

**UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES
- CAMPUS DE ERECHIM**

**DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO CURSO
DE CIÊNCIA DA COMPUTAÇÃO**

JACKSON FELIPE MAGNABOSCO

**DESENVOLVIMENTO DE APLICATIVO DE CONTROLE AUTOMATIZADO NO
PROCESSO DE PRODUÇÃO DE CERVEJA ARTESANAL**

ERECHIM - RS 2020

JACKSON FELIPE MAGNABOSCO

**DESENVOLVIMENTO DE APLICATIVO DE CONTROLE AUTOMATIZADO NO
PROCESSO DE PRODUÇÃO DE CERVEJA ARTESANAL**

**Projeto de conclusão de curso, apresentado ao
Departamento de Engenharias e Ciências da
Computação Universidade Regional Integrada do
Alto Uruguai e das Missões – Campus de Erechim.**

ERECHIM - RS 2020

RESUMO

O objetivo principal deste projeto consiste em apresentar uma solução para o controle automatizado no processo de produção de cerveja artesanal, levando em conta uma produção de até cem litros, proporcionando maior segurança, agilidade, produtividade e redução de custos. Este trabalho apresenta o desenvolvimento de um aplicativo onde foram utilizados, a metodologia orientada a objeto e as ferramentas Android Studio com a plataforma de desenvolvimento Dart com o framework Flutter, juntamente com o banco de dados do Firebase. O presente trabalho visa aplicar a estrutura do módulo ESP8266 utilizando a linguagem de programação C no ambiente de desenvolvimento integrado do Arduino, juntamente com o sensor de temperatura DS18B20 a prova d'água para medir a temperatura em todos os processos de produção da cerveja, desde a fervura, resfriamento, fermentação até a maturação, avaliando os dados obtidos por meio da temperatura no sensor serão processados e transformados em gráficos e tabelas, para que possam ser analisados, possibilitando concluir se todos os processos de produção da cerveja artesanal ocorreram de forma correta.

Palavras-chave: Automação. Aplicativo. Serviço Web

ABSTRACT

The main objective of this project is to present a solution for automated control in the craft beer production process, taking into account a production of up to one hundred liters, providing greater safety, agility, productivity and cost reduction. This work presents the development of an application where they were used, the object-oriented methodology and the Android Studio tools with the Dart development platform with the Flutter framework, together with the Firebase database. The present work aims to apply the ESP8266 module structure using the C programming language in the Arduino integrated development environment, together with the waterproof DS18B20 temperature sensor to measure the temperature in all beer production processes, from boiling, cooling, fermentation until maturation, evaluating the data obtained through the temperature in the sensor will be processed and transformed into graphs and tables, so that they can be analyzed, making it possible to conclude if all the craft beer production processes occurred correctly.

Keywords: Automation. App. Web Service

LISTA DE TABELAS

Tabela 1 - Ferramentas integradas.....	17
--	----

LISTA DE QUADROS

Quadro 01 – Cronograma de atividades	21
--	----

LISTA DE ILUSTRAÇÕES

Figura 01 – Arquitetura.....	09
Figura 02 – Analyzer.....	10
Figura 03 – Widget.....	11
Figura 04 – HTML/CSS análogos em Flutter.....	11
Figura 05 - Cupertino e material.....	12
Figura 06 - OneSignal - push notification.....	13
Figura 07 - NoSQL.....	14
Figura 08 - Provedores de login.....	16
Figura 09 - Sensor DS18B20.....	16
Figura 10 - Microcontrolador ESP8266.....	17
Figura 11 - Módulo ESP8266 NodeMCU.....	18
Figura 12 - BreadBoard.....	19

SUMÁRIO

1 Introdução	8
2 Objetivos	9
2.1 Objetivo Geral	9
2.2 Objetivos Específicos	9
3 Justificativa	10
4 Referencial Teórico	11
4.1 Flutter	11
4.1.1 OneSignal	14
4.2 Firebase	15
4.2.1 Autenticação	17
4.3 Sensor inteligente	18
4.3.1 Sensor DS18B20	18
4.3.1.1 Protocolo One-Wire	19
4.3.2 Módulo Wifi ESP8266 NodeMCU CP2102 ESP-12E	19
4.3.2.1 ESP8266	19
4.3.2.2 NodeMCU	20
4.3.3 BreadBoard	20
5 Metodologia	21
6 Cronograma	22
7 Resultados Esperados	23
8 Referência	24

1 Introdução

Com diferentes notas, aromas e sabores, as cervejas artesanais estão ganhando cada vez mais espaço no mercado brasileiro.

No mês de março, o Ministério da agricultura, pecuária e abastecimento lançou o anuário da cerveja 2019, o qual traz dados sobre a atividade cervejeira no Brasil nos últimos anos. Segundo o estudo, o crescimento no setor vem avançando de forma sustentada e traz números registrados de cervejarias e de cervejas que confirmam essa tendência. O país atingiu a marca de 1209 cervejarias registradas em 26 estados. Só em 2019 foram 320 novas cervejarias, ou seja quase uma nova marca foi aberta por dia no país (SALVADOR, 2020).

A busca pela cerveja artesanal cresceu, gerando demanda e consequentemente uma maior movimentação financeira no setor e participação na economia do país.

O Brasil é considerado o terceiro maior produtor de cerveja artesanal do mundo, estimulando uma indústria que fatura R\$ 100 bilhões por ano, atrás apenas dos Estados Unidos e da China (SARIS, 2019).

A partir dessa motivação o aplicativo vem se desenvolvendo para atender melhor os cervejeiros caseiros e trazer um controle mais adequado de todo o processo de produção.

Neste trabalho é desenvolvido um aplicativo, utilizando a linguagem de programação Dart com framework Flutter no ambiente de desenvolvimento integrado Android Studio e o banco de dados do Firebase que tem uma interface simples, objetiva e intuitiva.

Também é utilizado um módulo wifi ESP8266 juntamente com o sensor de temperatura DS18B20 a prova d'água para medir a temperatura em todos os processos de produção da cerveja enviando notificações por push para alertar alguma mudança brusca de temperatura nos processos ou alteração de processo,

Há várias vantagens em se adotar a abordagem de desenvolvimento multiplataforma, pois não depende do dispositivo, removendo uma barreira entre ambientes computacionais quanto à barreira física à utilização do aplicativo.

A principal vantagem é a facilidade no acesso, em qualquer local é possível facilmente acessar as informações contidas na aplicação.

Optou-se pelo modelo orientado à objetos pela sua facilidade e modularização dos componentes.

2 Objetivos

2.1 Objetivo Geral

O objetivo principal deste trabalho consiste em apresentar uma solução para controle automatizado no processo de produção de cerveja artesanal, levando em conta uma produção caseira de até cem litros.

2.2 Objetivos Específicos

Os objetivos específicos deste projeto são:

- Desenvolver um aplicativo utilizando a linguagem de programação Dart com o framework Flutter aplicando no banco de dados do Firebase.
- Estruturar o módulo ESP8266 com o sensor de temperatura DS18B20 à prova d'água para medir a temperatura em todos os processo de produção da cerveja (fervura, resfriamento, fermentação e maturação).
- Disparar push notification para o aplicativo informando sobre situações de alerta, como mudanças bruscas de temperatura.
- Implementar um cronômetro para estimar o tempo que cada processo deve ficar em determinada temperatura na rampa de fervura.
- Avaliar os dados obtidos do sensor de temperatura que foram enviados ao banco de dados pelo módulo ESP8266 para concluir que todos os processos ocorreram de forma correta na receita produzida.

3 Justificativa

Com o apreço a degustação da cerveja e o paladar apurado dos consumidores que procuram por uma cerveja mais lupulada e puro malte, as nano e micro cervejarias tentam suprir essa demanda, a qual, normalmente não é satisfeita pelos grandes fabricantes.

Nesse âmbito, afloram também os intitulados *homebrewing* ou seja os cervejeiros caseiros, que produzem a própria cerveja em casa, buscando desenvolver uma bebida de qualidade, seguindo as receitas tradicionais de produção dos antigos mestres cervejeiros e seguindo a lei da pureza alemã.

Esta nova plataforma propõe medir a temperatura em todos os processos de produção da cerveja, desde a fervura, resfriamento, fermentação até a maturação assim permitindo um acompanhamento em tempo real da temperatura do processo.

Para maior segurança, diferentes níveis de acesso estão em vigor, permitindo que diferentes usuários tenham acesso. Os níveis de acesso incluem gerentes, administradores e balconistas.

Este cenário acarretou o desenvolvimento deste projeto, utilizando hardware e software para a automação, com a finalidade de melhorar o controle do processo de produção de cerveja artesanal.

4 Referencial Teórico

Para produzir o projeto, é necessário ter conhecimento das tecnologias a serem utilizadas e também do ambiente de desenvolvimento do mesmo. As seções a seguir apresentam a base teórica que constitui a pesquisa a ser realizada neste projeto.

4.1 Flutter

Desenvolvido pela equipe da Google, o Flutter é um framework de desenvolvimento híbrido que utiliza a linguagem de programação Dart no ambiente de desenvolvimento integrado Android Studio para criação de seus aplicativos. A principal ideia de utilizar ele no projeto é, ele é totalmente multiplataforma, de código aberto e gratuito, com uma licença limpa e também é um padrão da ECMA com mais de 100 contribuidores externos.

Desenvolve-se apenas um código que vai rodar em diversos dispositivos. Uma tecnologia de fácil aprendizado, especialmente para quem já programou em alguma outra linguagem de programação, pois carrega consigo algumas características similares como classes, orientação a objetos e fortemente tipada.

O Flutter utiliza uma máquina virtual chamada Dart VM que serve de intérprete para imitar a arquitetura do hardware da máquina quando o software for executado. Esta máquina virtual facilita a portabilidade de um idioma para novas plataformas. Sua arquitetura é feita por uma engine que foi totalmente desenvolvida na linguagem de programação C++, como mostra a Figura 01, *Ahead Of Time*, convertendo o código para nativo, ou seja binário. (LIMA, 2019).

Figura 01 – Arquitetura



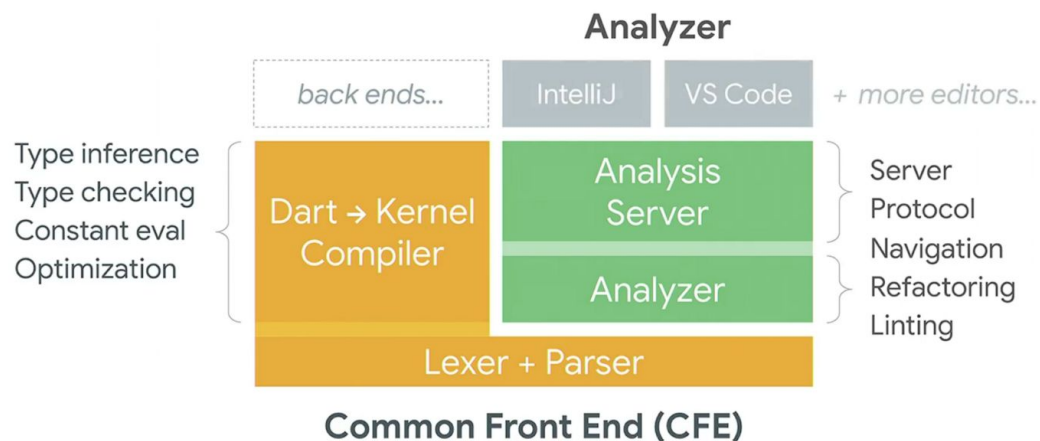
Fonte: Adaptado de (HONDA, 2019)

Outra vantagem que torna indispensável é o *Just in time*, a recompilação do código no dispositivo, enquanto o aplicativo está rodando, a aplicação não perde o estado de

desenvolvimento. Isso gera um ciclo de desenvolvimento muito rápido e produtivo, possibilitando o recarregamento expresso do aplicativo em tempo de execução. (LIMA, 2019).

O compilador do Dart dispõe de um Kernel, onde a linguagem é processada, o Analyzer, tem como objetivo realizar a navegação, refatoração e adequação do código, o Analysis Server, atua como um servidor local que apresenta o código para os editores. Todo o processo funciona de maneira eficiente como mostra a Figura 02. (HONDA, 2019)

Figura 02 – Analyzer

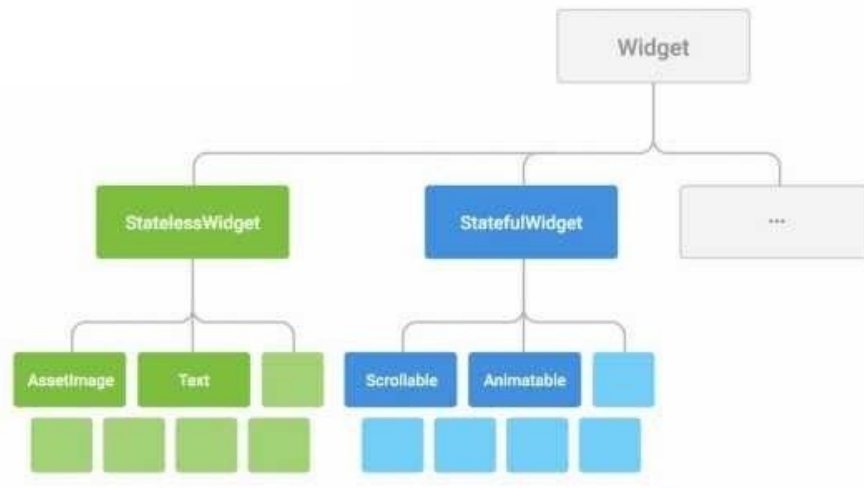


Fonte: Adaptado de (HONDA, 2019)

O Flutter se preocupa muito com a performance e limitação de hardware, assim utiliza 60 quadros por segundos nos aplicativos, Com o uso do código nativo, ele não exibe uma interface lenta, dispõe de um renderizador *Mobile First* apressurado por GPU para que dê-se contextura da UI entre as plataformas e o dispositivo.

Esta linguagem de programação é baseada em *widgets*, que são os componentes da aplicação que interagem com o APP. Tudo no Flutter são widgets, desde o texto até os botões, ou seja, uma árvore de widgets para o widget conforme a Figura 3 demonstra. (DIAS, 2018)

Figura 03 – Widget



Fonte: Adaptado de (DIAS, 2018)

A estrutura do Flutter é similar a estrutura do HTML/CSS na Figura 04 demonstra o HTML/CSS do lado esquerdo e o Flutter do lado direito.

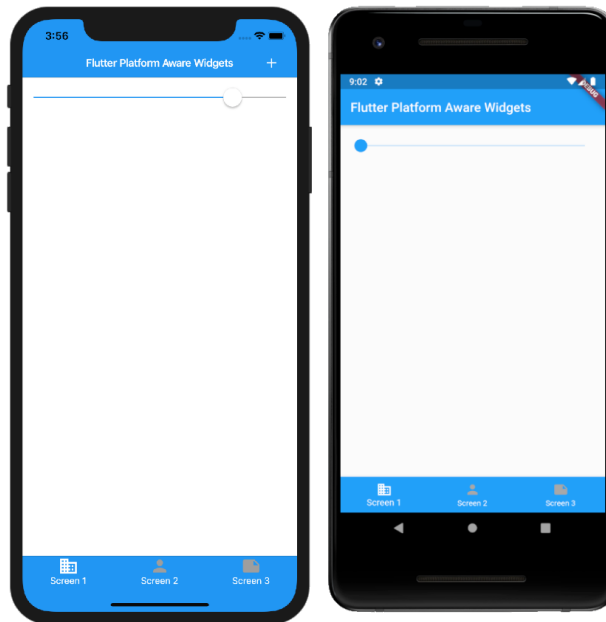
Figura 04 - HTML/CSS análogos em Flutter

<pre> <div class = "greybox" > Lorem ipsum </div> .greybox { background-color: #e0e0e0; width: 320px; height: 240px; font: 900 24px Georgia; } </pre>	<pre> var container = Container(child: Text("Lorem ipsum", style: TextStyle(fontSize: 24.0 fontWeight: FontWeight.w900, fontFamily: "Georgia",),), width: 320.0, height: 240.0, color: Colors.grey[300]); </pre>
--	--

Fonte: Adaptado de (DIAS, 2018)

O flutter tem tanto o sistema de design do Android, que é chamado de material, quanto o sistema de design do iOS, que é chamado de cupertino, assim simulando as duas interfaces como demonstra a Figura 05.

Figura 05 - Cupertino e material



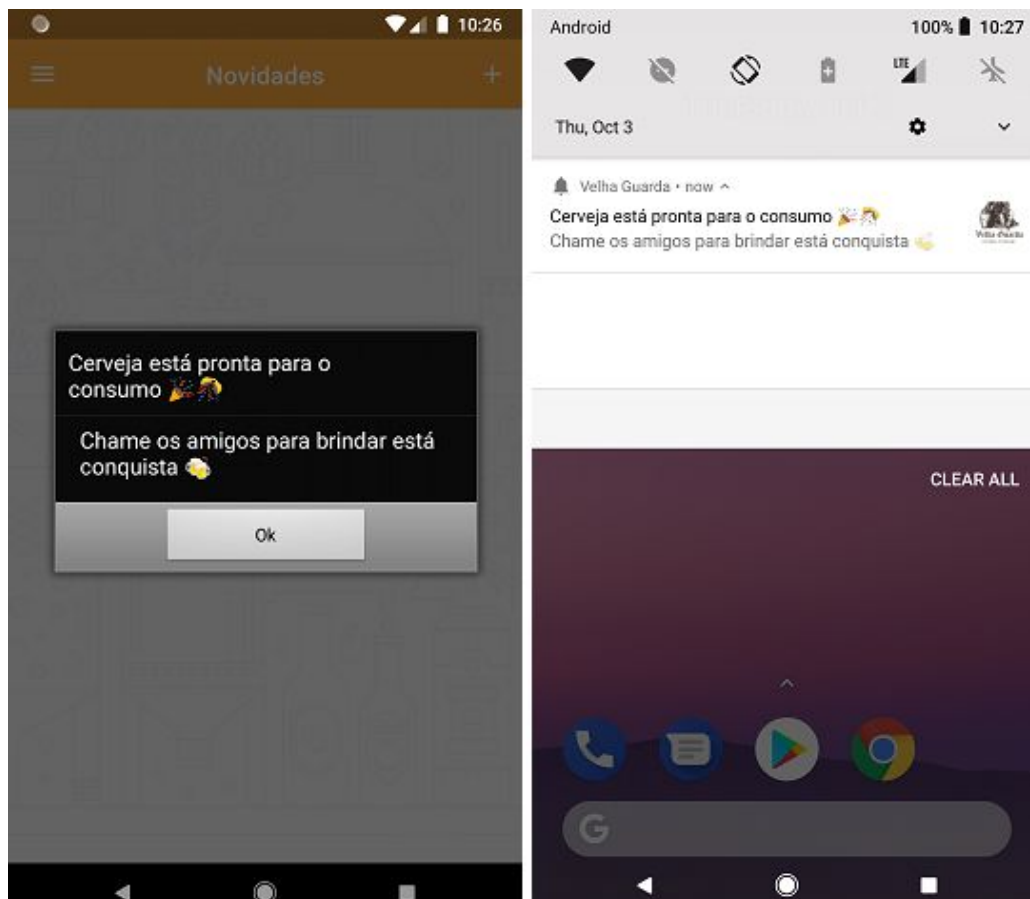
Fonte: Adaptado de (MOORE, 2019)

4.1.1 OneSignal

Segundo (SAVIO, 2019), o OneSignal é uma plataforma com plano gratuito que se comunica com as ferramentas do firebase e serve para impulsionar push móvel, web push, email e mensagens no aplicativo. O SDK facilita a integração dos aplicativos no Flutter.

As notificações por push são o principal fator para alertar alguma mudança brusca de temperatura nos processos de produção da cerveja artesanal ou mudança de processo como demonstra a Figura 06.

Figura 06 - OneSignal - push notification



Fonte: Adaptado de (JACKSON, 2019)

4.2 Firebase

O Firebase é uma plataforma móvel desenvolvida pela equipe da Google com diversas ferramentas integradas que servem para criar aplicativos de alta produtividade. Utiliza-se um banco de dados não relacional, ou seja, não utiliza SQL, quer dizer que não possui como padrão o sistema de tabelas e relacionamentos entre dados, sobretudo tratando cada informação como uma árvore, com um tronco principal e várias raízes como seus nós, onde cada raiz pode ter outras sub-raízes como demonstra a Figura 07. (PEDRO, 2018)

Figura 07 - NoSQL



Fonte: Adaptado de (PEDRO, 2018)

Existem vantagens por utilizar um banco de dados NoSQL no lugar do SQL comum, uma delas é o fácil desenvolvimento, utilização e ordenação. Ele é bem simples, com uma breve leitura da documentação todo o projeto estará arrematado para ser realizado, convertendo em um local impecável para propostas em fases introdutórias de validação.

A utilização e ordenação são bem simples de assimilar pois utiliza o padrão JSON para gravar os dados. Outro grande benefício é a agilidade em queries e updates, oposto aos bancos de dados comuns, a aplicação do acesso e nós ao contrário de tabelas amplia a rapidez de resposta da query pois ela está sendo realizada pelo próprio cliente.

No contexto, o Firebase não fará uma query em diferentes nós e não vai retornar os dados simplificados, basicamente é preciso estabelecer isso tudo manualmente. Assim, fazendo com que seja desnecessário qualquer comunicação direta entre nós para criar relacionamentos entre dados pois, basicamente criando essa comunicação no código.

O lado ruim é exatamente a falta de praticidade desse método pois tudo deve ser feito pelo seu código, então cabe avaliar os prós e os contras de projeto a projeto (PEDRO, 2018).

Melhor performance para grandes volumes de dados no formato NoSQL são os ideais para um número abundante de dados pois a não presença de comunicação entre os nós

diminui o tempo entre uma busca e a inserção, um grande exemplo de onde é utilizado é o BigData, bancos de dados comuns não foram feitos para este tipo de uso.

Uma das principais vantagens é a flexibilidade da estrutura, com a utilização do JSON permite que seus objetos não sejam limitados a colunas e linhas. Um objeto de um nó pode ter diversos outros sub-nós.

As ferramentas integradas podem ser divididas em dois grupos, o primeiro grupo é denominado de desenvolvimento e o segundo grupo é denominado qualidade.









Desenvolvimento	Qualidade
Realtime Database	Firebase Analytics
Auth	Invites
Test Lab	Cloud Messaging
Crashlytics	Predictions
Cloud Functions	AdMob
Firestore	Dynamic Links
Cloud Storage	Adwords
Performance Monitoring	Remote Config
Crash Reporting	App Indexing
Hosting	

Tabela 1 - Ferramentas integradas

4.2.1 Autenticação

Ela possibilita um sistema de registro simples e ágil, essa ferramenta possui aplicações integradas através de redes sociais, logins anônimos ou por usuário e senha criado na própria aplicação, a viabilidade de utilizar o sistema de recuperação de senhas interno do Firebase com envio de e-mails como demonstra a Figura 08.

Figura 08 - Provedores de login

Provedor	Status
 E-mail/senha	Ativado
 Smartphone	Desativado
 Google	Desativado
 Play Games	Desativado
 Facebook	Desativado
 Twitter	Desativado
 GitHub	Desativado
 Anônimo	Desativado

Fonte: Adaptado de (PEDRO, 2018)

A principal ideia de utilizar o Firebase no projeto é por ser uma plataforma móvel completa e robusta de fácil uso, com a viabilidade de trabalhar *offline* e sincronizar automaticamente ao retornar a conexão, utiliza sincronização em tempo de execução, tem plano gratuito e atende muitos cenários utilizando ferramentas que vão economizar tempo e ajudar a centralizar os recursos do sistema.

4.3 Sensor inteligente

4.3.1 Sensor DS18B20

O sensor DS18B20 é um termômetro digital à prova d'água produzido pela Dallas Instruments como demonstra a Figura 09.

Figura 09 - Sensor DS18B20



Fonte: Adaptado de (PEDRO, 2018)

Este dispositivo auxilia a monitorar o nível de temperatura de um determinado processo com a finalidade de gerar de forma correta e nas condições adequadas para o funcionamento apropriado do processo.

A principal motivação para usar este dispositivo no projeto é baseada no argumento de (MADEIRA, 2018), fundamenta que, o sensor é capaz de ler a temperatura, interpretá-la e enviar a informação do valor de temperatura em graus Celsius para o microcontrolador usando um barramento de apenas um fio utilizando o protocolo de comunicação One-Wire.

Este sensor está isolado e pode medir temperaturas entre -55 °C e 125 °C com uma precisão 0,5 °C. Existe uma ótima compatibilidade com o módulo ESP8266, pois utilizam unicamente um pino digital, tendo potencial de conectar múltiplos sensores no mesmo pino (ROVAI, 2019).

Uma grande vantagem em utilizar este sensor é o alarme programável, sendo possível disparar um sinal informando sobre situações de alerta, como mudanças bruscas de temperatura. Essa configuração é gravada em uma memória não volátil.

4.3.1.1 Protocolo One-Wire

O protocolo de comunicação aplicado para realizar com que os dados resultantes do sensor de temperatura DS18B20 se comunique com o módulo ESP8266 é o protocolo One-Wire.

Este protocolo consiste na comunicação através de um barramento, ou seja, uma linha única para transmissão de dados, na qual, podem ser conectados vários dispositivos, de modo que estes possam trocar informações com o módulo.

4.3.2 Módulo Wifi ESP8266 NodeMCU CP2102 ESP-12E

4.3.2.1 ESP8266

O ESP8266 é um microcontrolador com conexão a Internet. Ou seja, ele é similar a um arduino com integração Wi-Fi como demonstra a Figura 10.

Figura 10 - Microcontrolador ESP8266



Fonte: Adaptado de (MORAIS, 2017)

Segundo o argumento de (MORAIS, 2017), esta placa minúscula que também é chamada de módulo tem um potencial muito maior do que aparenta comparando diretamente com o arduino.

4.3.2.2 NodeMCU

Esta versão do ESP8266-12 inclui uma placa de desenvolvimento com uma estrutura que pode ser utilizada na criação de diversos projetos de automação no ambiente de desenvolvimento integrado Arduino. Este módulo contém um conversor serial mais um regulador de tensão 3.3V próprio, não sendo necessário ambos como nas outras versões anteriores. Também há pinos próprios para I2C, SPI, Analógico e outros, conforme apresenta a Figura 11.

Figura 11 - Módulo ESP8266 NodeMCU

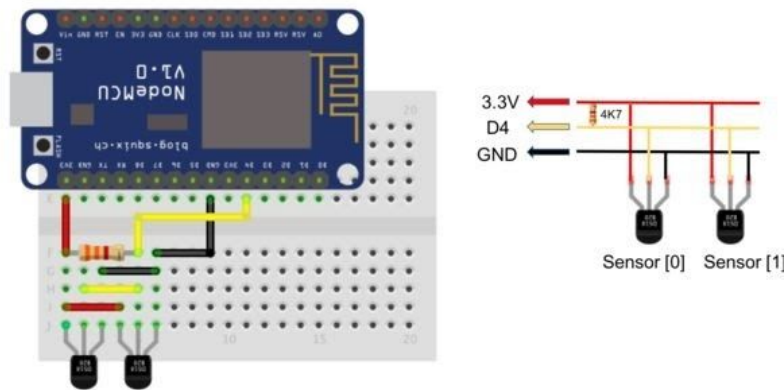


Fonte: Adaptado de (MADEIRA, 2018)

4.3.3 BreadBoard

Esta pequena placa retangular que possui muitos orifícios, serve para alojar componentes eletrônicos nela. Os furos na placa são interconectados de maneira a facilitar a conexão dos componentes como demonstra a Figura 12.

Figura 12 -BreadBoard



Fonte: Adaptado de (ROVAI, 2019)

5 Metodologia

A metodologia a ser empregada neste trabalho consiste no estudo, especificação, implementação e análise dos resultados obtidos a partir da implementação de um aplicativo desenvolvido na linguagem de programação dart com o framework flutter para automação de uma cervejaria, integrando um sensor DS18B20 e um módulo wifi ESP8266 para fazer a comunicação entre o sensor e a aplicação.

Primeiramente, será realizado um estudo aprofundado das tecnologias de flutter e arduino a serem utilizadas, a fim de se adquirir conhecimento para o desenvolvimento da integração do hardware com o software.

Em paralelo, também será estudado o funcionamento da plataforma móvel do firebase, com o objetivo de se ter domínio de suas ferramentas integradas. Posterior a estes estudos, será discutida e planejada a autenticação dos usuários e o método para impulsionar push móvel, web push, email e mensagens no aplicativo.

Em seguida será planejada a arquitetura a ser empregada, será dado início a escolha dos hardwares e testes de desempenho e segurança do mesmo. Primeiro será avaliado o tempo de execução das soluções em si.

Para isso, será comparado o desempenho de execução do aplicativo em dispositivo emulado virtualmente e depois em um dispositivo físico.

Este teste tem por objetivo verificar se a aplicação não apresentaram um resultado diferente quando executadas em algum dos dispositivos.

Os dados obtidos por meio da temperatura no sensor serão processados e transformados em gráficos e tabelas, para que possam ser analisados, possibilitando concluir se todos os processos de produção da cerveja artesanal ocorreram de forma correta.

6 Cronograma

O Quadro 1 apresenta o cronograma de atividades para o projeto proposto.

Quadro 1 – Cronograma de Atividades.

Atividades	Abr	Mai	Jun	Jul	Set	Out	Nov	Dez
Revisão bibliográfica	X	X						
Revisão da monografia		X	X	X	X	X	X	
Estudos das tecnologias envolvidas	X	X	X					
Testes de desempenho da linguagem em Dart	X	X	X					
Análise dos Resultados		X	X	X				
Avaliação de desempenho do aplicativo			X	X	X			
Entrega da monografia								X
Elaboração da apresentação							X	X
Defesa da monografia								X
Entrega da versão final								X

Sequência para o desenvolvimento deste trabalho:

1. Revisão Bibliográfica - Pesquisa sobre o assunto, análise de trabalhos relacionados e estudo do conteúdo bibliográfico encontrado.
2. Revisão da monografia - Escrever a monografia, baseado no Levantamento Bibliográfico e nas atividades referentes ao projeto.
3. Estudos das tecnologias envolvidas - Estudo sobre o Flutter, Firebase e automação com o módulo ESP8266 NodeMCU e sensor de temperatura DS18B20
4. Testes de desempenho da linguagem em Dart - Realização de testes de funcionalidade e coleta de dados para análise.
5. Análise dos Resultados - Analisar os resultados obtidos.
6. Avaliação de desempenho do aplicativo - Avaliar teste em um dispositivo físico e dispositivo emulado.
7. Entrega da Monografia: Entrega da Monografia para o orientador e demais professor do Curso.
8. Elaboração da apresentação: montar apresentação da Monografia.
9. Defesa da Monografia: Apresentação do Trabalho de Conclusão de Curso.
10. Entrega da versão final da Monografia: Entrega da Monografia, com as correções realizadas.

7 Resultados Esperados

Com a conclusão deste trabalho espera-se obter uma análise dos dados obtidos através das ferramentas utilizadas, também presume-se proporcionar um serviço com maior segurança, agilidade, produtividade e redução de custos.

8 Referência

SALVADOR, Douglas. Novidades do mercado. Cerveja de todos os jeitos, Curitiba, v. 86, n. 7, p. 3-3, abril. 2020.

SOMMERVILLE, Ian. Engenharia de Software. 8. Ed. São Paulo: Pearson Addison Wesley, 2007

MCLAUGHLIN, B.; POLLISE, G.; WESTHEAD, D. Use a Cabeça! Análise e Projeto Orientado a Objeto. 1ª ed., Alta Books, 2007.

HUNT, Andrew; Thomas, David. O Programador Pragmático. 1ª ed., Bookman, 2010.

MEDEIROS, Ernani Sales de. Desenvolvendo Software com UML 2.0: definitivo.1. ed. São Paulo, 2004

SARIS, Simoni. Um olhar mais econômico para a cerveja artesanal. Disponível em: <<https://www.folhadelondrina.com.br/economia/um-olhar-mais-economico-para-a-cerveja-artesanal-2971424e.html>> . Acesso em: 24 abril. 2020

MARCUSSO, Eduardo. Anuário da cerveja 2019. Disponível em: <<https://www.cervesia.com.br/noticias/noticias-de-mercado-ervejeiro/7759-anuario-da-cerveja-2019.html>> . Acesso em: 29 abril. 2020

Hipsters ponto tech: Flutter – Hipsters #183. Entrevistadores: Paulo Silveira, Igor Borges, Alexandre Freire, Guilherme Silveira, Alex Vieira, Fausto Blanco, Gabriel Sávio, Roberta Arcoverde. Produtora: Alura, 14 jan. 2020. Podcast. Disponível em: <<https://open.spotify.com/episode/08VJEPbd6EmQhyle3ktT9n>>. Acesso em: 24 abril. 2020.

DIAS, Diego. Por que usar Flutter. Disponível em: <https://www.flutterbrasil.com/read-blog/1_1-por-que-usar-flutter.html>. Acesso em: 24 abril. 2020

LIMA, Alexandre. Por Flutter usa dart. Disponível em: <<https://alexandredslima.com/2019/10/06/por-que-flutter-usa-dart/>>. Acesso em: 24 abril. 2020

HONDA, Rafael. Nossa reunião sobre Flutter e Dart. Disponível em: <<https://medium.com/mega-senior/google-i-o-19-nosso-resum%C3%A3o-sobre-flutter-e-dart-402611573b23>>. Acesso em: 24 abril. 2020

MOORE, Kevin. Platform-Aware Widgets in Flutter. Disponível em: <<https://www.raywenderlich.com/4968762-platform-aware-widgets-in-flutter>> . Acesso em: 25 abril. 2020

PEDRO, João. Firebase — como, quando e porque utilizar esse Banco de Dados do Google. Disponível em: <<https://medium.com/@dezembro/firebase-como-quando-e-porque-utilizar-esse-banco-de-dados-do-google-f65ab5ae182a>> . Acesso em: 25 abril. 2020

SAVIO, Gabriel. Push Notifications - Flutter com OneSignal. Disponível em: <<https://medium.com/@gbrlsavio2/push-notifications-flutter-com-onesignal-fd5f68ead24d>> . Acesso em: 05 maio. 2020

MAGNABOSCO, Jackson. Aplicativo para automação de uma micro cervejaria. Disponível em: <https://github.com/jacksonn455/velha_guarda> . Acesso em: 04 maio. 2020

MADEIRA, Daniel. DS18B20 – Sensor de temperatura inteligente. Disponível em: <<https://portal.vidadesilicio.com.br/sensor-de-temperatura-ds18b20/5>> . Acesso em: 27 abril. 2020

ROVAI, Marcelo. O IOT FEITO SIMPLES: MONITORANDO MÚLTIPLOS SENSORES. Disponível em: <<https://tudosobreiot.com.br/o-iot-feito-simples-monitorando-multiplos-sensores/>> . Acesso em: 27 abril. 2020

ROVAI, Marcelo. O IoT feito simples: Monitorando a temperatura desde qualquer lugar. Disponível em: <<http://labdegaragem.com/profiles/blogs/o-iot-feito-simples-monitorando-a-temperatura-desde-qualquer>> . Acesso em: 27 abril. 2020

MORAIS, José. O QUE É ESP8266 - A FAMÍLIA ESP E O NODEMCU. Disponível em: <<https://portal.vidadesilicio.com.br/o-que-esp8266-nodemcu/>>. Acesso em: 27 abril. 2020