



**UFOP**

Universidade Federal  
de Ouro Preto

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

**Um estudo sobre abordagens dirigidas  
por modelos para o desenvolvimento  
de aplicações multiplataforma para  
dispositivos móveis**

**Felipe Dias Mendes**

**TRABALHO DE  
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:

Euler Horta Marinho

COORIENTAÇÃO:

George Henrique Godim da Fonseca

**Abril, 2021**

**João Monlevade–MG**

**Felipe Dias Mendes**

**Um estudo sobre abordagens dirigidas por  
modelos para o desenvolvimento de aplicações  
multiplataforma para dispositivos móveis**

Orientador: Euler Horta Marinho

Coorientador: George Henrique Godim da Fonseca

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

**Abril de 2021**

## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

M538e Mendes, Felipe Dias .

Um estudo sobre abordagens dirigidas por modelos para o desenvolvimento de aplicações multiplataforma para dispositivos móveis. [manuscrito] / Felipe Dias Mendes. - 2021.

59 f.: il.: color., tab..

Orientador: Prof. Me. Euler Horta Marinho.

Coorientador: Prof. Dr. George Henrique Godim Fonseca.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Sistemas de Informação .

1. Aplicativos móveis. 2. Engenharia de software. 3. Software de aplicação - Desenvolvimento. I. Fonseca, George Henrique Godim. II. Marinho, Euler Horta. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 004.41

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



## FOLHA DE APROVAÇÃO

**Felipe Dias Mendes**

Um estudo sobre abordagens dirigidas por modelos para o desenvolvimento de aplicações multiplataforma para dispositivos móveis

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 20 de abril de 2021

### Membros da banca

Mestre - Euler Horta Marinho - Orientador (Universidade Federal de Ouro Preto)  
Doutor - George Henrique Godim da Fonseca - Coorientador (Universidade Federal de Ouro Preto)  
Doutor - Diego Zuquim Guimarães Garcia - (Universidade Federal de Ouro Preto)  
Mestra - Daniela Rodrigues Dias - (Doutoranda em Educação - Universidade Federal de Ouro Preto)

Euler Horta Marinho, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 28/04/2021



Documento assinado eletronicamente por **Euler Horta Marinho, PROFESSOR DE MAGISTERIO SUPERIOR**, em 28/04/2021, às 10:16, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0164487** e o código CRC **7F1235D1**.

*Este trabalho é dedicado especialmente a minha família, amigos e todos envolvidos pelo incentivo e pelo apoio constante.*

# Agradecimentos

Agradeço primeiramente a Deus, por sempre guiar os meus caminhos, dar-me saúde e coragem, aos meus pais Adilson e Vanilza, e minha irmã Camila, que deram-me todo incentivo e apoio necessários para chegar até aqui e conquistar meus sonhos. Agradeço aos amigos que conheci durante a graduação pelo apoio na vida acadêmica e aos meus amigos pessoais.

Agradeço a minha mãe Vanilza, proprietária do Studio Vanilza Mendes, um estúdio de beleza em João Monlevade, pela confiança e pela contribuição significativa para o desenvolvimento deste trabalho.

Aos meus orientadores, Euler e George, por todo o conhecimento ministrado em sala de aula, que foi fundamental para a concretização deste trabalho, e por terem me acompanhado nesta etapa de grande aprendizado e crescimento pessoal. A todos vocês, muito obrigado.

“Agir, eis a inteligência verdadeira. Serei o que quiser. Mas tenho que querer o que for. O êxito está em ter êxito, e não em ter condições de êxito.”

— Fernando Pessoa.

# Resumo

O avanço tecnológico observado nos últimos anos possibilitou novas abordagens de desenvolvimento de aplicações móveis. Assim as organizações têm objetivos de desenvolver com menos custo e em menor tempo. Com os dispositivos móveis, há o maior alcance para um sistema de informação, logo um dos mercados mais rentáveis para *softwares*. Este trabalho propõe um estudo sobre *Model-Driven Development* (MDD) em aplicações para dispositivos móveis. A aplicação multiplataforma foi desenvolvida no ambiente WebRatio 8.10.4 *Mobile Community*. Nesta plataforma utiliza-se *Interaction Flow Modeling Language* (IFML), com objetivo de representar interações entre usuários com a aplicação. Como estudo de caso, foi desenvolvida uma aplicação para o gerenciamento de um estúdio de beleza, Studio Vanilza Mendes. A aplicação foi validada através de testes automatizados no *Firebase Test Lab* e proprietária do estúdio.

**Palavras-chaves:** Aplicação móvel. Desenvolvimento de software. IFML.

# Abstract

The technological advances observed in recent years have enabled new approaches to the development of mobile applications. Organizations aim to develop at less cost and in a shorter period of time. With mobile devices, there is the greatest reach for an information system, therefore one of the most profitable markets for software. This work proposes a study on model-driven development (MDD) in applications for mobile devices. The multiplatform application developed in the WebRatio 8.10.4 Mobile Community environment. In this platform, Interaction flow Modeling language (IFML) is used, in order to represent interactions between users with the application. As a case study, an application was developed for the management of a beauty studio, Studio Vanilza Mendes. Validated by automated tests in Firebase Test Lab and owner of the studio.

**Key-words:** Mobile application. Software development. IFML.

# Lista de ilustrações

Figura 1 – View Components . . . . .	23
Figura 2 – Action . . . . .	23
Figura 3 – Exemplo . . . . .	23
Figura 4 – Exemplo de DataFlow . . . . .	24
Figura 5 – Utility Components . . . . .	24
Figura 6 – Operation . . . . .	25
Figura 7 – Pocket Agenda . . . . .	25
Figura 8 – Salon Soft . . . . .	26
Figura 9 – Corthy . . . . .	26
Figura 10 – Protótipo da Tela Home . . . . .	34
Figura 11 – Protótipo da Tela Agenda . . . . .	35
Figura 12 – Protótipo da Tela criar agendamento . . . . .	35
Figura 13 – Modelo Entidade Relacionamento . . . . .	36
Figura 14 – Modelo de domínio . . . . .	37
Figura 15 – Diagrama IFML para <i>login</i> . . . . .	38
Figura 16 – Diagrama IFML para gestão do cliente . . . . .	39
Figura 17 – <i>Action</i> de salvar cliente . . . . .	40
Figura 18 – <i>Action</i> de <i>delete</i> cliente . . . . .	41
Figura 19 – Lista de clientes . . . . .	42
Figura 20 – Cadastro de cliente . . . . .	42
Figura 21 – Tela Home . . . . .	43
Figura 22 – Visão geral de teste automatizado . . . . .	44
Figura 23 – Sugestão de acessibilidade . . . . .	44
Figura 24 – Diagrama IFML para gestão de comanda . . . . .	51
Figura 25 – Diagrama IFML para gestão de despesas . . . . .	52
Figura 26 – Diagrama IFML agenda . . . . .	52
Figura 27 – Diagrama IFML tela Home . . . . .	53
Figura 28 – Tela cliente formulário . . . . .	54
Figura 29 – Tela lista de despesas . . . . .	54
Figura 30 – Tela detalhes despesa . . . . .	55
Figura 31 – Tela despesa formulário . . . . .	55

# Lista de tabelas

Tabela 1 – Visão das plataformas móveis . . . . .	20
Tabela 2 – Visão dos planos da aplicação Salon Soft . . . . .	26
Tabela 3 – Requisitos levantados . . . . .	29
Tabela 4 – Funcionalidades abrangidas . . . . .	45

# Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
BPMN	<i>Business Process Model and Notation</i>
CSS	<i>Cascading Style Sheets</i>
DECSI	Departamento de Computação e Sistemas
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IFML	<i>Interaction Flow Modeling Language</i>
MDD	<i>Model-Driven Development</i>
MDE	<i>Model-Driven Engineering</i>
MVC	<i>Model-View-Controller</i>
OMG	<i>ObjectManagement Group</i>
QR Code	<i>Quick Response Code</i>
SDK	<i>Software Development kit</i>
SQL	<i>Structured Query Language</i>
UFOP	Universidade Federal de Ouro Preto
UI	User Interface
UML	<i>Unified Modeling Language</i>
XP	<i>Extreme Programming</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Justificativa</b>	<b>15</b>
<b>1.2</b>	<b>Objetivos</b>	<b>16</b>
1.2.1	Objetivos Gerais	16
1.2.2	Objetivos Específicos	16
<b>1.3</b>	<b>Metodologia</b>	<b>17</b>
<b>1.4</b>	<b>Organização do trabalho</b>	<b>17</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>18</b>
<b>2.1</b>	<b>Processo de desenvolvimento de software</b>	<b>18</b>
<b>2.2</b>	<b>Desenvolvimento de Aplicações Móveis</b>	<b>19</b>
2.2.1	Desenvolvimento Nativo	20
2.2.2	Desenvolvimento Híbrido	20
2.2.3	Desenvolvimento Web	20
<b>2.3</b>	<b>IFML</b>	<b>20</b>
2.3.1	Extensão da IFML para Aplicações Móveis	21
2.3.1.1	Processo de modelagem	22
2.3.1.2	Componentes de modelagem	22
2.3.2	Aplicações semelhantes	25
2.3.2.1	Pocket Agenda	25
2.3.2.2	Salon Soft	26
2.3.2.3	Corthy	26
2.3.3	Considerações finais de revisão	27
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>28</b>
<b>3.1</b>	<b>Tecnologias utilizadas</b>	<b>28</b>
3.1.1	PostgreSQL	28
3.1.2	WebRatio	28
<b>3.2</b>	<b>Processo de desenvolvimento</b>	<b>28</b>
3.2.1	Levantamento de requisitos	29
3.2.2	Histórias de usuário	30
3.2.3	Protótipos	33
3.2.4	Modelagem do <i>back-end</i>	36
3.2.4.1	Modelo de Banco de Dados	36
3.2.4.2	Modelo de domínio	36
3.2.5	Modelagem do <i>front-end</i>	37

3.2.5.1	Modelagem do <i>Controller</i> . . . . .	39
<b>4</b>	<b>RESULTADOS</b> . . . . .	<b>42</b>
<b>4.1</b>	<b>Apresentação do sistema</b> . . . . .	<b>42</b>
<b>4.2</b>	<b>Validação da Aplicação</b> . . . . .	<b>43</b>
4.2.1	Firebase Test Lab . . . . .	44
4.2.2	Validação da proprietária . . . . .	45
4.2.3	Comparação . . . . .	45
<b>4.3</b>	<b>Dificuldades</b> . . . . .	<b>45</b>
<b>5</b>	<b>CONCLUSÃO</b> . . . . .	<b>47</b>
<b>5.1</b>	<b>Trabalhos futuros</b> . . . . .	<b>47</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>49</b>
	<b>APÊNDICES</b> . . . . .	<b>50</b>
	<b>APÊNDICE A – MODELOS IFML</b> . . . . .	<b>51</b>
	<b>APÊNDICE B – TELAS</b> . . . . .	<b>54</b>
	<b>APÊNDICE C – QUESTIONÁRIO DE AVALIAÇÃO DO PROJECTS- TUDIO</b> . . . . .	<b>56</b>

# 1 Introdução

A cada ano que se passa, *hardware* e software sofrem intensas modificações e inovações, o que leva ao surgimento de novos sistemas operacionais e linguagens de programação de alto nível, junto a novas funcionalidades, que vem permitindo que aplicações sofisticadas e complexas sejam desenvolvidas. O desenvolvimento de aplicações móveis é algo bastante enigmático, e as organizações, de forma geral, apresentam uma alta demanda por esses softwares.

Diante de um ambiente móvel complexo com vários ecossistemas e contextos, pesquisadores e organizações, através de seus estudos, buscam encontrar ferramentas e métodos que auxiliam na produção de software de forma mais rápida, prática e simples. Desse modo, o *Model-Driven Development* (MDD) tem o potencial para acelerar o desenvolvimento de aplicações móveis. No domínio das aplicações móveis, existem duas grandes plataformas com participação significativa de mercado, as quais são alvos da maioria das aplicações feitas. Cada plataforma possui sistemas operacionais distintos e linguagens de programação, sendo o Android com a Linguagem Java e C++ e o iOS com Objective-C e Swift. Entretanto, no momento em que os desenvolvedores são alocados para fazer duas versões diferentes do mesmo software, implica em um maior prazo para a entrega do projeto e maior investimento financeiro. Sendo assim, novos métodos estão surgindo de modo a contornar esse problema, onde são utilizados principalmente *frameworks* para o desenvolvimento multiplataforma. Podemos ver em [Rieger e Majchrzak \(2019\)](#), que quando falamos de uma abordagem de desenvolvimento multiplataforma, não consideramos a implementação concreta do software, mas antes, o desafio de resolver de uma maneira geral o desenvolvimento de uma aplicação funcional, que execute em várias plataformas. Em [Raj e Tolety \(2012\)](#), classificam algumas possibilidades de desenvolvimento de aplicações multiplataforma: abordagem *Interpreted*, abordagem *Hybrid*, abordagem *Web* ou abordagem *Cross-Compiled*.

Na Engenharia de Software, modelar é extremamente importante. De acordo com [Sommerville \(2011\)](#), um modelo é uma abstração de um sistema ignorando alguns detalhes do mesmo, mas por outro lado, podem ser desenvolvidos para mostrar as interações, a estrutura, o contexto e o comportamento do sistema. Visto que modelar software é uma atividade no processo de desenvolvimento de grande importância, logo os modelos gerados, que serviam apenas para expressar características específicas de um software, poderiam contribuir ativamente na construção do mesmo, gerando códigos funcionais através desses modelos. O MDD é uma abordagem que consiste na geração de aplicações utilizando diretamente os modelos possivelmente criados durante o processo de desenvolvimento.

O MDD veio de forma acoplada para criarmos apenas uma versão da aplicação. Enquanto modela a estrutura de dados, processos, funcionalidades e *layouts*, aplicações executáveis podem ser gerados a partir desses modelos. O MDD não se limita a detalhes técnicos e início de projeto, portanto se estende em todo processo de desenvolvimento de software, assim elevando o nível de abstração, conseqüentemente diminuindo a complexidade. Segundo [Whittle, Hutchinson e Rouncefield \(2013\)](#), “a modelagem de software é, sem dúvida, uma atividade central no desenvolvimento de software”. De acordo com [Vaupel et al. \(2018\)](#), o MDD tem alto potencial de acelerar o processo de desenvolvimento de aplicações nativas e fornece uma adaptabilidade mais fácil a características específicas de determinadas plataformas.

Este trabalho consiste no estudo de uma metodologia de MDD utilizando como estudo de caso o desenvolvimento de uma aplicação que visa dar suporte ao gerenciamento do Studio Vanilza Mendes. A aplicação foi desenvolvida para dispositivos móveis, utilizando a *Interaction Flow Modeling Language* (IFML), a qual foi adotada como padrão pela *Object Management Group* (OMG) em março de 2013, através do auxílio da Ferramenta WebRatio Mobile, a qual permite que a aplicação seja gerada de forma compatível com diversos dispositivos móveis. As principais funcionalidades da aplicação envolvem controle da agenda do estúdio, fluxo de caixa e fidelização dos clientes. Durante o desenvolvimento do projeto, será analisado como o MDD apoiará todo o processo de desenvolvimento da aplicação, além da apresentação das etapas da criação da aplicação, envolvendo conceitos de Engenharia de Software, aplicações para dispositivos móveis e testes manuais e automatizados.

## 1.1 Justificativa

Projetar, desenvolver e reutilizar software orientado a objetos é um desafio. Com a finalidade de projetar um software, busca-se minimizar refazer módulos ou componentes do zero. Portanto, um projetista de software, geralmente, reutiliza implementações existentes e adequam ao escopo da nova aplicação. Reúso de software traz ao projeto redução de custos, desenvolvimento mais rápido e diminuição de riscos, além de maior confiabilidade no sistema ([SOMMERVILLE, 2011](#)).

A fim de minimizar a dificuldade de desenvolvimento de software, foram investigadas as metodologias de MDD e *Model-Driven Engineering* (MDE). O MDD utiliza modelos na geração de aplicações executáveis com alta abstração. Já o MDE inclui outros usos de modelos para apoiar o processo de desenvolvimento, como em engenharia reversa e modelos dirigidos à evolução do software.

A abordagem dirigida por modelos utilizada se baseia no uso da Linguagem IFML. A extensão da linguagem para plataformas móveis, permite acesso simples aos recursos nativos das plataformas e a comunicação da aplicação criada com outras aplicações. Um

exemplo é marcar um evento no calendário padrão do dispositivo através da aplicação criada. Logo, não precisamos ficar preocupados com os detalhes de como essa comunicação é realizada.

No estudo de caso, iremos apoiar a gestão do Studio Vanilza Mendes, que necessita de um suporte tecnológico, a fim de obter uma melhor gestão do negócio, visto que seus processos são manuais, dificultando a agilidade em perceber algumas informações estratégicas. Pode-se ver que a falta de uma boa administração no empreendimento, pode resultar em sérios problemas, inclusive causar falência do negócio.

Segundo Chiavenato (2007), o sucesso de uma organização não apenas sorte, e sim de uma série de decisões, união de recursos, estratégias, competências e a busca permanente de objetivos para alcançar resultados cada vez melhores. Dessa forma, este projeto busca melhorar os resultados do estúdio com a aplicação a criada, ou seja, algo que irá ser utilizado em um caso real.

## 1.2 Objetivos

O presente trabalho consiste no desenvolvimento de um Sistema de Informação móvel que visa facilitar a tarefa de gestão financeira, controle de agendamentos, os quais são atendidos por funcionários do estúdio e fidelização de clientes. O sistema será desenvolvido através da plataforma WebRatio, permitindo assim que seja acessado em qualquer dispositivo móvel compatível, oferecendo mobilidade e praticidade.

Durante o desenvolvimento do projeto serão analisadas as ferramentas usadas de forma a compreender como a abordagem MDD e MDE com o apoio da linguagem *Interaction Flow Modeling Language* IFML, auxilia no desenvolvimento de aplicações multiplataforma para dispositivos móveis. Também será verificado o impacto da abordagem em incrementos de novos recursos, manutenibilidade e reuso.

### 1.2.1 Objetivos Gerais

Este trabalho tem como objetivo um estudo sobre abordagens dirigidas por modelos para o desenvolvimento de uma aplicação multiplataforma para dispositivos móveis.

### 1.2.2 Objetivos Específicos

Esse trabalho tem como objetivos específicos:

- Desenvolver um Sistema de Informação para o gerenciamento do Studio Vanilza Mendes.
- Realizar um breve estudo sobre o funcionamento e a aplicação da linguagem IFML;

- Identificar e especificar os requisitos da aplicação;
- Realizar a modelagem do *front-end* do sistema utilizando a linguagem IFML;
- Desenvolver o sistema utilizando a IDE WebRatio Mobile;
- Realizar testes manuais e testes automatizados;
- Validar a aplicação junto à proprietária do estúdio de beleza;

### 1.3 Metodologia

O presente trabalho adota uma metodologia apoiada nos conceitos do desenvolvimento de software com o principal foco na abordagem MDD. O software desenvolvido com diversos incrementos alinhado ao cliente e com funcionalidades com atributos de prioridade, com o intuito de direcionar os esforços nos pontos mais críticos do sistema.

### 1.4 Organização do trabalho

O restante deste trabalho é organizado como se segue. O Capítulo 2 apresenta toda revisão bibliográfica e estudo feito antes de iniciar o desenvolvimento do projeto. O Capítulo 3 descreve todo o processo de desenvolvimento do sistema, desde levantamento de requisitos, modelagem do software, desenvolvimento, validação e apresentação da aplicação. No Capítulo 4 são apresentados as considerações finais sobre o projeto e propostas para trabalhos futuros.

## 2 Revisão bibliográfica

Este capítulo apresenta uma revisão bibliográfica, bem como trabalhos correlatos e aplicações semelhantes.

### 2.1 Processo de desenvolvimento de software

O desenvolvimento de software tem o objetivo principal de resolver problemas da vida real. No mundo contemporâneo há vários tipos de problemas a serem resolvidos, logo existem vários tipos de softwares, desde simples softwares até sistemas de informações mais complexos. Junto a evolução de *hardware*, desenvolvimento de microarquiteturas mais eficientes, aparecimento e evolução dos sistemas operacionais e as linguagens de programação de alto nível de abstração. Observa-se que no decorrer dos anos os softwares vem recebendo novas possibilidades de escopo.

O conceito de sistema, é definido por um grupo de componentes inter-relacionados que trabalham juntos com o mesmo objetivo. Um sistema, possui três funções básicas principais. Começando pela entrada, onde captura e reúne os elementos que entram no sistema. Seguido do processamento, sendo responsável por transformar as entradas em produto. Por fim a saída, que apresenta aos usuários o produto que foi trabalhado no processamento, de modo que seja fácil entender um resultado esperado ou um comportamento esperado. Os sistemas mais robustos com diversos módulos e funcionalidades, chamamos de sistemas de informação. Segundo [Laudon e Laudon \(2004\)](#), um sistema de informação pode ser definido como um conjunto de componentes ou subsistemas interconectados, que recupera e ou coleta, processa, armazena e distribui informações entre os módulos para apoiar a tomada de decisão, à coordenação e ao controle de uma organização. Analisando a complexidade de um sistema grande como o de informação, mais o impacto que incide sobre a organizações e vantagens competitivas, devemos efetuar um planejamento e uma boa organização no desenvolvimento. Devido ao número de áreas relacionadas a um sistema de informação, naturalmente é tarefa da área de Engenharia de software se preocupar com esses aspectos.

No início do projeto de software, sem uma metodologia definida no processo de desenvolvimento, pode aumentar o risco de acontecer o que chamamos de "falhas de software", conhecidas também como "bugs". Essas falhas podem causar simples anomalias até problemas graves na aplicação. É evidente que, pode atrapalhar a experiência do usuário. A Engenharia de software aparece a fim de criar técnicas e padrões para resolver problemas muito recorrentes e mútuos nos projetos de criar um software, buscando minimizar a ocorrência das "falhas de software". Segundo [Sommerville \(2011\)](#), a Engenharia de software tem o objetivo central apoiar o desenvolvimento profissional de software, dessa forma

pode-se criar sistemas de informação complexos e estáveis.

Os engenheiros de software se preocupam em desenvolver produtos de software. Existem produtos de software genéricos, que consistem em sistemas com diversas funcionalidades, sendo os engenheiros responsáveis por definir o escopo do projeto e são colocados a venda para vários clientes. Outro tipo de produto são os criados sob encomenda, esses em questão são sistemas criados a pedido de um cliente específico e o engenheiro de software encontra os requisitos do sistema junto ao seu cliente de acordo com suas necessidades, nesta abordagem, no cenário ideal, o cliente deve participar ativamente de todo o processo de desenvolvimento da aplicação.

Na Engenharia de Software, modelar é extremamente importante. De acordo com [Sommerville \(2011\)](#), um modelo é uma abstração de um sistema ignorando alguns detalhes do mesmo, por outro lado, podem ser desenvolvidos para mostrar as interações, a estrutura, o contexto e o comportamento do sistema. Visto que modelar software é uma atividade no processo de desenvolvimento de grande importância, logo perceberam que os modelos gerados, que serviam apenas para expressar características específicas de um software, poderiam contribuir ativamente em sua construção, gerando códigos funcionais e executáveis.

## 2.2 Desenvolvimento de Aplicações Móveis

Para entender como funciona o processo de desenvolvimento de aplicações móveis, primeiramente deve-se conhecer como funciona o ambiente de dispositivos móveis e suas tecnologias. É certo que, a cada ano que passa, a quantidade de pessoas que têm acesso a algum dispositivo móvel cresce aceleradamente, dessa forma, tornando-se plataforma alvo dos desenvolvedores e organizações.

Aplicações móveis são desenvolvidas para diversos fins, desde simples entretenimento, a aplicações comerciais sérias. O desafio de criar softwares móveis é lidarmos com o contexto móvel e alguns recursos que a maioria deles contém. Um exemplo destes aspectos, o *hardware*, que nessa plataforma há limitações em relação ao desempenho bruto, preocupa-se também com a área de tela, na maioria dos casos, é menor que em outros dispositivos, problemas de rede sem fio, largura de banda menor e bateria limitada, influencia os requisitos funcionais e não funcionais da aplicação.

De acordo com [Marinho e Resende \(2015\)](#), o desempenho de aplicações móveis varia consideravelmente dependendo da ferramenta adotada no desenvolvimento. Desse modo, há algumas categorias no desenvolvimento de software para dispositivos móveis. No ambiente móvel temos algumas abordagens de desenvolvimento de aplicações, nativos, baseando em *Web* e híbridos.

### 2.2.1 Desenvolvimento Nativo

As aplicações nativas geralmente são desenvolvidas com um *Software Development kit* (SDK) específico, distribuído por um fornecedor de plataforma. A Tabela 1 tem algumas características das duas plataformas mais relevantes no mercado de dispositivos móveis.

Tabela 1 – Visão das plataformas móveis

Plataforma	Linguagem de Programação	GUI <i>development</i>	IDE
Android	Java, Kotlin	Android UI, <i>Material Design</i>	Eclipse, Android Studio
iOS	Objective-C, Swift	Cocoa Touch	Xcode

Fonte: (MARINHO; RESENDE, 2015)

As aplicações nativas podem ter benefícios sob outras abordagens, pelo fato de serem programadas na linguagem nativa e compiladas nativamente. Em relação a usabilidade da aplicação, apresentam uma *interface* mais consistente e com aspectos mais próximos da plataforma em si. Isso significa que o usuário terá uma experiência intuitiva, já que está acostumado com os padrões adotados da plataforma que está usando, mesmo que seja um novo software.

### 2.2.2 Desenvolvimento Híbrido

As aplicações híbridas são aquelas que combinam tecnologias nativas e da *Web*. Por exemplo, através do *Cordova* e *Phonegap*, serve para gerar uma aplicação nativa capaz de abrir uma *webview* executando o HTML/CSS e *JavaScript* e exibir ao usuário.

### 2.2.3 Desenvolvimento Web

As aplicações *Web* têm aparência específica que pode conter estruturas e tecnologias de UI (*User Interface*), tais como JQuery, CSS e JavaScript. Por outro lado, a abordagem *Web* pode apresentar problemas de inconsistência de aparência ou funcionalidades que não se comportam como esperado.

## 2.3 IFML

A *Interaction Flow Modeling Language* (IFML) é uma linguagem de modelagem proposta para expressar as interações do usuário com o software. Além disso, veio com novas possibilidades de expressar mais interações do que a WebML, visto que, os autores da IFML tinham extensa experiência de 10 anos na WebML junto a empresa WebRatio.

Logo sabiam quais eram suas limitações e como resolvê-las. Nesse sentido, a IFML se tornou uma linguagem de modelagem padrão da *Object management Group* (OMG) em Março de 2013.

Uma vez que a IFML é independente de plataforma, os projetistas de software podem especificar a *interface* do usuário e as opções de interação, independentemente da tecnologia de implementação e a plataforma de implementação escolhida, ou seja, pode gerar diversos sistemas executáveis a partir de um único núcleo de desenvolvimento e projeto. Os diagramas IFML permitem gerar aplicações móveis, aplicações *Web* tradicionais, sistemas cliente-servidor, aplicações para *desktop*, software multicanal e sensível ao contexto.

O padrão IFML complementa outras linguagens de modelagem OMG, contendo *Business Process Model and Notation* (BPMN) e *Unified Modeling Language* (UML). Por exemplo, pode-se projetar as *interfaces* de usuário e importar para especificações de processo de negócio no padrão BPMN. Segundo [White e Miers \(2008\)](#), o BPMN é a combinação de dois fluxos de trabalho. Um com foco em gerenciamento de fluxo de trabalho e planejamento, já o outro com foco em modelagem e arquitetura. É um padrão de modelagem de gerenciamento de processos de negócio, que visa facilitar o entendimento do usuário.

### 2.3.1 Extensão da IFML para Aplicações Móveis

As aplicações móveis têm *interfaces* ricas, que juntamente com os requisitos de mobilidade e vários sensores, como câmeras e GPS, disponibilizam recursos que são melhor utilizados por extensões do núcleo IFML, conforme o padrão estabelecido em [Franzago, Muccini e Malavolta \(2014\)](#). Segundo [Acerbis et al. \(2015\)](#), apresenta três categorias principais de extensões:

- **Extensões de Contexto** cobrir dimensões de contexto (permitindo a seleção de um contexto ou de outro) e variáveis de contexto (permitindo o uso de seu valor como parâmetros dentro dos modelos IFML).
- **Extensões de *Container*** incluem a especificação de tipos especiais de *containers* para dispositivos móveis. Normalmente, definimos o conceito de *Screen* como principal na IFML, o *ViewContainer* para aplicações móveis, e o *System Containers* que fornecem uso consistente de recursos do sistema operacional. Exemplos são os sistemas de "Notificação" ou o painel de "Configurações".
- **Extensões de Componentes** incluem o conceito de Componentes de Sistemas, elementos de *interface* padrão fornece funcionalidade básica consistente para o desenvolvimento da aplicação.

### 2.3.1.1 Processo de modelagem

Segundo [Brambilla e Fraternali \(2014\)](#), modelagem de software é um dos processos mais consolidados da tecnologia da informação. O processo de modelagem da IFML visa, além de definições de *interfaces*, relacionar modelo de domínio como algo complementar a esta atividade. O modelo de domínio produzido também auxilia na implementação física de estruturas para armazenamento, atualização e recuperação de dados. Desse modo, a equipe de desenvolvimento poderá visualizar e compreender os conceitos e funcionalidades mais importantes de todo o projeto, assim como o fluxo de dados e relacionamento entre os módulos do sistema.

A motivação de escolha da IFML vem pelo complemento da *Unified Modeling Language* (UML), diferentemente da UML, com a IFML, permite construção de modelos de navegação e interação com o usuário. Além disso, com base na semelhança com a UML, diagrama de classe e com a compatibilidade ao banco de dados de modelo entidade relacionamento, IFML tem o seu lugar e suas características próprias.

O modelo de domínio representa o sistema definido com entidades, associações e relações, de modo conceitual, serve como esqueleto de todo sistema. Já a modelagem IFML tem a capacidade de expressar *interfaces* que inclui componentes funcionais de conteúdo, entradas de dados em formulários, ações entre módulos do sistema, como exibir mensagens e dados, bem como interações entre elementos, operações com o banco de dados e comunicação entre outras aplicações nativas do dispositivo ou funções móveis. Cada elemento ou componente possui uma notação gráfica usada para o desenvolvimento dos modelos de *interface front-end*.

### 2.3.1.2 Componentes de modelagem

A ferramenta da plataforma *WebRatio* utiliza notações gráficas que diferenciam com os padrões originais da linguagem IFML criada originalmente. A Lista abaixo apresenta as conotações gráficas utilizadas na etapa de modelagem de software no *front-end* da aplicação, relacionando o componente e sua função.

- *View Component*

São elementos de *interface* diretamente relacionados com a tela do usuário, utilizado para exibir informações e realizar interações. A Figura 1 exibe *View Components*, que incluem formulários, mensagens, listas, listas encadeadas, detalhes (ex: todos os atributos de um objeto) dentre outros.

Figura 1 – View Components

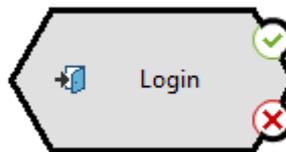


Fonte: Web Ratio.

- *Action*

É uma ação que modifica o estado da aplicação. A Figura 2 representa uma ação de *login*.

Figura 2 – Action

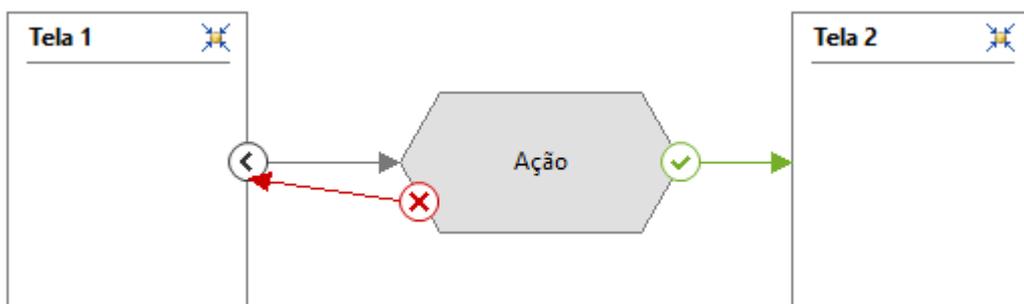


Fonte: Web Ratio.

- *Flows*

Representa fluxos de navegação. Na cor verde *OK Flow*, onde há caminhos positivos de ações. Na cor vermelha *KO Flow*, serve para falhas, o fluxo padrão na cor preta é de uso geral. A Figura 3 mostra um exemplo utilizando os *Flows*.

Figura 3 – Exemplo

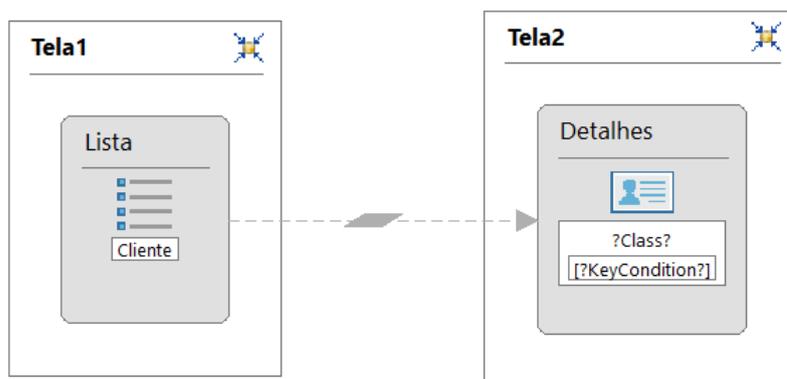


Fonte: Web Ratio.

- *Data Flow*

Representa o fluxo de dados entre *ViewComponents* ou uma consequência de uma interação do usuário. A Figura 4 mostra um exemplo utilizando o *Data Flow*.

Figura 4 – Exemplo de DataFlow



Fonte: Web Ratio.

- *Utility Components*

Representa funcionalidades que acessam recursos nativos dos dispositivos móveis. A Figura 5 retrata calendário, mapas, *QR Code* e pesquisa de itens.

Figura 5 – Utility Components

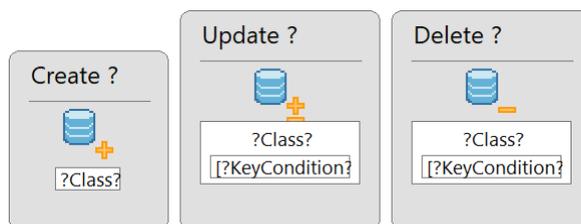


Fonte: Web Ratio.

- *Operation*

Representa operações sobre o modelo relacional de banco de dados (Create, Update e Delete, por exemplo). A Figura 6 apresenta as operações de armazenamento.

Figura 6 – Operation



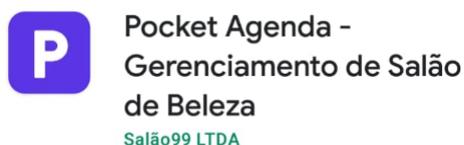
Fonte: Web Ratio.

### 2.3.2 Aplicações semelhantes

Nesta subseção, serão apresentados algumas aplicações semelhantes proposta deste trabalho, porém, analisados apenas a versão contida na Play Store, plataforma do Android.

#### 2.3.2.1 Pocket Agenda

Figura 7 – Pocket Agenda



Fonte: Salão99 LTDA.

O Pocket Agenda - Gerenciamento de Salão é uma aplicação gratuita de gestão para salões de beleza e prestadores de serviço. Contém as funcionalidades de cadastrar clientes, controlar agendamentos, criar comandas, calculo de comissões, cadastrar despesas e visualizar relatórios. Contudo, algumas funcionalidades estão presentes apenas na versão paga da aplicação, logo para desfrutar de todo potencial da ferramenta de gestão devemos pagar um plano mensal, dentre as funções pagas estão o backup em nuvem e os lembretes enviados via WhatsApp ilimitados, os quais servem para lembrar os clientes dos horários agendados.

## 2.3.2.2 Salon Soft

Figura 8 – Salon Soft



Fonte: Essenti softwares.

O Salon Soft Agenda e Sistema para Salão de Beleza é uma aplicação com o mesmo fim da aplicação citada anteriormente. Suas funções segundo os desenvolvedores você poderá cadastrar clientes, serviços e profissionais com facilidade, porém, há também integração com um programa de computador, sendo assim um sistema distribuído. Há uma versão de teste grátis por 7 dias. Além disso, nesta aplicação contem 3 planos mensais apresentados na Tabela 2.

Tabela 2 – Visão dos planos da aplicação Salon Soft

Módulos	Básico	Ideal	Avançado
Agenda	X	X	X
Cadastro de clientes	X	X	X
Caixa/Comandas		X	X
Cálculo de comissão		X	X
Controle de fiados		X	X
Gestão financeira		X	X
Controle de estoque			X
Fornecedores			X

Fonte: Salon Soft

## 2.3.2.3 Corthy

Figura 9 – Corthy



Fonte: Inovhy.

O Corthy: Gestão de Salão de Beleza, Estética + é uma aplicação mais genérica, o qual possui diversos setores que o abrange, como clínica de podologia, estúdio de tatuagem e barbearia. É uma ferramenta para organizar o negócio, voltado para melhorar a comunicação com os clientes e aumentar o lucro da empresa. Visto que seu maior foco é no negócio e deve atender bem a mesma ideia do projeto abordado neste estudo. Porém, há limitações, como as outras aplicações estudadas anteriormente, este tem funcionalidades básicas grátis e planos mensais para o uso das funções mais avançadas.

### 2.3.3 Considerações finais de revisão

Neste capítulo foram abordados conceitos relacionados ao desenvolvimento deste trabalho. Começando com a Engenharia de software no processo de desenvolvimento de software, descrevendo também o processo de desenvolvimento focado para aplicações móveis, suas abordagens e tecnologias. Levantado a importância da modelagem de software, incluindo a *Interaction Flow Modeling Language* (IFML), focado a expressar as interações entre o sistema e o usuário. Além disso, foi estudado e levantados características de aplicações com o escopo semelhante ao proposto por este projeto. O capítulo seguinte descreve o desenvolvimento da aplicação proposta desde o estágio inicial, contendo as tecnologias utilizadas, levantamento de requisitos, histórias de usuário e protótipos, também a modelagem *back-end* e *front-end*, até a validação da aplicação.

## 3 Desenvolvimento

Este capítulo descreve o desenvolvimento do projeto e da aplicação proposta, chamada ProjectStudio. Serão descritas as tecnologias utilizadas, o levantamento de requisitos, modelagem *front-end* e modelagem *back-end*.

### 3.1 Tecnologias utilizadas

Esta seção apresenta as principais tecnologias e ferramentas que foram utilizadas no decorrer do desenvolvimento deste trabalho.

#### 3.1.1 PostgreSQL

O PostgreSQL é um poderoso Sistema de Gerenciamento de Banco de Dados (SGBD) relacional, de código aberto, que estende a linguagem SQL combinada a diversos recursos de armazenamento e escalabilidade com segurança de dados mais complexos. PostgreSQL teve origem em 1986, como parte do projeto POSTGRES na Universidade da Califórnia em Berkeley e tem mais de 30 anos de desenvolvimento ativo na plataforma central.

#### 3.1.2 WebRatio

Segundo os distribuidores da WebRatio, ela é uma plataforma de baixo código que permite desenvolver aplicações Web 7 vezes mais rápido, personalizando todo tipo de detalhe de operações e aparência. As ferramentas da plataforma possibilitam criar o *front-end*, *controller* e *back-end*. Dessa forma, se encaixa na arquitetura MVC (Model, View and Controller), que nada mais é uma arquitetura que separa o *software* em 3 camadas. *Model* destinada à manipulação de dados, *View* para a interação com o usuário e *Controller* para o controle e execução das operações.

### 3.2 Processo de desenvolvimento

Nesta seção será apresentado o processo de desenvolvimento da aplicação ProjectStudio, desde as etapas iniciais, o levantamento de requisitos, a elaboração das histórias de usuários, protótipo, até a modelagem da aplicação do *front-end* e do *back-end*.

### 3.2.1 Levantamento de requisitos

De acordo com Sommerville (2011), os requisitos de um sistema são a descrição de um comportamento que o sistema deve oferecer, os serviços e suas restrições de funcionamento. Esses requisitos são identificados através de uma análise das necessidades dos clientes com uma finalidade determinada, como controlar um dispositivo, colocar um pedido, registrar informações ou encontrá-las, por exemplo. Esse processo de analisar, descobrir, documentar e verificar funções e restrições é chamado Engenharia de Requisitos.

Para efetuar o levantamento de requisitos da aplicação, primeiramente foram realizadas reuniões com a proprietária do Studio Vanilza Mendes.

A lista abaixo mostra algumas características importantes sobre a aplicação:

- A proprietária marca os atendimentos em uma agenda física. A mesma tarefa deverá ser realizada na aplicação, com a descrição do atendimento, especificando qual serviço foi utilizado e conseqüentemente com o valor pago/devido.
- Foi identificado que há o mínimo de controle de fluxo de caixa, pois, é anotado os valores pendentes de cada cliente. Então, para gerenciar a parte econômica do estúdio, deveremos colocar um módulo financeiro, com os valores pagos e devidos de cada serviço prestado e cada produto vendido.

A Tabela 3 apresenta alguns exemplos de requisitos levantados inicialmente.

Tabela 3 – Requisitos levantados

Funcionalidade	Descrição
Cliente	Gerenciar clientes, desse modo relacionando os clientes com agendamentos e comandas.
Agendamento	O cliente informa o dia e o horário específico que quer realizar o atendimento. Com o horário disponível, é criado um agendamento. O histórico dos agendamentos serão mantidos, possibilitando a visualização do status de pagamento (aberto/fechado).
Despesa	No Studio há necessidade de registrar a saída de dinheiro. Logo, precisamos de um módulo para gerenciar e manter o histórico de despesas.
Comanda	Pode acontecer de algum cliente comprar algum produto ou realizar algum serviço sem ter marcado previamente. Então, temos a comanda para gerenciar a entrada de dinheiro, diferente do agendamento.
Tela <i>Home</i>	A tela inicial ( <i>Home</i> ) será responsável por exibir informações resumidas dos módulos do sistema de forma estratégica.

Fonte: produzido pelo autor.

### 3.2.2 Histórias de usuário

As histórias de usuário têm origem em abordagens de desenvolvimento ágil de *software*, tais como o *Extreme Programming* (XP). Apesar de não adotarmos o XP neste projeto, as histórias ajudam a representar uma visão melhor dos papéis do usuário em relação às funcionalidades da aplicação.

Importante ressaltar que, existem diferentes formatos para a escrita de histórias de usuário. O modelo utilizado durante o desenvolvimento deste projeto seguiu aquele proposto em (LONGO; SILVA, 2014):

- **Como um...** (Quem? - papel/ator)
- **eu quero...** (O que? - funcionalidade pretendida)
- **de modo que...** (Porque? - valor de negócio/benefício)

A seguir apresenta-se as histórias de usuário da aplicação neste período do desenvolvimento.

- **Cadastrar agendamento**

**Como um** administrador

**eu quero** cadastrar agendamentos na aplicação

**de modo que** possibilite os clientes marcarem horários de atendimentos no estúdio.

- **Listar despesas**

**Como um** administrador

**eu quero** visualizar todas despesas na aplicação

**de modo que** tenha uma lista de todas as despesas filtradas por um período. Em cada item da lista, deverá ter opção de ver todos os atributos de uma determinada despesa.

- **Editar dados de comanda**

**Como um** administrador

**eu quero** editar os dados de uma comanda na aplicação

**de modo que** permita alterar todos os dados da comanda, a fim de corrigir eventuais erros e/ou mudanças.

- **Tela home**

**Como um** administrador  
**eu quero** a visão geral de todos os módulos  
**de modo que** tenha informações estratégicas e resumidas de todo estúdio.

- **Cadastrar agendamento**

**Como um** administrador  
**eu quero** cadastrar agendamentos na aplicação  
**de modo que** possibilita os clientes marcarem horários de atendimentos no estúdio.

- **Listar agendamento**

**Como um** administrador  
**eu quero** visualizar agendamentos na aplicação  
**de modo que** possibilita identificar quais clientes estão com horários marcados em determinado dia e horário no estúdio.

- **Editar agendamento**

**Como um** administrador  
**eu quero** editar os dados de um agendamento na aplicação  
**de modo que** permita alterar todos os dados, a fim de corrigir eventuais erros e/ou mudanças.

- **Excluir agendamento**

**Como um** administrador  
**eu quero** excluir um agendamento na aplicação  
**de modo que** permita excluir permanentemente o horário agendado.

- **Cadastrar comanda**

**Como um** administrador  
**eu quero** cadastrar comandas na aplicação  
**de modo que** possibilita os clientes comprar produtos e realizar serviços sem necessidade de agendamento.

- **Listar comanda**

**Como um** administrador  
**eu quero** visualizar comandas na aplicação  
**de modo que** possibilite identificar produtos e serviços associados a cada cliente.

- **Editar comanda**

**Como um** administrador  
**eu quero** editar os dados de uma comanda na aplicação  
**de modo que** permita alterar todos os dados, a fim de corrigir eventuais erros e/ou mudanças.

- **Excluir comanda**

**Como um** administrador  
**eu quero** excluir uma comanda na aplicação  
**de modo que** permita excluir permanentemente.

- **Cadastrar despesa**

**Como um** administrador  
**eu quero** cadastrar despesa na aplicação  
**de modo que** permita o estúdio registrar custos fixos e custos variáveis.

- **Listar despesa**

**Como um** administrador  
**eu quero** visualizar despesas na aplicação  
**de modo que** possibilite identificar onde estão os maiores gastos, assim focar na eficiência na operação do estúdio.

- **Editar despesa**

**Como um** administrador  
**eu quero** editar os dados de uma despesa na aplicação  
**de modo que** permita alterar todos os dados, a fim de corrigir eventuais erros e/ou mudanças.

- **Excluir despesa**

**Como um** administrador  
**eu quero** excluir uma despesa na aplicação  
**de modo que** permita excluir permanentemente.

- **Cadastrar cliente**

**Como um** administrador  
**eu quero** cadastrar clientes na aplicação  
**de modo que** permita registrar todas as ações do cliente associadas ao estúdio.

- **Listar cliente**

**Como um** administrador  
**eu quero** visualizar clientes na aplicação  
**de modo que** possibilita identificar comandas, agendamentos e valores pendentes.

- **Editar cliente**

**Como um** administrador  
**eu quero** editar os dados de um cliente na aplicação  
**de modo que** permita alterar todos os dados, a fim de corrigir eventuais erros e/ou mudanças.

- **Excluir cliente**

**Como um** administrador  
**eu quero** excluir uma cliente na aplicação  
**de modo que** permita excluir permanentemente.

### 3.2.3 Protótipos

Segundo [Sommerville \(2011, p. 30\)](#), um protótipo é uma versão inicial de sistema, a fim de demonstrar conceitos e validar requisitos e restrições, experimentar detalhes do projeto, explorar o problema e suas possibilidades de soluções. Nas etapas iniciais de desenvolvimento, podemos usar o protótipo para validar os requisitos junto aos *stakeholders* do sistema. No processo de projeto de sistema, usamos para estudar soluções específicas do *software* e apoiar o desenvolvimento de *interfaces* de usuário.

A Figura 10 representa a tela **Home** da aplicação. Ao topo, fica um menu de navegação entre os módulos, agrupando funcionalidades da mesma "família", ou seja, funcionalidades que sejam semelhantes. Logo abaixo, fica o logotipo do Studio Vanilza

Mendes. Em seguida, o dia da semana com a data atual. Do lado direito, fica dois botões, um para compartilhar propaganda do estúdio e outra para sincronizar os dados locais com os dados na nuvem. No quadro de próximos horários fica os 5 próximos horários do dia. No quadro seguinte, fica um resumo financeiro total sobre o estúdio, porém, com muito mais detalhes no módulo próprio, do mesmo jeito os agendamentos.

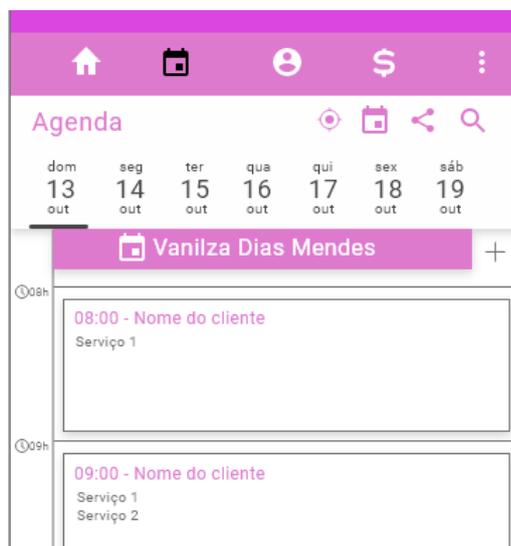
Figura 10 – Protótipo da Tela Home



Fonte: Produzida pelo autor, 2019.

A Figura 11 representa a tela **Agenda** da aplicação. Pode-se ver que no topo temos alguns ícones, os quais têm funcionalidade de busca, restaurar o local da agenda para a data atual e compartilhamento da agenda. O intuito é gerar uma imagem de fácil compartilhamento, pronta para o envio entre outras aplicações. Assim, o cliente poderá visualizar os horários disponíveis da página. Uma função de extrema importância é alterar entre agendas dos funcionários.

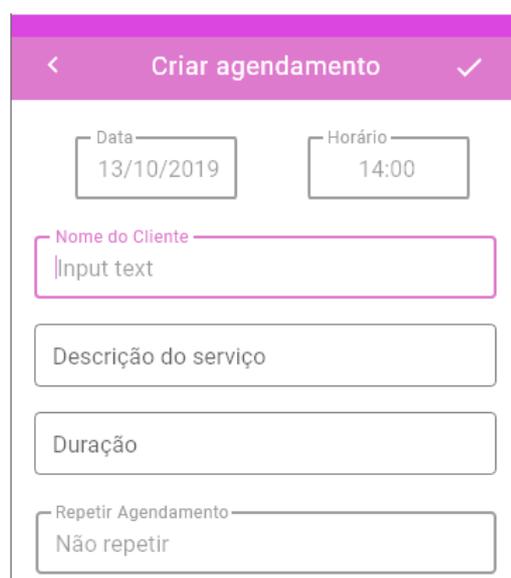
Figura 11 – Protótipo da Tela Agenda



Fonte: Produzida pelo autor, 2019.

A Figura 12 representa a tela **Criar agendamento** da aplicação. Pode-se ver que no topo há o ícone de confirmar e de voltar para tela anterior. O ícone de voltar, é importante, pois, há dispositivos que não vêm nativamente com essa função. Nesta tela, os campos são preenchidos para criar um horário no estúdio. Os dados devem ser validados para não causar conflitos.

Figura 12 – Protótipo da Tela criar agendamento



Fonte: Produzida pelo autor, 2019.

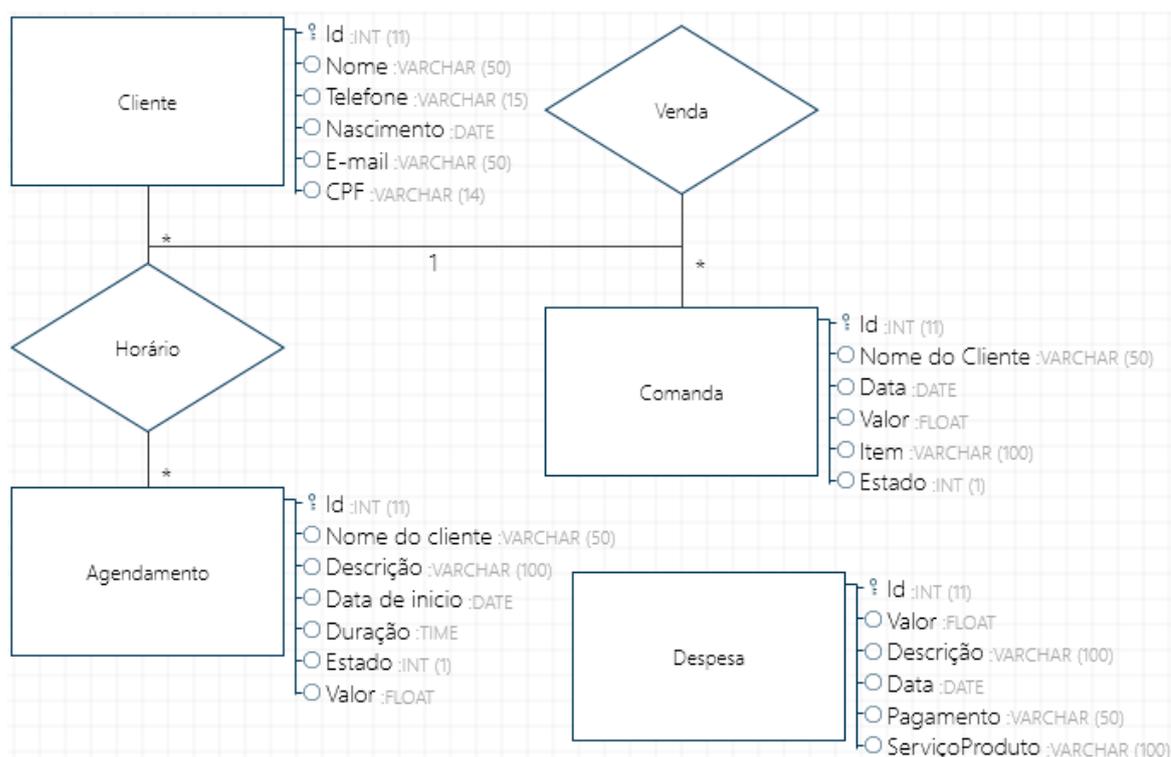
### 3.2.4 Modelagem do *back-end*

Esta subseção apresentará alguns diagramas correspondentes à modelagem do *back-end* do ProjectStudio, ou seja, representa o lado que fica atrás do sistema acessado pelo usuário.

#### 3.2.4.1 Modelo de Banco de Dados

Para a persistência dos dados e informações da aplicação, foi utilizado o sistema de gerenciamento de banco de dados relacional PostgreSQL, o qual é suportado pela *Web Ratio* e cumpre os requisitos da aplicação e de modelagem do projeto. O modelo em questão, relacional, é ortogonal, armazena dados em tabelas e toda operação resulta em novas tabelas. A Figura 13 apresenta a modelagem Entidade Relacionamento geral da aplicação.

Figura 13 – Modelo Entidade Relacionamento



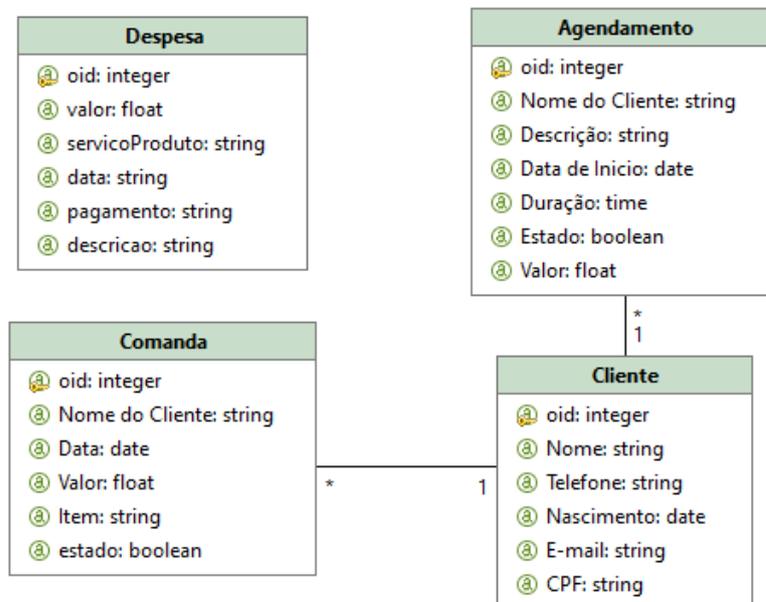
Fonte: Produzida pelo autor, 2021.

#### 3.2.4.2 Modelo de domínio

No decorrer de um projeto de software, os diagramas de classes são muito utilizados, segundo [Sommerville \(2011\)](#). Nestes diagramas, são representados *Controllers* e *Models*, com métodos e atributos, bem como os relacionamentos. Porém, o modelo de domínio

vai além do diagrama de classes. O modelo de domínio utiliza o diagrama de classes para expressar informações de um certo domínio da aplicação. Visualmente, não há diferenças entre eles. A Figura 14 mostra o modelo de domínio referente à aplicação.

Figura 14 – Modelo de domínio

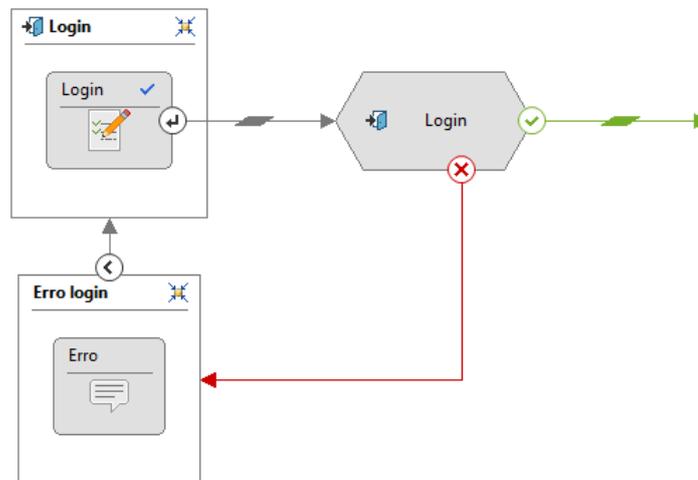


Fonte: Produzida pelo autor, 2020.

### 3.2.5 Modelagem do *front-end*

Para realizar a modelagem do *front-end* do ProjectStudio, foi utilizada a *Interaction Flow Modeling Language* (IFML). Os diagramas foram criados na ferramenta *WebRatio Mobile Community*, que especifica as *views* apresentadas aos usuários. Como sugerido pela IFML, a primeira fase deste processo foi a definição do modelo de domínio do sistema, apresentado na seção anterior. Em seguida, as *views* foram elaboradas. Com o intuito de facilitar a visualização dos modelos, eles foram agrupados quanto as funcionalidades.

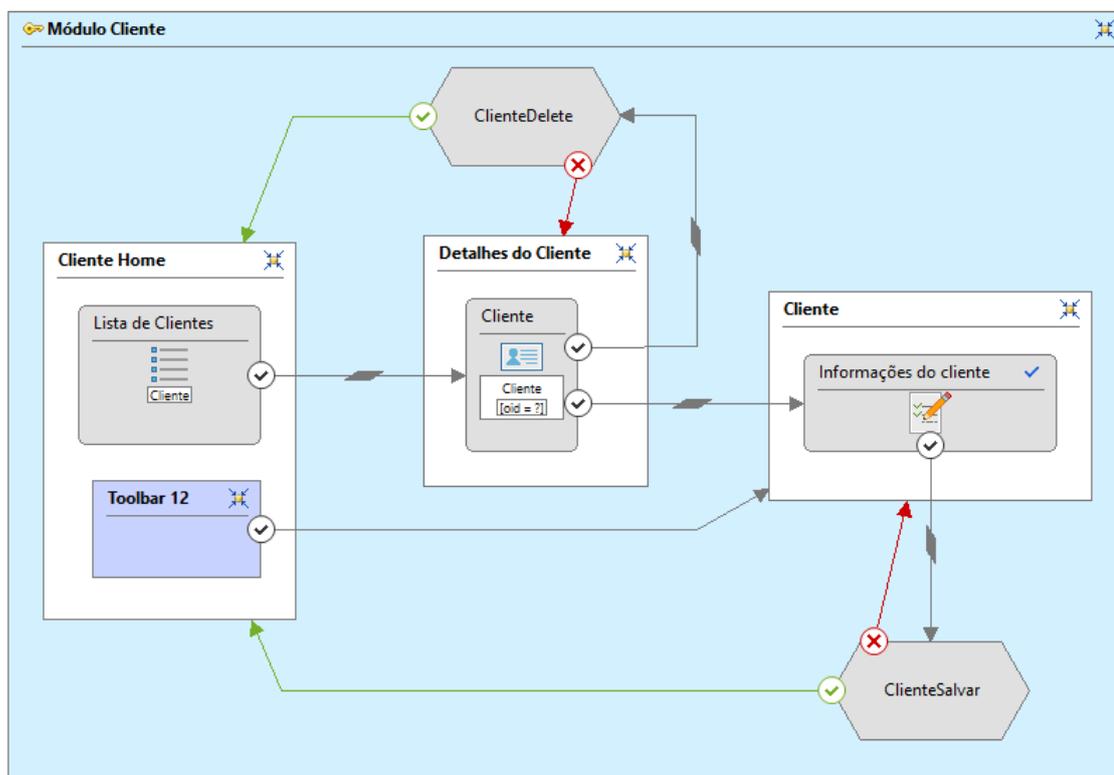
A Figura 15 mostra o diagrama IFML no processo de *login* efetuado pelo usuário. Na tela inicial da aplicação, o usuário preenche as informações de nome do usuário e senha. Caso sejam válidas as informações, segue o fluxo *Flow OK* em verde passando por parâmetro a chave de acesso à aplicação. Caso seja inválida a credencial, segue pelo *Flow KO* em vermelho, e uma mensagem de erro é exibida e, em sequência, o usuário é redirecionado para a tela inicial.

Figura 15 – Diagrama IFML para *login*

Fonte: Produzida pelo autor, 2019.

A Figura 16 mostra o diagrama IFML no processo da gestão de clientes. A primeira tela deste módulo, chamada *Cliente Home*, há uma lista com todos os usuários cadastrados na aplicação. Ela contém apenas informações básicas, tais como o nome e o telefone. Para visualizar todas as informações, basta tocar no item da lista e o usuário é direcionado para a tela de detalhes do cliente, podendo visualizar todos os atributos. Nesta tela, pode-se acionar o botão delete, o qual apaga permanentemente o objeto e retorna o usuário para a tela anterior. Caso queira atualizar as informações, deve-se tocar no botão de atualizar dentro de detalhes do cliente. As informações serão passadas para o formulário na tela cliente. Nesta mesma tela, pode-se adicionar um novo cliente.

Figura 16 – Diagrama IFML para gestão do cliente

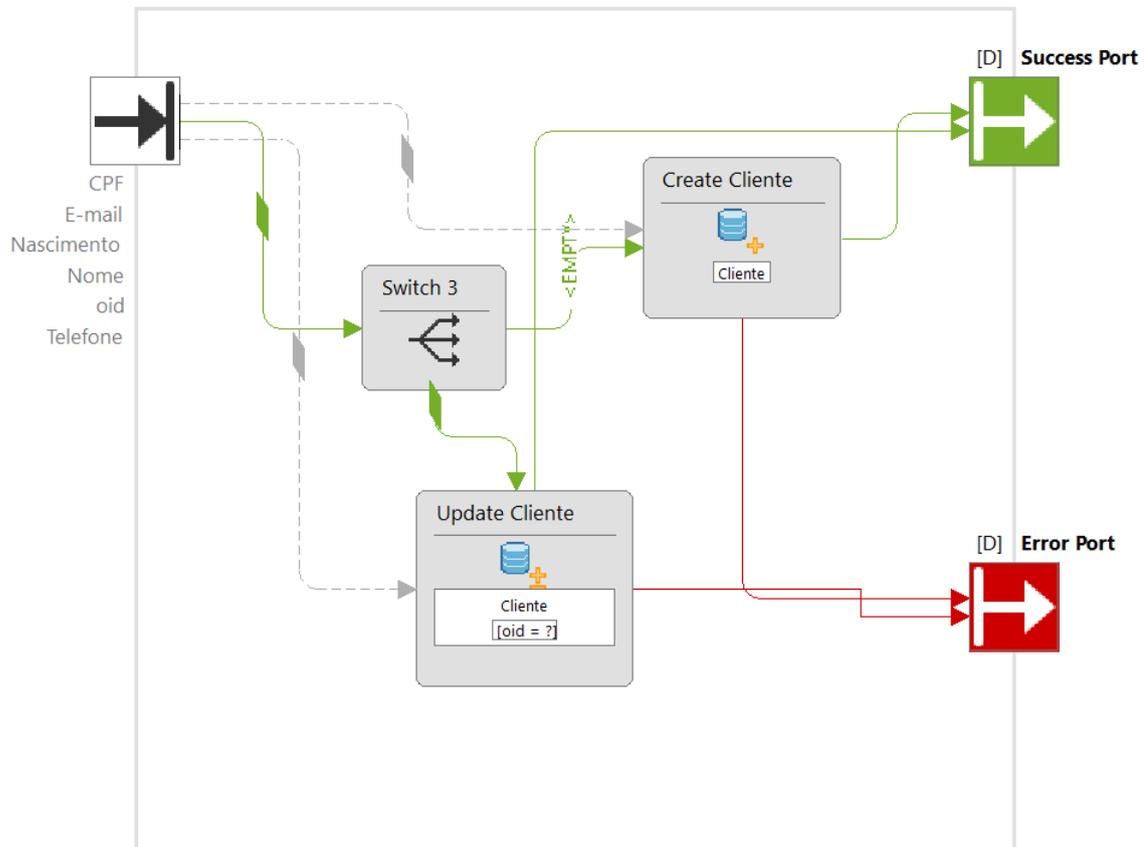


Fonte: Produzida pelo autor, 2021.

A seção de apêndices A, consta o restante da modelagem IFML deste projeto.

### 3.2.5.1 Modelagem do *Controller*

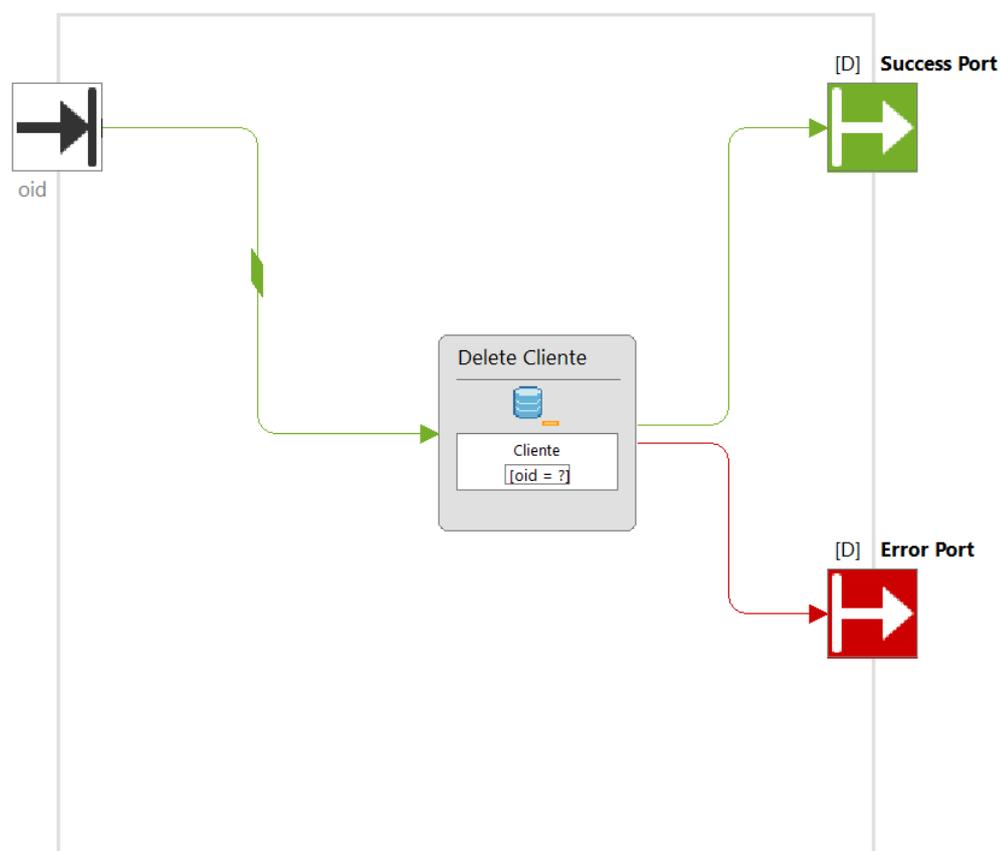
*Action* na plataforma WebRatio tem o principal papel de realizar ações. Portanto, pertence à camada de *Controller*. A Figura 17 expõe o ato de *create* e *update* do objeto cliente. O ícone da esquerda significa os parâmetros de entrada da ação. Após a entrada, o ícone de *Switch* irá consultar se o ID do objeto já existe ou é um novo ID. Logo, o fluxo irá para o *Create* Cliente, ou para *Update* Cliente. A saída da operação será um resultado verdadeiro ou falso.

Figura 17 – *Action* de salvar cliente

Fonte: Produzida pelo autor, 2021.

Similarmente, o *Action* para *delete* tem a mesma estrutura no anterior, porém, mais simples. A ação de *delete*, recebe apenas o ID do objeto e esse objeto simplesmente será deletado da base de dados, resultando em sucesso ou não da operação.

Figura 18 – Action de delete cliente



Fonte: Produzida pelo autor, 2021.

Neste capítulo, foram documentadas todas as etapas do processo de desenvolvimento do projeto, que recebeu o nome de ProjectSudio. Desde o levantamento de requisitos, e histórias de usuário, até a apresentação da modelagem do *back-end* e *front-end* da aplicação. O capítulo seguinte irá apresentar os resultados, não só apresentar a aplicação, e comparar, mas também validar resultados de testes automatizados.

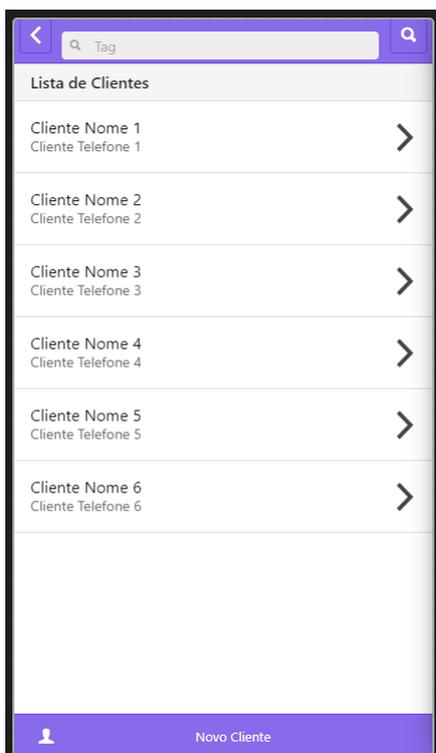
## 4 Resultados

Este capítulo apresenta os resultados do projeto e a aplicação desenvolvida. Descreve as dificuldades que surgiram durante o desenvolvimento, exhibe a aplicação e compara as funcionalidades com a versão padrão (menor custo mensal) das aplicações semelhantes descritas no Capítulo 2. Mesmo com escopo um pouco menor, não foi o objetivo deste projeto cobrir inicialmente todos os pontos abrangidos pelas aplicações semelhantes e sim estudar a abordagem MDD. Os testes automatizados foram realizados no *Firebase Test Lab*, sistema que permite testar a aplicação em uma variedade de dispositivos e configurações. Logo após, houve a avaliação da proprietária do Studio Vanilza Mendes.

### 4.1 Apresentação do sistema

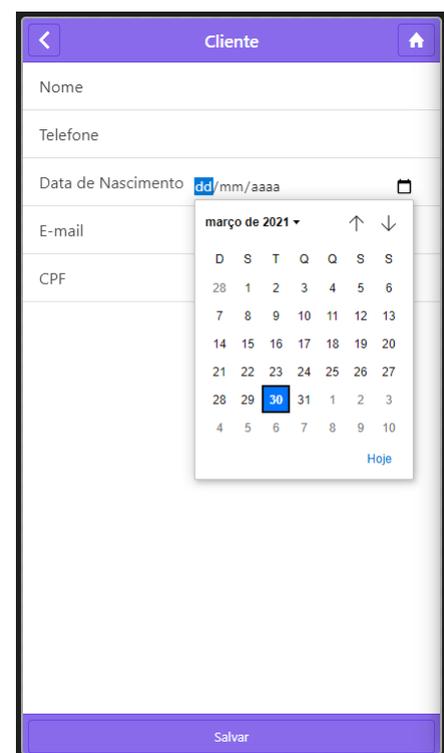
A aplicação utilizou o *template* padrão da plataforma WebRatio. A Figura 19 exhibe a tela inicial do módulo de clientes, na qual há a lista de clientes do estúdio e uma barra de pesquisar. A Figura 20 apresenta a tela de adicionar novo cliente. A mesma pode ser utilizada para editar as informações de cliente. De modo simples, o *input* do tipo *date* já facilita a escolha de uma data via o *gadget*.

Figura 19 – Lista de clientes



Fonte: Produzido pelo autor, 2021

Figura 20 – Cadastro de cliente



Fonte: Produzido pelo autor, 2021

A Figura 21 apresenta a tela de menu do ProjectStudio, que irá apresenta, resumidamente, algumas informações objetivas sobre o estúdio. A barra de navegação dos módulos Cliente, Agenda, Despesas e Comanda, foram passadas para a parte inferior da tela. A barra superior possibilita retornar para a tela anterior e ir diretamente para a tela principal.

Figura 21 – Tela Home



Fonte: Produzido pelo autor, 2021

A seção de apêndices B, contém mais telas da aplicação desenvolvida.

## 4.2 Validação da Aplicação

Segundo Sommerville (2011), a fase de testes em um *software* possui dois objetivos distintos. A primeira frase, é demonstrar ao desenvolvedor e ao cliente, que o software atende aos requisitos levantados e acordados. Esses testes são relacionados à validação, onde são utilizados casos de testes que representam o uso esperado do sistema. O segundo propósito envolve descobrir possíveis situações em que o *software* se comporta de maneira inesperada, indesejável ou fora do que foi especificado. Para isso, podem ser utilizado casos de teste que representam situações inesperadas de uso do sistema, por exemplo, envolvendo dados e ações inválidas.

### 4.2.1 Firebase Test Lab

O ambiente do *Firebase Test Lab* permite configurar o teste com diversos dispositivos móveis, além disso, pode executar o teste em até 60 min e 5 vezes por dia na versão gratuita do serviço. Neste teste, foi utilizado o dispositivo Pixel 2, com a API 28 do Android. A Figura 22 mostra o resumo do teste automatizado realizado.

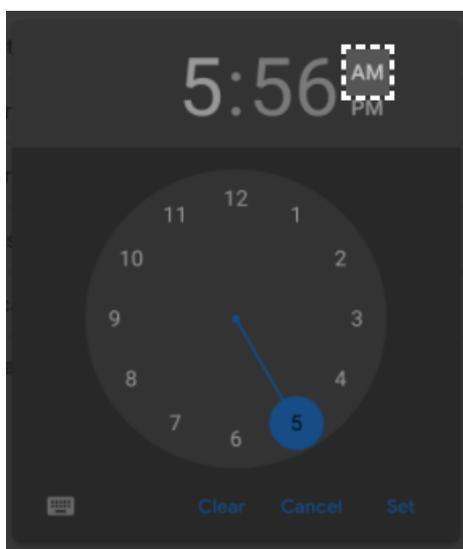
Figura 22 – Visão geral de teste automatizado



Fonte: Firebase Test Lab.

A Figura 23 destaca uma sugestão do robô de teste com relação à acessibilidade. Segundo o resultado do teste, este é um aviso do tamanho da área de toque. Esses elementos precisam ter uma altura e largura com pelo menos 48dp, segundo as diretrizes de acessibilidade do *Material Design*.

Figura 23 – Sugestão de acessibilidade



Fonte: Firebase Test Lab.

Outro ponto levantado, é em relação ao mesmo item na Figura 23, na questão de contraste, categoria de problemas secundários. A combinação das cores das palavras e do fundo, segundo o resultado do teste, podem prejudicar a leitura e compreensão do texto para o usuário.

## 4.2.2 Validação da proprietária

O *feedback* da proprietária do Studio Vanilza Mendes foi realizado através de um formulário criado no Formulários Google. Com isso, foram realizadas perguntas, a fim de avaliar alguns aspectos da aplicação desenvolvida deste projeto. A seção de apêndices C apresenta o questionário completo e as respostas da proprietária. Em destaque, pontos apontados por ela, sobre controle de estoque de produtos e mudanças na *interface* de usuário. Em seguida, o bom desempenho da aplicação, visto que, não apresentou travamentos ou lentidão. De acordo com a proprietária, o ProjectStudio poderá trazer impactos positivos na gestão do estúdio. Os testes foram realizados por simulações, pois devido a situação de pandemia, o estúdio encontra-se fechado por tempo indeterminado.

## 4.2.3 Comparação

Visto que a aplicação deste projeto não foi desenvolvida, inicialmente, com o objetivo de abranger funcionalidades existentes nas aplicações semelhantes, pode-se futuramente incrementar novos requisitos por meio de atualizações. A Tabela 4 exibe as funcionalidades do ProjectStudio e compara aplicações semelhantes apresentadas no Capítulo 2.

Tabela 4 – Funcionalidades abrangidas

Módulos	ProjectStudio	Pocket Agenda	Salon Soft	Corthy
Agenda	X	X	X	X
Cadastro de clientes	X	X	X	X
Caixa/Comandas	X	X		X
Gestão financeira	X	X		X
Calculo de comissões		X		X
Controle de estoque				X

Fonte: Produzido pelo autor, 2021.

## 4.3 Dificuldades

Sem dúvidas, o desenvolvimento deste projeto foi desafiador, devido a abordagem única de desenvolvimento. O autor sempre desenvolveu aplicações diretamente pela linguagem de programação. Com a ideia de acelerar o desenvolvimento de uma aplicação multiplataforma para dispositivos móveis, a abordagem utilizada cumpre bem sua proposta. Porém, há alguns contratemplos nesta plataforma da WebRatio. Um ponto bastante relevante é a falta de material de estudo, visto que, muitas coisas que podem ser feitas, não são de fato ensinadas no material de referência. Como consequência, isso dificultou o processo de aprendizagem, refletindo na dificuldade da conclusão de alguns requisitos. É de extrema importância ressaltar, a versão utilizada neste estudo é a WebRatio 8.10.4

*Mobile Community*. Por outro lado, há uma versão paga da plataforma que pode ter melhor suporte e matérias de estudo.

Além disso, houve um problema na integração com o SGBD MySQL. Por algum motivo, não foi possível integrar o banco de dados na aplicação. Com isso, o uso do PostgreSQL possibilitou a resolução deste problema.

Neste capítulo, foram apresentados os resultados do projeto. Conforme mencionado, as dificuldades trouxeram impacto no processo de desenvolvimento, porém, não inviabilizam o estudo da abordagem. Na visão do autor, a abordagem MDD tem extremo potencial de acelerar e facilitar o desenvolvimento de *software*. Foram apresentados testes automatizados do *Firebase Test Lab* e a comparação com as aplicações semelhantes. No próximo capítulo, serão apontadas as considerações finais, bem como os trabalhos futuros.

## 5 Considerações finais

Este trabalho apresentou o desenvolvimento de uma aplicação por meio da abordagem *Model-Driven Development* (MDD), para apoio da gestão do Studio Vanilza Mendes. Primeiramente, foi realizado um estudo sobre a Engenharia de *Software*, desde conceitos de sistema, e desenvolvimento de aplicações móveis, até a modelagem de *software Interaction Flow Modeling Language* (IFML). Com objetivo de analisar quais impactos a abordagem MDD com a modelagem na IFML tem no desenvolvimento de aplicações. A IFML foi utilizada para representar a interação dos usuários com a aplicação. De modo simples, os modelos possibilitam a geração de códigos executáveis.

Posteriormente, foram abordadas as tecnologias utilizadas neste projeto, como o PostgreSQL e a plataforma de desenvolvimento WebRatio 8.10.4 *Mobile Community*, logo tiveram grande impacto no desenvolvimento do trabalho. Também foi descrito o processo de levantamento de requisitos e a construção de modelos *back-end* e *front-end*, bem como a apresentação da aplicação, a validação por testes automatizados, a validação da proprietária do Studio Vanilza Mendes e as dificuldades encontradas no desenvolvimento. Durante o desenvolvimento, foram realizadas reuniões para discussão de aspectos importantes da aplicação.

Sabe-se que uma aplicação não pode ficar sem evoluir. Portanto, a aplicação deste projeto continuará a ser desenvolvida e, as eventuais falhas serão corrigidas. Juntamente as novas tecnologias, pode-se utilizar novos meios de prosseguir com o projeto, de acordo com a necessidade do estúdio. Em síntese, espera-se que a aplicação desenvolvida auxilie o Studio Vanilza Mendes a gerenciar melhor seu negócio.

A seguir, serão apresentados algumas ideias de propostas de trabalhos futuros.

### 5.1 Trabalhos futuros

Foram identificadas propostas de melhorias e novos recursos, assim gerando novas possibilidades para trabalhos futuros:

- Realizar melhorias no *design* gráfico das telas.
- Desenvolver módulo de controle de estoque dos produtos, de modo a permitir alertar o baixo estoque de determinado produto.
- Enviar mensagens automáticas de lembretes aos clientes via WhatsApp, assim lembrando-o do horário agendado no Studio.

- Estudo comparativo entre ferramentas que utilizam a abordagem MDD.
- Estudo de reusabilidade da abordagem MDD.

# Referências

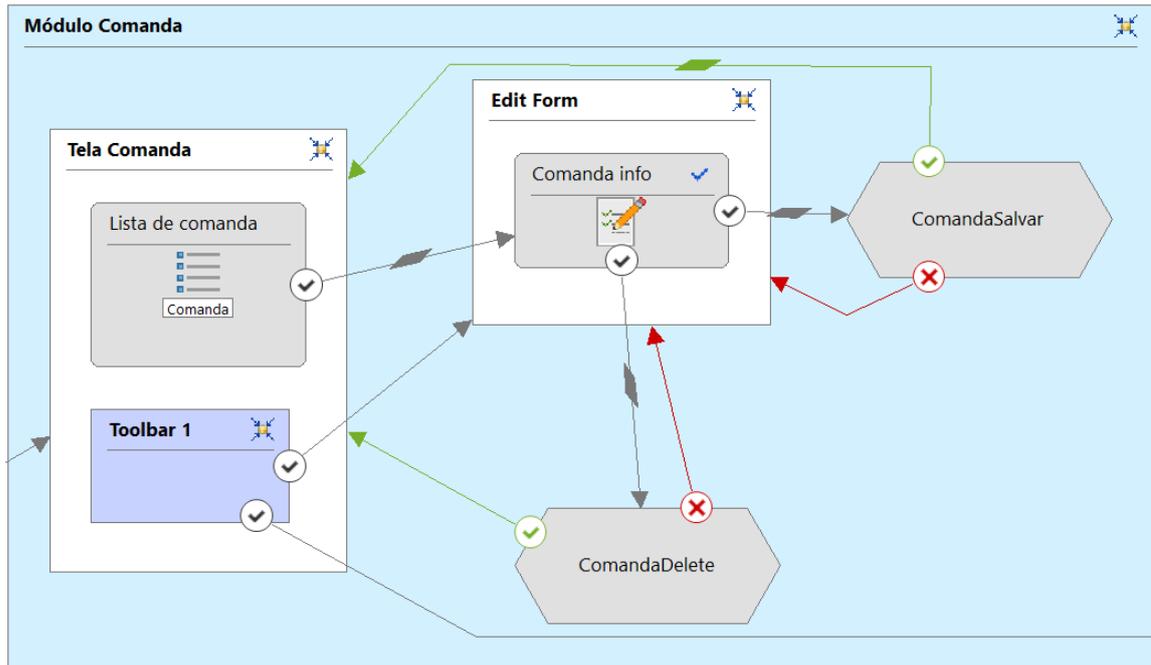
- ACERBIS, R. et al. Model-driven development of cross-platform mobile applications with webratio and ifml. In: IEEE PRESS. *Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems*. [S.l.], 2015. p. 170–171. Citado na página 21.
- BRAMBILLA, M.; FRATERNALI, P. *Interaction flow modeling language: Model-driven UI engineering of web and mobile apps with IFML*. [S.l.]: Morgan Kaufmann, 2014. Citado na página 22.
- CHIAVENATO, I. *Administração*. [S.l.]: Elsevier Brasil, 2007. Citado na página 16.
- FRANZAGO, M.; MUCCINI, H.; MALAVOLTA, I. Towards a collaborative framework for the design and development of data-intensive mobile applications. In: *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems*. [S.l.: s.n.], 2014. p. 58–61. Citado na página 21.
- LAUDON, K. C.; LAUDON. *Sistemas de informação gerenciais*. [S.l.]: Pearson Educação, 2004. Citado na página 18.
- LONGO, H. E. R.; SILVA, M. P. A utilização de histórias de usuários no levantamento de requisitos ágeis para o desenvolvimento de software. *International Journal of Knowledge Engineering and Management (IJKEM)*, v. 3, n. 6, p. 1–30, 2014. Citado na página 30.
- MARINHO, E. H.; RESENDE, R. F. Native and multiple targeted mobile applications. In: SPRINGER. *International Conference on Computational Science and Its Applications*. [S.l.], 2015. p. 544–558. Citado 2 vezes nas páginas 19 e 20.
- RAJ, C. R.; TOLETY, S. B. A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach. In: IEEE. *2012 Annual IEEE India Conference (INDICON)*. [S.l.], 2012. p. 625–629. Citado na página 14.
- RIEGER, C.; MAJCHRZAK, T. A. Towards the definitive evaluation framework for cross-platform app development approaches. *Journal of Systems and Software*, Elsevier, v. 153, p. 175–199, 2019. Citado na página 14.
- SOMMERVILLE, I. *Software engineering*. [S.l.]: Addison-wesley, 2011. Citado 8 vezes nas páginas 14, 15, 18, 19, 29, 33, 36 e 43.
- VAUPEL, S. et al. Model-driven development of mobile applications for android and ios supporting role-based app variability. *Software & Systems Modeling*, Springer, v. 17, n. 1, p. 35–63, 2018. Citado na página 15.
- WHITE, S. A.; MIERS, D. *BPMN modeling and reference guide: understanding and using BPMN*. [S.l.]: Future Strategies Inc., 2008. Citado na página 21.
- WHITTLE, J.; HUTCHINSON, J.; ROUNCEFIELD, M. The state of practice in model-driven engineering. *IEEE software*, IEEE, v. 31, n. 3, p. 79–85, 2013. Citado na página 15.

# Apêndices

# APÊNDICE A – Modelos IFML

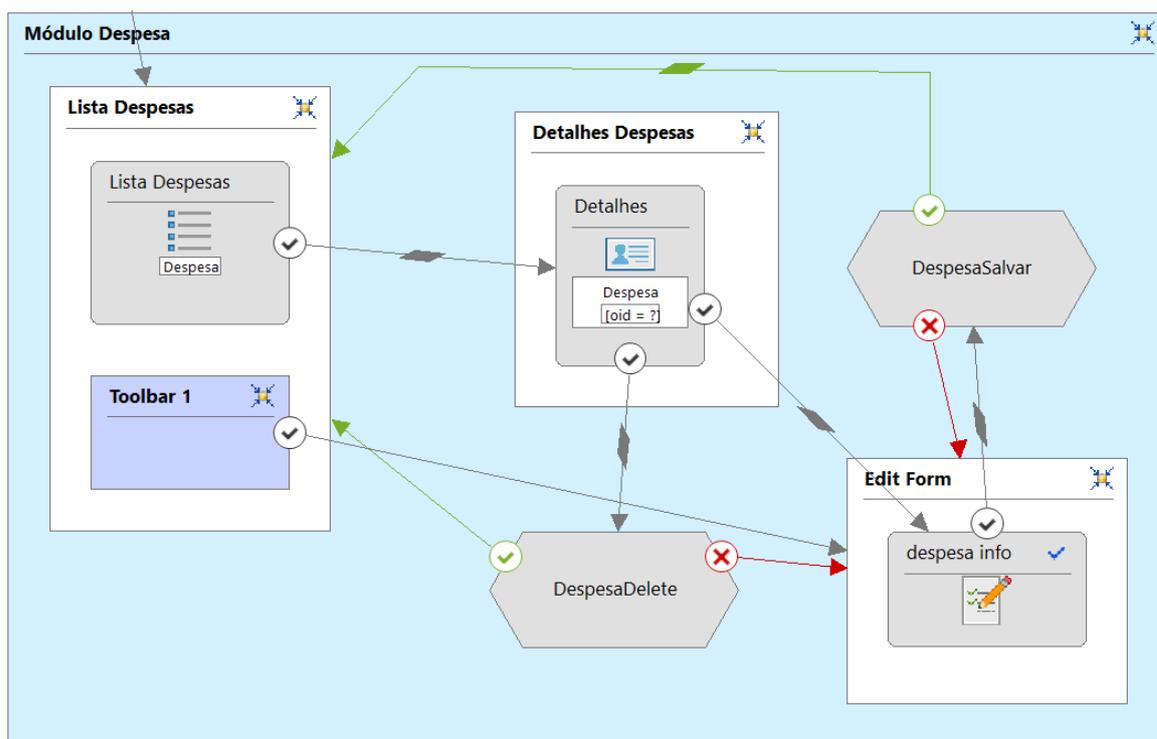
Os modelos IFML utilizados no desenvolvimento da aplicação seguem abaixo.

Figura 24 – Diagrama IFML para gestão de comanda



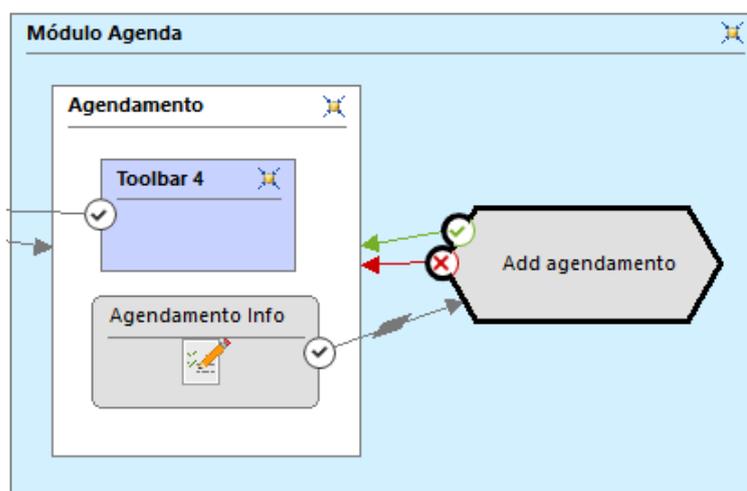
Fonte: Produzida pelo autor, 2021.

Figura 25 – Diagrama IFML para gestão de despesas



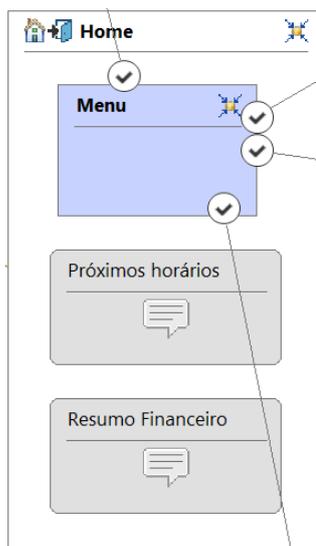
Fonte: Produzida pelo autor, 2021.

Figura 26 – Diagrama IFML agenda



Fonte: Produzida pelo autor, 2021.

Figura 27 – Diagrama IFML tela Home



Fonte: Produzida pelo autor, 2021.

## APÊNDICE B – Telas

A seguir é apresentado as *interfaces* de usuário que representam algumas funcionalidades do ProjectStudio.

Figura 28 – Tela cliente formulário

Nome	Cliente Nome 1
Telefone	Cliente Telefone 1
Data de Nascimento	03/07/2014
E-mail	Cliente E-mail 1
CPF	Cliente CPF 1

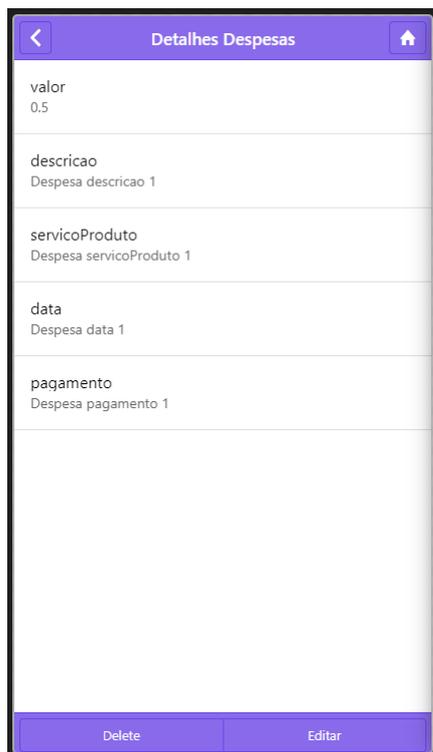
Fonte: Produzido pelo autor, 2021

Figura 29 – Tela lista de despesas

0.5	Despesa servicoProduto 1	Despesa data 1	>
0.667	Despesa servicoProduto 2	Despesa data 2	>
0.75	Despesa servicoProduto 3	Despesa data 3	>
0.8	Despesa servicoProduto 4	Despesa data 4	>
0.833	Despesa servicoProduto 5	Despesa data 5	>
0.857	Despesa servicoProduto 6	Despesa data 6	>

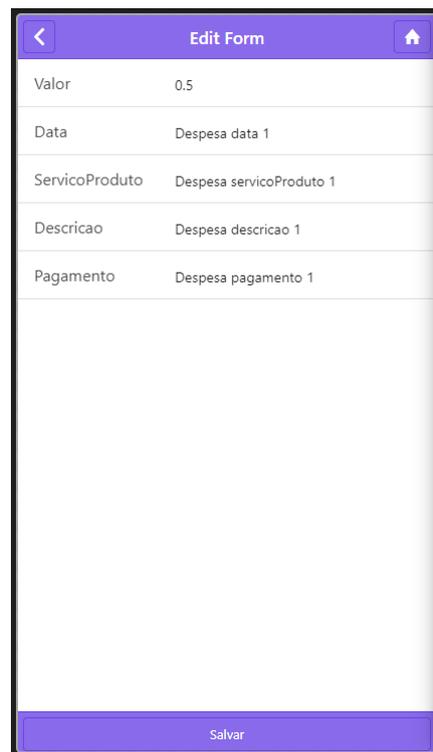
Fonte: Produzido pelo autor, 2021

Figura 30 – Tela detalhes despesa



Fonte: Produzido pelo autor, 2021

Figura 31 – Tela despesa formulário



Fonte: Produzido pelo autor, 2021

A demais telas geradas seguem o mesmo padrão, ou seja, estilo de *design*.

# APÊNDICE C – Questionário de avaliação do ProjectStudio

O formulário de avaliação, foi criado através da plataforma Google. Respondido pela proprietária do Studio Vanilza Mendes.

- Nome completo.  
Vanilza Alves Dias Mendes.
- Achei a aplicação fácil de usar. (Escala linear de 1 a 5)  
5 Bom.
- As funções da aplicação cumprem seus objetivos. (Escala linear de 1 a 5)  
3 Neutro.
- Estabilidade da aplicação. (apresentou falhas e/ou travamentos). (Escala linear de 1 a 5)  
5 Bom.
- Você mudaria algum aspecto de *interface*? (Ex: cor, posições de botões...)  
Tela inicial não atendeu o esperando.
- Deseja adicionar novas funcionalidades?  
Sim.
- Caso seja sim, qual / quais?  
Controle de estoque de produtos.
- Acredita que a aplicação terá impacto positivo no gerenciamento do estúdio?  
Concordo Totalmente.
- Sugestões ou comentários.  
Sem respostas para essa pergunta.

# TERMO DE RESPONSABILIDADE

Eu, **Felipe Dias Mendes** declaro que o texto do trabalho de conclusão de curso intitulado “*Um estudo sobre abordagens dirigidas por modelos para o desenvolvimento de aplicações multiplataforma para dispositivos móveis*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 12 de abril de 2021



---

Felipe Dias Mendes

# DECLARAÇÃO DE CONFORMIDADE

Certifico que o(a) aluno(a) **Felipe Dias Mendes**, autor do trabalho de conclusão de curso intitulado “*Um estudo sobre abordagens dirigidas por modelos para o desenvolvimento de aplicações multiplataforma para dispositivos móveis*” efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

João Monlevade, 30 de abril de 2021.



---

Euler Horta Marinho