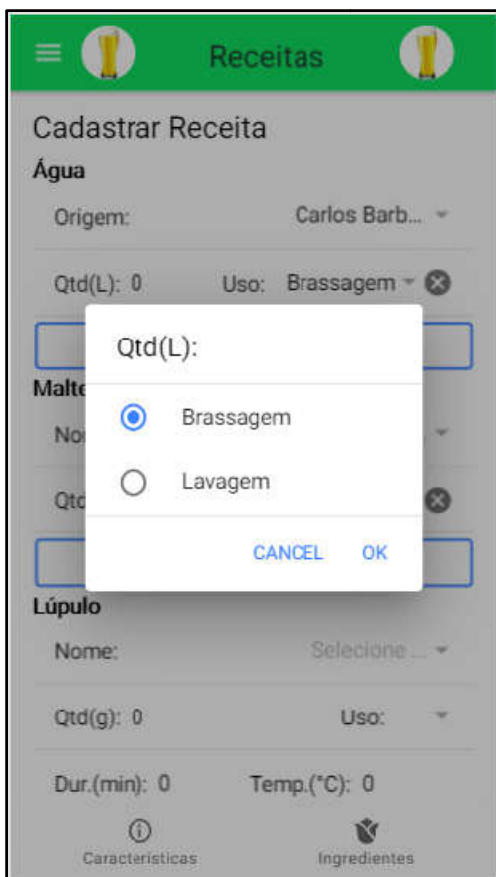


Figura 37: Tela Cadastro Receita, aba Água.  
Fonte: Autor, 2019.

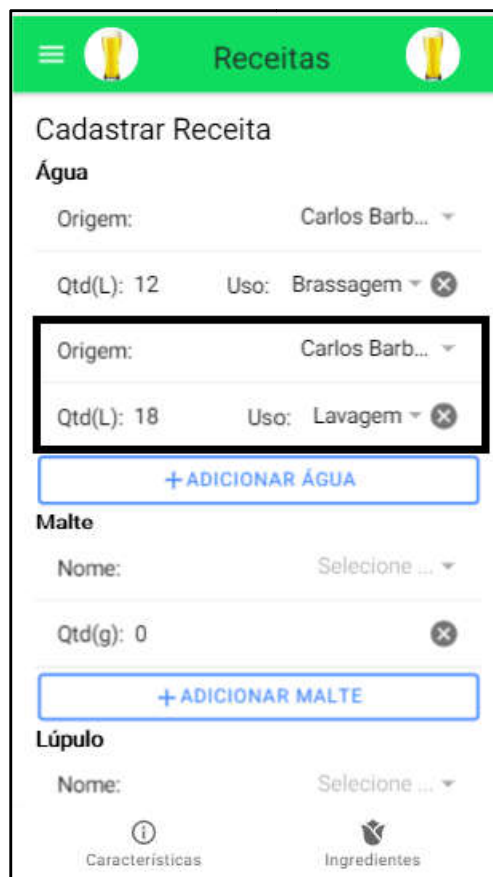
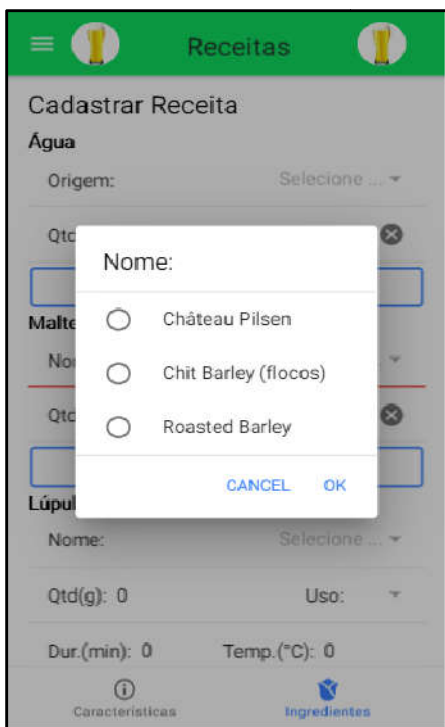


Figura 38: Tela Cadastro Receita, aba Ingredientes, Adicionar Água.  
Fonte: Autor, 2019.

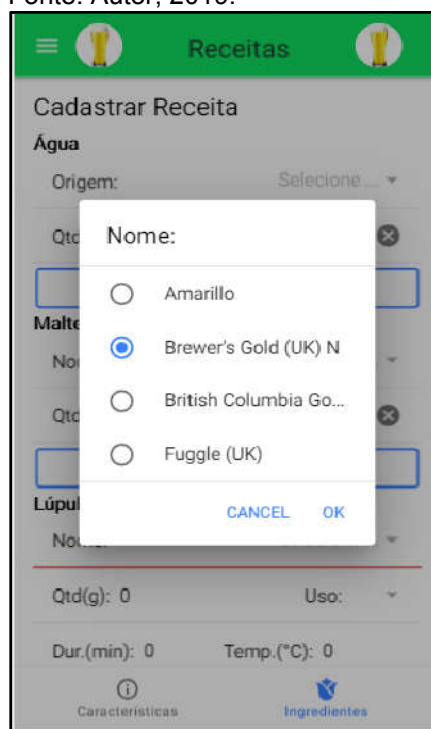
Para inserir o malte clicar no campo “Nome” selecionar no *popup* dentro a lista de maltes cadastrados, aquele que for necessário para a receita. Após inserir-se a quantidade desejada, conforme se pode verificar na Figura 39.

O lúpulo, da mesma forma que o malte, é selecionado dentro uma lista de lúpulos previamente cadastrados (Figura 41). Informar a quantidade que será utilizada e qual a finalidade do uso, se é para fervura ou para *Dry Hop* (Figura 42 ). Segundo Komar (2017) Essa técnica consiste em adicionar o lúpulo nas etapas após o resfriamento da cerveja, ou seja, durante a fermentação ou maturação da bebida. O *Dry Hopping* tem sido muito usado na produção de cervejas especiais, principalmente em estilo de cerveja que buscam ressaltar os sabores e especialmente os aromas de lúpulo fresco.

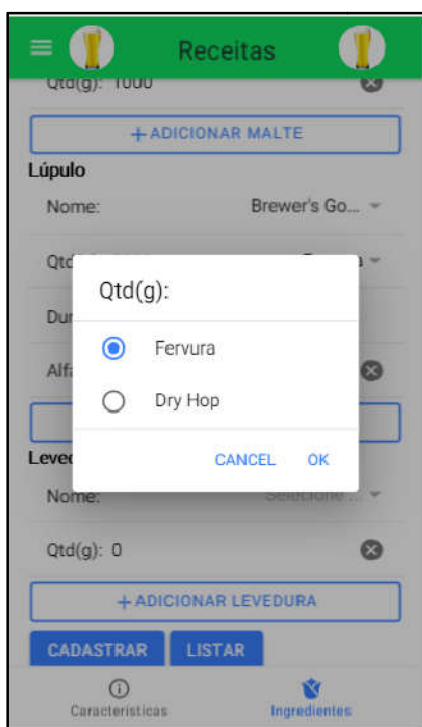


**Figura 39: Tela Cadastro Receita, aba Ingredientes, Malte.**  
Fonte: Autor, 2019

Fonte: Autor, 2019.



**Figura 41: Tela Cadastro Receita, aba Ingredientes, Lúpulo.**  
Fonte: Autor, 2019



**Figura 40: Tela Cadastro Receita, aba Ingredientes, Uso do Lúpulo.**



**Figura 42: Tela Listar Receitas.**  
Fonte: Autor, 2019.

Para cadastrar a levedura, seleciona-se dentre uma lista de leveduras pré-cadastradas e insere a quantidade desejada para a receita. Feito estes passos todos, finalmente pode-se cadastrar a receita no banco de dados do aplicativo



clicando no botão “Cadastrar”. Neste momento, a aplicação direciona para a tela onde tem a lista de receitas cadastradas (Figura 42). Ao clicar sobre uma receita o aplicativo direciona para tela com todos os dados cadastrados para aquela receita (Figura 43). Tela de edição está em desenvolvimento, porém por uma questão de tempo, ainda não está funcionando.



The screenshot shows a mobile application interface for 'Receitas'. The header is green with a hamburger menu icon on the left, a bell icon in the center, and another bell icon on the right. Below the header, the recipe details are listed in a white box with horizontal dividers between each field:

- Nome:** Cream ale (teste 1)
- Descrição:** Primeira receita Cream Ale
- Tipo:** 1
- Cervejeiros:** Marcelo, Rosito
- Data Início:** 2019-10-16
- Data Fim:** 2019-10-30
- Volume:** 70
- Densidade Original (OG):** 1044
- Densidade Final (FG):** 1050
- Amargor:** 44 IBUs
- Cor:** 3 SRM
- Gradação Alcoólica (ABV):** 4.5

At the bottom of the form, there are two blue buttons: 'VOLTAR' and 'EDITAR'.

**Figura 43: Tela Detalhe Receita**

Fonte: Autor, 2019.

## 6. ANÁLISE DO PROTÓTIPO

Nesta seção, apresentam-se os pontos altos e baixos do desenvolvimento do aplicativo utilizando o *Framework Ionic* e a verificação de como o protótipo desenvolvido pode afetar a maneira como os cervejeiros produzem suas cervejas artesanais. Tal análise foi baseada em uma entrevista com seis grupos de cervejeiros. Esta entrevista ocorreu em novembro de 2019, onde o protótipo foi apresentado para avaliação. Após foi realizados um questionário *on-line* com quinze perguntas.

A proposta do estudo era desenvolver um aplicativo híbrido usando o *Framework Ionic*, para poder auxiliar os cervejeiros caseiros, na formulação das receitas, e também poder ser utilizado como um repositório de receitas e ingredientes usados na produção de cerveja artesanal caseira, com a vantagem de poder rodar nas plataformas mais utilizadas atualmente, *Android*, *IOs* e *Windows Phone*.

Conforme abordado na seção 4, para o desenvolvimento do protótipo do aplicativo híbrido foi levado em consideração um questionário aplicado às duas equipes de cervejeiros. Neste questionário, foram levantados dados sobre o perfil e experiência na produção de cerveja artesanal. Quando questionados sobre o uso de algum *software* para auxiliar de alguma forma na produção de cervejas, verificou-se que o aplicativo mais conhecido entre o meio, é o *BeerSmith*, este *software* auxilia na formulação de receitas assim como banco de dados de insumos, porém a licença é paga, e ele é um aplicativo bastante técnico, requerendo conhecimento mais avançado na área.

Utilizando as técnicas e conhecimentos em programação adquiridos ao longo do curso, e com o auxílio do *Framework Ionic*, iniciou-se o desenvolvimento do código fonte que resulta no aplicativo. Neste ponto o *Ionic* é ótimo, o *framework* gera automaticamente todo o código base para o aplicativo, o desenvolvedor deve

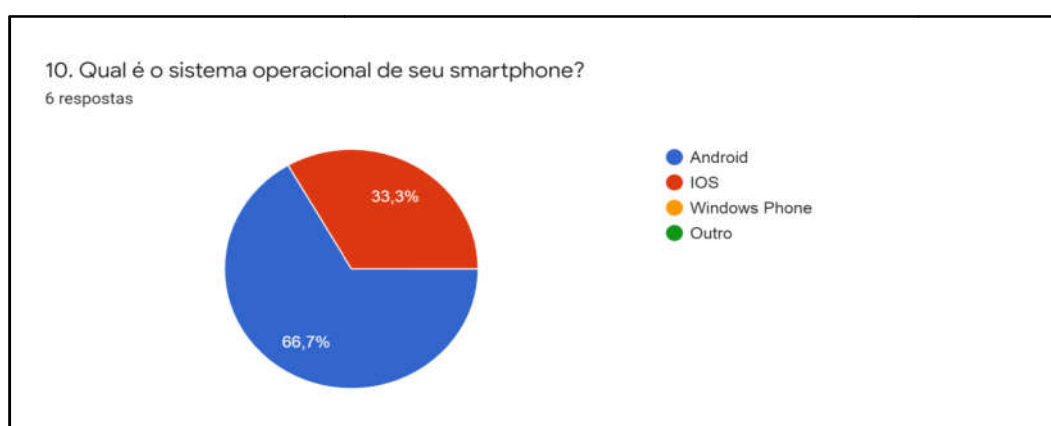
apenas definir o *template* a ser utilizado, a partir, daí ele faz todas as ligações no padrão *MVC (Model-View-Controller)*. Conforme Massari(2017) o *MVC* é um padrão de arquitetura de aplicações que divide a aplicação em três camadas: a visão (*view*), o modelo (*model*), e o controlador (*controller*). O padrão *MVC* foi desenvolvido em 1979 por Trygve Reenskaug com a finalidade de ser utilizado como arquitetura para aplicativos *desktop*. Entretanto, o padrão se popularizou para uso em sistemas *web*, a partir da adesão de milhares de *Frameworks* de mercado.

A utilização do *json-server* criar uma *API REST fake* para armazenamento de dados local, também é uma solução que agiliza muito o processo de desenvolvimento, pois, com menos código fonte consegue-se criar o banco de dados e facilmente operações *CRUD (Create, Read, Update, Delete)*.

De acordo com as respostas do questionário, verificou-se que a faixa etária dos cervejeiros ficou entre 25 e 54 anos de idade e a experiência média com produção de cerveja artesanal ficou em cerca de 4 anos.

Pode-se constatar também que alguns deles sequer usam algum tipo de aplicativo, simplesmente pegam receitas prontas de livros ou na *internet* e produzem a cerveja baseado nestes conteúdos.

Confirmou-se também, a informação de que o *Android* é o sistema operacional mais popular, incluindo o meio cervejeiro, 66,7% utiliza *Android* e 33,3% usa *iOS* (Figura 45).



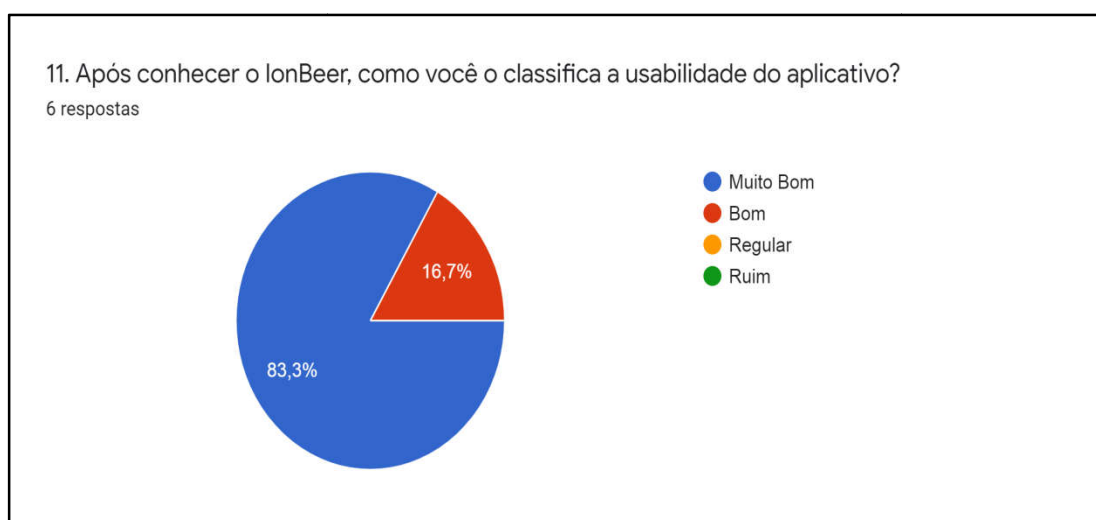
**Figura 44: Gráfico Sistema operacional mais utilizado pelos cervejeiros.**

Fonte: Autor, 2019.

Com base nas respostas dos entrevistados, obteve-se um *feedback* bastante positivo (Figura 45), 83,3% dos entrevistados respondeu que o aplicativo é muito bom e 16,7% disseram que é bom. No geral eles gostaram da interface, que é bem

amigável, navegabilidade muito prática, o *menu* lateral facilitou muito esta questão, pois, a partir dele e do ponto onde o usuário estiver pode-se acessar a todas telas chave. Eles consideram o aplicativo como sendo de fácil aprendizado, intuitivo e direto ao ponto. Quando questionados se às informações armazenadas eram suficientes para formulação da receita, 99% afirmaram positivamente.

Um entrevistado afirmou que pelo fato de o aplicativo não fazer alguns cálculos ele não seria tão preciso para formulação da receita, mas em contraponto ele disse que seria um bom aplicativo para armazenar receitas e insumos para consulta durante a brassagem. Algumas pessoas sugeriram algumas melhorias, inserção informações, itens que poderiam ser considerados para *upgrade* do aplicativo.



**Figura 45: Índice de aprovação do aplicativo.**

Fonte: Autor, 2019.

## 7. CONSIDERAÇÕES FINAIS

O desenvolvimento do presente estudo possibilitou uma análise sobre como o *Framework Ionic* pode auxiliar no desenvolvimento de um aplicativo híbrido para dispositivos móveis para auxiliar na produção de cerveja artesanal. Este estudo mostrou que o Ionic é uma poderosa e completa ferramenta para desenvolvimento de aplicativos híbridos para dispositivos móveis, que está conectada a outros *frameworks*. Pode-se comprovar que a curva de aprendizado é reduzida quando comparada a curva de aprendizado de plataformas nativas, pois com um único código fonte pode-se gerar aplicativos para as plataformas *Android*, *IOs* e *Windows Phone* sem necessidade de nova programação. Já na concepção do projeto, o *Ionic* gera toda a estrutura base da aplicação de acordo com o *template* selecionado. Em questionário com perguntas abertas (APÊNDICE D), o *feedback* do protótipo apresentado aos cervejeiros caseiros foi bastante positivo conforme descrito na seção 6. Protótipo teve uma ótima avaliação quanto a interface e navegabilidade proporcionada ao usuário, uma navegação simples e direta ao ponto. Foi evidenciado de modo geral que o aplicativo atende a necessidade a qual se propõe, que é armazenar ingredientes e armazenar e montar receitas, possibilidade de alterar ou ajustar os dados para que o cervejeiro possa aprimorar sua cerveja artesanal, que é o grande diferencial neste mercado .

O aplicativo foi testado em uma brassagem e os cervejeiros, de modo geral, falaram que a aplicação facilita muito a produção de cervejas, pois com todas as informações na mão deixa o processo mais assertivo com base nas anotações de tempo que eles tinham, agilizou a produção em 20%. No caso de necessitar ajustar a receita, a interface do aplicativo é outro ponto forte, com uma navegação simples e rápida permite fazer correções a qualquer momento, aperfeiçoando assim às receitas. Pode-se dizer então, que aplicativos híbridos têm performance muito similar ao dos aplicativos nativos, com a vantagem de ser multiplataforma, o que é um ótimo negócio, pois viabiliza o desenvolvimento com preços mais competitivos.

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

AAKER, D. A., KUMAR, V., DAY, G. S. Pesquisa de marketing. 2ª ed. São Paulo: Atlas, 2007.

AGASTHYAN Roy, Criando uma API REST Falsa Com json-server. [S.l.] 2017. Traduzido por Eric Patrick. Disponível em: <<https://code.tutsplus.com/pt/tutorials/fake-rest-api-up-and-running-using-json-server--cms-27871>> Acessado em 10 de nov. 2019.

ANGULAR, what is Angular? [S.l.] [2018?] Disponível em <<https://angular.io/docs>>. Acessado em 01 dez. 2018.

BONACCORSI Mauro Manzali, GUIA DE ESTILOS DE CERVEJAS BJCP 2015. [S.l.] 2016. Disponível em <<http://www.brauakademie.com.br/assets/bjcp-2015-beer-pt-br.pdf>>. Acessado em 10 out. 2019.

BOOCH Grady; RUMBAUGH James; JACOBSON Ivair. UML: Guia do Usuário. Tradução Fábio Freitas. 2ª Tiragem. Rio de Janeiro: Campus, 2000. 472p.

BRAUAKADEMIE Guia de Estilos de Cerveja BJCP 2015. [S.l.] [2015]. Disponível em <<http://www.brauakademie.com.br/assets/bjcp-2015-beer-pt-br.pdf>>. Acessado em 05 ago. 2019.

COMOFAZERCERVEJA Matéria Prima. [S.l.] [2019?]. Disponível em <<https://www.comofazercerveja.com.br/materia-prima-cerveja>>. Acessado em 14 nov. 2019.

( \_\_\_\_\_ ) Lei da Pureza, [S.l.][2019?] Disponível em <<https://www.comofazercerveja.com.br/lei-da-pureza>>. Acessado 10 out. 2019.

CORDOVA Apache, Visão Geral [S.I.] [2017?]. Disponível em:

<<https://cordova.apache.org/docs/en/latest/guide/overview/index.html#architecture>>

Acessado em 02 nov. 2019.

CRAFT , Alex, Beer Lúpulo: Pellet X Cryo Hops,[S.I.] 2018. Disponível em:

<<https://alexcraftbeer.wordpress.com/2018/03/25/lupulo-pellet-x-cryo-hops/>>

Acessado em 20 out. 2019.

DEVMEDIA Introdução ao Ionic, [S.I.] [2018?]. Disponível em

<<https://www.devmedia.com.br/guia/ionic/38372>> Acessado em 10 set. 2018.

GONÇALVES Débora, Compreenda as diferenças entre o desenvolvimento nativo e híbrido, [S.I.] 2018 . Disponível em <<https://blog.cronapp.io/compreenda-as-diferencas-entre-o-desenvolvimento-nativo-e-hibrido/>>

Acessado em 06 nov. 2019.

GONÇALVES Eduardo Correia, JSON Tutorial. [S.I.] 2012. Disponível em

<<https://www.devmedia.com.br/json-tutorial/25275>>. Acessado em 12 nov. 2019.

GONÇALVES, Allan J. R., Livro Anais - Artigos e Minicursos, v. 1, n. 1, p. 500-515, jun. 2017.

IDC Analyze The Future, Smartphone Market Share[S.I.] [2017?]. Disponível em:

<<https://www.idc.com/promo/smartphone-market-share/os>> Acessado em 02 dez.

2018;

IONICFRAMEWORK, O que é framework Ionic? [S.I.] [2019?] Disponível em

<<https://ionicframework.com/docs/intro>>, acessado em 01 nov. 2019.

KOMAR Ana Paula, Você sabe o que é Dry Hopping?. [S.I.] 2017. Disponível

em<<https://blog.clubedomalte.com.br/curiosidades-cervejeiras/o-que-e-dry-hopping/>>

Acessado em 14 nov. 2019

KUNZE, Wolfgang; Tecnología Para Cerveceros y Malteros. Primeira edição, p.72, 2006.

LARA Carlos, Cerveja Artesanal: tudo o que você precisa saber a respeito, [S.I] 2018. Disponível em <<https://www.hominilupulo.com.br/cervejas-caseiras/artigos/cerveja-artesanal/>> Acessado em 05 dez. 2018.

LARA Carlos, Quantidade de água na cerveja: aprenda como calcular. [S.I] 2018. Disponível em <<https://www.hominilupulo.com.br/cervejas-caseiras/como-calcular-a-quantidade-de-agua/>> Acessado em 14 nov. 2019.

MASSARI, Jorge. O que é Model-View-Controller(MVC). [S.I] 2017. Disponível em <<https://www.portalgsti.com.br/2017/08/padrao-mvc-arquitetura-model-view-controller.html>> Acessado em 17 nov. 2019.

MORIMOTO, C. E. Smartphones Guia Prático. [S.I.]: Sul Editores, 2009.

NPMJS, TypeScript. [S.I] [2018?]. Disponível em <<https://www.npmjs.com/package/typescript#installing>>. Acessado em 15 nov. 2019.

OLIVEIRA Alexandre, Ions Labs [S.I] 2017. Disponível em <<https://pt.slideshare.net/AlexMontanha/ionic-labs>>. Acessado em 15 out. 2019.

ORTIZ Paulo Rodolfo Buffon, monografia - Análise do Consumo Energético do Processo de Produção de Cerveja Artesanal por Bateladas, p. 1, novembro de 2014.

PORTALGSTI, O Que é Ionic? [S.I] [2018?]. Disponível em <<https://www.portalgsti.com.br/ionic/sobre/>>. Acessado em nov. 2018.

PREZOTTO Ezequiel Douglas, BONIATI Bruno Batista, Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas, [S.I] [2014]. Disponível em <<http://www.eati.info/eati/2014/assets/anais/artigo8.pdf>> Acessado em 08 nov. 2019.

RICARTE Ivan Luiz Marques. Programação Orientada a Objetos: Uma Abordagem com Java [S.I] 2001. Disponível em <<http://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>>. Acessado em 09 de nov. 2019.

ROSENTHAL Rodolfo, HOMIMILUPULO Leveduras cervejeiras: as verdadeiras “mestres cervejeiras” [S.I] 2018. Disponível em



<<https://www.hominilupulo.com.br/cervejas-caseiras/guia-basico/leveduras/>>

acessado em 15 out. 2019.

(\_\_\_\_\_.) Rodolfo, HOMIMILUPULO Malte: o que é e qual seu papel na cerveja? [S.I.] 2018. Disponível em < <https://www.hominilupulo.com.br/cervejas-caseiras/guia-basico/malte/> > acessado em 15 out. de 2019.

SEBRAE Nacional, Microcervejarias ganham espaço no mercado nacional. [S.I.] 2015. Disponível em <<https://respostas.sebrae.com.br/microcervejarias-ganham-espaco-no-mercado-nacional/>> Acessado em 03 dez. 2018.

SILVEIRA, Denise Tolfo; CÓRDOVA, Fernanda Peixoto, Métodos de Pesquisa, Livro 2009. Disponível em <<http://hdl.handle.net/10183/52806>>. Acessado em 01 dez. 2018.

STATCOUNTER, Operating System Market Share Worldwide [S.I.][2019?]. Disponível em <<https://gs.statcounter.com/os-market-share#monthly-201810-201910>> Acessado em 01 nov. 2019.

YOUNG Thomas W., Beer Alcoholic Beverage.[S.I.] [2019?] Disponível em <<https://www.britannica.com/topic/beer>>. Acessado em 15 out. 2019.

ZURIARRAIN José Mendiola, Android já é o sistema operacional mais usado do mundo. [S.I.] 2017. Disponível em <[https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467\\_396232.html](https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467_396232.html)> Acessado em 07 dez. 2018.

## 9. APÊNDICE A

### Questionário sobre utilização de algum software na produção de cerveja artesanal

1. Nome:
2. Idade:
3. Tempo de experiência com produção de cerveja:
4. Qual é o sistema operacional do seu celular?
5. Onde você salva suas receitas?
6. Você conhece algum aplicativo para auxiliar na fabricação de cerveja artesanal que atenda suas necessidades? Explique sua resposta.
7. Como um aplicativo poderia contribuir na produção de cerveja artesanal?
8. Você utiliza um aplicativo que acompanha passo a passo o processo de produção de cerveja artesanal? Se sim, qual?
9. No momento da adição do malte, como o cervejeiro sabe qual o PH da água que está utilizando?
10. Um aplicativo que pudesse cadastrar o PH da água seria útil? SIM/ NÃO.
11. Como o cervejeiro valida o teste de iodo, para conferência dos açúcares?
12. Um aplicativo poderia auxiliar na validação do teste de iodo mostrando o resultado esperado para cada receita? SIM/ NÃO.

13. Um aplicativo que fizesse o acompanhamento passo a passo, durante o processo de produção de cerveja, emitindo alarmes e telas com tempos estimados e medições a serem feitas em cada etapa da produção, seria útil?
14. Como o cervejeiro faz para evoluir sua receita? Como é feito este registro?  
Por exemplo, depois da cerveja pronta, percebe que poderia ajustar a quantidade de determinado ingrediente para uma próxima produção.
15. Além dos processos citados acima, existe mais algum processo que necessite de aferição durante o processo de produção de cerveja artesanal? Como é feita esta aferição?
16. Seria interessante o cervejeiro ter a possibilidade de cadastrar todos insumos de produção de cerveja em um aplicativo?
17. Como é monitorado o tempo de fermentação?
18. Existe alguma funcionalidade no *Beersmith* que poderia ser melhorada?
19. Como é feito o controle dos tempos de cada etapa de produção?
20. Comentários, sugestões?

## 10. APÊNDICE B

### Casos de uso

**Caso de uso:** Manter Lúpulo

**Visão geral:** O cervejeiro realiza o cadastro do Lúpulo no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar e editar os lúpulos cadastrados.

**Caso de uso:** Manter Malte

**Visão geral:** O cervejeiro realiza o cadastro do Malte no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar e editar os Maltes cadastrados.

**Caso de uso:** Manter Levedura

**Visão geral:** O cervejeiro realiza o cadastro da Levedura no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar e editar as Leveduras cadastradas.

**Caso de uso:** Manter Água

**Visão geral:** O cervejeiro realiza o cadastro da Água no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar e editar as Águas cadastradas.

**Caso de uso:** Manter Tipos de Cerveja

**Visão geral:** O cervejeiro realiza o cadastro do tipo de cerveja no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar e editar os tipos de cervejas cadastradas.

**Caso de uso:** Manter Receitas

**Visão geral:** O cervejeiro realiza o cadastro da receita cerveja no aplicativo. O cervejeiro pode realizar quantos cadastros forem necessários, pode também consultar e editar as receitas de cerveja cadastradas. Para realizar o cadastro da receita, o cervejeiro deverá definir um nome, uma breve descrição, data de início e data final de execução da receita. Deverá informar também, a quantidade de litros a serem produzidos e as características da cerveja. O cervejeiro ainda precisa inserir os ingredientes previamente cadastrados e quantificá-los, necessita informar temperatura para adição do malte, tempo de fervura e rampas de temperatura.

## 11. APÊNDICE C

### Estrutura banco de dados JSON

```
{
  "lupulos": [
    {
      "nome": "Brewer's Gold (UK) N",
      "pais": "Reino Unido",
      "alpha": "6",
      "beta": "4",
      "tipo": "Aroma",
      "formato": "Pellet",
      "id": 3
    }
  ],
  "maltes": [
    {
      "nome": "Château Pilsen",
      "cor": 2.5,
      "potencial": 100,
      "id": 13
    }
  ],
  "leveduras": [
    {
      "nome": "levedura",
      "atenuacao": "6",
      "tempMin": "67",
      "tempMax": "85",
      "id": 2
    }
  ],
  "aguas": [
    {
      "origem": "Garibaldi",
      "ph": 5,
      "calcio": 7,
      "magnesio": 2,
      "sodio": 6,
      "sulfato": 2,
    }
  ]
}
```

```
        "bicarbonato": 2,  
        "id": 1  
    }  
],  
"cervejas": [  
    {  
        "id": 1,  
        "nome": "Cream Ale",  
        "descricao": "Cream Ale é um estilo americano de c  
erveja que surgiu no final do século 19, como uma variação das  
populares Pilsners.",  
        "og": 1050,  
        "fg": 1002,  
        "ibu": 1234,  
        "srm": 234,  
        "abv": 4.7  
    }  
],  
"receitas": [  
    {  
        "nome": "Nova Beer",  
        "descricao": "Teste",  
        "tipo": 1,  
        "cervejeiros": "caras",  
        "data_inicio": "2019-08-21",  
        "data_fim": "2019-10-22",  
        "volume": 30,  
        "og": 1090,  
        "fg": 1045,  
        "ibu": 78,  
        "srm": 65,  
        "abv": 7,  
        "aguas": [  
            {  
                "quantidade": 1,  
                "uso": "Brassagem"  
            }  
        ],  
        "maltes": [  
            {  
                "quantidade": 1  
            },  
            {  
                "quantidade": 2  
            }  
        ],  
        "lupulos": [  
            {
```

```
        "quantidade": 1,  
        "uso": "Dry Hop",  
        "tempo": 2,  
        "temperatura": 22,  
        "alfa": 9  
    },  
    ],  
    "leveduras": [  
        {  
            "quantidade": 292  
        }  
    ]  
}  
]  
}
```



## 12. APÊNDICE D

### Convite para Participação em Pesquisa

Caro(a) entrevistado,

Este estudo tem por objetivo refletir sobre a utilização do aplicativo *IonBeer* na produção de cerveja artesanal, por isso sua participação é muito importante. O questionário é anônimo, você não será identificado em momento algum da pesquisa. Algumas informações sobre o perfil (faixa etária, regime de trabalho, formação, entre outras) serão coletadas com a única finalidade de caracterizar a amostra. Além disso, NÃO serão coletados endereços de e-mails dos participantes.

O resultado será tomado para construção dos dados da minha defesa de TCC(Trabalho de Conclusão de Curso), no Programa de Graduação Tecnólogo em Análise e Desenvolvimento de Sistemas, no Estudo sobre o Desenvolvimento de um Aplicativo para Fabricação de Cerveja Artesanal usando *Framework Ionic*.

Antes de proceder com o preenchimento do Questionário solicito, gentilmente, a leitura do Termo de Consentimento Livre e Esclarecido (TCLE). Se concordar em participar, basta assinalar a Manifestação de conhecimento e concordância para dar seguimento à pesquisa, que ficará disponível de 26/10/2019 a 10/11/2019.

Contatos

a) Marcelo Milani - discente no curso de Análise e Desenvolvimento de Sistema no Instituto Federal do Rio Grande do Sul - IFRS - Bento Gonçalves e-mail: marcelomilani1@outlook.com.br

b) Prof.Dr. Maurício Covolan Rosito - Professor e orientador do curso de Análise e Desenvolvimento de Sistema no Instituto Federal do Rio Grande do Sul - IFRS - Bento Gonçalves e-mail: mauricio.rosito@bento.ifrs.edu.br

Obrigado!

Marcelo Milani

### \*Obrigatório

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO – TCLE Esta pesquisa versa sobre utilização do aplicativo *IonBeer* na produção de cerveja artesanal e está sendo desenvolvida por Marcelo Milani, estudante do curso de Análise e Desenvolvimento de Sistemas no Instituto Federal do Rio Grande do Sul - IFRS - Bento Gonçalves, no Estudo sobre o Desenvolvimento de um Aplicativo para Fabricação de Cerveja Artesanal usando *Framework Ionic*. Sob a orientação do Prof. Dr. Maurício Covolan Rosito. A finalidade desta pesquisa é construir parte dos dados para tese de TCC, bem como verificar possíveis melhorias na elaboração do aplicativo. Os benefícios da pesquisa vão além da construção de dados, pois entende-se os dados coletados será possível futuramente aperfeiçoar as funcionalidades do aplicativo. Para amenizar os riscos de danos, as medidas tomadas são as seguintes: utilização de questionário não identificado, sem coleta de e-mails dos respondentes, resguardando o sigilo dos participantes; não apresentação de perfil participante de forma isolada. Ressalta-se, ainda, que a análise será tomada conjuntamente, sem haver juízo de valor ou qualquer interpretação que possa desmerecer, expor, ou indicar os respondentes, e/ou colocá-los em situação de comparação entre colegas de trabalho. Outrossim, haverá zelo pela integridade dos participantes. Solicitamos a sua colaboração para participar da pesquisa por meio do preenchimento dos questionários fechado e aberto, bem como sua autorização para apresentar os resultados deste estudo em eventos da área de tecnologia, publicações em revistas científicas nacionais e/ou internacionais e respectiva tese. Sua adesão é voluntária, portanto, se quiser participar solicita-se, gentilmente, sua concordância aos termos deste instrumento,

bastando clicar na Manifestação de conhecimento e concordância aos termos do instrumento de pesquisa. Atenciosamente, Marcelo Milani. \*

Manifesto conhecimento e concordância aos termos deste instrumento de pesquisa.

## Questionário

### Perfil:

1. Nome: \*

2. Idade: \*

18 anos a 24 anos.

25 anos a 34 anos.

35 anos a 44 anos.

45 anos a 54 anos.

55 anos a 64 anos.

3. Gênero: \*

Masculino.

Feminino.

Outro.

Prefiro não dizer.

4. Cidade/Estado:

5. Ano em que começou a produção de cerveja artesanal (data aproximada)?

6. Nome de seu grupo de cervejeiros artesanal: \*

7. Informe quantos integrantes fazem parte deste grupo: \*

8. Você utiliza algum aplicativo para acompanhar a produção de cerveja? Qual? \*

10. Qual é o sistema operacional de seu *smartphone*? \*

*Android*

*IOS*

*Windows Phone*

Outro

11. Após conhecer o *IonBeer*, como você o classifica a usabilidade do aplicativo?

Muito Bom

Bom

Regular

Ruim

12. As informações armazenadas no aplicativo são suficientes para formulação de uma receita para cerveja artesanal? Explique. \*

13. Você indicaria o aplicativo a um amigo?

Com certeza

Sim

Talvez

Não

14. Quais são os pontos positivos observados neste aplicativo? \*

15. Quais são os pontos negativos observados neste aplicativo? \*