

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 da 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 por que 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. É 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, utiliza-se *Ahead Of Time* convertendo o código para nativo ou seja em binário. (LIMA, 2019).

Figura 01 – Arquitetura



Fonte: Adaptado de (HONDA, 2019)

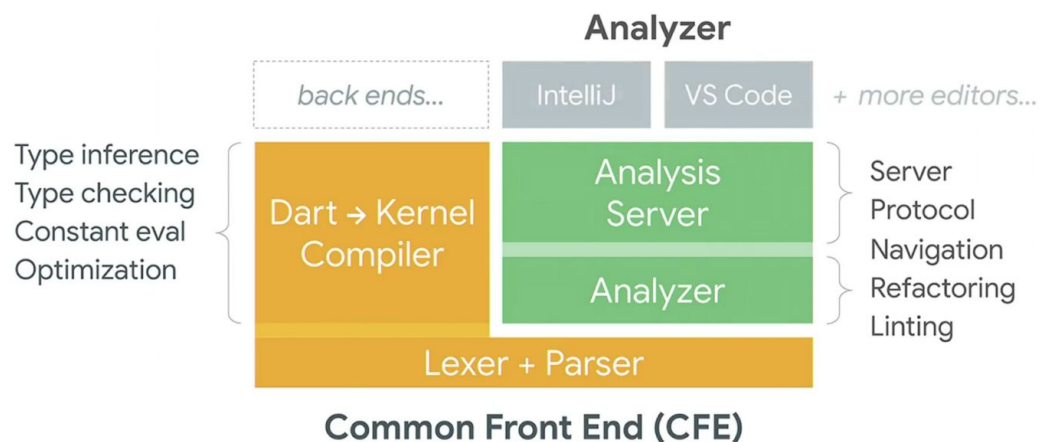
Outra vantagem que torna indispensável é o *Just in time*, possibilita a recompilação do código no dispositivo, enquanto o aplicativo está rodando, isso faz com que a aplicação não perca o estado de desenvolvimento. Isso gera um ciclo de desenvolvimento muito rápido e

produtivo, possibilitando o re-carregamento expresso do aplicativo em tempo de execução. (LIMA, 2019)

Este recurso é chamado *hot reload* sendo mais um benefício pois quando uma linha de código é alterada, ao salvar, ele automaticamente executa a alteração no dispositivo virtual ou físico.

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

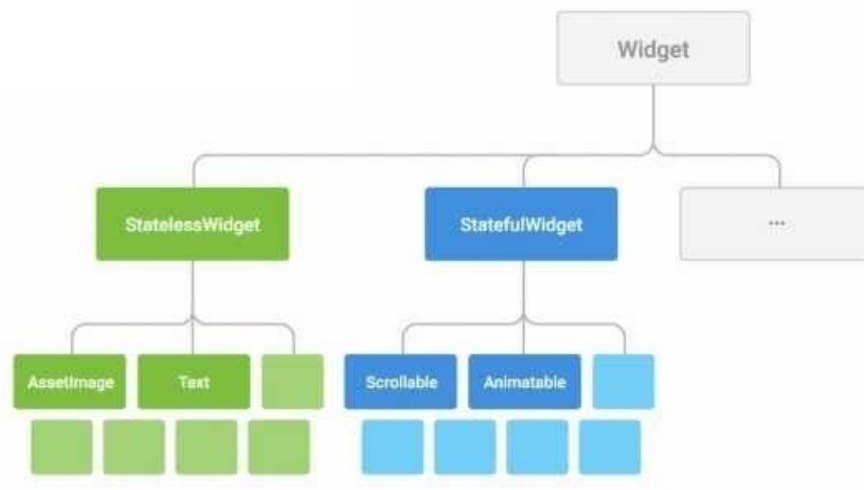


Fonte: Adaptado de (HONDA, 2019)

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, utiliza um renderizador *Mobile First* acelerado por GPU para que haja consistência 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 vai criando 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.

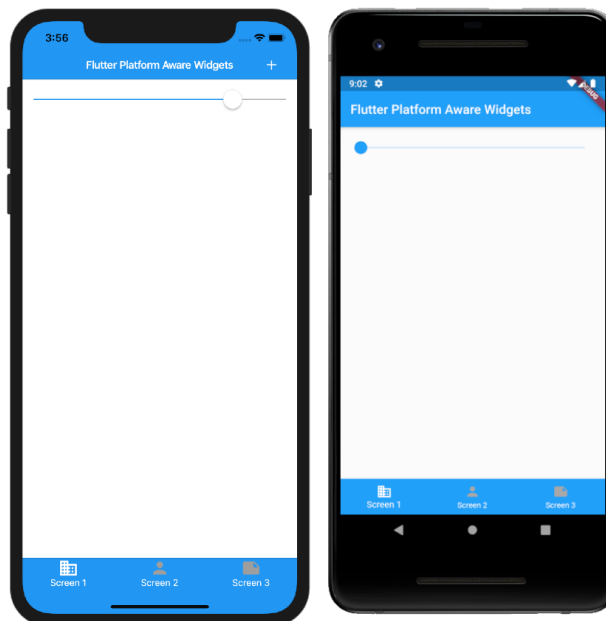
Figura 04 - HTML/CSS análogos em Flutter



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 na Figura 05.

Figura 05 - Cupertino e material



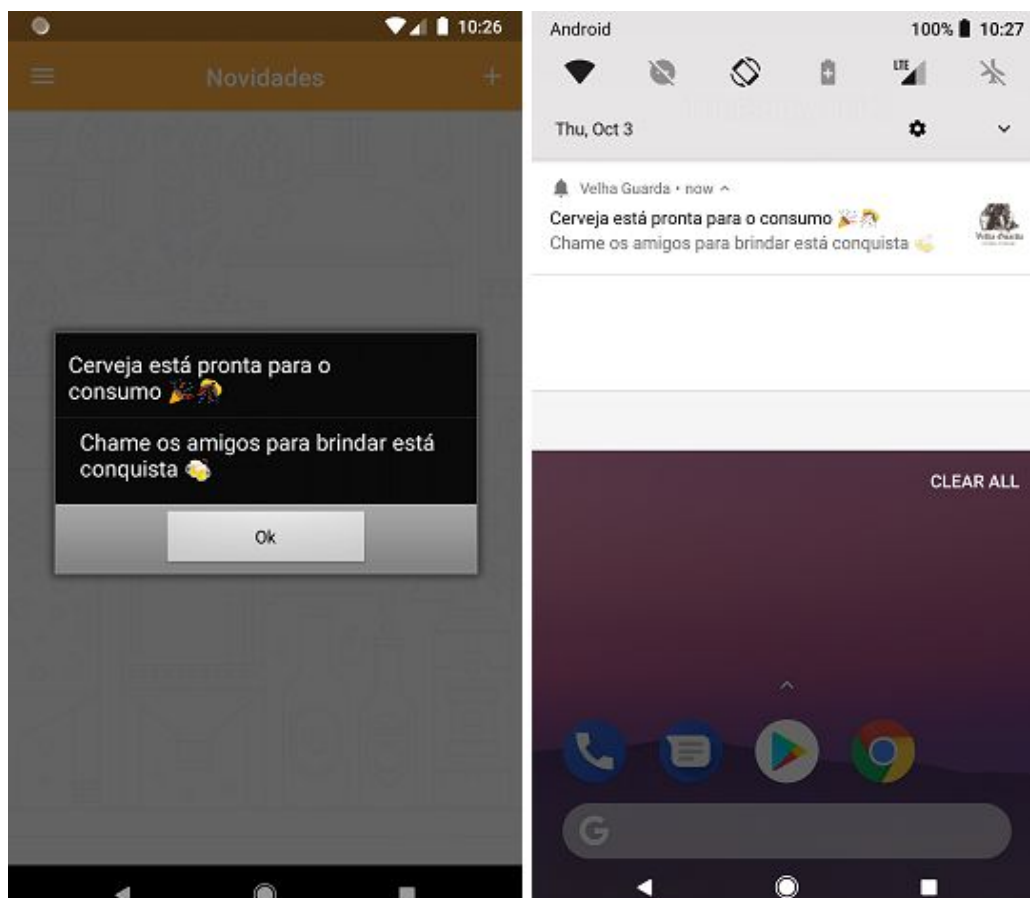
Fonte: Adaptado de (MOORE, 2019)

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, esse 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

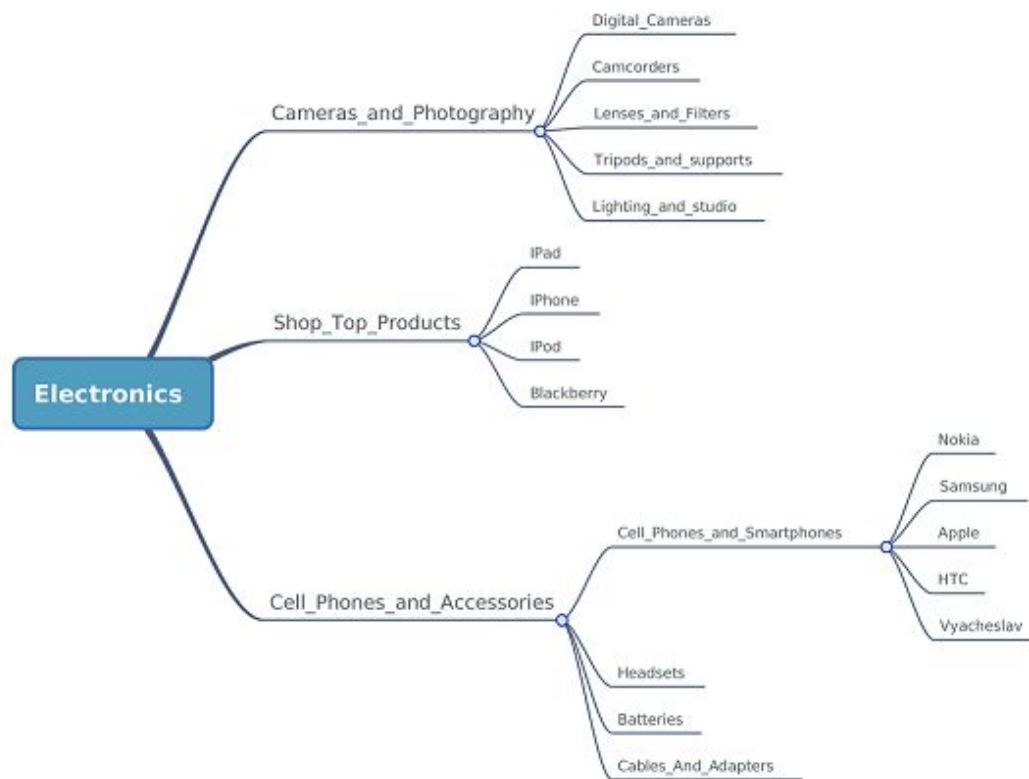


Fonte: Adaptado de (JACKSON, 2019)

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. Nele utiliza-se um banco de dados não relacional, ou seja não utiliza SQL, isso quer dizer que não possui como padrão o sistema de tabelas e relacionamentos entre dados, basicamente 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 podendo criar 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 a você 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.

4.2.1 Desenvolvimento

- Realtime Database
- Auth
- Test Lab
- Crashlytics
- Cloud Functions
- Firestore
- Cloud Storage
- Performance Monitoring
- Crash Reporting
- Hosting









4.2.2 Qualidade

- Firebase Analytics
- Invites
- Cloud Messaging
- Predictions
- AdMob
- Dynamic Links
- Adwords
- Remote Config
- App Indexing

4.2.3 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



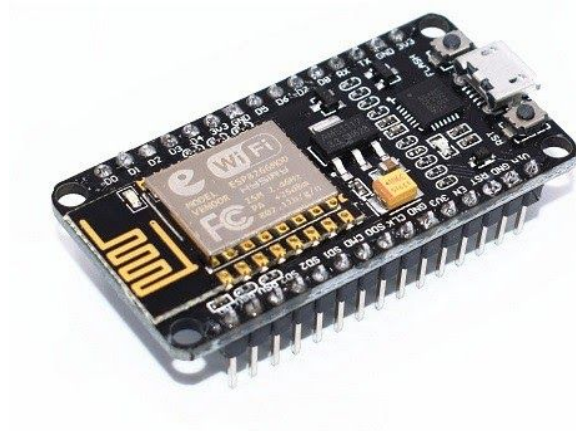
Fonte: Adaptado de (MORAIS, 2017)

Segundo o argumento de (MORAIS, 2017) afirma que, 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

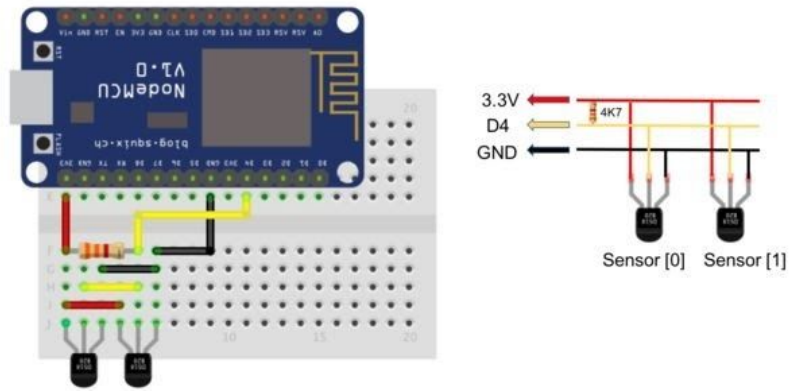


Fonte: Adaptado de (MADEIRA, 2018)

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)