



## **PROTÓTIPO DE APLICAÇÃO MÓVEL MULTIPLATAFORMA PARA APRESENTAÇÃO DOS DADOS DE MONITORAMENTO DE TANQUES DE RESFRIAMENTO DE LEITE**

### ***PROTOTYPE OF CROSS-PLATFORM MOBILE APPLICATION TO PRESENT MONITORING DATA FROM MILK COOLING TANKS***

Submetido em: 28/02/2021

Aprovado em: 21/03/2021

Diogo Lourenço Pereira <sup>1</sup>

Pedro Alves de Oliveira <sup>2</sup>

### **RESUMO**

Mudanças organizacionais e estruturais ocorrem na agropecuária brasileira em face da transformação digital do setor, onde uma nova agropecuária mais tecnológica apresenta-se como possibilidade para aumentar a competitividade do setor. Este estudo propõe-se a projetar o desenvolvimento de um protótipo em uma arquitetura distribuída multiplataforma para dispositivos móveis orientada a microsserviços de modo a obter maior flexibilidade, acessibilidade e usabilidade, através da adaptação de tecnologias *web* e sensores baseados em um microcontrolador integrado a modelos de computação em nuvem. O sistema consiste em capturar dados progressivos de temperatura e volume, além de renderizar esses dados em relatórios para pecuaristas ou especialistas em agropecuária leiteira por meio de dispositivos baseados em sistemas operacionais Android ou iOS. Assim torna-se possível a um usuário analisar a condição do leite manipulando os dados para tomar a decisão mais correta cujo efeito pode ser prejudicial a qualidade do leite e, consequentemente a saúde humana.

**Palavras-chave:** Agropecuária leiteira. Pecuarista. Aplicativos. Transformação digital. Computação em nuvem.

<sup>1</sup>Graduado em Sistemas de Informação pela Pontifícia Universidade Católica de Minas Gerais, unidade São Gabriel. E-mail: [diogo.lourenco@sga.pucminas.br](mailto:diogo.lourenco@sga.pucminas.br)

<sup>2</sup>Doutor em Geografia pela Pontifícia Universidade Católica de Minas Gerais e mestre em Engenharia de Produção pela Universidade Federal de Minas Gerais. Professor Adjunto do Instituto de Ciências Exatas e Informática da PUC Minas. E-mail: [pedroalves@pucminas.br](mailto:pedroalves@pucminas.br)

## ABSTRACT

*Organizational changes and exhibitions in Brazilian agriculture in the digital transformation of the sector, where a new, more technological agriculture is presented as a possibility to increase the sector's sales. This study proposes to design or develop a prototype in a multiplatform distributed architecture for microservice oriented mobile devices in order to obtain greater flexibility, accessibility and usability, by adapting web technologies and tests used in an integrated microcontroller. to cloud computing models. The system consists of capturing progressive temperature and volume data as well as rendering this data in reports to ranchers or dairy specialists using devices built on Android or IOS operating systems. Thus, it is possible for a user to analyze the milk condition by manipulating the data to make a more correct decision, the effect of which may be detrimental to milk quality and, consequently, to human health.*

**Keywords:** Dairy Farming. Farmer. Applications. Digital transformation. Cloud computing.

## 1 INTRODUÇÃO

A utilização, pelos produtores rurais, de recursos digitais tem se mostrado um caminho para todos que buscam um agronegócio de alta rentabilidade econômica e sustentável. Contudo, existe uma necessidade maior entre os pequenos agricultores e pecuaristas de adotarem as vantagens proporcionadas pelos recursos digitais, sendo um dos principais motivos a falta de infraestrutura, uma vez que eles não possuem um aparato tecnológico apropriado como os agricultores de médio e grande porte, dificultando a concorrência com eles.

De acordo com Souza et al. (2012), o Brasil tem demonstrado ao mundo, os avanços de um modelo sustentável para o desenvolvimento do agronegócio, capaz de aliar tecnologia, rentabilidade econômica e preservação ambiental. Assim, o agronegócio brasileiro desponta como um campo propício para a transformação digital, uma vez que é notório o crescimento acelerado advindo da demanda do mercado consumidor.

Segundo dados apresentados pela ABIA (2019), relativos ao mercado agroindustrial alimentício brasileiro, o setor de laticínios faturou cerca de R\$ 68,7 bilhões, ocupando o segundo lugar na posição do *ranking* de faturamento, ficando entre os principais setores alimentícios do país no ano de 2018.

A digitalização do agronegócio pode eliminar muitos efeitos indesejáveis existentes no setor, uma vez que a cadeia de produção do leite possui diversos processos, e por isso diversas dores, sendo uma dessas a conservação do leite após a ordenha, que precisa ser armazenado em tanques de resfriamento a temperatura adequada para garantir a qualidade microbiológica do mesmo. Porém atualmente esse processo é manual e trabalhoso, apresentando erros na aferição da temperatura e também do volume de leite armazenado, além de sofrer influência de fatores externos, pois alguns produtores não ficam nas fazendas em tempo integral.

Analisando as dificuldades apresentadas anteriormente, o objetivo desse trabalho é construir um protótipo de aplicação móvel multiplataforma para apresentação dos dados de monitoramento de tanques de resfriamento de leite em tempo real.

## **2 REFERENCIAL TEÓRICO**

Esta seção apresenta os conceitos teóricos das áreas de conhecimento que compõem este trabalho e, para isso, as subseções Agronegócio, Agronegócio 4.0, Sistemas Distribuídos, Aplicação móvel multiplataforma, Aplicação móvel híbrida e Ionic Framework foram criadas.

### **2.1 Agronegócio**

Com forte pressão do mercado, o agronegócio está passando por uma transformação onde precisará sair de sua zona de conforto e se reinventar para acompanhar as tendências e atender às demandas que se renovam de maneira constante. A aplicação das tecnologias digitais existentes, contribui para a quebra desta barreira e evolução de todo o ecossistema de estímulo a inovação, que possibilitou a criação de programas de fomento como o "Ideas For Milk". Trata-se de um projeto da Embrapa Gado de Leite que propõem uma competição entre empreendedores na busca pelas melhores inovações tecnológicas para o setor de laticínios (ARBEX; MARTINS, 2019).

#### **2.1.1 Agronegócio 4.0**

Para Schwab (2019) a revolução agrícola foi seguida por uma série de revoluções industriais iniciadas na segunda metade do século XVIII. A marca dessas revoluções foi a transição da força muscular para a energia mecânica, a qual evolui até a atual quarta revolução industrial, quando a produção humana foi aumentada por meio da potência aprimorada da cognição.

Segundo Arbex e Martins (2019) a integração de todos os agentes e membros deste movimento de estruturação do leite com foco na revolução 4.0 consiste na convergência e sinergia de diversas tecnologias e a relação de trabalho homem-máquina a partir das chamadas "tecnologias habilitadoras" tais como computação em nuvem, *big data*, automação industrial, simulação (Ambiente Virtual), IOT (Internet das Coisas), entre outras. Utilizadas para promo-

ver a transformação digital em diversas áreas da sociedade, principalmente no setor lácteo por vezes considerado a parcela menos evoluída do agronegócio brasileiro.

Para o setor de agropecuária no Brasil este é o momento de aproveitar as oportunidades e garantir a competitividade frente a mercados globais que estão avançando com auxílio de tecnologias emergentes e inovações em processos, segundo Kaminski (2018). Deste modo, o impacto do denominado agro 4.0, é considerado muito abrangente, pois atinge de forma integral toda a população e a cadeia produtiva: produtores, fabricantes, fornecedores e trabalhadores. Essa nova realidade faz com que seja necessária maior qualificação e maior preparação dos profissionais para lidarem com determinadas mudanças, Milagre et al. (2018).

## **2.2 Sistemas Distribuídos**

Os sistemas distribuídos surgiram como consequência dos avanços tecnológicos, inovações em processos e aumento na complexidade apresentada pelas aplicações, não somente relacionadas ao custo computacional cada vez maior, mas também por causa do gerenciamento e necessidade de recursos que têm impulsionado o desenvolvimento de técnicas para compartilhamento dos mesmos através do desacoplamento das partes de um sistema.

De acordo com Garcia et al. (2015) a arquitetura em sistemas distribuídos trata dos componentes que constituem uma aplicação, suas características visíveis e as ligações em seus mais diversos níveis. Uma boa arquitetura na utilização de recursos compartilhados, tolerância a falhas e concorrência, aumentando a escalabilidade do sistema tanto em tamanho quanto no que se refere à adição de recursos e usuários.

Uma das propriedades mais importantes existente em sistemas distribuídos é a transparência, sendo uma das maiores colaboradoras do compartilhamento otimizado de recursos, pois permite que o sistema seja gerenciado e processado de forma componentizada mas invisível para o usuário, garantindo a ilusão de que toda a aplicação é processada em um sistema único, ou seja, um sistema monolítico (STEEN; TANENBAUM, 2009).

Segundo Steen e Tanenbaum (2009) existem cinco tipos diferentes de transparências em sistemas distribuídos que formam um estado válido de transparência única. São elas:

- **Transparência de acesso:** é a forma sobre a qual é realizada a interação com a possibilidade de acessar e também transformar o acesso em utilização.

- Transparência de localização: os recursos, dados e dispositivos que são compartilhados e/ou acessados, não contém nenhuma informação que permita ao usuário encontrar a localização onde este recurso está contido.
- Transparência de migração: os recursos podem precisar ser alocados para outro servidor de modo a garantir o funcionamento da aplicação, mas essa movimentação não deve interferir no acesso a um arquivo ou recurso.
- Transparência de replicação: é habitual, em aplicações distribuídas que recursos e dados sejam transacionados como objetos. Por esse motivo é necessário que a aplicação seja capaz de gerenciar a replicação de informação de maneira invisível para usuário.
- Transparência de falhas: este é um aspecto de grande importância, ao se utilizar recursos compartilhados. É fundamental a capacidade de recuperação de falhas mantendo o problema isolado e transparente para o usuário.

Sistemas distribuídos tem adquirido uma importância crescente na indústria da computação devido à ampla difusão da internet e às características diferenciadas quando comparados aos sistemas monolíticos (PASCUTTI, 2002). Porém, existem alguns desafios relacionados a sistemas distribuídos como, por exemplo, o monitoramento que envolve um processo de extração, transformação e carregamento de informações acerca dos componentes do sistema e sobre os processos executados, além de sua apresentação em formato adequado aos usuários. Também apresenta-se de maneira não menos importante garantir flexibilidade, escalabilidade, portabilidade, entre outros requisitos arquiteturais.

### **2.3 Aplicação móvel multiplataforma**

De acordo com Palmieri et al. (2012) o crescimento intensivo da indústria de dispositivos móveis tem exigido alto desempenho dos sistemas operacionais móveis nesse contexto que grandes empresas do setor tecnológico como, por exemplo, Apple, Microsoft e Google estão desempenhando um papel importante. Cada uma delas introduziu seu próprio produto para competir nesse mercado. Todos eles fornecem recursos únicos para atrair desenvolvedores. Num cenário de diversidade de sistemas operacionais e plataformas de desenvolvimento, desenvolver uma aplicação móvel de código único para utilizá-lo em varias plataformas é um dos maiores anseios da computação móvel (SILVA; SANTOS, 2014). Existe uma variedade de *frameworks* para construção de aplicativos multiplataforma para dispositivos móveis que

foram criados com propósitos, características e abordagens distintas. A seguir é abordado o conceito de aplicação híbridas, sendo apresentado o *framework* Ionic uma das ferramentas mais conhecidas para desenvolvimento móvel híbrido.

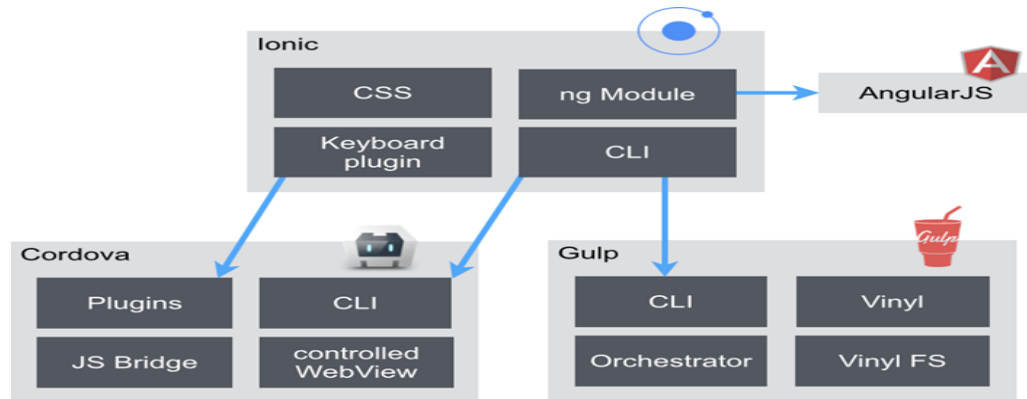
### 2.3.2 Ionic Framework

É um *framework* para desenvolvimento multiplataforma de código aberto e mantido pela Drifty Co. Foi desenvolvido em 2012 para oferecer desempenho e velocidade em aplicativos híbridos, quando o uso de tecnologias *Web* como plataforma de aplicativos com funções nativas ainda estava nascendo. O objetivo era criar uma maneira melhor de os desenvolvedores *Web* usarem suas habilidades para criar aplicativos para as lojas de aplicativos (IONIC, 2019).

Atualmente, o Ionic constitui a pilha tecnológica de desenvolvimento móvel híbrido mais popular do mercado, promovendo o desenvolvimento rápido de aplicações em todo mundo. A plataforma do *framework* permite que desenvolvedores usem tecnologias *Web* para criar e fornecer aplicativos para dispositivos móveis, com tudo o que precisam, desde o *design* do *front-end* (Camada de Apresentação) até o processo de *continuous delivery* (Entrega Contínua) (IONIC, 2019).

Na Figura 1 é possível ver em detalhes a maneira pela qual os diferentes componentes do Ionic operam mutuamente.

. **Figura 1 – Arquitetura de alto nível Ionic Framework**



**Fonte: Ripkens (2014)**

De acordo com Ripkens (2014) a base arquitetural do Ionic, contém quatro componentes fundamentais:

- O CSS, que define um *layout* básico para cada tipo de aparelho.
- O módulo *Angular* que define os componentes de navegação que contam com um funcionamento padrão.
- O CLI que opera como um tipo de *proxy* para a CLI do Apache Cordova e Gulp.
- O *plugin-in* de teclado que embora não seja indispensável, permite acessar informações como o estado atual do aplicativo, entre outros.

## 2.4 Trabalhos relacionados

A transformação tecnológica do agronegócio acontece progressivamente e nesse processo uma parcela de produtores se antecipa aos demais na inovação, podendo alcançar lucros maiores, à medida que novas tecnologias surgem para reduzir custos e o preço do produto permanece constante. Os produtores que antecipam a inovação e buscam a redução dos custos, a partir da evolução dos processos, elevam seus ganhos, pois a melhoria tecnológica vem, geralmente, acompanhada do aumento na escala da produção, impactando a quantidade produzida (KONZEN, 2019).

Segundo Araújo (2014), o momento é favorável a inovação e sustentabilidade do agronegócio, acreditando na importância do pequeno produtor, uma vez que as micros e pequenas

empresas são um dos principais pilares de sustentação da economia, quer pela sua enorme capacidade geradora de empregos.

O autor em seu trabalho, desenvolveu um estudo de caso para identificar os indicadores das inovações aplicadas a 6 laticínios de pequeno e médio porte para entender as dificuldades em incorporar a inovação e a sustentabilidade em sua cadeia de produção do leite, seja por falta de maturidade gerencial, condições estruturais e/ou financeiras, falta de conhecimento técnico, entre outros.

### **3 METODOLOGIA**

Esta seção tem por objetivo descrever a metodologia utilizada no desenvolvimento deste trabalho.

A partir da utilização de técnicas das áreas de conhecimento citadas na seção 2 este trabalho foi desenvolvido com base em uma pesquisa do mercado agropecuário brasileiro, de modo a construir um protótipo de aplicação móvel multiplataforma para a apresentação dos dados de monitoramento de tanques de resfriamento de leite. A metodologia foi dividida como descrita a seguir para alcançar os objetivos do trabalho:

- Revisão bibliográfica;
- Obtenção e estruturação da informação;
- Definição das tecnologias da arquitetura da aplicação;
- Metodologia de desenvolvimento ágil;

#### **3.1 Revisão bibliográfica**

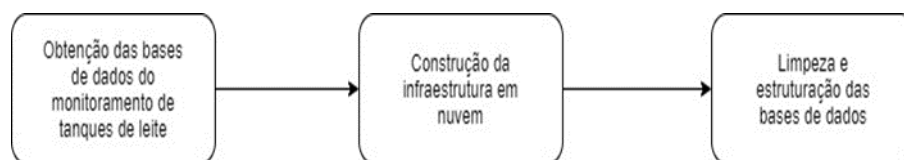
A construção do protótipo proposto foi realizada a partir de pesquisas de referências bibliográficas sobre os conteúdos básicos para o desenvolvimento do trabalho. Dentre os conteúdos pesquisados estão, desenvolvimento móvel, ferramentas de desenvolvimento móvel híbrido, padrões arquiteturais de desenvolvimento de *software*, persistência de dados, hospedagem e consumo de serviços em nuvem. Foram destacados conceitos nas visões de diferentes autores com o objetivo de apoiar teoricamente o desenvolvimento do protótipo, além de definir quais tecnologias seriam utilizadas em seu desenvolvimento.

#### **3.2 Obtenção e estruturação da informação**



Está etapa serviu como uma fase de filtragem e entendimento das bases de dados de monitoramento de tanques de resfriamento de leite, conforme relevância após limpeza e estruturação das mesmas, foi escolhida uma infraestrutura de armazenamento dos dados em nuvem, de modo a simplificar o acesso e proporcionar um nível adequado de segurança das informações. Sendo o fluxo da etapa descrito na imagem Figura 2:

**Figura 2 – Etapas de obtenção e estruturação da informação**



**Fonte : Elaborado pelo autor**

Para a coleta dos dados de monitoramento foi instalado um aparelho em tampas de tanques de resfriamento de leite que afere a temperatura, volume, data da coleta, entre outros. Utilizando uma estrutura no formato JSON (*JavaScript Object Notation*) envia os dados ao servidor tal como, mostra o exemplo:

```
{  
  " mac Address " :    " 3 0 : AE: A4 : 1 8 : 2 8 : D8 " ,  
  " volume " :   1 7 3 . 1 0 ,  
  " t e m p e r a t u r a " :  2 5 . 3 1 ,  
  " d a t a C o l e t a " :  " 1 6 / 1 1 / 2 0 1 9   1 8 : 1 6 : 1 2 " ,  
  " v i b r a c a o D e t e c t a d a " :  f a l s e ,  
  " t a m p a A b e r t a " :    t r u e ,  
  " d e s n i v e l a m e n t o " : f a l s e
```

}

**Fonte: Elaborado pelo autor**

Após o envio dos dados de coletados ao servidor, são armazenados no banco de dados na nuvem e passando por um processo de limpeza e estruturação em modelo relacional para que a informação gerada seja recuperada posteriormente pelo protótipo construído, de acordo com o exemplo:

```
{
"mac Address ":" 3 0 : AE: A4 : 1 8 : 2 8 : D8 ",
"volume ":    98.7,
"temperatura":  25.31,
"data Coleta":  "14/11/2019      11:14:12",

"vibracao Detectada": true, "tampa Aberta ":true,
"desnivelamento":    false
},

{
"mac Address ":" 3 0 : AE: A4 : 1 8 : 2 6 : D6 ",
"volume ":    43.1,
"temperatura":  25.31,
"data Coleta":  "15/11/2019      11:14:12", "vibracao Detectada":  true,
"tampa Aberta ":    false, "desnivelamento":    true
},
{
"mac Address ":" 3 0 : AE: A4 : 1 8 : 2 8 : D8 ",
"volume ":    29.23,
"temperatura":  25.31,
"data Coleta":  "15/11/2019      11:14:12", "vibracao Detectada":  true,
"tampa Aberta ":    true,
"desnivelamento":    false
}
}
```

**Fonte: Elaborado pelo autor**

### 3.3 Definição das tecnologias da arquitetura do projeto

Para o desenvolvimento do protótipo proposto neste trabalho foram utilizadas as seguintes tecnologias, de modo a atender as necessidades arquiteturais do projeto.

- **Ionic Framework 4.1.1:** O *framework* de desenvolvimento Ionic, permite a criação de um aplicativo móvel, acessível de qualquer uma das plataformas *mobile* Android ou iOS.
- **PostgreSQL 12.1:** O PostgreSQL é um banco de dados relacional de código aberto utilizado por diversos desenvolvedores e empresas, possui alta performance, confiabilidade e escalabilidade, além de recursos para *backup* e recuperação de dados.
- **Spring Boot 1.5.22:** É um framework para o desenvolvimento de microsserviços utilizando a linguagem de programação Java, de modo a auxiliar no processo de configuração e distribuição de projetos.
- **Amazon S3:** É um serviço para armazenamento de dados que fornece escalabilidade, alta disponibilidade e baixa latência. Torna possível armazenar diferentes tipos de dados como sites estáticos, imagens e backup de dados.
- **Heroku** É uma plataforma de serviços de computação em nuvem que permite a utilização de recursos privados e/ou compartilhados, tais como, hospedagem de microsserviços com suporte a várias linguagens de programação, execução de diferentes tarefas remotamente, consumo de recursos computacionais sob demanda, infraestrutura auto gerenciada, entre outros.

### 3.4 Metodologia de desenvolvimento

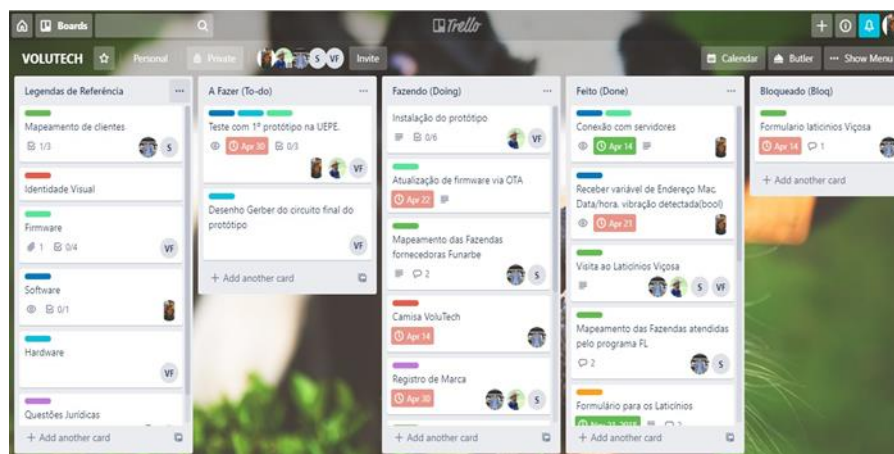
- Foi utilizado na construção do protótipo a metodologia de desenvolvimento ágil *Scrum*. Um *framework* para desenvolvimento de *software* pouco burocrático com foco na qualidade do produto, comunicação iterativa e entrega de pequenos artefatos. Funcionando dentro do contexto proposto em conjunto com outras técnicas e práticas, tais como, o quadro *Kanban* para acompanhamento das tarefas das *sprints* do *Scrum*.

Segundo Sabbagh (2014) os benefícios no uso do *Scrum* incluem: Entregas frequentes de retorno ao investimento dos clientes;

- Redução dos riscos do projeto;
- Maior qualidade no produto gerado;
- Mudanças utilizadas como vantagem competitiva;
- Visibilidade do progresso do projeto;
- Redução do desperdício;
- Aumento de produtividade.

A Figura 3 apresenta um exemplo do quadro *Kanban* utilizado para acompanhamento das tarefas das *sprints* realizadas para alcançar o objetivo do trabalho na ferramenta para gerenciamento de projeto Trello.

**Figura 3 – Quadro Kanban do protótipo**



**Fonte: Elaborado pelo autor**

#### **4 DESENVOLVIMENTO**

Esta seção tem por objetivo descrever o desenvolvimento do trabalho e apontar as dificuldades encontradas. A construção do protótipo foi dividida em 4 *sprints*, sendo compostas pelas seguintes atividades:

- *Sprint 1* - Definição de requisitos funcionais e não funcionais;

- *Sprint 2* - Definição das tecnologias;
- *Sprint 3* - Desenho da arquitetura do protótipo;
- *Sprint 4* - Desenvolvimento do protótipo;

#### 4.1 Definição de requisitos funcionais e não funcionais

O protótipo desenvolvido possui diversos componentes, seguindo o princípio da responsabilidade única, de modo a atender os requisitos funcionais e não funcionais estipulados nas tabelas, 1 e 2:

#### 4.2 Definição das tecnologias

Para o desenvolvimento desta aplicação foram utilizadas as tecnologias citadas na seção 3 deste trabalho, tais como, *Ionic framework*, *PostgreSQL*, *Spring Boot*, *Amazon S3* e *Heroku*.

| Requisitos Funcionais |                                         |                                                                                                                                                          |
|-----------------------|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID                    | Requisito                               | Descrição do Requisito                                                                                                                                   |
| RF01                  | Monitoramento de temperatura            | O protótipo deve exibir os dados de monitoramento de temperatura de tanques de resfriamento de leite.                                                    |
| RF02                  | Monitoramento de volume                 | O protótipo deve exibir os dados de monitoramento de volume de tanques de resfriamento de leite.                                                         |
| RF03                  | Monitoramento de status                 | O protótipo deve exibir os dados de monitoramento de status de tanques de resfriamento de leite.                                                         |
| RF04                  | Envio de Notificações                   | O protótipo deve enviar notificações de alertas em cenários críticos de monitoramento, por exemplo, alta temperatura do tanque de resfriamento de leite. |
| RF05                  | Geração de relatório de temperatura     | O protótipo deve gerar um relatório de temperatura por tanque de resfriamento de leite.                                                                  |
| RF06                  | Geração de relatório de volume diário   | O protótipo deve gerar um relatório de volume de leite produzido diariamente.                                                                            |
| RF07                  | Geração de relatório de evolução mensal | O protótipo deve gerar um relatório de evolução mensal do volume de leite produzido.                                                                     |

**Tabela 1: Requisitos Funcionais**

Fonte: Elaborado pelo autor Tabela

## 2: Requisitos Não Funcionais

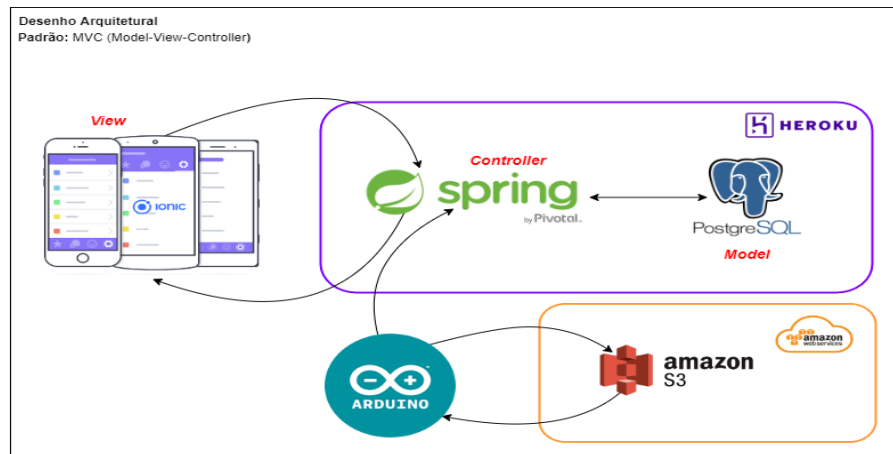
| Requisitos Não Funcionais |                             |                                                                                                                                       |
|---------------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| ID                        | Requisito                   | Descrição do Requisito                                                                                                                |
| RNF01                     | Espaço de armazenamento     | Requisito de Produto: O protótipo deve ocupar um tamanho máximo de 10 MB.                                                             |
| RNF02                     | Monitoramento em tempo real | Requisito de Produto: A protótipo deve exibir os dados de monitoramento de tanques de leite em tempo real.                            |
| RNF03                     | Desenvolvimento em Ionic    | Requisito de Implementação: A protótipo deve ser desenvolvida utilizando o <i>framework</i> Ionic.                                    |
| RNF04                     | Biblioteca Lodash           | Requisito de Implementação: A protótipo deve utilizar a biblioteca <i>JavaScript</i> de funções utilitárias <i>Lodash</i> .           |
| RNF05                     | Escalabilidade              | Requisito de Desempenho: O protótipo deve possuir uma arquitetura escalável horizontalmente.                                          |
| RNF06                     | Padrão de projeto MVC       | Requisito de Implementação: A protótipo deve ser desenvolvida utilizando o padrão arquitetural MVC ( <i>Model, View, Controller</i> ) |
| RNF07                     | Disponibilidade             | Requisito de Produto: A protótipo deve possuir uma disponibilidade de 99,99%                                                          |

Fonte: Elaborado pelo autor

### 4.3 Desenho arquitetural do protótipo

A Figura 4 apresenta o modelo de arquitetura distribuída com padrão MVC (*Model-View-Controller*) desenvolvido para o protótipo.

**Figura 4 – Desenho arquitetural do protótipo**



**Fonte: Elaborado pelo autor**

A arquitetura do protótipo objetiva entregar flexibilidade, escalabilidade, disponibilidade e segurança em uma solução estruturada que atenda os aspectos técnicos e de negócios. O modelo arquitetural foi baseado no padrão MVC (*model, view, controller*) com componentes distribuídos, onde é atribuído ao *framework* Ionic a responsabilidade pela interface e interação com toda a tecnologia de *front-end*, a plataforma de serviços de computação em nuvem Heroku, a hospedagem do banco de dados PostgreSQL e serviços com as regras de negócios desenvolvidos utilizando o *framework* Spring Boot em Java, o *storage* S3 com os arquivos de atualização do aparelho responsável pelo monitoramento dos tanques de resfriamento de leite.

#### 4.4 Desenvolvimento do protótipo

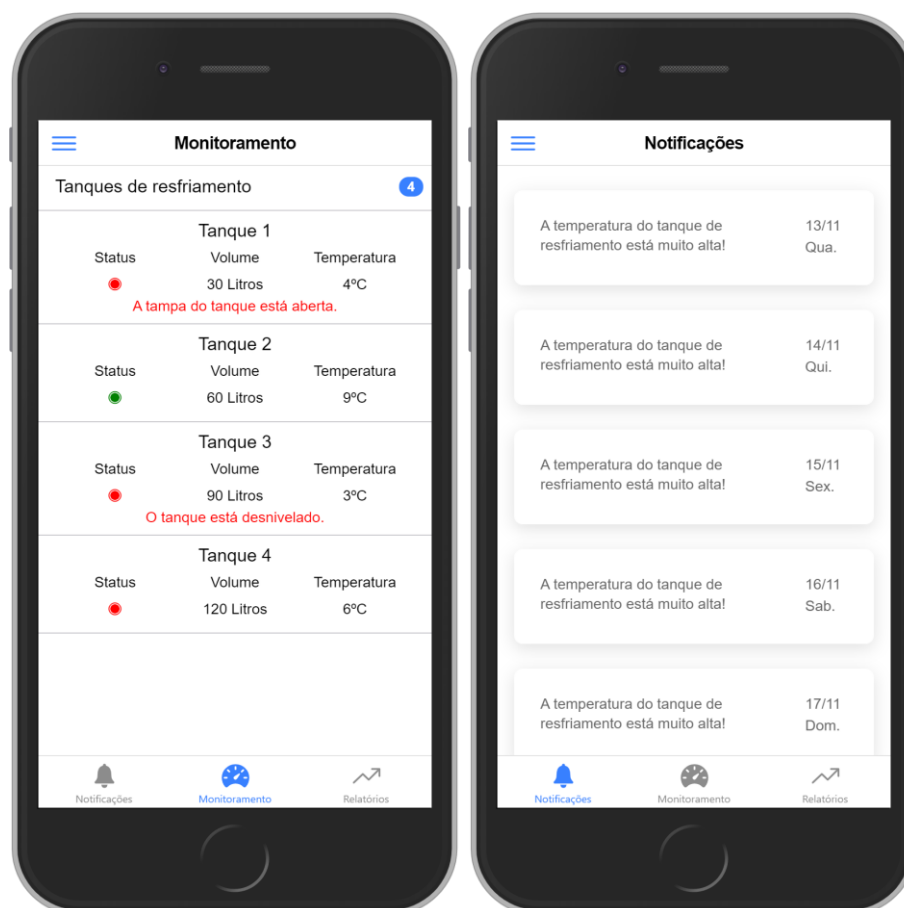
O protótipo do aplicativo foi desenvolvido utilizando o *framework* Ionic, que permite a construção de aplicações móveis híbridas, a partir de uma pilha tecnológica composta de ferramentas para desenvolvimento *web* como HTML, CSS e *JavaScript*. O banco de dados utilizado foi o PostgreSQL, que é um banco de dados relacional de código aberto, mantido pela comunidade de desenvolvedores voluntários. O servidor foi desenvolvido utilizando o *framework* Spring Boot, que é um projeto Java para simplificar o processo de configuração e publicação de API's com o padrão *REST* (*Representational State Transfer*) e está hospedado no serviço de nuvem Heroku juntamente com o banco de dados.

O aplicativo possui uma organização arquitetural do projeto necessária pelo *framework*

Ionic, que é dividida em quatro partes principais: *views*, *components*, *providers* e *models*. As *views* são responsáveis pela visualização de toda a camada de apresentação, sendo constituídas de *components* que são elementos com comportamento único que podem ser reutilizados na aplicação, os *providers* são responsáveis pela comunicação com o servidor, a partir das API's disponibilizadas, devolvendo a informação recuperada aos *models* dos *components* para exibição da informação nas *views*.

A Figura 5 apresenta duas telas do protótipo, a tela de monitoramento, onde é possível acompanhar os dados de status, volume e temperatura do leite, além de informações como desnivelamento e tampa aberta. E também a tela de notificações, que permite a visualização de alertas de cenários críticos de monitoramento, por exemplo, alta temperatura do tanque de resfriamento de leite.

**Figura 5 – Telas de monitoramento e notificações do protótipo**

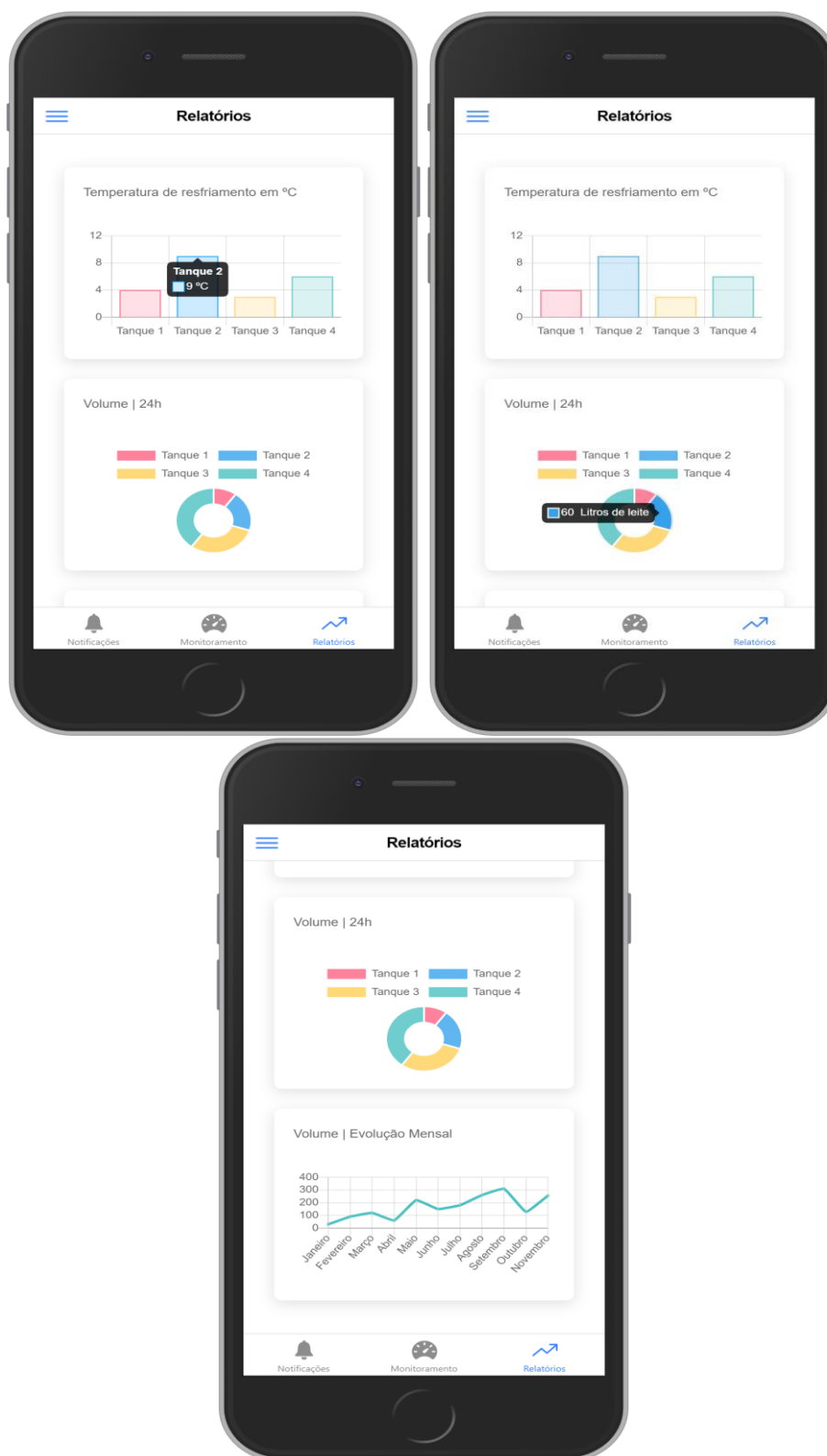


**Fonte: Elaborado pelo autor**



A Figura 6 apresenta a tela de relatórios desenvolvida para o protótipo, onde é possível visualizar a temperatura dos tanques de resfriamento de leite em °C, volume da produção diária e a evolução mensal da produção.

**Figura 6 – Tela de relatórios do protótipo**



**Fonte: Elaborado pelo autor**

## 5 EXPERIMENTOS E RESULTADOS

Após o desenvolvimento do protótipo, ele foi implantado em uma ordenha real para identificar se os objetivos foram cumpridos ou se ainda possuía pontos a melhorar. Para isso houve um período de quinze dias de observação da produção de leite, a partir do monitoramento pelo protótipo, que em determinado momento apresentou falha na coleta e apresentação dos dados dos tanques de resfriamento, uma vez que o serviço de nuvem *Heroku*, não se apresentou como a melhor plataforma para hospedagem gratuita do servidor, pois o alto número de demandas durante o tempo de monitoramento da ordenha, excedeu o limite da hospedagem rapidamente, o que impossibilitou a utilização do protótipo até a renovação da mesma que acontece de forma automática ao final de todo mês.

Dentre as rotinas responsáveis pela geração deste problema, o maior ofensor é a coleta de dados dos tanques de resfriamento de leite, que envia um volume muito alto de dados em um intervalo curto de tempo para o *backend*, o que indica a necessidade de revisar a arquitetura desenhada inicialmente para atender as necessidades do projeto.

As Figuras 5 e 6 foram capturadas durante o período de monitoramento da ordenha, o que permitiu a validação das funcionalidades do protótipo.

O protótipo também foi utilizado na competição nacional de *startups* do agro, *Ideas For Milk*, que aconteceu no dia 22 de novembro de 2019 em São Paulo, no CUBO, como parte de uma solução de inovação no monitoramento da cadeia do leite, com foco em atender as necessidades de produtores em relação a conservação do leite. O protótipo foi avaliado pela comissão julgadora da competição, na qual eles aprovaram o mesmo e fizeram observações para novas funcionalidades, que tornariam o protótipo mais satisfatório e atrativo como, por exemplo, a inclusão de uma tela para acompanhamento do itinerário dos caminhões de coleta de leite das cooperativas pelo produtor. Dentre os 8 projetos finalistas da competição dos mais de 60 inscritos, a ganhadora do primeiro lugar foi a *startup* Volutech que utilizou o protótipo desenvolvido nesse trabalho. Comprovando uma alta taxa de aceitação quanto ao uso do protótipo, que apesar dos *feedbacks* de melhorias, alcançou um resultado positivo na entrega da experiência de uso.

## 6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho tem como principal contribuição, a elaboração de um protótipo de aplicação baseado em tecnologias de desenvolvimento móvel híbridas para validação de uma

solução de inovação para monitoramento da produção de leite em fazendas, concluindo-se que apesar do problema gerado pela falta de capacidade do *backend* diante do cenário de monitoramento da ordenha, de modo geral o protótipo demonstrou a viabilidade de construção de uma aplicação para o acompanhamento de ordenhas em tempo real.

Como trabalhos futuros, pretende-se desenvolver uma versão *web* da aplicação, implementar a funcionalidade de acompanhamento de itinerário dos caminhões de coleta de leite, implementar a funcionalidade de monitoramento pela perspectiva das cooperativas de leite, melhoria na apresentação dos dados dos relatórios e migração da arquitetura de servidor para a *Amazon Web Services*(AWS).

## REFERÊNCIAS

ABIA. **Associação Brasileira da Indústria de Alimentos**. 2019. Acesso em: 07 set. 2019. Disponível em: <<https://www.abia.org.br/vsn/tmp/6.aspx?id=16>>.

ARAÚJO, Alcione Lino de. **Inovações nos laticínios de pequeno e médio porte na região dos Campos Gerais-PR com vistas à manutenção no mercado**. 2014. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná.

ARBEX, Wagner; MARTINS, Paulo do Carmo. Anuário do leite 2019. **Anuário do Leite 2019**, 2019.

GARCIA, CM; ABILIO, R; MALHEIROS, N. Abordagens e tecnologias para integração de sistemas: Um estudo de caso na universidade federal de lavras. **Revista de Sistemas de Informação da FSMA**,(15), p. 11–22, 2015.

IONIC. **Página oficial do framework**. 2019. Acesso em: 11 set. 2019. Disponível em: <<https://ionicframework.com/about>>.

KAMINSKI, Natanael. Indústria 4.0 no brasil: 2018 promete avanços significativos. 2018. Acesso em: 20 set. 2019. Disponível em: <<https://administradores.com.br/noticias/industria-4-0-no-brasil-2018-promete-avancos-significativos>>.

KONZEN, Otto G. Modernização e competitividade entre sistemas na produção de leite. **Revista de Economia e Sociologia Rural**, Editora Cubo, v. 36, n. 1, p. 105–130, 2019.

MILAGRE, Ana Paula Almeida et al. Mapeamento do uso de tecnologias no agronegócio na microrregião de frutal. **GESTÃO, INOVAÇÃO E EMPREENDEDORISMO**, v. 1, n. 1, 2018.

PALMIERI, Manuel; SINGH, Inderjeet; CICCHETTI, Antonio. Comparison of cross-platform mobile development tools. **2012 16th International Conference on Intelligence in Next Generation Networks, ICIN 2012**, s. 179-186, 2012.

PASCUTTI, Márcia Cristina Dadalto. Uma proposta de arquitetura de um ambiente de desenvolvimento de software distribuído baseada em agentes. 2002.

RIPKENS, Ben. **Ionic: An AngularJS based framework on the rise**. 2014. Acesso em: 11 set. 2019. Disponível em: <<https://blog.codecentric.de/en/2014/11/ionic-angularjs-framework-on-the-rise/>>.

SABBAGH, Rafael. **Scrum: Gestão ágil para projetos de sucesso**. [S.l.]: Editora Casa do Código, 2014.

SCHWAB, Klaus. **A quarta revolução industrial**. [S.l.]: Edipro, 2019.

SILVA, Marcelo Moro da; SANTOS, Marilde Terezinha Prado. Os paradigmas de desenvolvimento de aplicativos para aparelhos celulares. **Revista TIS**, v. 3, n. 2, 2014.

SOUZA, Fábila et al. Perfil dos artigos sobre agronegócio publicados nos periódicos de contabilidade com estratocapes (agribusiness profile of articles published in journal of accounting with stratum capes). **Contexto, Porto Alegre**, v. 12, n. 22, p. 87–102, 2012.

STEEN, Maarten Van; TANENBAUM, Andrew S. **Sistemas Distribuídos: princípios e paradigmas**. [S.l.]: São Paulo: Prentice Hall, 2009.