



**LUIZ GUSTAVO FRANCO SOUZA**

**SISTEMA PARA GERENCIAMENTO DE AFERIÇÃO DE BALANÇAS UTILIZANDO  
O FRAMEWORK IONIC PARA DESENVOLVIMENTO DE APLICATIVOS  
HÍBRIDOS**

Ji-Paraná

2019

**LUIZ GUSTAVO FRANCO SOUZA**

**SISTEMA PARA GERENCIAMENTO DE AFERIÇÃO DE BALANÇAS UTILIZANDO  
O FRAMEWORK IONIC PARA DESENVOLVIMENTO DE APLICATIVOS  
HÍBRIDOS**

Monografia apresentada à Banca examinadora do Centro Universitário São Lucas, como requisito de aprovação para obtenção do Título de Bacharel em Sistemas de Informação.

Orientador: Prof. Maigon Nacib Pontuschka.

Ji-Paraná

2019

Dados Internacionais de Catalogação na Publicação  
Gerada automaticamente mediante informações fornecidas pelo(a) autor(a)

---

S729s Souza, Luiz Gustavo Franco

Sistema para gerenciamento de aferição de balanças utilizando o Framework Ionic para desenvolvimento de aplicativos híbridos / Luiz Gustavo Franco Souza-- Ji-Paraná, RO, 2019.

55 p.

Orientador(a): Prof. Me Maigon Nacib Pontuschka

Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) - Centro Universitário São Lucas

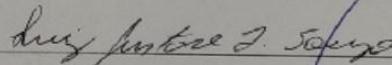
1. Processos de automação. 2. Segurança. 3. Processos de aferição. I. Pontuschka, Maigon Nacib. II. Título.

CDU 004.45:681.2

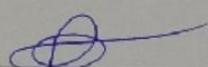
**ATA N° 05/2019 DE TRABALHO DE CONCLUSÃO DE CURSO EM  
SISTEMAS DE INFORMAÇÃO**

No segundo dia do mês de dezembro de 2019, das 17h as 22h reuniram-se na sala de Inovação Tecnológica 7 o(a) professor(a) orientador(a) Maigon Nacib Pontuschka e os(as) professores(as) José Rodolfo Milazzotto Olivas e Willian Alves de Oliveira Fachetti para comporem Banca Examinadora de Trabalho de Conclusão de Curso em Sistemas de Informação sob presidência do(a) primeiro(a), para analisarem a apresentação do trabalho " Sistema para gerenciamento de aferição de balanças utilizando o framework Ionic para desenvolvimento de aplicativos híbridos". Após as arguições e apreciação sobre o trabalho exposto foi atribuída a menção como nota do Trabalho e Concluso do curso do Acadêmico(a) Luiz Gustavo Franco Souza.

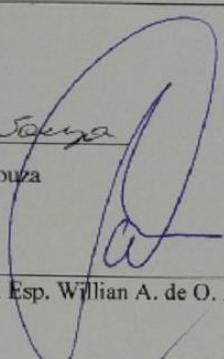
OBS: Trabalho de Conclusão de Curso  Aprovado ou  Reprovado com nota total de 9,1, atribuido o valor de 8,6 (Oito vírgula seis pontos) para o trabalho escrito e o valor de 9,5 (nove vírgula cinco pontos) para a apresentação oral.



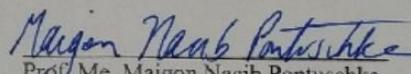
Luiz Gustavo Franco Souza



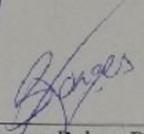
Prof. Esp. José Rodolfo M. Olivas



Prof. Esp. Willian A. de O. Fachetti



Prof. Me. Maigon Nacib Pontuschka  
Orientador



Prof. Me. Thyago Bohrer Borges  
Coord. Sistemas de Informação

## RESUMO

O presente trabalho trata do desenvolvimento de um sistema para a automatização de processos de uma empresa que efetua aferições e manutenções em balanças, a fim de proporcionar uma maior agilidade e segurança dos processos. A aferição de balanças é um tema pouco conhecido pelo público em geral, porém tem um impacto muito grande no dia a dia das pessoas, pois é a partir dela que se descobre se o equipamento está informando valores verdadeiros para o cliente. Para garantir que as balanças estejam sempre em perfeito funcionamento, existe o IPEM (Instituto de pesos e medidas) que fiscaliza se as empresas e seus prestadores de serviço estão cumprindo as normas descritas pelo INMETRO. Neste trabalho são apresentadas as tecnologias, processos e métodos utilizados para o desenvolvimento desta aplicação.

**Palavras-chave:** Automatização, aferição de balanças, desenvolvimento híbrido

## **ABSTRACT**

The present work deals with the development of a system for the automation of processes of a company that performs measurements and maintenance on scales in order to provide greater agility and safety for these processes. Scale calibration is a subject that is little known by the general public, but it has a very large impact on people's daily lives, because scale calibration determines if the equipment is informing true values for the client. To ensure that the scales are always in perfect working order, the IPEM (The Brazilian Institute of Weights and Measures) monitors whether companies and their service providers are complying with the standards described by INMETRO. This work presents the technologies, processes and methods used during the development of this application.

**Keywords:** Automation, scale calibration, hybrid development

## LISTA DE SIGLAS

API	-	Application Programming Interface
GPS	-	Global Positioning System
HTML	-	HyperText Markup Language
IDE	-	Integrated Development Environment
INMETRO	-	Instituto Nacional de Metrologia, Qualidade e Tecnologia
IPEM	-	Instituto de Pesos e Medidas
IOS	-	iPhone Operation System
OS	-	Operation System
UML	-	Unified Modeling Language
VSC	-	Visual Studio Code
WEB	-	World Wide Web

## LISTA DE FIGURAS

Figura 1 - Componente de um sistema de computação .....	15
Figura 2 – Visão geral do Visual Studio Code .....	19
Figura 3 - Visão geral da plataforma Firebase .....	20
Figura 4 - Caso de uso geral do sistema.....	23
Figura 5 - Diagrama de atividades geral do sistema .....	33
Figura 6 - Diagrama de sequência .....	34
Figura 7 - Diagrama de classe .....	35
Figura 8 - Código de comunicação.....	37
Figura 9 – Tela de autenticação android .....	38
Figura 10 – Tela de autenticação IOS.....	39
Figura 11 – Tela de navegação.....	40
Figura 12 - Tela de consulta de clientes.....	41
Figura 13 – Tela de consulta de balanças.....	42
Figura 14 – Tela de consulta de ferramentas.....	43
Figura 15 – Tela de cadastro de clientes .....	44
Figura 16 – Tela de cadastro de balanças .....	45
Figura 17 – Tela de cadastro de ferramentas.....	46
Figura 18 - Formulário de aferição 1 .....	47
Figura 19 - Formulário de aferição 2 .....	48
Figura 20 - Tela de documentos.....	48
Figura 21 - Confirmação para baixar .....	49
Figura 22 - Certificado da aferição .....	50

## LISTA DE TABELAS

Tabela 1 - Modelo de documento de caso de uso.....	24
Tabela 2 - Documentação do caso de uso Realizar aferição .....	24
Tabela 3 - Documentação do caso de uso Cadastrar cliente .....	25
Tabela 4 - Documentação do caso de uso Cadastrar balanças .....	26
Tabela 5 - Documentação do caso de uso Cadastrar ferramentas de aferição .....	26
Tabela 6 - Documentação do caso de uso Realizar login .....	27
Tabela 7 - Documentação do caso de uso Baixar relatório .....	27
Tabela 8 - Documentação do caso de uso gerar relatório.....	28
Tabela 9 - Documentação do caso de uso realizar calibragem.....	29
Tabela 10 - Exemplo de especificações funcionais e não funcionais.....	31
Tabela 11 - F1 - Autenticação no sistema.....	31
Tabela 12 - F2 - Seleção de cliente para aferição.....	31
Tabela 13 - F3 - Seleção de ferramentas para aferição .....	32

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	PROBLEMATIZAÇÃO	10
1.2	DELIMITAÇÃO	11
1.3	OBJETIVOS	11
<b>1.3.1</b>	<b>Objetivo Geral</b>	<b>11</b>
<b>1.3.2</b>	<b>Objetivos Específicos</b>	<b>11</b>
1.4	JUSTIFICATIVA	12
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>13</b>
2.1	BALANÇAS	13
2.2	LEGISLAÇÃO	13
2.3	O QUE É AFERIÇÃO	13
<b>2.3.1</b>	<b>Processo de aferição</b>	<b>13</b>
2.3.1.1	Nivelamento	14
2.3.1.2	Repetibilidade	14
2.3.1.3	Teste de pesagem de canto	14
2.3.1.4	Carga máxima	14
2.4	SISTEMAS OPERACIONAIS	15
2.5	SISTEMAS OPERACIONAIS MÓVEIS	16
<b>2.5.1</b>	<b>Android</b>	<b>16</b>
<b>2.5.2</b>	<b>IOS</b>	<b>16</b>
2.6	DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS	16
<b>2.6.1</b>	<b>Abordagem Nativa</b>	<b>16</b>
<b>2.6.2</b>	<b>Abordagem Multiplataforma</b>	<b>17</b>
2.6.2.1	Híbrida	17
2.6.2.2	Web	17
2.6.2.3	Interpretada	18
2.6.2.4	Compilação-cruzada	18
<b>3</b>	<b>FERRAMENTAS DE DESENVOLVIMENTO</b>	<b>19</b>
3.1	IONIC	19
3.2	VISUAL STUDIO CODE (VSC)	19
3.3	FIREBASE	20
3.4	HTML 5	20

3.5	TYPESCRIPT .....	20
<b>4</b>	<b>METODOLOGIA DE DESENVOLVIMENTO.....</b>	<b>22</b>
4.1	CASO DE USO.....	23
4.2	DOCUMENTAÇÃO DO CASO DE USO.....	24
4.3	REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS .....	29
4.4	DIAGRAMA DE ATIVIDADE.....	32
4.5	DIAGRAMA DE SEQUÊNCIA .....	33
4.6	DIAGRAMA DE CLASSE .....	34
<b>5</b>	<b>DESENVOLVIMENTO .....</b>	<b>36</b>
5.1	COMUNICAÇÃO ENTRE OS SERVIÇOS.....	36
<b>5.1.1</b>	<b>Exemplo de comunicação.....</b>	<b>36</b>
5.2	FUNCIONALIDADES .....	38
<b>5.2.1</b>	<b>Autenticação .....</b>	<b>38</b>
<b>5.2.2</b>	<b>Menu de navegação .....</b>	<b>39</b>
<b>5.2.3</b>	<b>Consultas .....</b>	<b>39</b>
<b>5.2.4</b>	<b>Cadastros .....</b>	<b>43</b>
<b>5.2.5</b>	<b>Aferição .....</b>	<b>46</b>
<b>5.2.6</b>	<b>Relatório .....</b>	<b>47</b>
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>51</b>
<b>7</b>	<b>REFERÊNCIAS.....</b>	<b>52</b>

## 1 INTRODUÇÃO

O processo de aferição de balança é algo que deve ocorrer de tempos em tempos para verificar se o equipamento presente em uma empresa está trabalhando de forma correta, a fim de garantir o bom funcionamento do equipamento e a veracidade das informações apresentadas por ele.

Para alguns gestores de empresas, a presença de um fiscal do IPEM (Instituto de Pesos e Medidas) em seu estabelecimento é motivo de preocupação, muitas das vezes por não manterem um calendário correto de aferições ou muito pior, saber que o equipamento está fora dos padrões e não efetuarem a calibragem do mesmo. Balanças que não estejam em conformidade podem gerar advertências e multas por parte do fiscal.

Com o objetivo de eliminar os possíveis problemas, existem empresas especializadas em efetuar a aferição e calibragem de balanças periodicamente de maneira que o equipamento sempre esteja em perfeito funcionamento e dentro dos critérios da Portaria do INMETRO nº 236 (1994). A validade da verificação é limitada em um ano, com exceção de casos especiais que podem ser definidos pelo INMETRO.

O presente trabalho vem para sanar um problema apresentado por uma das empresas que prestam serviço de aferição e calibragem de balanças. Com o crescimento da empresa prestadora de serviço, conseqüentemente sua demanda de aferições aumento, porém o processo de trabalho da empresa continua sendo no papel de forma manual, e isso causa um grande transtorno pela falta de agilidade e problemas na perda de documentos gerados durante o processo de aferição. O objetivo é se utilizar de tecnologias atuais, mais especificamente, um sistema de gerenciamento de aferições utilizando a tecnologia híbrida de desenvolvimento oferecida pelo Framework Ionic (2019) juntamente com a plataforma Firebase (2019).

### 1.1 PROBLEMATIZAÇÃO

A empresa Balanças Baltec conta com uma vasta experiência no mercado e efetua tanto o processo de aferição quanto o processo de calibração das balanças.

Para que as balanças sempre estejam em bom funcionamento, é necessário que haja periodicamente uma verificação do equipamento para garantir que o mesmo sempre esteja informando os dados corretos conforme descrito pelo fabricante. A aferição das balanças é feita por profissionais técnicos que verificam o equipamento seguindo uma legislação e utilizando ferramentas homologadas para esse fim. Hoje o processo de aferição, desde a escolha das ferramentas necessárias, verificação do equipamento e até a geração do certificado é feito de forma manual e anotado em documentos de papel que são posteriormente digitados em uma planilha. O processo todo pode demorar cerca de 15 dias por conta do grande volume de aferições feitas por essa empresa.

## 1.2 DELIMITAÇÃO

Uma série de módulos relacionados a aferição e calibragem poderiam ser adicionados ao projeto com intuito de automatizá-los, como a utilização de ferramentas para fazer a conexão diretamente entre a balança e o sistema mobile, desta maneira, agilizando ainda mais o processo e diminuindo possíveis riscos de erros humanos no processo de informar os dados ao sistema durante a aferição. Porém, neste trabalho iremos apenas desenvolver os módulos de inserção, verificação e geração relatórios utilizando o Framework IONIC (2019) para desenvolvimento multiplataforma.

## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

Desenvolver um sistema híbrido utilizando o framework IONIC para gerenciar o processo de aferição de balanças.

### 1.3.2 Objetivos Específicos

- Desenvolver um sistema híbrido

- Desenvolver um sistema que possa ser utilizado sem conexão com a internet.
- Acelerar o processo de aferição com a transferência de informações via internet
- Possibilitar mais segurança nos processos de aferição de balanças se utilizando da tecnologia.

#### 1.4 JUSTIFICATIVA

As dificuldades atuais e a morosidade do processo de aferição feito atualmente no papel e por processos manuais foram os fatores principais para que se pensasse em utilizar dispositivos móveis com o intuito de assegurar a exatidão e agilizar os testes de aferição e, ao mesmo tempo, facilitar futuras consultas de dados e geração de relatórios, tanto por parte dos funcionários, quanto por parte das empresas clientes.

A escolha do framework Ionic para o desenvolvimento desse projeto se deu pelo fato de o Ionic proporcionar ao desenvolvedor escrever código uma única vez e, a partir dele, compilar o projeto para ser utilizado em diversos tipos de dispositivos, sejam Android ou IOS, como também compilar o projeto para a web, que pode ser acessado por diversos navegadores. Desta maneira o tempo de produção é reduzido e é possível alcançar uma vasta gama de dispositivos e, conseqüentemente, usuários.

## 2 REFERENCIAL TEÓRICO

### 2.1 BALANÇAS

A balança é um instrumento desenvolvido e utilizado para pesar ou medir massas, diferentemente do que as pessoas utilizam no vocabulário do dia a dia, a balança não faz a medição do peso, porém através de cálculos matemáticos, pode ser obtido o peso.

### 2.2 LEGISLAÇÃO

Segundo a Lei 1.521/51, é crime fraudar pesos e medidas, ou comercializar produtos que se siba que foram fraudados. A pena prevista para esse tipo de crime é de 6 meses a 2 anos de detenção e multa. (BRAZIL, 1951)

### 2.3 O QUE É AFERIÇÃO

A aferição é o processo que se faz para verificar se um equipamento está calibrado com base em processos e medidas previamente determinadas. Para a aferição é necessário instrumentos específicos e certificados por órgãos reguladores.

#### 2.3.1 Processo de aferição

Para que se obtenha uma passagem correta utilizando uma balança, é necessário que o dispositivo utilizado esteja calculando e informando os valores corretos para o operador, por esse motivo existe o processo de aferição.

Segundo Victorino (2018) a aferição verifica se a balança verificada está realmente calculando e exibindo os valores corretos. Os testes são feitos baseados em padrões que são regulamentados por lei, além de que se deve utilizar ferramentas apropriadas que são regulamentadas pelo Instituto de pesos e medidas - IPEM.

Dentre os pontos a serem testados durante a aferição, quatro deles têm um alto grau de importância: nivelamento, repetibilidade, pesagem de canto e carga máxima.

#### 2.3.1.1 Nivelamento

O local onde o equipamento deve ser instalado é de suma importância para a aferição. As balanças respondem de acordo com o peso inserido, desta maneira é importante que a superfície ou a base que está por baixo esteja plana e forte, de maneira que suporte o peso demandado (VICTORINO, 2018).

#### 2.3.1.2 Repetibilidade

Dentre os diversos testes de aferição realizados pelos técnicos e fiscais certificados pelo IPEM, um dos testes é o de repetibilidade de resultados. O teste de repetibilidade é simples, porém de suma importância. Com o auxílio de um peso padrão e certificado é verificado se o equipamento exibe o mesmo resultado ao se realizar o teste 3 vezes seguidas (VICTORINO, 2018).

#### 2.3.1.3 Teste de pesagem de canto

Dependendo do tipo e tamanho da balança ela pode possuir vários pontos de sensores para garantir uma pesagem fiel. O teste de pesagem de canto consiste em verificar todas as extremidades da balança para saber se os seus resultados conferem com o do peso utilizado para o teste.

Dependendo do tipo da balança, ela pode possuir uma determinada tolerância de variação nas suas extremidades. Os valores de variação são definidos de acordo com a faixa de pesagem de cada balança e os padrões de aceitação são apresentados na portaria do INMETRO (VICTORINO, 2018).

#### 2.3.1.4 Carga máxima

O teste de carga máxima é a parte da aferição em que se verifica se a balança está apresentando os resultados corretos referentes ao peso de sua capacidade máxima. Se a balança tem capacidade máxima de 1 tonelada, é colocado um ou mais pesos certificados, que chegue a o valor total correspondente à carga máxima e a

partir daí, verifica-se se a balança está exibindo o resultado esperado (VICTORINO, 2018).

## 2.4 SISTEMAS OPERACIONAIS

O OS (Sistema Operacional) é um programa que atua como intermediador entre o hardware de computador e o usuário. O OS tem dois propósitos principais, primeiro, fornecer um ambiente conveniente no qual o usuário possa utilizar seus programas e, segundo, utilizar o hardware do computador de uma maneira que aproveite todo o seu potencial (SILBERSCHATZ,1998).

Segundo Silberschatz (1998), o componente mais importante de um sistema de computação é o sistema operacional. O sistema de computação pode ser dividido em quatro componentes que são de suma importância para o funcionamento, sendo eles, hardware, sistema operacional, software e usuário e se comunicam entre si, conforme pode ser visto na FIGURA 1 abaixo.

Figura 1 - Componente de um sistema de computação



Fonte: WIKIMEDIA, 2019

## 2.5 SISTEMAS OPERACIONAIS MÓVEIS

Os sistemas operacionais móveis trazem uma nova maneira de se interagir com um sistema operacional, pois não atendem apenas a uma tarefa específica, como a dos sistemas embarcados, eles proporcionam um desenvolvimento totalmente diferente por se utilizarem de sensores, GPS, acelerômetros, teclados virtuais, widgets etc. (WASSERMAN, 2010).

### 2.5.1 Android

Segundo Lecheta (2013), o Android é um sistema operacional baseado no kernel Linux que foi desenvolvido para ser utilizado, a princípio, em smartphones, porém hoje está presente em uma vasta gama de dispositivos diferentes como, por exemplo, em TVs e automóveis.

### 2.5.2 IOS

O IOS (iPhone Operation System) é o sistema operacional desenvolvido e utilizado pela empresa Apple. O sistema foi baseado no sistema operacional do MAC OS X e criado para atender os smartphones produzidos pela Apple (LECHETA, 2013).

## 2.6 DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS

Segundo Barros (2015), existem dois tipos de abordagens para o desenvolvimento de aplicativo para dispositivos móveis, abordagem nativa e multiplataforma.

### 2.6.1 Abordagem Nativa

A abordagem nativa se refere aos aplicativos que são desenvolvidos especificamente para um tipo de plataforma móvel, por exemplo, quando uma aplicação é desenvolvida nativamente para Android seu desenvolvimento é baseado

na linguagem Java e tem como IDE o Android SDK. Já para o desenvolvimento em IOS, se utiliza a linguagem Objective-C juntamente com a Xcode IDE (BARROS, 2015)

## **2.6.2 Abordagem Multiplataforma**

A abordagem multiplataforma tem como objetivo desenvolver aplicações para várias plataformas diferentes utilizando apenas um único código fonte. Segundo Barros (2015), existem 4 tipos de abordagem multiplataforma, sendo elas, abordagem WEB, híbrida, interpretada e compilação-cruzada.

### **2.6.2.1 Híbrida**

A abordagem híbrida é desenvolvida utilizando a abordagem WEB, porém é executada de forma nativa, utilizando do motor WEBKit para exibição da interface e intermédio entre o utilizador. Desta maneira, a abordagem híbrida consegue extrair o melhor das outras duas abordagens, utilizando de forma eficiente os navegadores e possibilitando a utilização de API's nativas dos dispositivos (BARROS, 2015, apud, SOUZA, 2017).

### **2.6.2.2 Web**

Todas as aplicações desenvolvidas utilizando das tecnologias web, HTML, CSS e Javascript com o foco em serem executadas em navegadores dos dispositivos dos clientes, são consideradas parte da abordagem WEB.

A desvantagem da utilização dessa abordagem é pelo fato de não conseguir acesso as API's nativas dos dispositivos, desta maneira tendo que utilizar os recursos dos servidores e por isso ficando refém do acesso a internet. Suas vantagens ficam por conta de as aplicações estarem rodando em um servidor, dessa maneira não depende do dispositivo do cliente para efetuar atualizações e podem ser utilizados em qualquer tipo de dispositivo que tenha um navegador com suporte aos padrões da aplicação (BARROS, 2015, apud, SOUZA, 2017).

### 2.6.2.3 Interpretada

As aplicações desenvolvidas utilizando a abordagem interpretada, na maioria das vezes são escritas utilizando uma linguagem específica e possuem um interpretador que a converte para a plataforma na qual será utilizada. O acesso aos recursos nativos é intermediado por uma camada de abstração. A vantagem é que a abordagem interpretada fornece aparência nativa, porém por se tratar de um aplicativo que necessita ser interpretado antes de ser executado, seu desempenho é inferior ao de um nativo (BARROS, 2015, apud, SOUZA, 2017).

### 2.6.2.4 Compilação-cruzada

A abordagem de desenvolvimento cruzada, cria códigos nativos, convertendo o código fonte para o da plataforma alvo, considerando, assim, os aplicativos como nativos e possuindo todos os recursos disponíveis nas plataformas. A desvantagem fica por conta das interfaces não podem ser reutilizadas em outras plataformas, desta maneira o programador tem que desenvolver cada interface para cada uma das plataformas alvo (BARROS, 2015, apud, SOUZA, 2017).

### 3 FERRAMENTAS DE DESENVOLVIMENTO

#### 3.1 IONIC

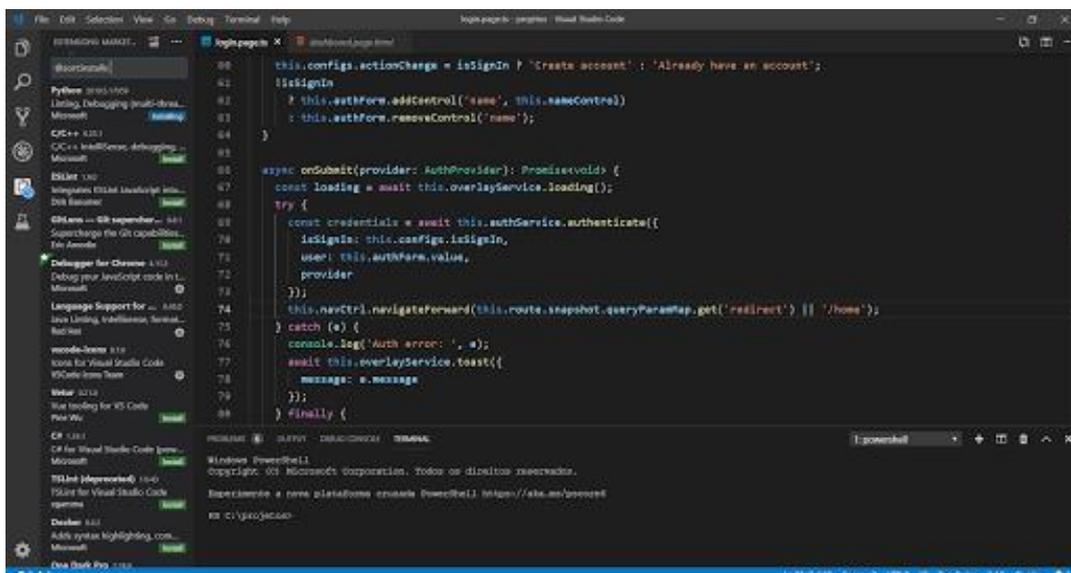
O Ionic Framework (2019) é uma poderosa ferramenta de código aberto que utiliza as tecnologias HTML, CSS e JavaScript para desenvolver aplicativos híbridos (Mobile e Desktop) de alta qualidade e desempenho.

O Ionic Framework é focado na experiência do usuário do front-end, ou interação da interface do usuário de um aplicativo (controles, interações, gestos, animações). É fácil de aprender e se integra muito bem com outras bibliotecas ou estruturas, como o Angular, ou pode ser usado como autônomo sem uma estrutura frontend usando um simples script de inclusão (IONIC, 2019)

#### 3.2 VISUAL STUDIO CODE (VSC)

Segundo o site Visual Studio Code (2019), O VSC é um editor de código-fonte gratuito leve e poderoso, ele está disponível para as plataformas Windows, macOS e Linux. Por padrão, já conta com suporte a JavaScript, TypeScript e Node.js e conta com diversas extensões para outras linguagens (como C #, Java, C ++, Python, PHP, Go) e tempos de execução (como .NET e Unity), como pode ser visto na Figura 2.

Figura 2 – Visão geral do Visual Studio Code

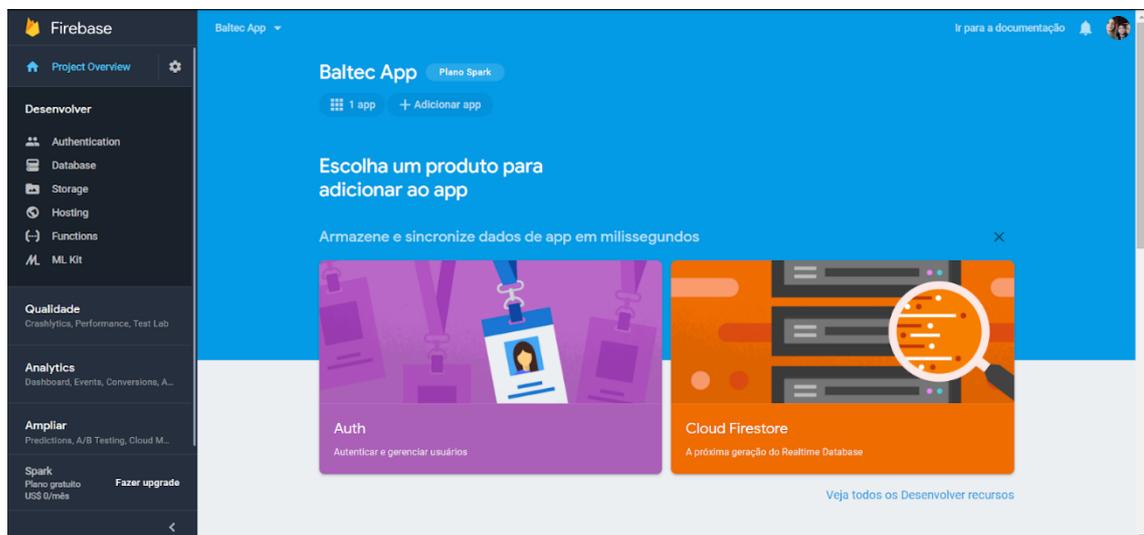


Fonte: Próprio Autor a partir de tela do software

### 3.3 FIREBASE

Segundo Adriano (2018), o Firebase é uma plataforma do Google que nos auxilia a desenvolver aplicativos de forma rápida e de alta qualidade. A plataforma nos disponibiliza de forma gratuita várias ferramentas e serviços que são simples de implementar. Com apenas alguns passos, o usuário já consegue disponibilizar funcionalidades no seu aplicativo.

Figura 3 - Visão geral da plataforma Firebase



Fonte: Próprio Autor a partir da tela do software

### 3.4 HTML 5

O HTML5 veio para revolucionar a forma como interagimos com os navegadores. Este proporcionou uma maior segurança e navegabilidade entre as páginas dentro dos navegadores. HTML5 é a mais recente evolução do padrão que define o HTML

### 3.5 TYPESCRIPT

Segundo Guedes (2017), o TypeScript, com sua estrutura fortemente tipada, veio para facilitar o desenvolvimento JavaScript, de forma a diminuir os erros de codificação. O TypeScript faz com que a linguagem fique mais produtiva e reutilizável.

Pelo fato de ser compilada em JavaScript, tem uma vasta compatibilidade com os navegadores.

## 4 METODOLOGIA DE DESENVOLVIMENTO

Para resolver o problema apresentado, ou seja, desenvolver o sistema proposto, foi utilizada metodologia baseada no Processo Unificado (UP). O UP é um processo de desenvolvimento de software orientado a objetos, iterativo e incremental, dirigido por casos de uso (JACOBSON, BOOCH e RUMBAUGH, 1999). O UP busca aproveitar ao máximo a Linguagem de Modelagem Unificada (UML).

O UP divide o desenvolvimento de um sistema em quatro fases:

- **Concepção:** em que são descobertos os principais requisitos do sistema e identificados os casos de uso de alto nível, buscando entender sua extensão e verificar sua viabilidade;
- **Elaboração:** onde é realizada uma análise detalhada dos requisitos, através da expansão dos casos de uso e geração do modelo conceitual;
- **Construção:** fase em que é efetuada a maior parte da implementação e realização de testes;
- **Transição:** que consiste, basicamente, nos testes finais e implantação do sistema.

Iniciando a fase de concepção, a primeira atividade realizada foi a modelagem de negócio, que tratou da identificação dos processos a serem automatizados. Com base nas informações obtidas nessa atividade, foi iniciada a modelagem de sistema. O primeiro passo para modelagem do sistema foi o levantamento de requisitos, seguido da elaboração do diagrama de caso de usos. Concluídos esses dois passos, a etapa de concepção foi, portanto, finalizada.

Na fase de elaboração, dando continuidade a modelagem do sistema, foi efetuada a expansão dos casos de uso e elaborado o diagrama de classes.

Durante a fase de construção, foi realizada, a implementação do sistema. A camada de apresentação foi implementada se utilizando do Ionic Framework. Já para a camada de aplicação, foi utilizado o Angular. Enfim, a camada de dados foi implementada utilizando o Firebase.

A fase de transição tem menor foco neste trabalho, pois a implantação do sistema será feita em um futuro próximo.

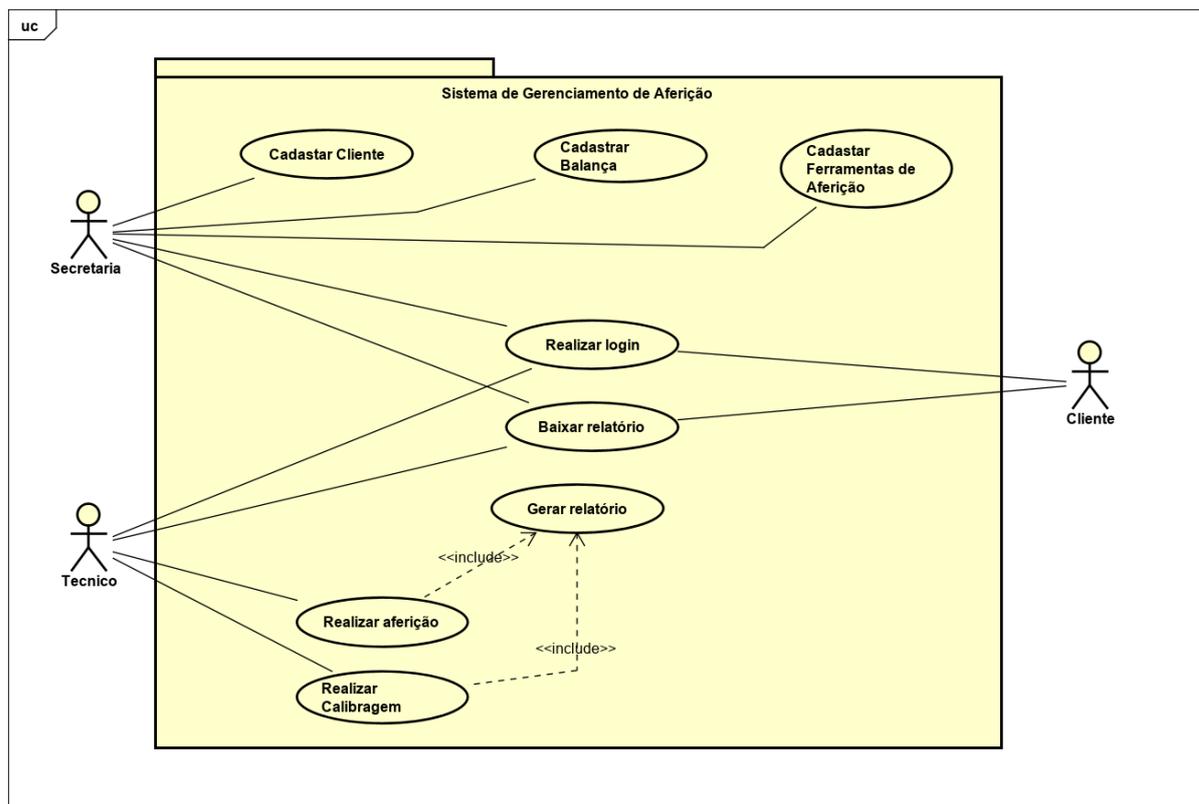
#### 4.1 CASO DE USO

Para Guedes (2008, apud SOUZA, H., 2017), por ser o diagrama mais abstrato entre todos os diagramas da UML (Unified Modeling Language ou linguagem de modelagem unificada), o Diagrama de caso de uso consegue apresentar de forma simples para qualquer pessoa os comportamentos e funcionalidades principais do sistema que será desenvolvido.

O diagrama de caso de uso é de grande importância na análise de requisitos, pois possibilita ao analista identificar e documentar as funcionalidades, características e pessoas envolvidas em cada caso de uso do sistema.

A figura 4 mostra o diagrama do caso de uso do Sistema de Gerenciamento de Aferição que está sendo desenvolvido.

Figura 4 - Caso de uso geral do sistema



Fonte: Próprio autor

## 4.2 DOCUMENTAÇÃO DO CASO DE USO

Para Guedes (2008, p. 48), uma documentação de caso de uso deve ser uma descrição bem básica das etapas, parâmetros, pós-condições, pré-condições, restrições e validações que determinado ator deverá executar para concluir o objetivo do caso de uso.

A partir do modelo descrito por Guedes (2008), a tabela 1 irá definir o modelo de documentação que iremos utilizar para documentar o sistema que está sendo desenvolvido.

Tabela 1 - Modelo de documento de caso de uso

<b>Nome</b>		
<b>Caso de uso geral</b>		
<b>Ator principal</b>		
<b>Atores secundários</b>		
<b>Resumo</b>		
<b>Pré-condição</b>		
<b>Pós-condição</b>		
	<b>Ação do ator</b>	<b>Ação do sistema</b>
<b>Restrições/Validações</b>		

Os modelos a seguir irão mostrar e especificar as funcionalidades do sistema em desenvolvimento.

Tabela 2 - Documentação do caso de uso Realizar aferição

<b>Nome</b>	Realizar aferição
<b>Caso de uso geral</b>	
<b>Ator principal</b>	Técnico
<b>Atores secundários</b>	

<b>Resumo</b>	O caso de uso mencionado descreve as etapas a do processo de escolha e testes das balanças.	
<b>Pré-condição</b>	As informações do cliente, balança e equipamento devem ser cadastradas previamente.	
<b>Pós-condição</b>	O sistema irá efetuar uma verificação nos dados informados durante o teste.	
	<b>Ação do ator</b>	<b>Ação do sistema</b>
	1 - Clicar no botão “Realizar aferição” e informar os dados solicitados pelo sistema.	2 - Solicitar dados do teste
		3 - Validar informações
		4 - Exibir mensagens conclusão
<b>Restrições/Validações</b>	O usuário deve estar logado; O usuário deve ter permissão de técnico	

Tabela 3 - Documentação do caso de uso Cadastrar cliente

<b>Nome</b>	Cadastrar cliente	
<b>Caso de uso geral</b>		
<b>Ator principal</b>	Secretária	
<b>Atores secundários</b>		
<b>Resumo</b>	O caso de uso mencionado descreve as etapas do cadastro das informações do cliente	
<b>Pré-condição</b>		
<b>Pós-condição</b>	O sistema irá efetuar uma verificação se o cliente já está cadastrado e se tem seu CNPJ valido.	
	<b>Ação do ator</b>	<b>Ação do sistema</b>
	1 - Clicar no botão “Cadastrar Clientes” e informar os dados solicitados pelo sistema.	2 - Validar informações do cliente
		3 - Salvar cadastro no dispositivo

<b>Restrições/Validações</b>	O usuário deve estar logado; O usuário deve ter permissão de secretária O dispositivo deve ter acesso a internet
------------------------------	--

Tabela 4 - Documentação do caso de uso Cadastrar balanças

<b>Nome</b>	Cadastrar balança	
<b>Caso de uso geral</b>		
<b>Ator principal</b>	Secretária	
<b>Atores secundários</b>		
<b>Resumo</b>	O caso de uso mencionado descreve as etapas do cadastro das balanças com suas informações únicas	
<b>Pré-condição</b>		
<b>Pós-condição</b>	O sistema irá efetuar uma verificação se a balança já está cadastrada com base nos dados únicos	
	<b>Ação do ator</b>	<b>Ação do sistema</b>
	1 - Clicar no botão “Cadastrar Balanças” e informar os dados solicitados pelo sistema.	2 - Validar informações da balança
		3 - Salvar cadastro no dispositivo
<b>Restrições/Validações</b>	O usuário deve estar logado; O usuário deve ter permissão de secretária O dispositivo deve ter acesso a internet	

Tabela 5 - Documentação do caso de uso Cadastrar ferramentas de aferição

<b>Nome</b>	Cadastrar ferramentas de aferição
<b>Caso de uso geral</b>	
<b>Ator principal</b>	Secretária
<b>Atores secundários</b>	
<b>Resumo</b>	O caso de uso mencionado descreve as etapas do cadastro das ferramentas de aferição
<b>Pré-condição</b>	

<b>Pós-condição</b>	O sistema irá efetuar uma verificação se a ferramenta de aferição já está cadastrada	
	<b>Ação do ator</b>	<b>Ação do sistema</b>
	1 - Clicar no botão “Cadastrar ferramentas de aferição” e informar os dados solicitados pelo sistema.	2 - Validar informações das ferramentas de aferição
		3 - Salvar cadastro no dispositivo
<b>Restrições/Validações</b>	O usuário deve estar logado; O usuário deve ter permissão de secretária O dispositivo deve ter acesso a internet	

Tabela 6 - Documentação do caso de uso Realizar login

<b>Nome</b>	Realizar login	
<b>Caso de uso geral</b>		
<b>Ator principal</b>	Secretária, Técnico e Cliente	
<b>Atores secundários</b>		
<b>Resumo</b>	O caso de uso mencionado descreve as etapas da autenticação dos usuários no sistema	
<b>Pré-condição</b>	O usuário deve existir	
<b>Pós-condição</b>	O sistema irá efetuar uma verificação da permissão do usuário e irá exibir uma mensagem de sucesso ou de erro.	
	<b>Ação do ator</b>	<b>Ação do sistema</b>
	1 - Clicar no botão “Entrar”	3 - Validar informações de permissão do usuário
	2 – Informar dados	4 - Inicia a sessão
<b>Restrições/Validações</b>	Os dados pelo usuário devem estar corretos	

Tabela 7 - Documentação do caso de uso Baixar relatório

<b>Nome</b>	Baixar relatório
<b>Caso de uso geral</b>	

<b>Ator principal</b>	Secretária, Técnico e Cliente	
<b>Atores secundários</b>		
<b>Resumo</b>	O caso de uso mencionado descreve as etapas para gerar e efetuar o download dos arquivos das aferições e relatórios do sistema	
<b>Pré-condição</b>	Tem que haver aferições concluídas.	
<b>Pós-condição</b>	Exibir mensagens de download concluído	
	Ação do ator	Ação do sistema
	1 – Seleciona aferições concluídas	2 – Gera PDF dos arquivos correspondentes e mostra na tela
	3 – Clica em “download de arquivo”	4 – Baixa arquivo
<b>Restrições/Validações</b>	O usuário deve estar logado;	

Tabela 8 - Documentação do caso de uso gerar relatório

<b>Nome</b>	Gerar relatório	
<b>Caso de uso geral</b>		
<b>Ator principal</b>	Técnico	
<b>Atores secundários</b>		
<b>Resumo</b>	O caso de uso mencionado descreve as etapas do processo que irá analisar as informações obtidas com a aferição concluída e irá gerar um relatório e um documento de certificação da balança.	
<b>Pré-condição</b>	Tem que haver aferições analisadas.	
<b>Pós-condição</b>	Informar ao cliente e secretária que os documentos estão disponíveis para download	
	Ação do ator	Ação do sistema
	1 – Seleciona aferições analisadas	2 – Analisa e processa informações
	3 – Clica em “Gerar documentos”	4 – Documentos são gerados e disponibilizados para download
<b>Restrições/Validações</b>	O usuário deve estar logado;	

Tabela 9 - Documentação do caso de uso realizar calibragem

<b>Nome</b>	Realizar calibragem	
<b>Caso de uso geral</b>		
<b>Ator principal</b>	Técnico	
<b>Atores secundários</b>		
<b>Resumo</b>	O caso de uso mencionado descreve as etapas do processo que irá efetuar a manutenção da balança com base nos problemas apresentados após a aferição	
<b>Pré-condição</b>	Tem que haver aferições pendentes de manutenção.	
<b>Pós-condição</b>	Perguntar se houve conserto da balança	
	<b>Ação do ator</b>	<b>Ação do sistema</b>
	1 – Seleciona aferições pendentes de aferição	2 – Perguntar se irá efetuar a manutenção
	3 – Clica em “Sim” ou “Não”	4 – Caso sim, executar o teste de aferição. Caso não, informar mensagem de que a balança está inoperante e guardar informações no histórico do cliente
<b>Restrições/Validações</b>	O usuário deve estar logado;	

#### 4.3 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Segundo Stellman e Greene (2005) a identificação das necessidades do usuário e dos objetivos que se deseja alcançar é de suma importância na concepção de um sistema. Embora isso pareça óbvio, uma grande quantidade de projetos sofre atrasos ou até mesmo fracassam por iniciar seu desenvolvimento sem que o comportamento desejado para o novo sistema esteja bem definido. Sendo assim, fica evidente a importância da identificação dos requisitos, já nas primeiras etapas de um projeto.

Requisitos descrevem funções a serem desempenhadas pelo sistema, comportamentos desejados ou restrições ao seu funcionamento. De acordo com Sommerville (2011) é útil a distinção de requisitos em dois tipos: requisitos de usuário, que são descrições mais abstratas de alto nível, e requisitos de sistema, os quais são mais detalhados, com propósito de definir exatamente o que será implementado.

Os requisitos de sistema podem, ainda, ser classificados em funcionais ou não funcionais. De forma resumida, requisitos funcionais descrevem o que o sistema deve fazer, ou seja, suas funções e comportamentos enquanto requisitos não funcionais representam restrições ou qualificações ao seu funcionamento. (Sommerville, 2011)

Este trabalho segue os conceitos de Wazlawick (2011, p. 38-44) para definir o modelo das especificações do aplicativo que desenvolvido. Para as categorias das especificações não funcionais, seguem os seguintes conceitos:

- **Usabilidade** – Engloba todos os fatores que influenciam na interação do cliente, por exemplo: mensagens de aviso e auxílio na utilização de funções.
- **Confiabilidade** – Diz respeito ao gerenciamento de falhas, como os tipos de falhas que o sistema devera gerenciar, consumo, entrada e saída de dados, comunicação etc.
- **Performance** – Essa categoria fala sobre os princípios de eficiência e precisão que o software deve seguir, como o tempo de início e fim de um processo.
- **Configurabilidade** – Tem a intenção de definir quais configurações poderão ser parametrizadas pelo usuário.
- **Segurança** – Definir quais os níveis de perfil de cada usuário para que se crie um padrão de segurança.
- **Implementação** – Definir e inserir regras no desenvolvimento que estarão presentes nas funcionalidades.
- **Interface** – Cria uma regra de como será o as normas e comportamento da interface.

Tabela 10 - Exemplo de especificações funcionais e não funcionais

<b>F1 – Nome da especificação</b>				<b>Oculto ( )</b>
<b>Descrição:</b> Descrever de forma detalhada o requisito				
<b>Requisitos não funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
NF1.1	Descrever a restrição do requisito não funcional	--	Sim/Não	Sim/Não

Os quadros mostrados a seguir iram especificar as funcionalidades do sistema

Tabela 11 - F1 - Autenticação no sistema

<b>F1 – Autenticação no sistema</b>				<b>Oculto ( )</b>
<b>Descrição:</b> Para poder utilizar as funções do sistema, o usuário deve se identificar				
<b>Requisitos não funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
NF1.1 – Controle de Usuário	O usuário deve ter um registro no sistema	Segurança	Não	Sim

Tabela 12 - F2 - Seleção de cliente para aferição

<b>F2 – Seleção de clientes para aferição</b>				<b>Oculto ( )</b>
<b>Descrição:</b> O sistema deve possibilitar ao usuário selecionar cliente para efetuar a aferição				
<b>Requisitos não funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
NF2.1 – Controle de cliente	O sistema deve exibir apenas os clientes que tenham um protocolo em aberto	Confiabilidade	Sim	Sim

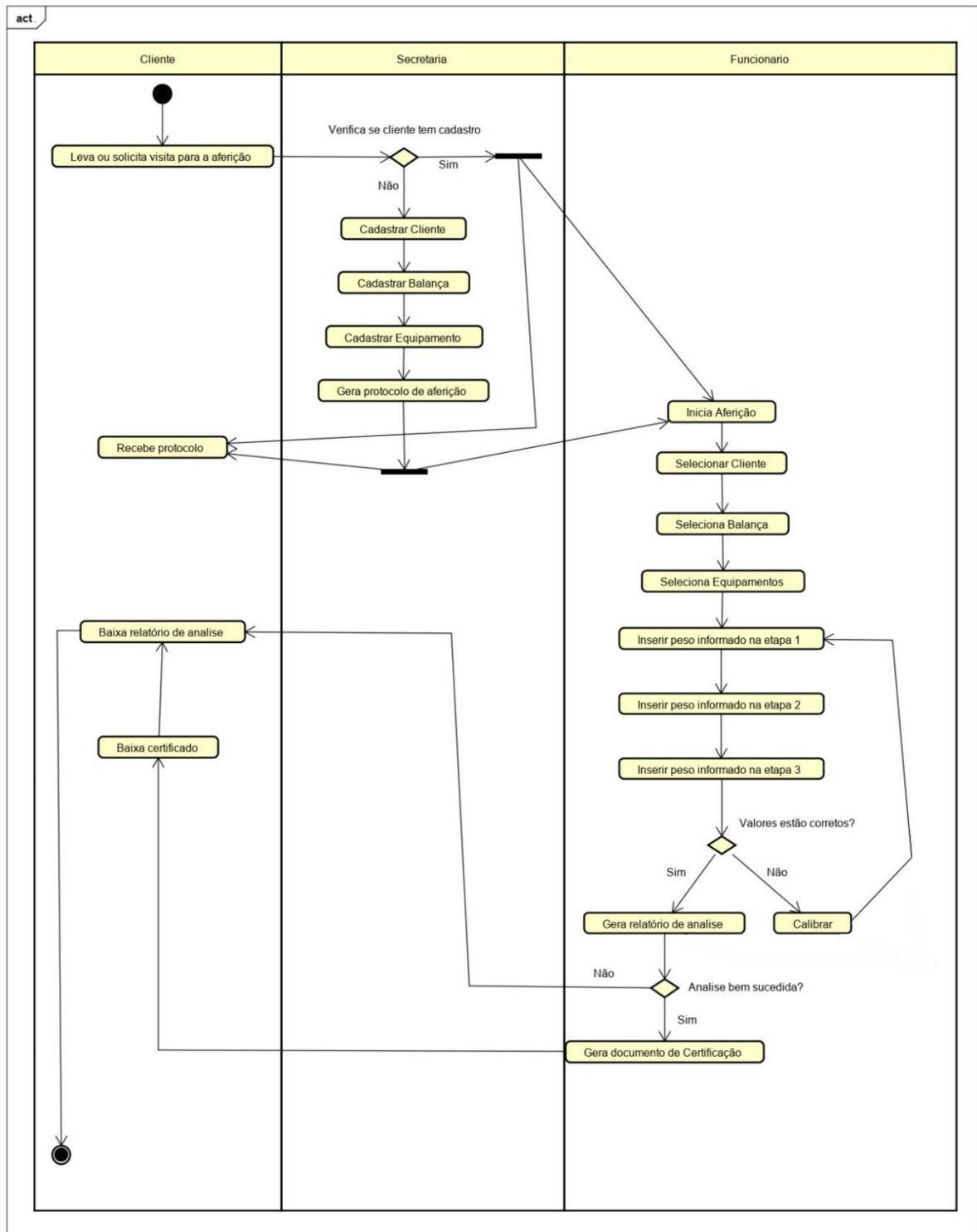
Tabela 13 - F3 - Seleção de ferramentas para aferição

<b>F3 – Seleção de ferramentas para aferição</b>				<b>Oculto ( )</b>
<b>Descrição:</b> O sistema deve possibilitar ao usuário selecionar ferramentas para efetuar a aferição				
<b>Requisitos não funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
NF3.1 – Controle de ferramentas	O sistema deve exibir apenas as ferramentas que estejam vinculadas a balança do cliente	Confiabilidade	Sim	Sim

#### 4.4 DIAGRAMA DE ATIVIDADE

Para melhor entendimento de cada um dos casos de uso, foi elaborado um diagrama de atividade, que descreve os passos realizados para conclusão do caso de uso em questão.

Figura 5 - Diagrama de atividades geral do sistema



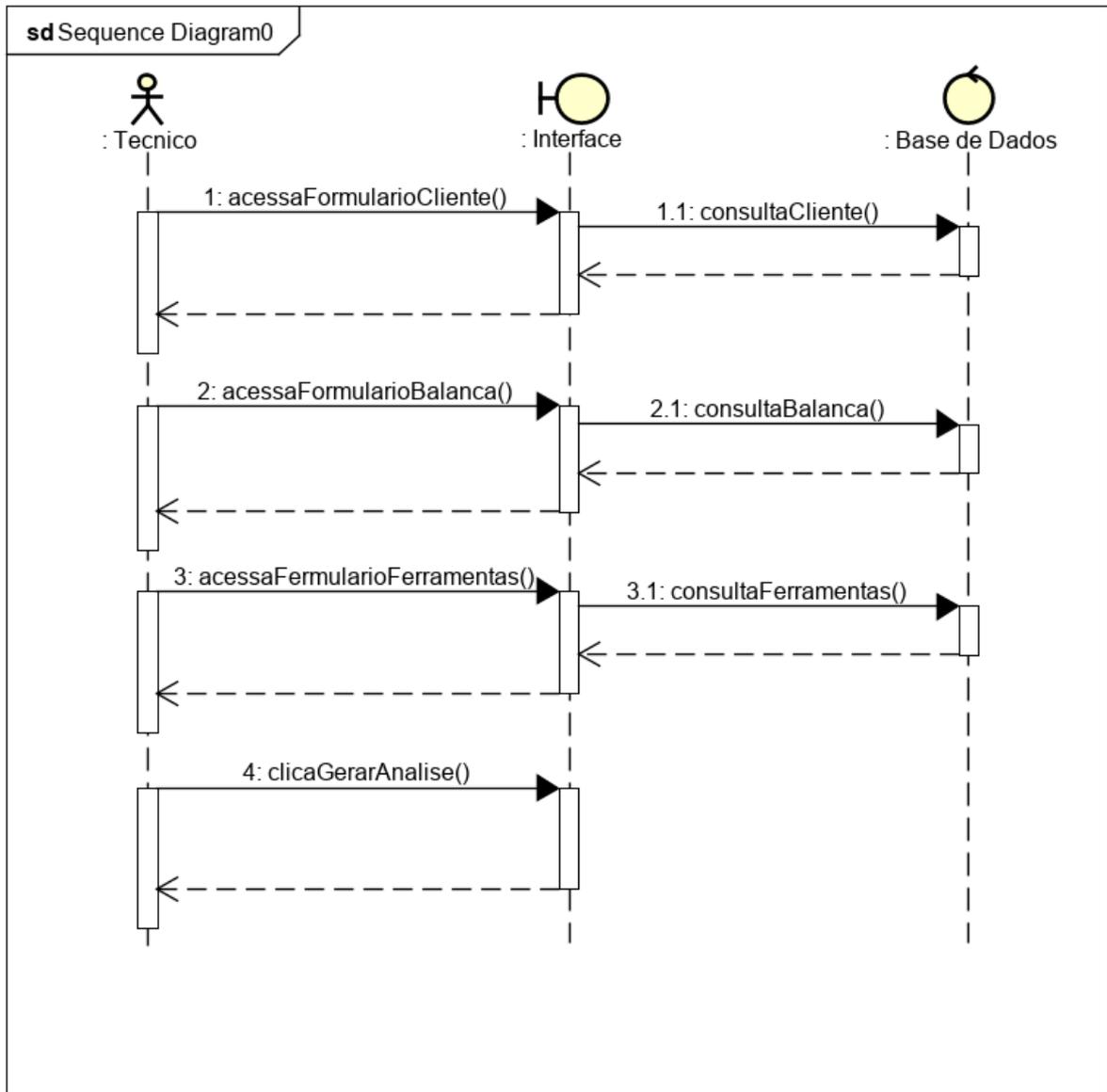
Fonte: Próprio Autor

#### 4.5 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência, representa sequência de mensagens trocadas entre objetos para a realização de um processo. Quando utilizado para representar um caso

de uso ele é então denominado diagrama de sequência de sistema. Segundo Larman (2002), o principal objetivo para descrição de um caso de uso por um diagrama de sequência é a identificação das operações de sistema.

Figura 6 - Diagrama de sequência



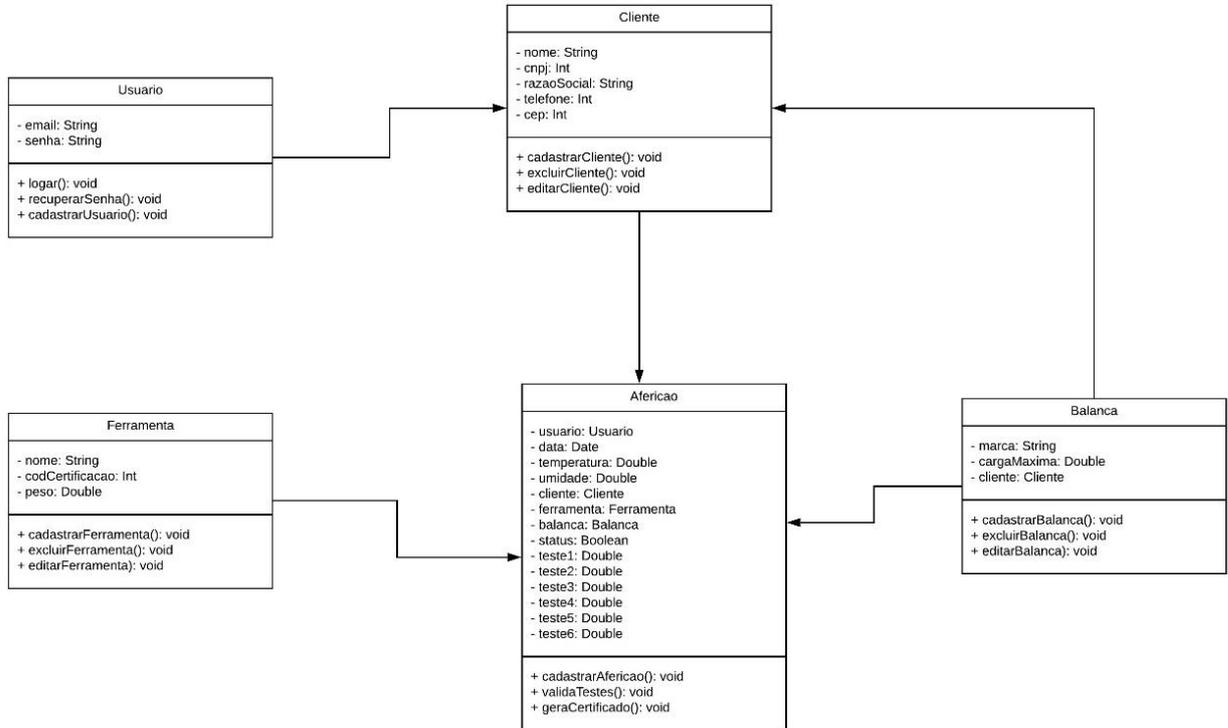
Fonte: Próprio Autor

#### 4.6 DIAGRAMA DE CLASSE

Ao fim da modelagem do diagrama de sequência, é construído o diagrama de classes que será refinado durante a fase de design lógico. O diagrama de classe não

deve incluir aspectos tecnológicos, apenas os aspectos relacionados a lógica de negócio são incluídos (WAZLAWICK, 2011).

Figura 7 - Diagrama de classe



Fonte: Próprio Autor

## **5 DESENVOLVIMENTO**

### **5.1 COMUNICAÇÃO ENTRE OS SERVIÇOS**

Com as ferramentas de desenvolvimento instaladas e configuradas, pudemos começar a escrever quais as informações que o sistema enviaria para o Firebase para que fossem salvas e ficassem disponíveis para consultas futuras. Para isso foram criados arquivos específicos para cada informação útil do sistema como, clientes, balanças, ferramentas, aferições e autenticação.

#### **5.1.1 Exemplo de comunicação**

Logo abaixo temos um exemplo que foi retirado do sistema, o código contido na imagem faz a comunicação entre o sistema e o Firebase.

Figura 8 - Código de comunicação

```
getClientes() {
  return this.clienteCollection.snapshotChanges().pipe(
    map(actions => {
      return actions.map(a => {
        const data = a.payload.doc.data();
        const id = a.payload.doc.id;

        return { id, ...data };
      });
    })
  );
}

addCliente(cliente: Cliente) {
  return this.clienteCollection.add(cliente);
}

getCliente(id: string) {
  return this.clienteCollection.doc<Cliente>(id).valueChanges();
}

updateCliente(id: string, cliente: Cliente) {
  return this.clienteCollection.doc<Cliente>(id).update(cliente);
}

deleteCliente(id: string) {
  return this.clienteCollection.doc(id).delete();
}
```

Fonte: Próprio Autor.

- O `getClientes()`, o Firebase recupera todos os clientes cadastrados no banco de dados.
- O `addCliente()` salvar no banco de dados um novo cliente a partir das informações passadas no parâmetro "Cliente".
- O `getCliente()` retorna apenas um cliente com o id igual ao passado no parâmetro "id".
- O `updateCliente()` atualizará as informações do cliente pelas informações novas passadas no parâmetro "cliente" e irá identificar em qual cliente fazer essa alteração a partir do parâmetro "id"

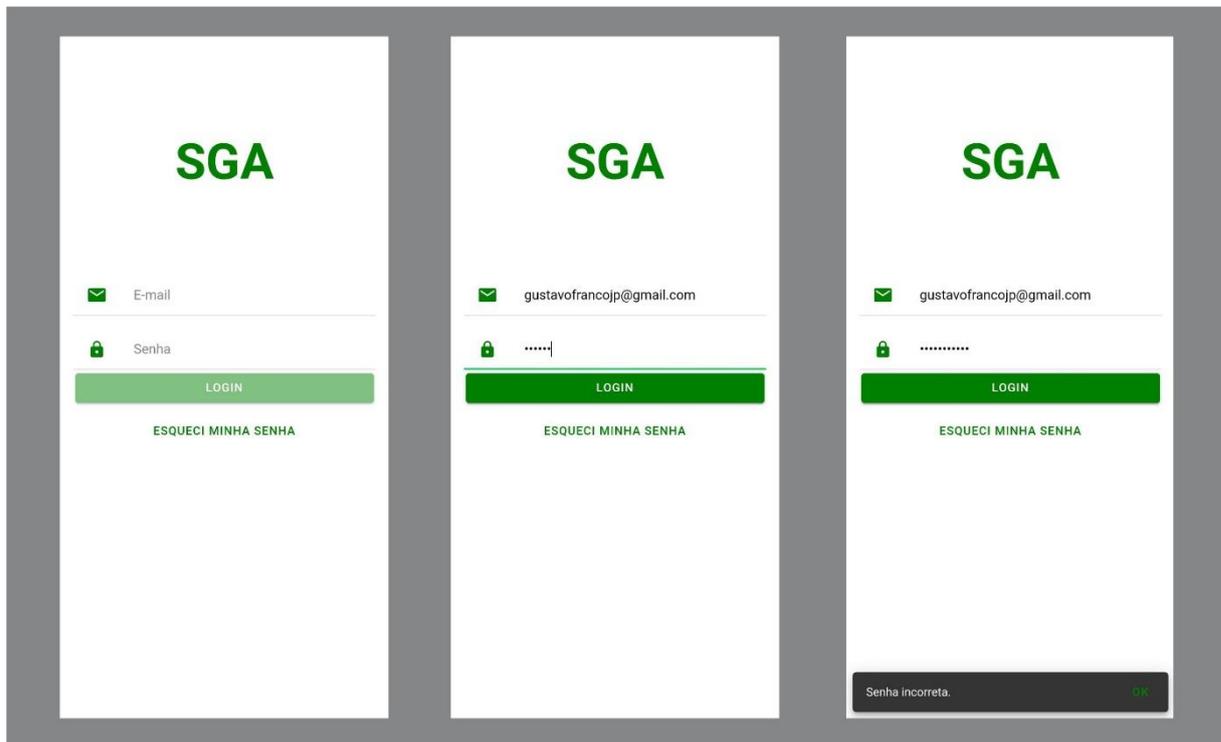
- Já o deleteCliente() excluirá o cliente existente no banco de dados com o id igual ao passado pelo parâmetro “id”.

## 5.2 FUNCIONALIDADES

Durante a explicação das funcionalidades do sistema, serão apresentadas imagens das telas geradas para o sistema operacional Android e IOS. Desta maneira poderemos fazer uma comparação visual de como ficou o resultado para cada compilação feita pelo Ionic.

### 5.2.1 Autenticação

Figura 9 – Tela de autenticação android

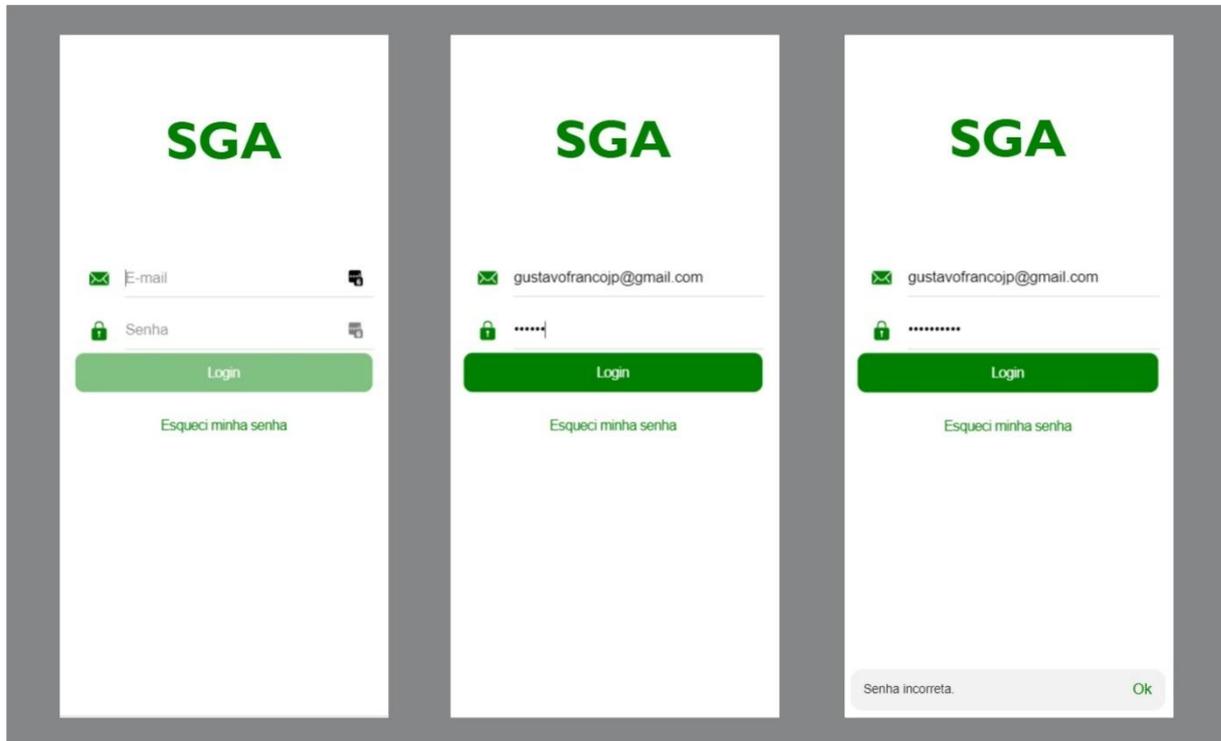


Fonte: Próprio Autor

A tela de autenticação é a porta de entrada para acessar as funcionalidades do sistema. Por esse motivo, é importante que essa tela efetue validações de acesso para que não ocorram acessos indevidos. Conforme mostrado nas figuras 13 e 14, é possível inserir na tela o e-mail e a senha para ser autenticado no sistema. Caso

alguma informação esteja incorreta, o sistema irá exibir alertas nos campos de digitação e em mensagens flutuantes no rodapé da tela.

Figura 10 – Tela de autenticação IOS



Fonte: Próprio Autor

### 5.2.2 Menu de navegação

A tela inicial de navegação do sistema é a primeira parada do usuário após efetuar a autenticação. A partir dela é possível acessar todos os serviços do sistema, sendo eles: cliente, balanças, ferramentas, realizar aferição e documentos, conforme mostrado na Figura 15.

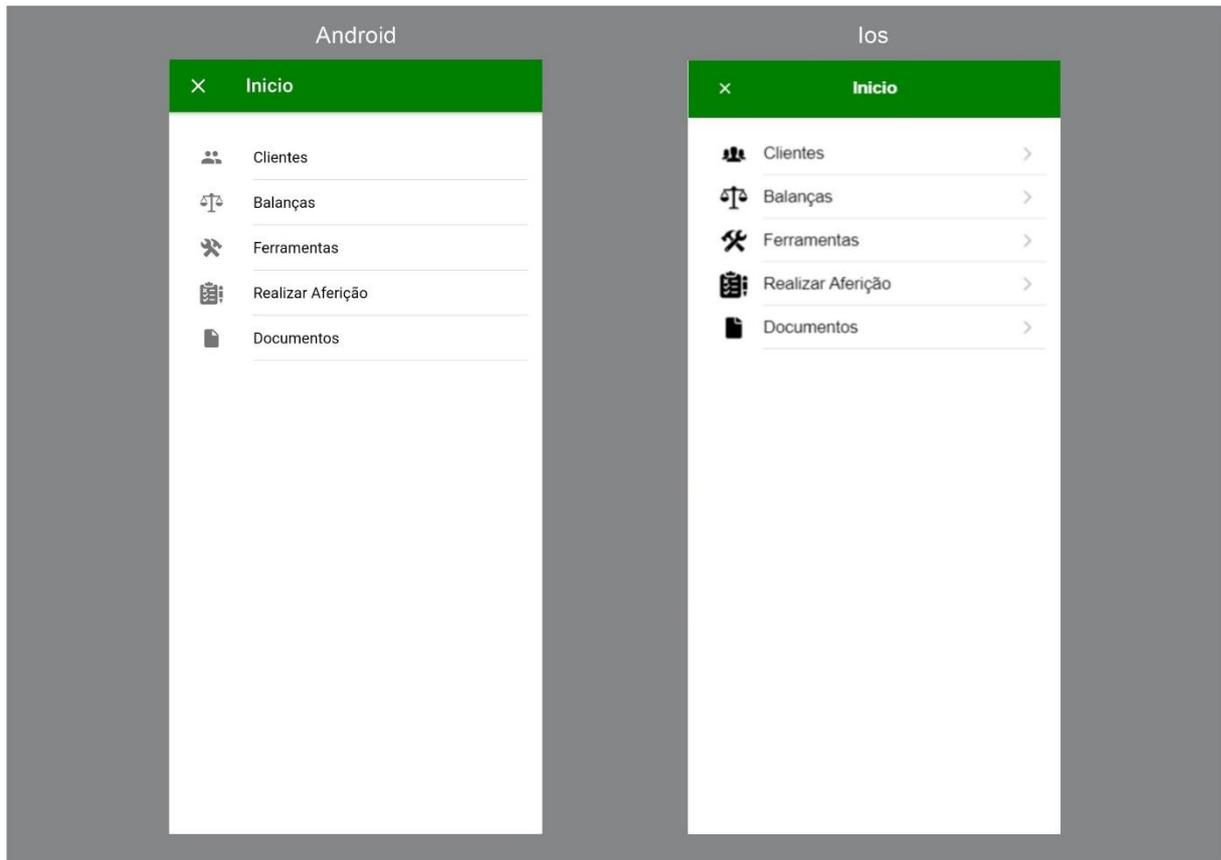
### 5.2.3 Consultas

As consultas do sistema são feitas através a primeira tela de cada funcionalidade, conforme mostrado na Figura 15, após clicar na opção de cliente o usuário será redirecionado para a tela de consulta do cliente, Figura 16. Nela é

possível consultar os clientes cadastrados pela lista e, também, pesquisar pelo nome do cliente utilizando o campo de pesquisa.

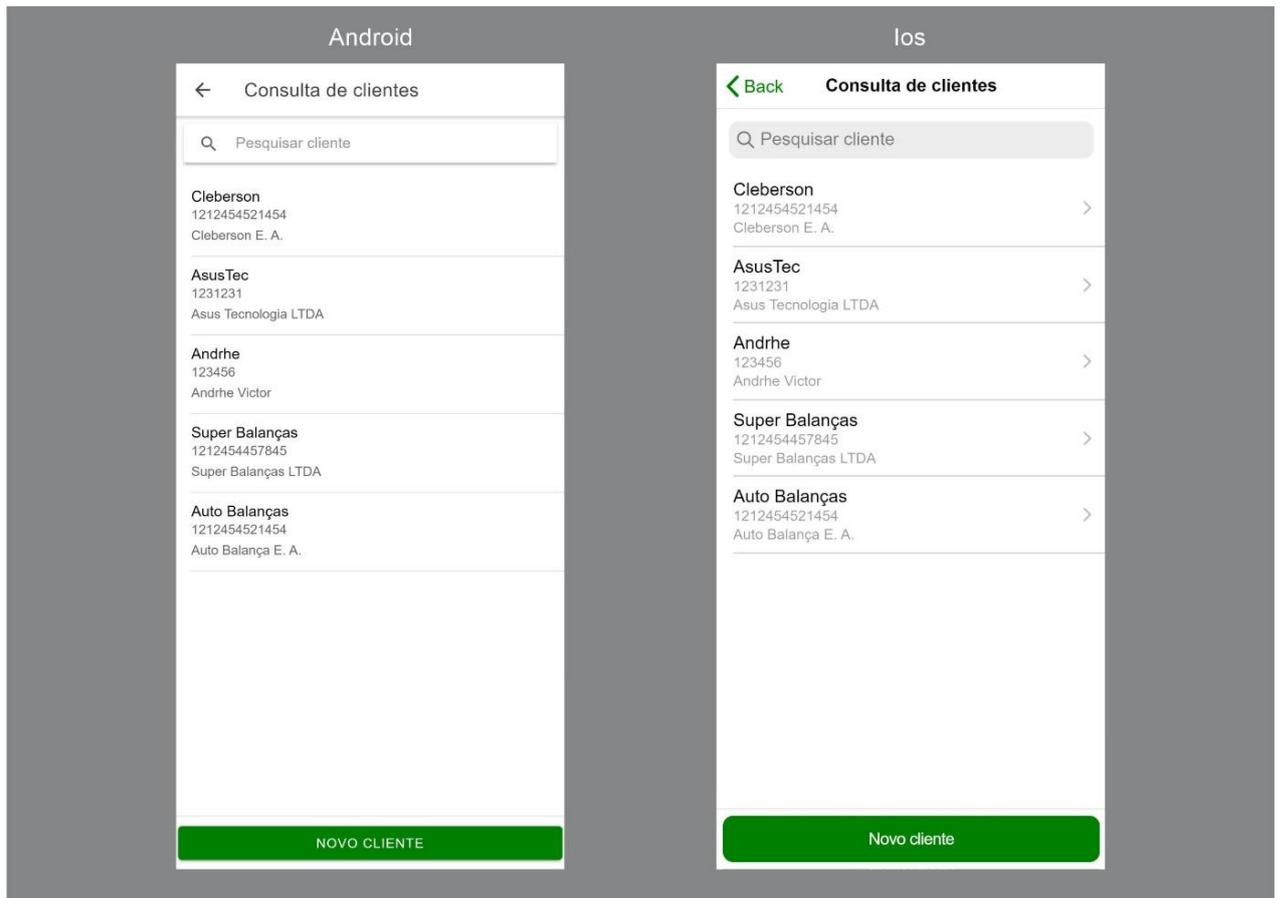
O padrão de consultas demonstrado na tela de consulta clientes vale também para as telas de consulta de balanças (Figura 17) e consulta ferramentas (Figura 18).

Figura 11 – Tela de navegação



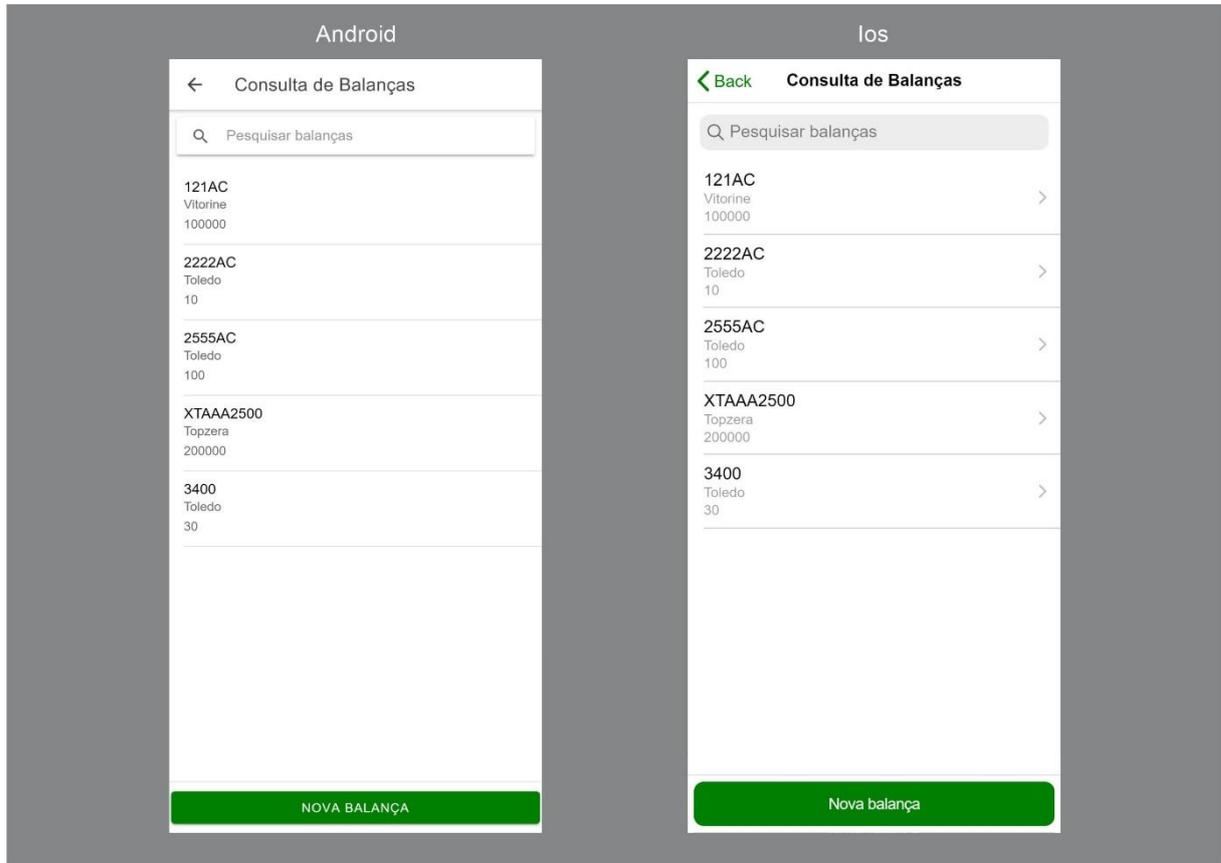
Fonte: Próprio Autor

Figura 12 - Tela de consulta de clientes



Fonte: Próprio Autor

Figura 13 – Tela de consulta de balanças



Fonte: Próprio Autor

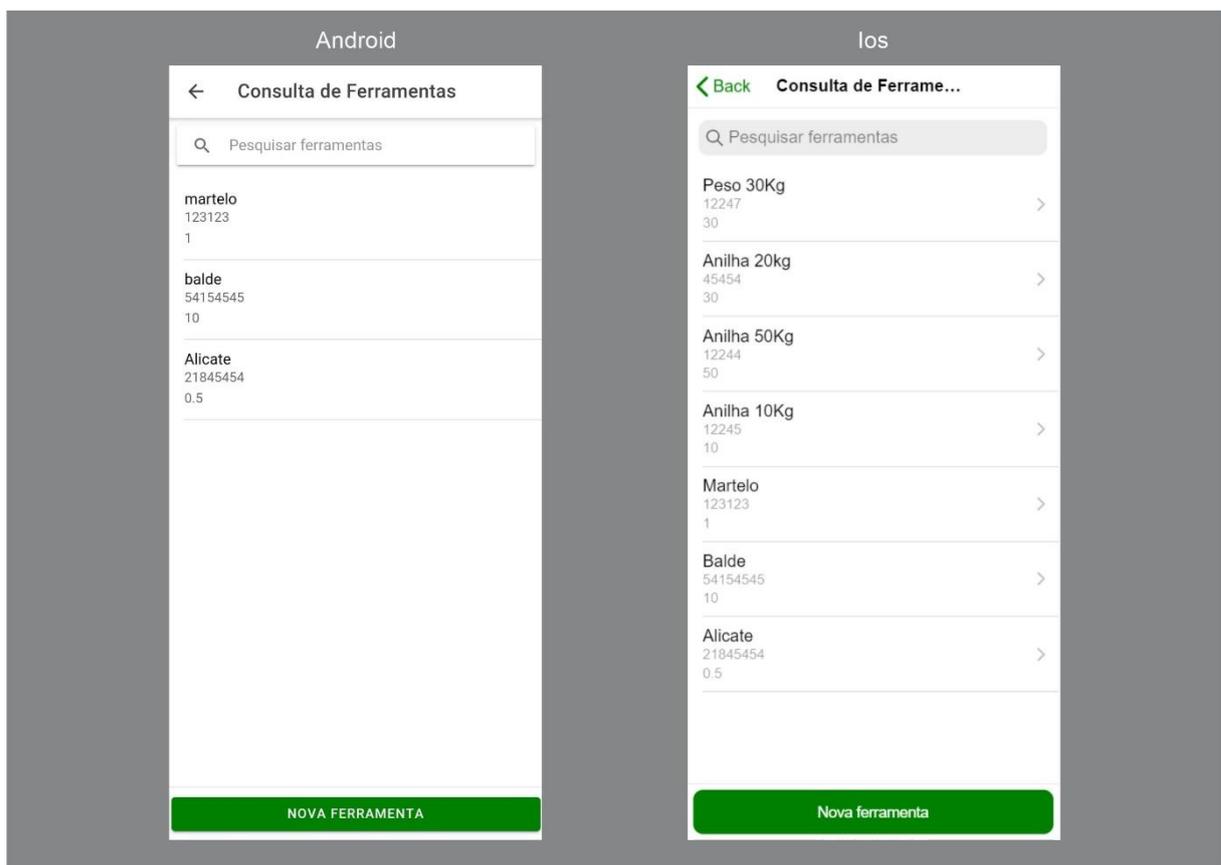
## 5.2.4 Cadastros

No rodapé de cada página existe um botão que redireciona o usuário para uma tela de cadastro, podendo ser cadastro de cliente, cadastro de balanças ou cadastro de ferramentas, dependendo da tela de consulta em que o usuário esteja.

Na tela de cadastro de cliente (Figura 19) é possível informar os dados do cliente que serão validados pelo formulário para que só seja possível finalizar o cadastro quando todas as informações estiverem de acordo com o exigido em cada campo.

O padrão de cadastro descrito acima vale para todas as telas de cadastro, sendo elas, cadastro de cliente, cadastro de balanças e cadastro de ferramentas.

Figura 14 – Tela de consulta de ferramentas



Fonte: Próprio Autor

Figura 15 – Tela de cadastro de clientes

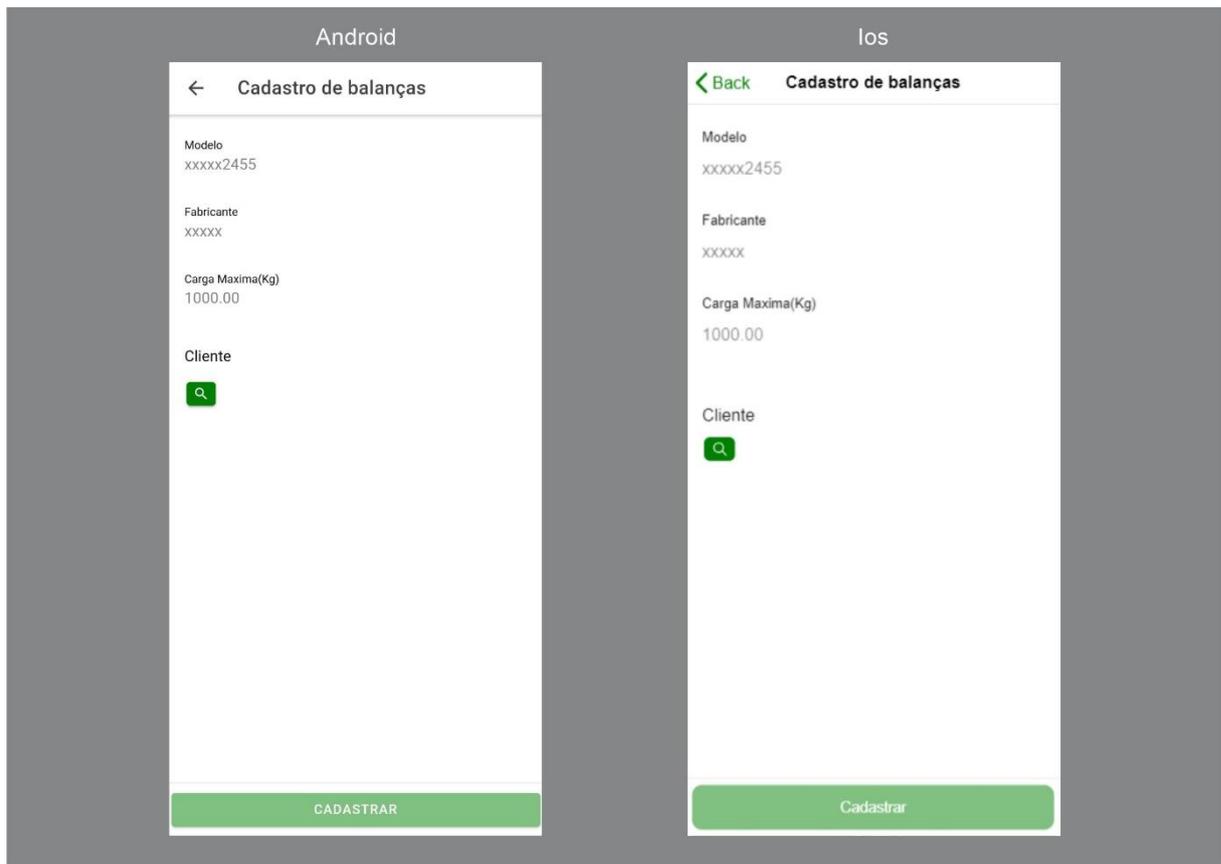
The image displays two mobile application screens for customer registration, side-by-side. The left screen is for Android and the right is for iOS. Both screens show the same data for a customer registration form.

Field	Value
CNPJ	000.000.000/0001-00
Nome	Nome do cliente
Razão Social	Razão Social
Telefone	(69) 90000-0000
CEP	00000-000

The Android screen (left) features a green button labeled "CADASTRAR" at the bottom. The iOS screen (right) features a green button labeled "Cadastrar" at the bottom. The iOS screen also includes a green back arrow icon and the text "Back" at the top left of the form area.

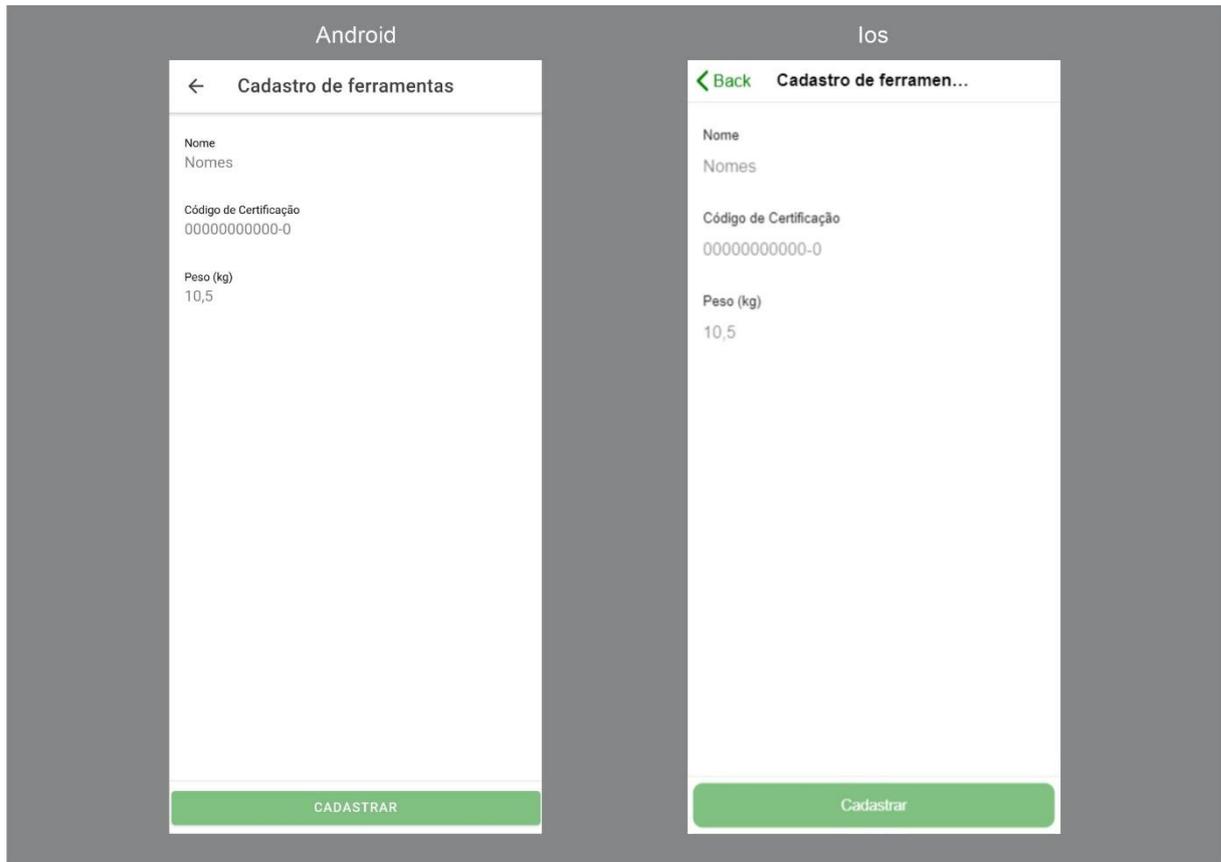
Fonte: Próprio Autor

Figura 16 – Tela de cadastro de balanças



Fonte: Próprio Autor

Figura 17 – Tela de cadastro de ferramentas



Fonte: Próprio Autor

### 5.2.5 Aferição

O processo de aferição do sistema é simples. Basta informar no formulário “Dados para Aferição” os dados do cliente, balança que será aferida, ferramentas utilizadas no processo de aferição e, nos dois últimos campos, adicionar as informações colhidas através dos sensores de temperatura e umidade que são exigidos para realização do procedimento (imagem da esquerda da Figura 22). Após concluído o primeiro formulário, basta clicar no botão “continue” para adicionar os dados obtidos durante os testes físicos feitos na balança. Ao concluir a etapa de testes e informá-los nos formulários do sistema, basta clicar no botão “finalizar” para que o sistema valide as informações inseridas e informe para o usuário se a balança necessita de calibração ou se ela está calibrada (Figura 23).

Figura 18 - Formulário de aferição 1

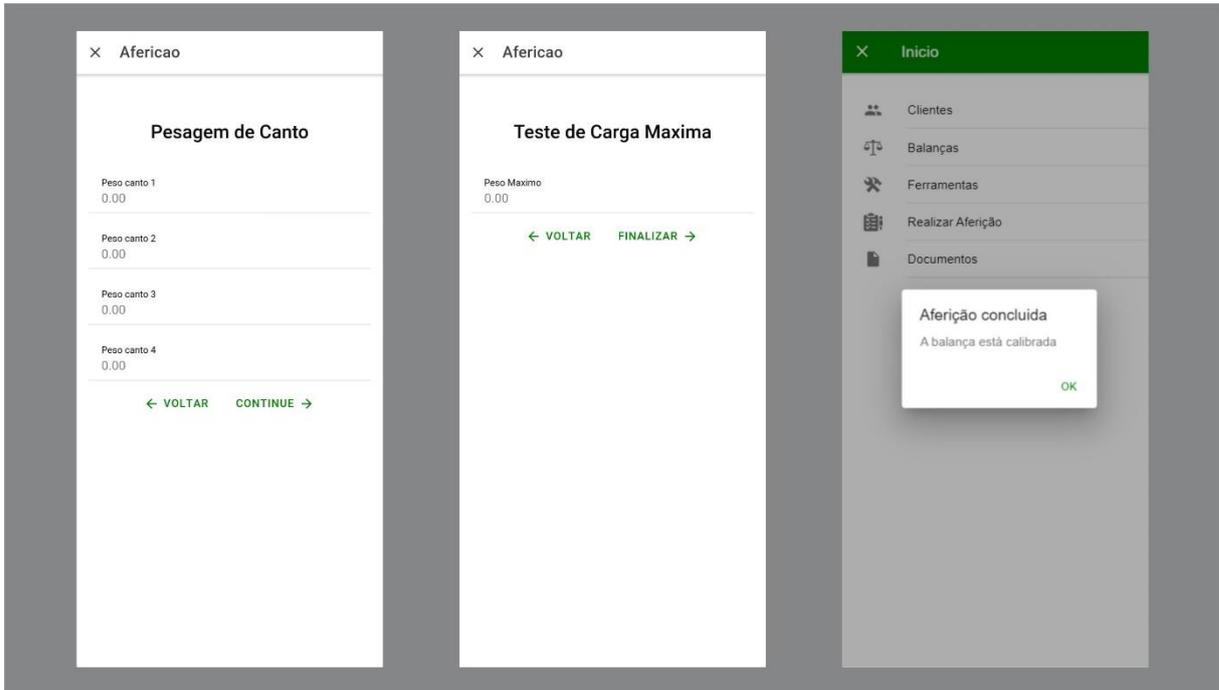
The image displays two sequential screens from a mobile application titled 'Afericao'. The first screen, 'Dados para Aferição', contains input fields for 'Cliente' (with a search icon), 'Balança' (with a dropdown arrow), 'Ferramentas' (with a dropdown arrow), 'Temperatura' (0.00), and 'Umidade do ar' (0.00). A green 'CONTINUE →' button is at the bottom. The second screen, 'Pesagem de Centro', shows 'Peso obtido' as 0.00 and features two green buttons: '← VOLTAR' and 'CONTINUE →'.

Fonte: Próprio Autor

### 5.2.6 Relatório

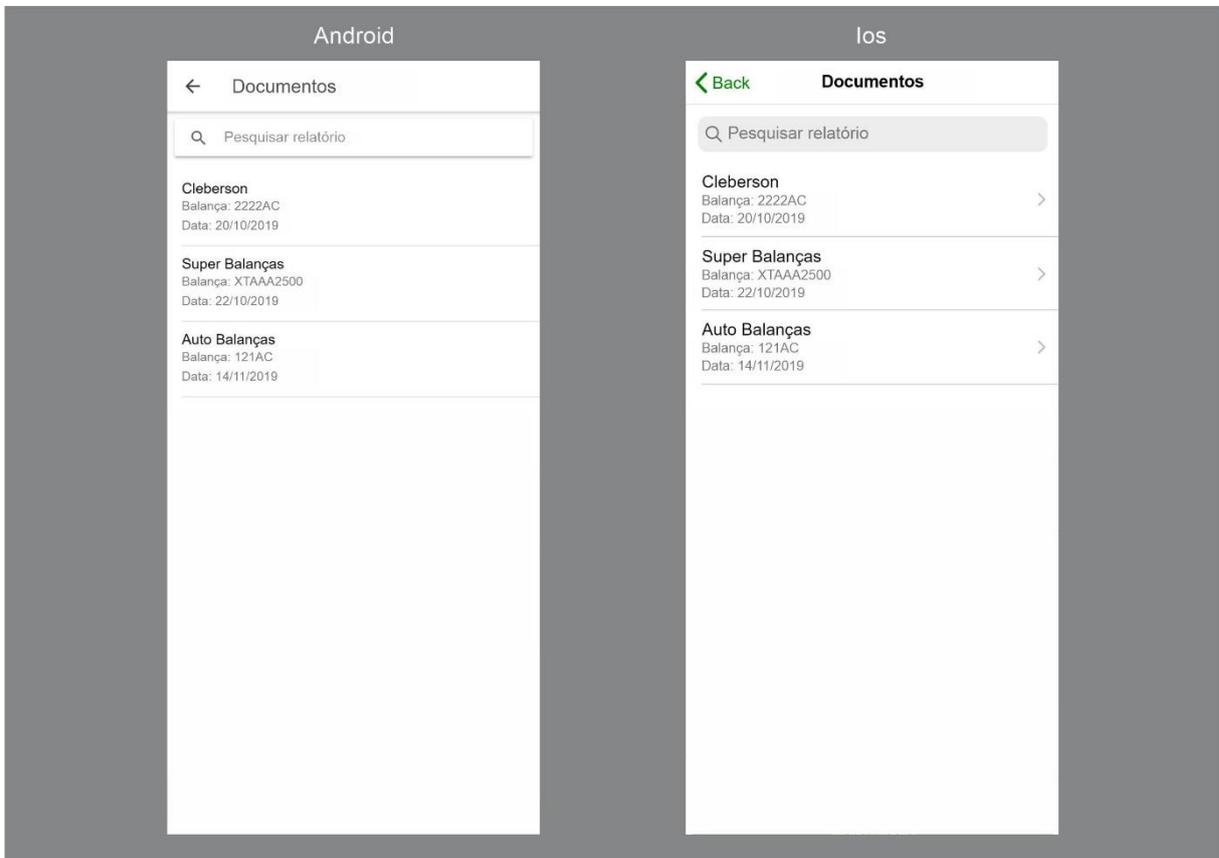
Ao finalizar a aferição conforme mostrado na Figura 23, é possível gerar um documento a partir da tela “Documentos” que comprovará a aferição da balança, acessível pela tela de navegação, conforme mostra a Figura 25.

Figura 19 - Formulário de aferição 2



Fonte: Próprio Autor

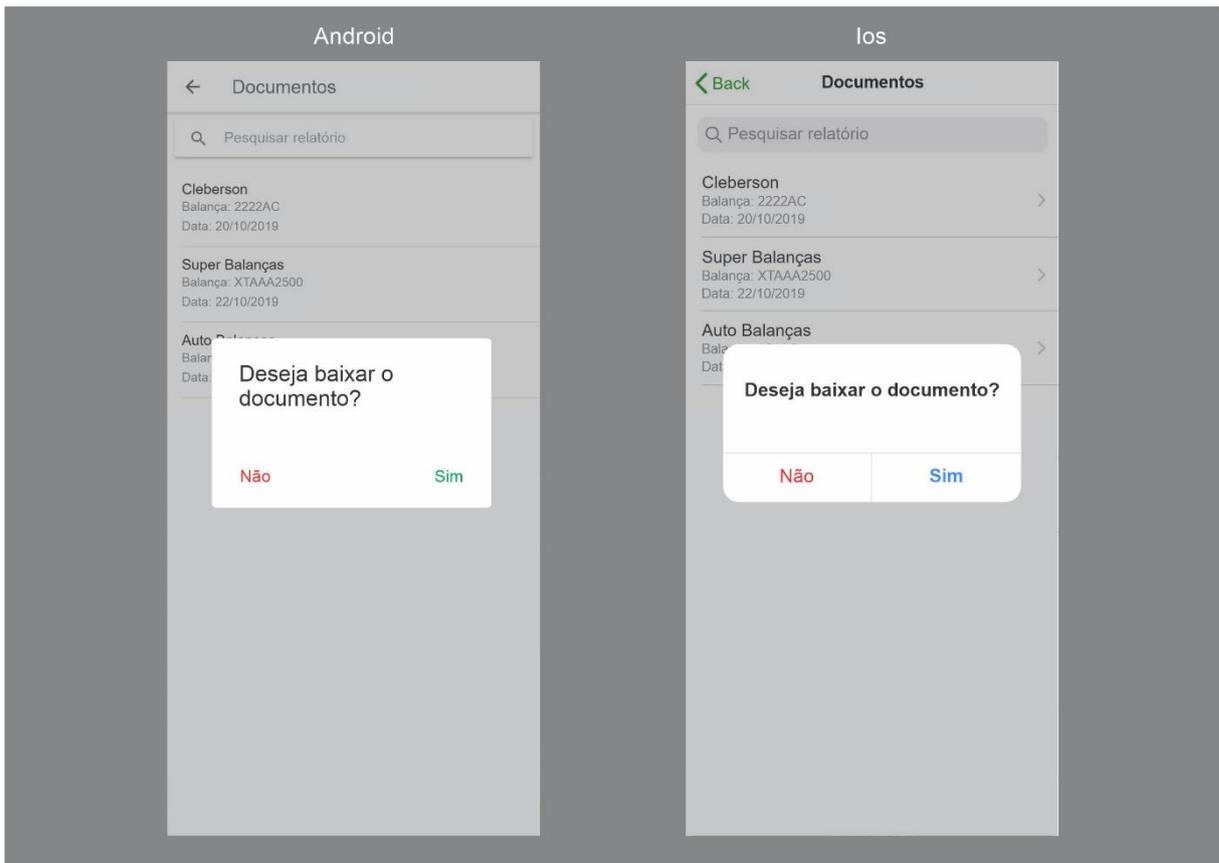
Figura 20 - Tela de documentos



Fonte: Próprio Autor

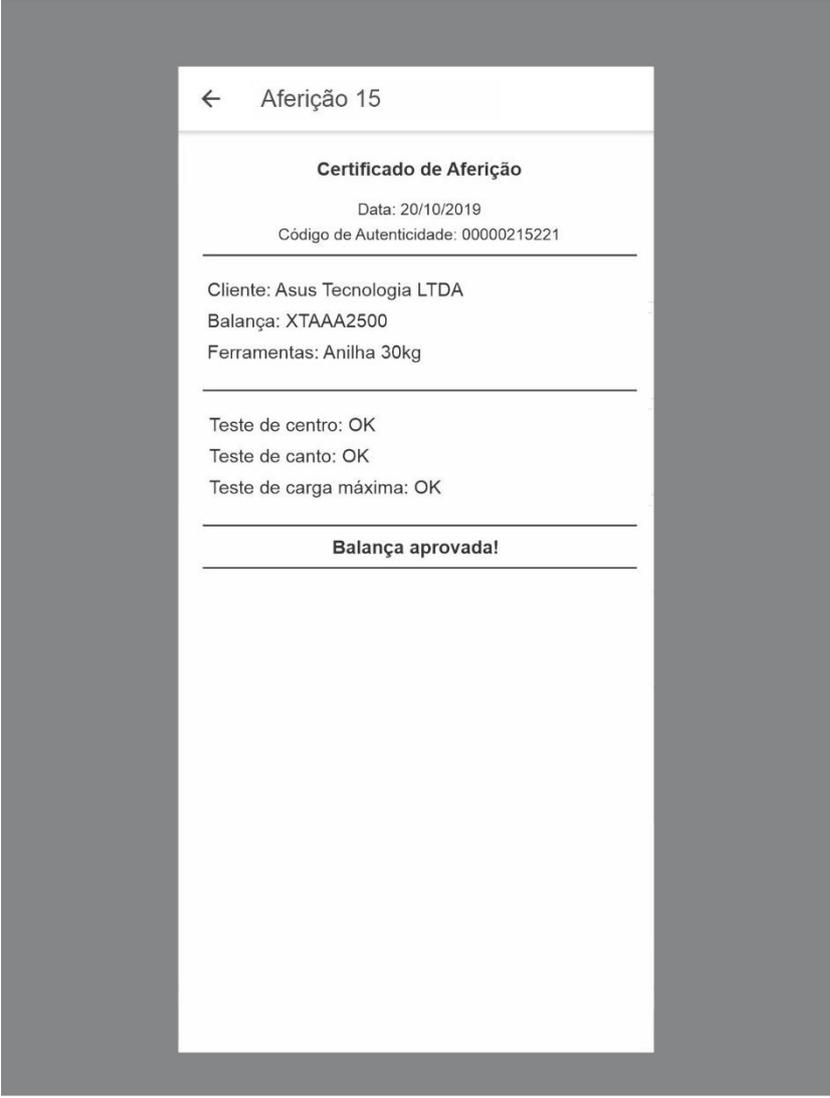
Ao clicar sobre uma das aferições listadas, irá surgir uma mensagem na tela (Figura 25), perguntando se o usuário deseja realizar o download do documento certificador da aferição em formato PDF. Caso o usuário selecione “Sim”, o arquivo da Figura 26 será baixado.

Figura 21 - Confirmação para baixar



Fonte: Próprio Autor

Figura 22 - Certificado da aferição



← Aferição 15

**Certificado de Aferição**

Data: 20/10/2019  
Código de Autenticidade: 00000215221

---

Cliente: Asus Tecnologia LTDA  
Balança: XTAAA2500  
Ferramentas: Anilha 30kg

---

Teste de centro: OK  
Teste de canto: OK  
Teste de carga máxima: OK

---

**Balança aprovada!**

---

Fonte: Próprio Autor

## 6 CONCLUSÃO

Na fundamentação teórica deste trabalho foram descritos os conceitos por trás dos sistemas operacionais para dispositivos móveis, o desenvolvimento de aplicações utilizando a abordagem multiplataforma e seu desenvolvimento. Também foi mostrado o processo de aferição e sua importância para o mercado, assim como os órgãos que regulamentam esse setor. Ao longo do desenvolvimento do trabalho, os métodos e procedimentos utilizados para o desenvolvimento da aplicação multiplataforma que utiliza o framework Ionic 4 e o Firebase foram explicitados e demonstrados, bem como a comunicação entre os serviços. Foram também demonstradas as funcionalidades do sistema de forma detalhada.

A utilização de tecnologias como o Firebase e Ionic acelerou o desenvolvimento, pois conta com muitas ferramentas e componentes previamente prontos, como a autenticação via e-mail do Firebase e os componentes de visualização do Ionic. Desta maneira, o aplicativo pode ter um visual bonito e padronizado para as plataformas web e mobile.

O sistema atende os objetivos propostos, dessa maneira possibilitando a digitalização das informações que antes eram apenas em papel e juntamente com isso, possibilitado uma maior segurança dos dados, pois agora eles estão no meio digital. Por não ter ocorrido a implantação, o sistema poderá receber novas funcionalidades e ajustes conforme ocorrer sua utilização, conforme a necessidade da empresa que irá utilizá-lo como ferramenta.

## 7 REFERÊNCIAS

- ADRIANO, Thiago S. **Introdução ao Firebase**. Disponível em: <<https://medium.com/@programadriano/introdu%C3%A7%C3%A3o-ao-firebase-bd59bfd03f29/>>.\_Acesso em: 29 mai. 2019
- ANGULARFIRE. Disponível em: <<https://github.com/angular/angularfire>> Acesso em: 14 de nov. 2019.
- BARROS, J. D. L. D. **Aplicação móvel Android de apoio a gestão de empréstimos e reservas das bibliotecas da Universidade do Minho**. Braga, Portugal: Universidade do Minho. 2015.
- BRASIL. Decreto n. 1.521, de 26 de dez. de 1951. **Serão punidos, na forma desta Lei, os crimes e as contravenções contra a economia popular**. dez, 2019.
- CORRAL, L.; SILLITTI, A.; SUCCI, G. **Mobile Multiplatform Development: An Experiment for Performance Analysis**. Procedia Computer Science, v. 10, p. 736-743, 2012. ISSN ISSN: 1877-0509. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050912004516>>. ANT\ 2012 and MobiWIS 2012
- FIREBASE. Disponível em: <<https://firebase.google.com/>> Acesso em: 14 de novembro. 2019.
- GUEDES, T. **Crie aplicações com angular o novo framework do google**. [S.l.]: Casa do código, 2017.
- GUEDES, G. T. **UML: uma abordagem prática**. [S.l.]: Novatec Editora, 2008.
- INMETRO. **Atualizar a legislação relativa aos instrumentos de pesagem, para proteção do consumidor, para facilidade de uso e exatidão das medições de massa, para prevenção contra a fraude e influências a que esses instrumentos estão sujeitos**. Portaria 236 de 22 de dezembro de 1994, 2019.
- IONIC FRAMEWORK. Disponível em: <<https://ionicframework.com/docs/intro>> Acesso em: 20 mai. 2019.
- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The Unified Software Development Process**. [S.l.]: Addison-Wesley Professional, 1999. 512 p.
- LARMAN, C. **Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development**. 2th. ed. [S.l.]: Prentice Hall, 2002. 627 p.
- LECHETA, Ricardo R. **Google Andoid: aprenda a criar aplicações para dispositivos móveis com Android SDK**: 3. ed. São Paulo: Novatec Editora, 2013.
- MAZZA, L. **HTML5 e CSS domine a web do futuro**. [S.l.]: Casa do Código, 2012.

SILBERSCHATZ, A. et al. **Operating system concepts**. [S.l.]: Addison-Wesley Reading, v. 4, 1998.

SOMMERVILLE, I. **Engenharia de Software**. 9th. ed. São Paulo: Pearson Prentice

SOUZA, Hiago Luan Flavio de. **Sistema de controle de conformidade de higienização para laticínios utilizando o framework ionic de desenvolvimento de aplicações móveis multiplataforma**. 2017. Trabalho de Conclusão de Curso (Graduação) – Universidade Luterana do Brasil, Ji-Paraná, 2017.

STELLMAN, A.; GREENE, J. **Applied Software Project Management**. 1. ed. Sebastopol: O'Reilly Media, Inc, 2005. 318 p.

VISUAL STUDIO CODE. Disponível em: <<https://code.visualstudio.com/docs>> Acesso em: 20 mai. 2019.

WASSERMAN, A. I. **Software engineering issues for mobile application development**. In **Proceedings of the FSE/SDP workshop on Future of software engineering research**, FoSER '10, pages 397–400, New York, NY, USA. ACM, 2010.

WAZLAWICK, R. **Análise e projeto de sistemas de informação orientados a objetos**. 2.ed. Rio de Janeiro: Elsevier, 2011.

WIKIMEDIA. Disponível em: <[https://upload.wikimedia.org/wikipedia/commons/f/fd/Operating\\_system\\_placement-pt.svg](https://upload.wikimedia.org/wikipedia/commons/f/fd/Operating_system_placement-pt.svg)> Acesso em: 29 de mai. 2019.