

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-
GRANDENSE - CÂMPUS PASSO FUNDO
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

MATHEUS XAVIER

**Desenvolvimento de uma aplicação móvel para fomentar a
educação financeira entre o público infantojuvenil**

Élder F. F. Bernardi

PASSO FUNDO

2018

MATHEUS XAVIER

**Desenvolvimento de uma aplicação móvel para fomentar a
educação financeira entre o público infantojuvenil**

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador (a): Élder F. F. Bernardi

PASSO FUNDO

2018

MATHEUS XAVIER

**Desenvolvimento de uma aplicação móvel para fomentar a
educação financeira entre o público infantojuvenil**

Trabalho de Conclusão de Curso aprovado em ____/____/____ como requisito
parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

Élder F. F. Bernardi

Anubis Graciela de Moraes Rossetto

José Antônio Oliveira de Figueiredo

Rafael Marisco Bertei

PASSO FUNDO

2018

DEDICATÓRIA

*Aos meus pais, por tudo que fizeram
e por sempre acreditarem em mim.*

AGRADECIMENTOS

Sobretudo, agradeço a minha família, pois o incentivo e amparo que eles me deram ao longo desses últimos anos, certamente, foram a força motriz para que eu conseguisse chegar até aqui.

Ademais, agradeço ao IFSUL pela oportunidade de ingressar neste curso e através dele obter grandes aprendizados sobre esta fascinante área de tecnologia da informação, aos professores pelos ensinamentos transmitidos durante o decorrer do curso e ao orientador por nortear desenvolvimento deste projeto.

RESUMO

O trabalho desenvolvido teve por objetivo criar o protótipo de um aplicativo de controle de tarefas que visa fomentar hábitos inerentes a educação financeira para o público infantojuvenil, integrando os dispositivos dos pais com o dos seus filhos para que fosse possível abordar de forma prática a temática no seio familiar. Para tanto, criou-se uma aplicação móvel utilizando o Ionic 3, AngularJS e Apache Cordova que pode ser executada de maneira híbrida. O *back-end* do aplicativo foi construído utilizando o Firebase, onde utilizou-se o banco de dados NoSQL Realtime Database para efetuar as operações CRUD do sistema e o serviço de autenticação provido pela plataforma BaaS do Google. O resultado deste estudo foi viabilizar no aplicativo um mecanismo onde os filhos podem cadastrar seus objetivos e os pais podem cadastrar, enviar e validar tarefas para que participem e auxiliem seus filhos no processo de busca das suas metas de forma lúdica e instigante. Dessa forma, os filhos podem ir praticando implicitamente desde cedo uma metodologia de educação financeira que visa melhorar sua perceptividade em relação âmbito financeiro na sua fase de vida adulta.

Palavras-chave: Aplicativo, Educação Financeira, Gamificação.

ABSTRACT

The objective of this work was to create a prototype of a task control application that aims to foster habits inherent to financial education for the child and adolescent public, integrating the parents' and their children's devices so that it is possible to approach in a practical way the theme in the family. For that, a mobile application was created using Ionic 3, AngularJS and Apache Cordova that can be executed in a hybrid way. The application's back-end was built using Firebase, where the NoSQL Realtime Database was used to perform CRUD system operations and the authentication service provided by Google's BaaS platform. The result of this study was to provide in the application a mechanism where children can register their goals and parents can register, send and validate tasks to participate and help their children in the process of pursuit of their goals in a playful and thought provoking. In this way, the children can implicitly practice an early financial education methodology that aims to improve their perceptiveness regarding financial scope in their adult life.

Keywords: Application, Financial Education, Gamification.

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS.....	15
1.1.1 Objetivo Geral.....	15
1.1.2 Objetivos Específicos.....	15
1.2 DESCRIÇÃO DO DOCUMENTO.....	16
2 REFERENCIAL TEÓRICO.....	17
2.1 EDUCAÇÃO FINANCEIRA.....	17
2.2 EDUCAÇÃO FINANCEIRA INFANTOJUVENIL.....	18
2.2.1 Alfabetização Financeira Infantil.....	18
2.3 METODOLOGIAS PARA EDUCAR FINANCEIRAMENTE NA INFÂNCIA.....	19
2.3.1 Metodologia DSOP.....	20
2.4 GAMIFICAÇÃO NO PROCESSO EDUCATIVO.....	20
2.5 DESENVOLVIMENTO MÓVEL.....	21
2.5.1 Web Mobile.....	21
2.5.2 PWA.....	22
2.5.3 Desenvolvimento Nativo.....	23
2.5.3.1 Plataformas para desenvolvimento nativo.....	23
2.5.4 Desenvolvimento Híbrido.....	24
2.5.4.1 Plataformas para desenvolvimento híbrido.....	25
2.6 TECNOLOGIAS UTILIZADAS.....	26
2.6.1 Tecnologias Backend.....	27
2.6.1.1 Node.js.....	27
2.6.1.2 Firebase.....	27
2.6.2 Tecnologias Front-end.....	28
2.6.2.1 Ionic Framework.....	29
2.6.2.2 AngularJS.....	30
2.6.2.3 Apache Cordova.....	31
3 TRABALHOS RELACIONADOS.....	32
4 METODOLOGIA.....	34
4.1 DESCRIÇÃO DO FUNCIONAMENTO IDEAL DO APLICATIVO.....	34

4.2 MODELAGEM DO APLICATIVO.....	36
4.2.1 Diagrama de Casos de Uso.....	36
4.2.2 Descrição dos Casos de Uso.....	37
4.2.3 Diagrama de Atividades.....	37
4.2.4 Diagrama de Classes.....	42
4.2.5 Diagrama de Implantação.....	43
4.3 AVALIAÇÃO DO APLICATIVO.....	44
5 DESENVOLVIMENTO DO PROTÓTIPO DA APLICACÃO.....	45
5.1 OBJETIVO DO PROTÓTIPO.....	45
5.2 FUNCIONALIDADES IMPLEMENTADAS.....	46
5.2.1 Autenticação dos usuários.....	46
5.2.2 Persistência dos dados.....	46
5.2.3 Usabilidade da aplicação.....	47
5.2.4 Mecanismo de validação do aplicativo.....	47
5.3 CONFIGURAÇÃO DO AMBIENTE.....	47
5.4 ESTRUTURA DO BAAS.....	51
5.5 DESENVOLVIMENTO FRONT-END.....	53
5.5.1 INTEGRAÇÃO DO FRONT-END COM O FIREBASE.....	55
5.6 CONSIDERAÇÕES FINAIS SOBRE O DESENVOLVIMENTO.....	57
6 RESULTADOS.....	59
6.1 TESTES EM UM DISPOSITIVO MÓVEL.....	59
7 AVALIAÇÃO EXTERNA DO APLICATIVO.....	69
8 CONSIDERAÇÕES FINAIS.....	72
9 REFERÊNCIAS.....	75
APÊNDICES	79

1 INTRODUÇÃO

A educação financeira trata-se de uma metodologia que orienta os consumidores a aprimorarem a sua relação com os recursos financeiros. Desta forma, objetiva-se compor o entendimento necessário para gerar consciência sobre oportunidades e riscos econômicos. Assim, as pessoas compreendem que o uso correto do dinheiro pode trazer melhorias em todos os sentidos de suas vidas.

Ao contrário disso, diante dos dados apresentados pela Confederação Nacional do Comércio, o percentual de famílias brasileiras com dívidas fechou o ano de 2017 em 62,2%, acima dos 59% do ano de 2016 (CNC, 2018). Pode-se verificar que o endividamento da sociedade tem crescido progressivamente e isso ocorre em função dos mais diversos fatores, tais como desemprego, falta de planejamento, facilidades para aquisição de crédito, mas, principalmente, no que se refere à falta de instrução financeira entre os indivíduos (PRADO, 2016).

De certo, verifica-se a necessidade de introduzir a temática na vida das pessoas, sobretudo abordando os mais jovens para que se familiarizem cada vez mais cedo com as ideias de domínio de dinheiro. Conforme KASSARDJIAN apud D'AQUINO (2013) o tema sobre educação financeira é algo relativamente novo no Brasil, sobretudo se voltada para o público jovem. Logo, uma possível maneira de instigar a relação com os ganhos e despesas monetárias é tratar do assunto no seio familiar, de forma integrada entre pais e filhos, tornando mais acessível e proveitoso o aprendizado de tal assunto.

Ademais, outra possível ferramenta de ensino é utilizar-se da tecnologia dos dispositivos móveis, que tornou-se algo trivial ao dia-a-dia dos indivíduos. É o que se pode observar nos dados divulgados na 29ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas em maio de 2018, realizada pela Fundação Getúlio Vargas de São Paulo (FGV-SP), onde estima-se que existam cerca de 220 milhões de smartphones ativos no Brasil. Superando a marca de mais de um celular inteligente por pessoa no país, isto sem contar outros tipos de dispositivos portáteis como *Tablets* e *Notebooks*, ao qual aproxima-se ao impressionante número de 306 milhões de dispositivos portáteis que estão em uso (MEIRELLES, 2018).

Com efeito, diante das ideias supracitadas, chegou-se ao entendimento de que a melhor forma de prender a atenção dos jovens ao tema é utilizar-se dos métodos de

gamificação, aproveitando-se da grande utilização da tecnologia pelas famílias como forma ferramenta alternativa de aprendizado frente ao processo educativo formal. Para tanto, propõe-se aqui o desenvolvimento de uma aplicação móvel de controle de tarefas ao qual se busca integrar dispositivos dos pais e filhos, onde os filhos delimitam os objetivos que pretendem alcançar em sua vida no aplicativo e os pais os viabilizam através do envio de tarefas. Compreende-se que esta é uma maneira para aproximar o filhos de suas metas, utilizando-se de atividades lúdicas que engajem os jovens na busca de tais intenções. Logo, estes podem praticar métodos de educação financeira de forma empírica. Portanto, pode-se utilizar da gama de potencialidades da tecnologia e assim propagar mecanismos de educação financeira para o público infantojuvenil.

1.1 OBJETIVOS

Nesta seção serão apresentados os objetivos gerais e específicos do projeto.

1.1.1 Objetivo Geral

Desenvolver um aplicativo para dispositivos móveis de controle de tarefas que estimule preceitos da educação financeira entre o público infantojuvenil.

1.1.2 Objetivos Específicos

- Integrar os dispositivos dos pais com o do seus filhos, onde os pais possam criar e enviar tarefas informando o grau de dificuldade delas para seus filhos, e que os jovens possam recebê-las e executá-las;
- Disponibilizar na aplicação uma espécie de bonificação virtual para que a cada tarefa realizada os jovens recebam no seu aplicativo uma forma de recompensa, utilizando-se deste mecanismo para aplicar conceitos de gamificação;
- Construir um mecanismo de controle para que os jovens recebam a bonificação quando as tarefas, de fato, tenham sido realizadas.

1.2 DESCRIÇÃO DO DOCUMENTO

O presente trabalho encontra-se organizado da seguinte maneira: no Capítulo 2 apresenta-se o referencial teórico sobre os conceitos que são abordados neste projeto; no Capítulo 3 são levantados alguns trabalhos relacionados que têm por objetivo disseminar a aplicação financeira para os jovens através de aplicações móveis; no Capítulo 4 é demonstrada a metodologia utilizada para desenvolver a o protótipo que servirá como critério de avaliação da aplicativo; no Capítulo 5 descreve-se como ocorreu o processo de desenvolvimento do protótipo; no Capítulo 6 apresenta-se os resultados obtidos a partir dos testes preliminares que foram realizados; no capítulo 7 é apresentado os resultados da avaliação externa feita por possíveis usuários sobre o funcionamento do aplicativo; e por fim, no Capítulo 8 são descritas as considerações finais sobre o panorama do projeto desenvolvido.

2 REFERENCIAL TEÓRICO

No decorrer deste capítulo serão abordados alguns dos principais conceitos e informações necessários para a elaboração do projeto, sendo subdividida em duas partes, onde primeiramente serão destacados algumas das definições, conceitos e metodologias que são inerentes à educação financeira. Posteriormente, serão introduzidas algumas das tecnologias que foram utilizadas no decorrer do processo de desenvolvimento do projeto.

2.1 EDUCAÇÃO FINANCEIRA

O conceito de educação financeira é bem abrangente, não restringindo-se apenas a aspectos que englobam termos financeiros. De acordo com SCOLARI (2014, p.20), "deve-se ter em mente os vários aspectos que estão ligados ao tema, como ética e dinheiro, consumo consciente, altas taxas de produção de lixo, impacto ambiental, exercício de cidadania e sustentabilidade". Não se limitando apenas ao ledor engano cometido por parte da população que acredita que educação financeira seja uma busca contínua por maior riqueza e fortuna (KASSARDJAN apud MODERNELL, 2013).

A Organização para a Cooperação e Desenvolvimento Econômico (OCDE), explana o tema de maneira formal e objetiva, explicando a diretriz principal dos seus objetivos:

A educação financeira é o processo pelo qual consumidores e investidores melhoram sua compreensão sobre conceitos e produtos financeiros e, por meio de informação, instrução e orientação objetiva, desenvolvem habilidades e adquirem confiança para se tornarem mais conscientes das oportunidades e dos riscos financeiros, para fazerem escolhas bem informadas e saberem onde procurar ajuda ao adotarem outras ações efetivas que melhorem o seu bem-estar e sua proteção. (OCDE, 2005, p.13, tradução nossa).

Observa-se a partir do conceito abordado acima, a relevância que esta temática pode ter na vida das pessoas, pois proporciona maior discernimento das responsabilidades que o âmbito financeiro implica na vida destes indivíduos, possibilitando a evolução para que as pessoas tenham maior bem-estar e segurança no seu respectivo meio de convívio social.

2.2 EDUCAÇÃO FINANCEIRA INFANTOJUVENIL

De acordo com KASSARDJIAN apud D'AQUINO (2013), a principal função de abordar a educação financeira entre os jovens é possibilitar que se crie raízes e que na vida adulta as crianças possam utilizar o dinheiro de forma sábia e eficaz. De acordo com a autora, educar financeiramente na infância e juventude serve como uma maneira de ensinar a gerir bem as finanças para quando crescerem e que possibilite direcioná-los para uma conduta ética na sua vida adulta.

A educação financeira abordada no âmbito infantil serve como a limiar de um processo para modificar sociedades financeiramente despreparadas, dando origem a uma geração que desenvolverá uma cultura de utilizar o dinheiro com maior austeridade, e isto irá capacitar os jovens a poupar seus recursos econômicos para que utilizem-os de forma consciente e responsável. (KASSARDJIAN, 2013)

KASSARDJIAN (2013) também relata, que ao introduzir a criança a lidar com dinheiro periodicamente, ajudará estes a se tornarem independentes, logo não será um fator limitante para quando eles se tornarem adultos, e pode-se alcançar este objetivo utilizando metodologias adequadas na utilização da abordagem financeira.

2.2.1 Alfabetização Financeira Infantil

SOUZA (2012) ressalta a importância de se ter uma alfabetização financeira, fazendo uma analogia com o ato de aprender a ler a escrever na infância, sendo que ambos os ensinamentos serão relevantes para a fase de vida adulta dos indivíduos. Conforme a autora, quando adultos, manipulamos dinheiro constantemente, e isto afeta a vida pessoal, sendo a razão da vida profissional das pessoas, entretanto, o que se vê são pessoas, geralmente os jovens, que são incapacitados de gerir suas finanças, que sucubem ao consumismo, sem conseguir planejar seu futuro.

Portanto, observa-se a grande importância de fomentar-se este tipo de estudo desde cedo, para que as crianças e os jovens tenham uma vida financeira estável e bem estruturada para que possam usufruir de um ambiente propício para evoluir e impactar em mudanças positivas na sociedade.

2.3 METODOLOGIAS PARA EDUCAR FINANCEIRAMENTE NA INFÂNCIA

De acordo com SCOLARI (2014), para que seja desenvolvido um efetivo processo de educação financeira infantil deve-se implementar métodos para educar financeiramente, sobretudo, nas escolas de educação básica. Aplicando-se uma didática que faça sentido para os alunos, possibilitando que os mesmos possam relacionar as situações aprendidas na escola com situações do cotidiano. Logo poderá ser explanado a aplicabilidade do conhecimento obtido em sala de aula com outras áreas do conhecimento. Segundo a autora, deve-se ressaltar a importância do papel da escola como formador de ser humano, bem como para sua inserção na sociedade e no mercado de trabalho. Isso fica explicitado no trecho a seguir:

“A abordagem de conteúdos ligados à educação financeira pode capacitar os alunos a entenderem melhor o mundo em que vivem, torná-los cidadãos críticos que conseguem entender as notícias veiculadas através dos meios de comunicação, prepará-los para ingressar no mundo do trabalho, consumir, indagar sobre seus direitos e analisar quais os seus deveres.”(SCOLARI apud OLIVEIRA, 2013 , p.21)

Conforme os referidos autores, um método de ensino eficaz para a educação financeira dos jovens, seria aplicá-los na disciplina de matemática, trabalhando juntamente com a formação dos valores éticos e construção da cidadania no processo educacional. Ademais, fora destacado pelos autores que a verdadeira educação financeira "continua não ocorrendo na maioria das escolas e, quando ocorre, propõe um ensino insatisfatório voltado unicamente ao estudo de conceitos de matemática financeira." (SCOLARI apud OLIVEIRA, 2013, p.22).

Porém, o ensino da educação financeira não precisa, exclusivamente, ser provida pela escola. Diferentes técnicas e métodos existentes para o ensino prático da educação financeira são disseminados através de materiais disponibilizados por livros e pela Internet. Dentre alguns métodos que tem como premissa divulgar os benefícios que são proporcionados pela educação financeira na vida das pessoas, tem-se na metodologia DSOP (Diagnosticar, Sonhar, Orçar e Poupar) como uma das principais no seu respectivo segmento, pois ensina de forma operacional como prover maior controle perante as finanças pessoais. Logo, caracteriza-se como uma forma de viabilizar efetivamente uma melhor gestão financeira, podendo ser praticada desde a infância.

2.3.1 Metodologia DSOP

De acordo com DOMINGOS (2017), criador da metodologia, ela fora desenvolvida para auxiliar as pessoas que por meio de hábitos e comportamentos adequados, sejam estimulados a poupar e conquistar, para que possam realizar seus anseios sem se endividar. Ela consiste em quatro pilares, sendo eles: Diagnosticar, Sonhar, Orçar e Poupar.

Conforme o referido autor, a primeira fase consiste em diagnosticar as finanças pessoais, para que tenham maior controle perante seus gastos, o mesmo sugere anotar todos os gastos durante um mês, para que o indivíduo saiba como gasta seu dinheiro e onde estão os possíveis excessos.

A segunda fase consiste em sonhar. O autor relata que de nada adianta trabalharmos todos os dias, se não for para realizar nossos sonhos; e para atingi-los, é sugerido que se estabeleçam três tipos de sonhos: um a curto prazo - a ser realizado dentro de um ano, outro a médio/longo prazo - a ser realizado no período de dez anos , e mais um de longo prazo - a ser realizado a um período superior a dez anos.

A terceira fase consiste em orçar. Conforme o autor, devemos tirar os sonhos do papel para saber quanto eles custam e quanto tempo será necessário economizar para realizá-lo, priorizando o sonho no orçamento mensal.

Por fim, a última fase consiste em poupar, o autor pauta que é insignificante traçar um planejamento, se não forem feitas as devidas reservas monetárias para realizar os sonhos, é sugerido que antes de comprar alguma coisa, compare-se os preços, também que tente-se reduzir as despesas fixas, para que não se tenham gastos excessivos.

2.4 GAMIFICAÇÃO NO PROCESSO EDUCATIVO

Segundo VIANNA et al. 2013, a gamificação corresponde em transformar os mecanismos que são utilizados nos jogos para que possibilite resolução de problemas práticos ou fomentar o envolvimento de um público específico. De acordo com os autores, esse tipo de técnica está sendo cada vez mais utilizada por empresas e entidades de diversos segmentos, de forma facultativa aos métodos tradicionais, principalmente, para despertar nas pessoas determinados

comportamentos, a habituá-las com novas tecnologias, bem como acelerar o processo de aprendizagem e transformar atividades consideradas tediosas em tarefas agradáveis e estimulantes.

Dentre umas das principais características da gamificação, pode-se verificar que geralmente possuem um sistema de recompensas como ocorre nos jogos, LEITE e SOUZA (apud SINGH, 2016, p.3) ressaltam que tal mecanismo tem no seu âmago uma forma "de aumentar a lealdade, motivação e satisfação de quem as recebe". Logo, pode-se salientar que uma implementação correta e efetiva de tal artifício pode melhorar o hábito comportamental das pessoas, sobretudo, implicando em mudanças gradativas para o desenvolvimento pessoal de cada indivíduo.

Dita a contextualização da educação financeira, bem como o cenário da educação financeira aplicada ao público infantojuvenil, além das metodologias que existem para educar financeiramente e descrever como é utilizada a gamificação no processo educativo, agora a seção subsequente abordará as principais maneiras que podem ser aplicadas para desenvolver aplicações móveis, bem como as respectivas plataformas que são utilizadas para desenvolvimento destas.

2.5 DESENVOLVIMENTO MÓVEL

O desenvolvimento móvel pode ser aplicado através de diferentes métodos, sendo as principais maneiras: *web mobile*, *PWA*, app nativo ou app híbrido, sendo que em cada tipo de abordagem existem características peculiares, bem como o respectivo tipo de experiência que pretendem proporcionar ao usuário. A seguir, serão detalhados os diferentes tipos de abordagens utilizadas no desenvolvimento de aplicações móveis.

2.5.1 Web Mobile

Trata-se de sites que são adaptados visando responder melhor a uma interface *mobile*, seja um *tablet* ou *smartphone*. São desenvolvidas com tecnologias *web*, tais como: HTML5, CSS, Javascript; que são executadas no navegador web do dispositivo móvel do usuário, sendo que quando o usuário acessa uma aplicação *web mobile*, o aparelho do usuário já carrega o site automaticamente para a versão *mobile* (BARAZZETTI, 2015).

Dentre os principais pontos positivos deste tipo de desenvolvimento *mobile*, pode-se destacar: aplicação pode rodar em múltiplas plataformas; atualização rápida e abrangente, pois não é preciso passar pelas lojas oficiais da plataforma do dispositivo ou ter que esperar o usuário baixar uma versão nova (SILVA, 2012).

E, como ponto negativo, pode ser destacado que por atender a várias plataformas *mobiles* diferentes, a UX da aplicação não terá o tom característico do dispositivo (SILVA, 2012).

2.5.2 PWA

Progressive Web Apps é uma forma de desenvolvimento móvel que mescla o melhor das aplicações *Web mobile* e o melhor dos aplicativos móveis (JUSTEN, 2018). Este tipo de aplicação é um website que utiliza as capacidades dos navegadores modernos para que à medida que o usuário for interagindo com ele, este se torne "progressivamente um aplicativo" e entregue ao usuário uma experiência muito próxima do que se obtém ao utilizar aplicativos nativos (TEIXEIRA, 2016).

Este tipo de aplicação oferece algumas vantagens em relação às demais formas de desenvolvimento móvel, dentre elas pode ser destacado as seguintes características: o usuário não precisa baixar o aplicativo das lojas oficiais para utilizá-lo; responsivo, pois se encaixa facilmente em qualquer resolução de tela; independente de conexão, pois como utiliza a tecnologia de Service Workers o aplicativo pode funcionar até quando o usuário está *offline*; interações com o usuário tão avançada quanto a de aplicativos nativos; atualizado, pois o usuário não precisa baixar uma atualização do aplicativo frequentemente, logo como a aplicação está na *Web*, sempre que o usuário abre o aplicativo a versão atualizada já está lá; seguro, pois o conteúdo do aplicativo é disponibilizado com o protocolo TLS; instalável, pois o *website* pode ser adicionado pelo usuário à *home screen* do celular; linkável, pois são mais fáceis de compartilhar conteúdo ao enviar o *link* para alguém, dentre outras características (TEIXEIRA, 2016).

Já como desvantagens desta abordagem de desenvolvimento *Mobile*, pode ser ressaltado: não possui controle total sobre o hardware do dispositivo; alguns navegadores ainda não suportam funcionalidades utilizadas pelas PWAs (LIMA, 2017).

2.5.3 Desenvolvimento Nativo

É o método tradicional de desenvolvimento mobile, onde é criado um aplicativo para cada sistema operacional no qual o aplicativo será lançado. Se forem lançados aplicativos para IOS e Android, serão dois aplicativos diferentes, por exemplo. Para cada uma dessas aplicações será criado com a SDK, IDE e os recursos próprios de cada sistema operacional. (SCUDERO, 2017).

As principais vantagens de desenvolver nativamente para os dispositivos móveis destas plataformas são as seguintes: proporcionar uma melhor experiência de usuário com a característica do sistema operacional do aplicativo; desenvolver utilizando SDKs e IDE do próprio sistema operacional, retirando o máximo do sistema operacional (SCUDERO, 2017).

Já as desvantagens de se desenvolver nativamente, pode-se destacar: a necessidade de equipes especializadas para o desenvolvimento para o sistema operacional em questão; custo elevada para manter mais de uma equipe de desenvolvimento; maior tempo dedicado na fase de desenvolvimento e de *deploy* do aplicativo (SCUDERO, 2017).

2.5.3.1 Plataformas para desenvolvimento nativo

Atualmente, existem diferentes tipos de plataformas para o desenvolvimento mobile nativo, cada uma direcionada ao respectivo sistema operacional que pretende ser disponibilizada a aplicação, bem como oferece diversas funcionalidades que auxiliam o desenvolvedor a criar sua aplicação *mobile*.

A principal plataforma de desenvolvimento *mobile* disponível para construir aplicações nativamente para o Android é a IDE Android Studio. Nela, o desenvolvedor pode desenvolver aplicativos utilizando a linguagem de programação Java podendo utilizar recursos como: sistema de compilação flexível no Gradle; emulador com inúmeros recursos; ambiente unificado para desenvolver para todos os dispositivos Android; Instant Run para aplicar alterações a aplicativos em execução sem precisar compilar um novo APK ferramentas e estruturas para realização de testes; ferramentas de verificação de código suspeito para detectar problemas de desempenho, usabilidade, compatibilidade com versões e outros; além de diversos outros recursos que são oferecidos (ANDROID, 2018).

Além disso, a principal plataforma para desenvolver aplicações para o IOS da Apple é a IDE Xcode. Através desta pode-se construir aplicações utilizando as linguagens de programação Swift ou Objective-C que possibilita executar nos dispositivos da Apple. Dentre os principais recursos oferecidos aos desenvolvedores que utilizam esta plataforma de desenvolvimento integrado, destacam-se: o editor assistente que mostra quais arquivos é mais útil para o desenvolvedor com base no trabalho que ele está desenvolvendo; o construtor de interface que auxilia o desenvolvedor no processo de construção de interfaces de usuário sem que ele precise escrever código; editor de controle de versões e controle de origem que permite o desenvolvedor comparar as versões de um arquivo, ver os logs das modificações, verificar quem fez alteração no código, dentre outras funcionalidades; auxilia o desenvolvimento de testes, sendo eles unitários ou um grupo de testes; além de customização para agilizar o fluxo de trabalho, dentre outras funcionalidades e recursos disponibilizados por esta plataforma (APPLE, 2018).

Ademais, existem outras IDEs que auxiliam a desenvolvimento *mobile* nativo para outras plataformas, como o Visual Studio utilizado para a construção de aplicações para Windows Phone, embora já não haja mais a produção de aparelhos com o sistema operacional do Windows pois ela foi descontinuada (BELIZÁRIO, 2018), mas estas não serão objetos de estudo deste projeto de pesquisa.

2.5.4 Desenvolvimento Híbrido

Neste tipo de desenvolvimento cria-se o aplicativo utilizando uma única codificação para o desenvolvimento e como resultado final são compilados aplicativos para plataformas diferentes, que poderão ser executadas em sistemas operacionais como o IOS e o Android, por exemplo. Esse mecanismo é possível devido a ferramentas e frameworks de desenvolvimento que utilizam linguagens Web como HTML5, CSS3 e Javascript (ou outras linguagens de programação). Os principais sistemas operacionais presentes no mercado mobile suportam essas linguagens e suas APIs (SCUDERO, 2017).

Dentre as principais vantagens de desenvolver aplicativos híbridos, podem ser destacadas as seguintes características: permite o desenvolvimento de aplicativos para diversas plataformas mobile, a partir de um único código; não demanda a necessidade de equipes de desenvolvimento específicas para sistemas

e plataformas diferentes; diminui o tempo de desenvolvimento, pois o aplicativo é desenvolvido para várias plataformas simultaneamente; custos diminuem, pois geralmente a equipe de desenvolvimento é mais enxuta (SCUDERO, 2017).

A possível desvantagem de criar aplicativos utilizando esta forma de desenvolvimento remete ao fato que ainda há algumas lacunas que prejudicam a experiência do usuário no seu sistema operacional. Embora os *frameworks* tenham evoluído e se aproximem das características gráficas e nativas de cada plataforma mobile (SCUDERO, 2017).

2.5.4.1 Plataformas para desenvolvimento híbrido

As plataformas para desenvolvimento híbrido têm crescido no últimos anos, pois estas possibilitam aumentar o nível de produtividade na equipe, além de diminuir o tempo despendido em na etapa de desenvolvimento no projeto *mobile*. Embora existam diversas plataformas distintas, irão ser abordados os seguintes frameworks nesta seção: Apache Cordova, Ionic 3, React Native e Xamarin.

O Apache Cordova torna o código produzido através dele portátil para múltiplas plataformas, ele trabalha mesclando os componentes nativos com componente *WebView* de cada dispositivo. Este funciona como um tipo de *browser*, onde possibilita acessar as APIs a nível de dispositivo, então ele permite acesso a recursos nativo de cada dispositivo, como acesso a câmera, acelerômetro, geolocalização, dentre outros (CORDOVA, 2018).

O framework Ionic é um SDK *open source* que auxilia os desenvolvedores a construir aplicativos de alta qualidade e desempenho utilizando tecnologias *web*, sendo elas: HTML, CSS e JavaScript. O *framework* foi originalmente construído por Ben Sperry, Adam Bradley e Max Lynch, onde inicialmente eles lançaram uma versão alfa da plataforma em novembro de 2013, depois lançaram uma versão beta 1.0 em março de 2014 e uma versão final 1.0 em maio de 2015. A plataforma é focada principalmente na aparência e na interação de interface de usuário de um aplicativo, logo seu principal objetivo é simplificar o processo de desenvolvimento visual do aplicativo, ou seja, o front-end. Ademais, vale ressaltar que este *framework* não é um substituto do Apache Cordova, mas é um componente que trabalha em conjunto com ele, ou seja, eles têm objetivos distintos, porém que se complementam

para facilitar o processo de desenvolvimento da aplicação (IONICFRAMEWORK, 2018).

O React Native é um *framework* que permite criar aplicativos móveis multiplataforma utilizando apenas JavaScript, ele utiliza o mesmo design do React (framework javascript para construir interfaces de usuário), permitindo construir uma interface rica a partir de componentes declarativos. Através dele, o desenvolvedor não criará um “*aplicativo Web*” ou um “*aplicativo HTML5*”, mas irá construir um aplicativo móvel real que é indistinguível de um aplicativo desenvolvido em Java para Android ou Objective-C para IOS, pois utiliza os mesmo blocos de construção fundamentais da interface do usuário dos aplicativos nativos, ele apenas integra esses blocos de construção utilizando JavaScript e React (FACEBOOK, 2018).

Através do Xamarin pode-se criar aplicativos multiplataformas maximizando a reutilização de código, além de oferecer uma experiência nativa de alta qualidade em todas as principais plataformas móveis: Android, IOS, Windows Phone. Com ele pode-se obter o melhor dos dois mundos: escrever os códigos dos dados de armazenamento e das regras de negócio, e apresentar interfaces de usuário nativas para cada plataforma, tudo isto sendo escrito na linguagem de programação C# (MICROSOFT, 2018).

Dita as principais plataformas de desenvolvimento móvel híbridas, na seção subsequente serão abordadas as tecnologias que foram utilizadas no projeto, bem como justificar o porquê da escolha destas ferramentas.

2.6 TECNOLOGIAS UTILIZADAS

Nesta seção serão apresentadas as tecnologias que foram utilizadas para o desenvolvimento do aplicativo móvel, tanto as que foram usadas no *backend* da aplicação, quanto às tecnologias para o desenvolvimento do *frontend*. Ademais, serão descritos os principais recursos e funcionalidades que cada plataforma provê aos programadores, auxiliando a etapa de desenvolvimento e otimizando o processo de construção do aplicativo.

2.6.1 Tecnologias Backend

As tecnologias utilizadas na camada de *backend* da aplicação foram o Node.js e o Firebase. A seguir, serão descritos os principais recursos que são ofertados nestas plataformas, bem como os benefícios que elas proporcionam aos desenvolvedores que utilizam as suas funcionalidades.

2.6.1.1 Node.js

O Node.js é um interpretador JavaScript que executa em tempo de execução nos servidores, ele foi construído sobre o motor JavaScript V8 do Google Chrome. Ele utiliza um modelo de entrada e saída sem bloqueio orientado a eventos que o torna leve e eficiente (NODEJS, 2018).

Dentro do pacote que contém toda a infraestrutura da plataforma está incluso o NPM (Node Package Manager), que é considerado o maior ecossistema de bibliotecas de código aberto do mundo (NODEJS, 2018). Basicamente, ele é um repositório online para publicação de projetos de código aberto. Ademais, ele é utilizado dentro dos projetos que utilizam o Node.js para auxiliar na instalação de pacotes, gerenciamento de versão e gerenciamento de dependências, sendo que estas funcionalidades são obtidas via linha de comando no terminal de cada sistema operacional, sendo que este interage com o repositório *online* do NPM (NODEBR, 2016).

2.6.1.2 Firebase

O Firebase é uma plataforma BaaS (Backend as a Service) que provê toda a infraestrutura *backend* para o desenvolvimento do aplicativo. Com ele, pode-se desenvolver para as plataformas IOS, Android ou Web, e permite o desenvolvedor focar apenas na camada visual da aplicação. Dentre as principais funcionalidades disponibilizadas pela plataforma, destacam-se: ferramentas de análises, banco de dados, mensagens, relatórios de erros, escalabilidade automática da aplicação, segurança e integração entre os produtos ofertados pela plataforma. Os projetos desenvolvidos no Firebase utilizam o Google Cloud Platform, que é a o serviço de

computação em nuvem do Google, e permite que a aplicação cresça até a escala de bilhões de usuários (FIREBASE, 2018).

O ecossistema de produtos do Firebase subdivide-se em dois grupos, sendo um deles direcionado ao desenvolvimento e testes do aplicativo, e outro grupo voltado à expansão e envolvimento com o público-alvo da aplicação. Os produtos disponibilizados para o desenvolvimento e testes dos aplicativos são os seguintes: Realtime Database, Crashlytics, Cloud Firestore, Authentication, Cloud Functions, Cloud Storage, Hosting, Test Lab para Android e Monitoramento de desempenho. Já os produtos da plataforma Firebase que possibilitam expandir e envolver o público-alvo do aplicativo, destacam-se: Google Analytics, Cloud Messaging, Previsões, Dynamic Links, Configuração remota, Invites, Indexação de Apps, AdMob e AdWords (FIREBASE, 2018). Neste projeto de pesquisa utilizou-se os seguintes produtos da plataforma Firebase: Realtime Database e Authentication.

O Realtime Database do Firebase permite armazenar e sincronizar dados hospedado na nuvem entre os usuários conectados pelos seus dispositivos em tempo real por meio de um banco de dados NoSQL, que é um mecanismo de persistência de dados não relacional (TOTH, 2011). Através dele os dados atualizados são sincronizados entre os dispositivos conectados em milissegundos, e os dados permanecem disponíveis se o aplicativo se desconecta, o que proporciona uma excelente experiência ao usuário, independentemente da conectividade da rede (FIREBASE, 2018).

O Authentication possibilita gerenciar os usuários do aplicativo de maneira simples e segura. O Firebase Auth oferece vários métodos de autenticação, incluindo e-mail, senha e provedores de terceiros, como Google ou Facebook, além de usar o sistema de contas atual de cada usuário diretamente. Pode-se utilizar a interface do usuário de código aberto da plataforma que é personalizável ou o desenvolvedor pode criar a sua (FIREBASE, 2018).

2.6.2 Tecnologias Front-end

Nesta seção serão descritas as tecnologias que utilizou-se para desenvolver a camada visual do aplicativo. A principal plataforma que foi utilizada para o desenvolvimento desta parte da aplicação será o Ionic Framework, entretanto esta

ferramenta possui como dependências o AngularJS e o Apache Cordova, e através da integração destas ferramentas obteve-se todos os recursos necessários para que o aplicativo funcionasse de forma correta.

2.6.2.1 Ionic Framework

É um SDK de código aberto que possibilita aos desenvolvedores construir aplicativos móveis com de alta qualidade e boa performance utilizando tecnologias web, como: HTML, CSS e JavaScript. Seu objetivo é simplificar a parte do desenvolvimento visual do aplicativo, pois possui grande variedade de componentes visuais ricos que auxiliam o desenvolvedor criar aplicativos com aparência elegante e com boa UX (IONICFRAMEWORK, 2018).

O *framework* também oferece uma ferramenta de interface de linha de comando, o Ionic CLI. A ferramenta fornece vários comandos úteis para os desenvolvedores, dentre alguns deles pode-se destacar: instalação e atualização do Ionic, servidor de desenvolvimento integrado, ferramentas de compilação e depuração, além de diversas outras funcionalidades (IONICFRAMEWORK, 2018).

O Ionic funciona através de componentes, que são elementos de interface de usuário reutilizáveis que servem como blocos de construção para o aplicativo móvel. Estes componentes são compostos de HTML, CSS e, às vezes, JavaScript. Todo componente do Ionic se adapta à plataforma na qual o aplicativo construído será executado (IONICFRAMEWORK, 2018).

A plataforma oferece uma estrutura de temas pré-definidos, estes são um conjunto de estilos que são aplicados ao aplicativo. O Ionic utiliza um tema padrão, mas este pode ser modificado e personalizável. Além da temática, a plataforma permite que os componentes tenham estilos específicos de cada sistema operacional. Isso significa que os estilos do aplicativo serão alterados com base na plataforma (IOS, Android, etc.) na qual está sendo exibido, assim oferecendo aos usuários uma experiência com a qual eles estão familiarizados (IONICFRAMEWORK, 2018).

2.6.2.2 AngularJS

É um *framework* para desenvolvimento *frontend* que utiliza HTML, CSS e TypeScript, este implementa funcionalidades do ES6 e permite o programador utilizar recursos como tipagens de variáveis e Orientação à Objetos similar a linguagens de programação Java ou C#, e quando este é compilado torna-se um arquivo JavaScript. Conforme GUEDES (2017), através deste *framework* pode-se construir aplicações reunindo trechos de código HTML que são denominados templates, juntamente com suas tags customizadas e classes definidas nos arquivos TypeScript que são marcadas com o decorador *@Component* para controlar o conteúdo que é exibido no template.

A ferramenta utiliza-se do conceito de SPA (Single Page Applications), que “são aplicações desenvolvidas em JavaScript que rodam quase que inteiras no lado do cliente (browser)” (GUEDES, 2017, p.3). Conforme o referido autor, quando o usuário acessa a aplicação o conteúdo ficará armazenado no navegador do usuário em forma de templates, e as requisições ao servidor acontecem apenas para buscar dados brutos de conteúdo que geralmente são enviados via JSON, então obtêm-se grandes benefícios ao utilizar este mecanismo, sendo os principais: performance, diminuição da carga de trabalho no servidor e melhor experiência de usuário.

Os componentes do AngularJS possuem como principal característica o conceito de modularização. Segundo GUEDES (2017), modularização trata-se de uma maneira de desenvolver a aplicação fragmentando-a em pequenas partes para que seja facilitado a forma de construir, testar e reutilizar os códigos do sistema. A principal vantagem de construir um sistema modularizado com AngularJS é reunir estes pequenos fragmentos para criar uma página completa. Ademais, através desta separação dos componentes, cada trecho de código da aplicação tem a sua responsabilidade, então simplifica-se os testes e as manutenções do sistema.

Logo, o Angular é uma estrutura subjacente que trabalha junto com o Ionic, por isto ele é uma dependência do framework. Ele é o responsável pela API de componentes que é utilizado como bloco de construção do Ionic, assim pode-se criar aplicativos modularizados através da integração dessas duas ferramentas (IONICFRAMEWORK, 2018).

2.6.2.3 Apache Cordova

O Apache Cordova é uma estrutura de desenvolvimento móvel de código aberto. Ele é uma ferramenta de desenvolvimento multiplataforma que utiliza linguagens *web*, como HTML5, CSS3 e JavaScript, evitando assim que utilize-se as linguagens nativas de cada plataforma *mobile*. A principal característica dessa plataforma é sua capacidade de acessar recursos nativos dos dispositivos móveis (CORDOVA, 2018).

Os aplicativos desenvolvidos com o Apache Cordova baseiam-se em um arquivo *config.xml* que fornece as informações sobre o aplicativo e especifica os parâmetros necessários para que ele funcione corretamente (CORDOVA, 2018).

Portanto, a integralização entre o Apache Cordova e o Ionic proporciona um conjunto completo de ferramentas para desenvolver aplicativos multiplataforma, pois o Cordova acessa os recursos a nível de *hardware* de cada dispositivo móvel, já o Ionic disponibiliza uma grande variedade de temas e componentes que permitem desenvolver interfaces da camada visual dos aplicativos com características de interfaces ricas e elegantes, deixando elas muito próximas das aparências de um aplicativo nativo. Assim, o Ionic encontra-se em uma camada superior e o Cordova em um nível mais baixo, e é justamente por isso que o Ionic possui o Cordova como uma de suas dependências, e a união destas duas plataformas provê todos os recursos necessários ao programador para que ele possa desenvolver aplicativos híbridos.

Dita as principais tecnologias que foram utilizadas no desenvolvimento deste projeto, no próximo capítulo serão apresentados os trabalhos relacionados que já se utilizaram de mecanismos similares para introduzir à educação financeira para o público infantojuvenil.

3 TRABALHOS RELACIONADOS

Existem poucos trabalhos desenvolvidos que têm por objetivo utilizar os dispositivos móveis como um meio de fomentar a educação financeira entre o público infantojuvenil, pois como já fora dito na introdução do projeto trata-se de um tema ainda pouco explorado no Brasil, sobretudo quando direcionada a este tipo de público.

O aplicativo para dispositivos móveis Trato desenvolvido pelo Banco do Brasil foi um dos pioneiros ao abordar o tema da educação financeira voltada para as crianças. Lançado no ano de 2016, ele tem por objetivo utilizar-se da gamificação para que os jovens aprendam conceitos de responsabilidade e organização para aprender o valor do dinheiro. O aplicativo se utiliza do cumprimento de tarefas para que os jovens recebam moedas virtuais no jogo, sendo que as tarefas estão divididas nas categorias: Educação, Tarefas Domésticas, Saúde e Bem-Estar e Comportamento (BB, 2016).

Além deste, os poucos que foram desenvolvidos têm obtido destaque na iniciativa privada, como é o caso do aplicativo Origininhos, aplicativo que ganhou o prêmio Hackaton no ano de 2017 promovido pelo Banco Original. A ideia do aplicativo consiste em permitir que os pais determinem as atividades que seus filhos deverão fazer, bem como a recompensa que será recebida pelo seus esforços, onde os pontos acumulados através da realização de tarefas reflete o quanto os jovens podem ganhar no final do mês através de uma conta no banco Original, caso o usuário tenha, e para o público adolescente também há dicas de investimentos, além de ensinamentos para poupar dinheiro (MILLER, 2017).

Ademais, também existe no mercado dos dispositivos móveis o aplicativo miniJOB com funcionalidades similares aos que já foram citados anteriormente. Entretanto, tal aplicativo encontra-se disponível apenas para o sistema operacional móvel da Apple, o IOS. O que difere este aplicativo dos citados anteriormente, é o fato de que a criança pode executar apenas 6 tarefas diárias, sendo que quatro delas são tarefas fixas: tomar banho, arrumar o quarto, lição de casa e a hora de dormir; além de duas tarefas customizáveis que podem ser disponibilizadas de acordo com a realidade e necessidade de cada família. (MINIJOBAPP, 2018).

Apresentados os trabalhos relacionados existentes, no próximo capítulo serão descritas as metodologias que foram utilizados no decorrer do processo de desenvolvimento do aplicativo móvel.

4 METODOLOGIA

Este projeto tem como objetivo principal o desenvolvimento de um protótipo de um aplicativo de controle de tarefas para dispositivos móveis que visa fomentar as diretrizes da educação financeira para o público infantojuvenil através de métodos de gamificação, haja vista que o índice de pessoas inadimplentes elevou-se nos últimos anos e isto acaba acarretando em um desequilíbrio financeiro que afeta todas as áreas do âmbito econômico.

Possivelmente, estes problemas podem ser equacionados a médio/longo prazo introduzindo o tema da educação financeira para os jovens, pois através destes ensinamentos quando chegarem na fase de vida adulta eles estarão preparados para gerir de forma adequada as suas finanças, e isso pode condicioná-los a ter uma melhor qualidade de vida nesta fase de sua existência.

Portanto, neste capítulo serão descritos o funcionamento ideal do sistema e os métodos utilizados na modelagem do aplicativo.

4.1 DESCRIÇÃO DO FUNCIONAMENTO IDEAL DO APLICATIVO

O aplicativo estará subdividido em dois diferentes níveis de acesso: uma parte do aplicativo será direcionada aos pais/responsáveis e o outro nível será viabilizado para os filhos. Inicialmente, os pais devem se cadastrar no sistema e depois realizar login com as credenciais que foram informadas. Quando estiverem logados no aplicativo eles poderão cadastrar seus filhos, informando o nome, email, senha e data de nascimento deles. Os pais também poderão cadastrar as principais tarefas que eles vão disponibilizar, apenas informando a descrição desta, por exemplo: arrumar o quarto.

Após terem cadastrado os filhos e as tarefas, os pais poderão acessar a funcionalidade do aplicativo que permitirá a eles vincularem os filhos às tarefas. Este recurso permitirá designar uma tarefa a um filho, informando o grau de dificuldade da tarefa. Esse nível de dificuldade já terá uma quantidade pré-determinada de moedas virtuais que o filho irá receber caso realize a tarefa corretamente, onde elas serão delimitadas da seguinte forma: tarefa de nível fácil - 25 moedas; tarefa de nível médio - 50 moedas; tarefa de nível difícil - 75 moedas; e tarefa de nível muito difícil - 100 moedas.

Opcionalmente, os pais também poderão estipular um intervalo de tempo para que o filho realize a tarefa, por exemplo: duas horas; e caso, o filho realize a tarefa dentro desse intervalo de tempo pré-definido, sua bonificação de moedas virtuais será duplicada. Após definirem os parâmetros para designar a tarefa para o filho, os pais poderão enviá-la para o seu filho e este receberá uma notificação no seu dispositivo, solicitando que efetue uma tarefa.

O filho poderá entrar no aplicativo através dos dados cadastrados no dispositivos dos seus pais, realizando o login no sistema informando o *e-mail* e senha que seus pais cadastraram. Posteriormente, caso queira, ele poderá modificar a senha de acesso padrão. Após logar no aplicativo, o filho terá acesso a uma funcionalidade que armazenará todas as tarefas que ele recebeu do(a) pai/mãe, e quando ele realizar a tarefa, deve marcar o componente *toggle* da tarefa como verdadeiro, que ativará a respectiva tarefa como efetuada. Quando selecionada esta opção, uma notificação será enviada para o dispositivo do seu responsável. Além disso, o filho também poderá traçar quais são seus objetivos, podendo ser tanto produtos ou sonhos que eles têm, por exemplo: um produto pode ser um tênis ou o sonho de ser um médico(a).

Quando ele for cadastrar o seu objetivo também deverá definir qual o é grau de dificuldade deste, categorizando-os em: objetivo de nível fácil - 250 moedas; objetivo de nível médio - 500 moedas; objetivo de nível difícil - 750 moedas; e objetivo de nível muito difícil - 1000 moedas. Após cadastrar o seu objetivo, bem como o grau de dificuldade, quando o jovem atingir o respectivo objetivo ele será notificado que atingiu sua meta, e isto irá habilitar a funcionalidade de receber conquistas.

Este recurso proverá ao jovem que ele receba uma espécie de troféu virtual, ao qual cada troféu indica o nível de progresso que ele obteve nas conquistas de moedas pela realização das tarefas e também irá habilitar troféus pelo número de tarefas realizadas. A conquista fará parte da galeria de troféus virtuais que já estarão com nomes pré-definidos e cadastradas no banco. Ademais, o aplicativo também disponibilizará uma funcionalidade para que pais e jovens recebam notificações de dicas sobre educação financeira, onde haverá um botão na página inicial que quando clicado será enviado uma dica financeira aleatória que já estará cadastrada no Realtime Database do Firebase.

Vale ressaltar que um aspecto importante para o funcionamento adequado do aplicativo é a questão de controlar a bonificação dos filhos, pois de nada vale o filho marcar a tarefa como realizada apenas para receber a recompensa. Para tanto, quando o jovem for notificado que recebeu uma tarefa e marcar ela como realizada, será enviado novamente uma notificação para seu responsável que irá avaliar se, de fato, a tarefa foi feita. Esta notificação estará disponível na lista de tarefas pendentes que será armazenada no dispositivo do responsável, e quando o pai/mãe verificar visualmente que a tarefa foi realizada pelo seu filho então poderá validar esta tarefa, para que possa mudar o status dela para realizada, e posteriormente esta ela armazenada na lista de tarefas feitas pelo seu filho, bem como o nome de quem efetuou ela e a data da realização.

Através deste aplicativo os pais podem proporcionar ao jovem uma maneira de utilizar os conceitos da gamificação para que seja estimulado de forma agradável e gradativa ao tema da educação financeira na sua vida. Além disso, vale ressaltar que ao introduzir uma lista de objetivos no aplicativo para ele, bem como o seu respectivo nível de dificuldade, pode-se aplicar uma maneira para que ele pratique tacitamente a metodologia DSOP, pois assim ele poderá estipular os seus objetivos e deverá realizar as tarefas para receber recompensas, e dessa forma ele pode atingir todos os seus sonhos e metas no mundo virtual, entretanto isso também servirá futuramente como uma metodologia que ele poderá aplicar no mundo real para atingir seus objetivos na fase de vida adulta.

4.2 MODELAGEM DO APLICATIVO

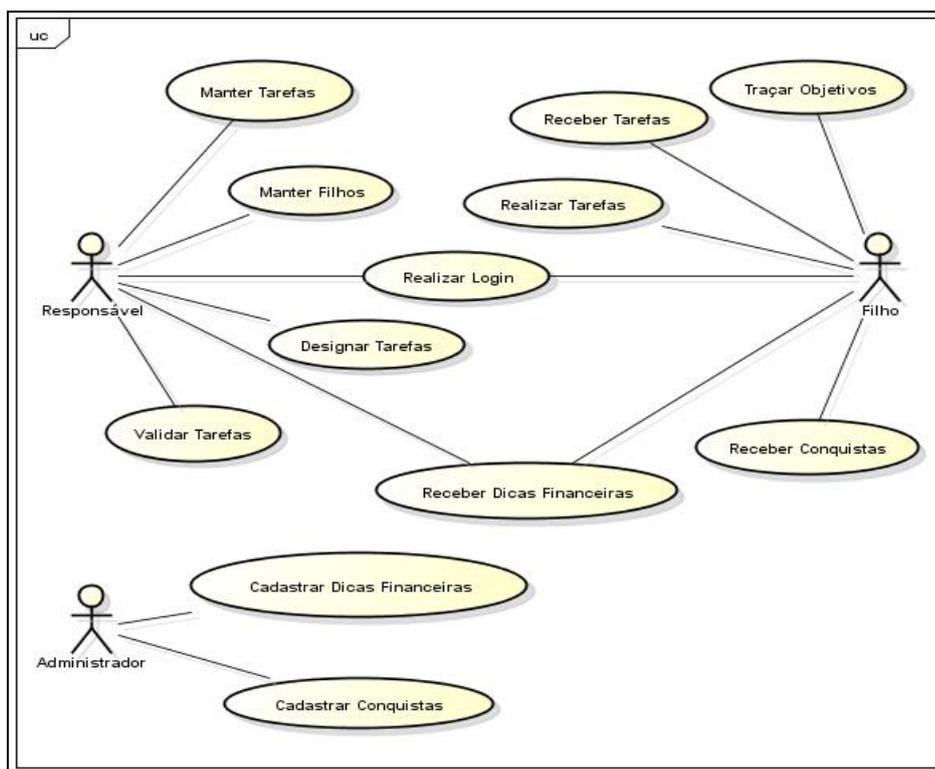
Nesta seção é ilustrado como foi realizada a modelagem da aplicação através de diagramas da UML, demonstrando de forma visual e detalhada como discorre o fluxo de funcionamento de cada funcionalidade existente no sistema móvel.

4.2.1 Diagrama de Casos de Uso

O Diagrama de Casos de Uso tem como principal objetivo demonstrar o fluxo de funcionamento do aplicativo através do ponto de vista do usuário, e isto fornece uma visão geral do funcionamento do sistema, além explicar as principais funcionalidades que o sistema deve disponibilizar para os usuários. O diagrama

apresentado na Figura 1 demonstra as principais funcionalidades que o aplicativo proverá aos usuários que irão utilizar o sistema.

Figura 1: Diagrama de Casos de Uso



Fonte: Do autor

4.2.2 Descrição dos Casos de Uso

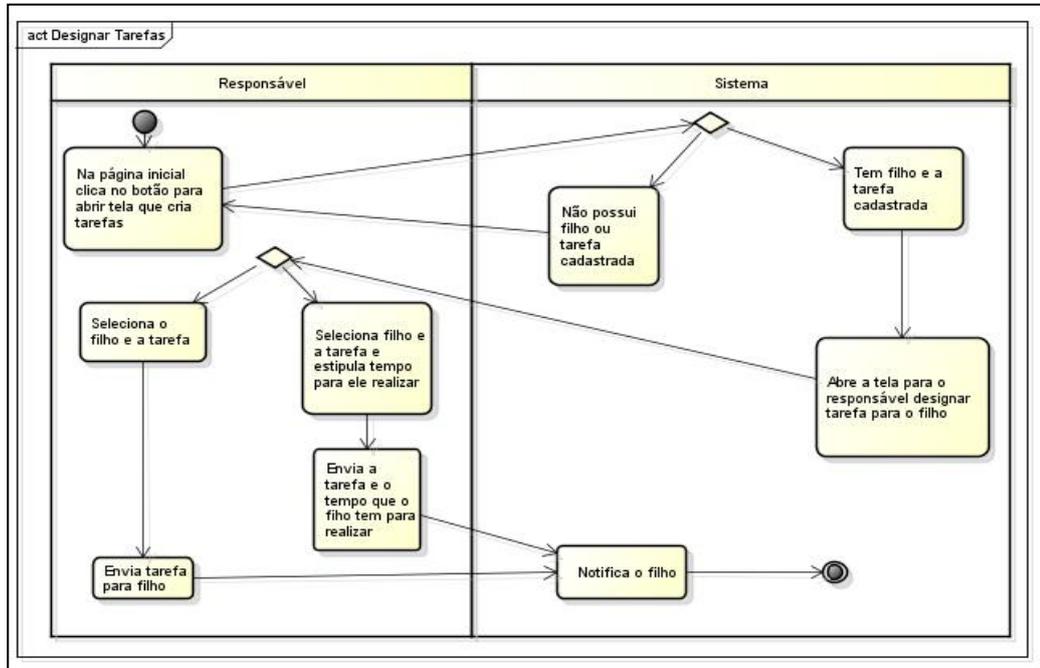
A descrição dos casos de uso tem por objetivo fornecer um panorama detalhado das principais funcionalidades que o sistema fornecerá aos atores através dos casos de uso que foram desenvolvidos no diagrama da Figura 1. No Apêndice A são descritos o funcionamento dos casos de uso: *Designar Tarefa*, *Validar Tarefas*, *Receber Tarefas* e *Realizar Tarefas*.

4.2.3 Diagrama de Atividades

O diagrama de atividades tem por objetivo descrever quais são as etapas necessárias que um determinado processo precisa realizar para que ele atinja o

objetivo que é pretendido através dele. O diagrama da Figura 2 exemplifica como vai funcionar a designação de tarefas por parte do ator Responsável.

Figura 2: Diagrama de Atividade Designar Tarefas

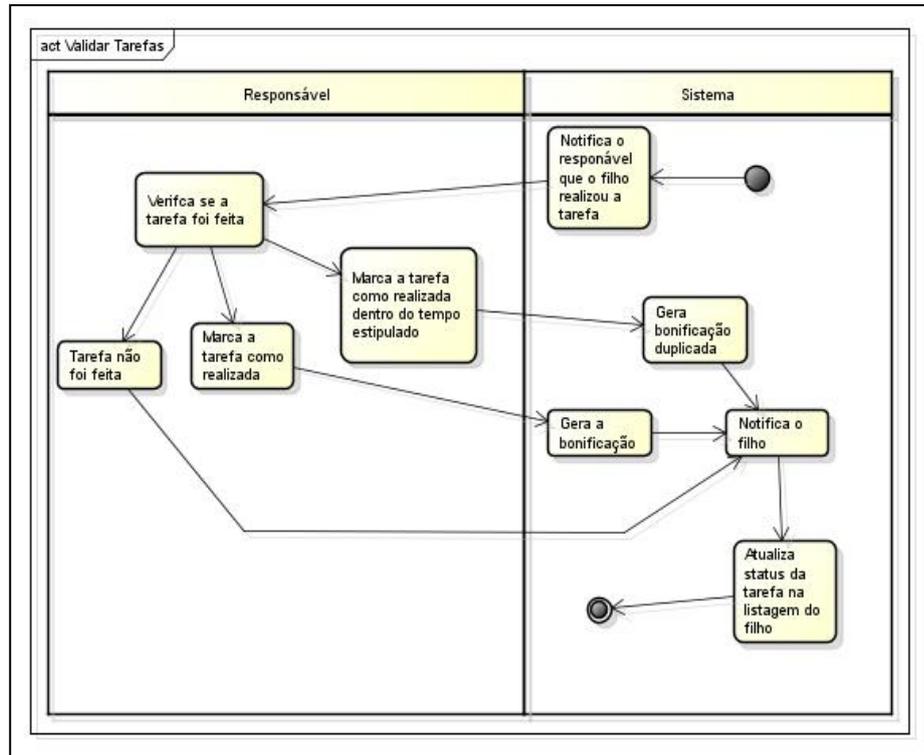


Fonte: Do autor

O diagrama descreve o fluxo que o responsável precisa fazer para designar uma tarefa para o seu filho. Onde, na página inicial poderá ser clicado em um botão que direciona para uma tela que vai habilitar o responsável vincular as tarefas ao seu filho. Caso ainda não tenha cadastrado tarefas ou filhos, o aplicativo redireciona o usuário para a tela inicial novamente e abre um alerta informando que eles devem ser cadastrados primeiramente, senão pode-se selecionar o filho e a tarefa na tela para a qual ele vai ser redirecionado. Nesta tela o usuário poderá simplesmente associar o filho a uma tarefa ou pré-definir um intervalo de tempo para que o filho realize-a. Após definir estes parâmetros o responsável poderá enviar a tarefa para o filho, e posteriormente o aplicativo mandará uma notificação para o dispositivos dos filhos informando que foi definida uma nova tarefa para ser realizada.

O diagrama de atividades da Figura 3 descreve como é o fluxo de funcionamento do caso de uso validar tarefas. Neste caso, o responsável poderá gerar a bonificação da tarefa somente quando ela realmente for realizada. Isto, pode ser verificado através da Figura 3.

Figura 3: Diagrama de Atividade Validar Tarefas

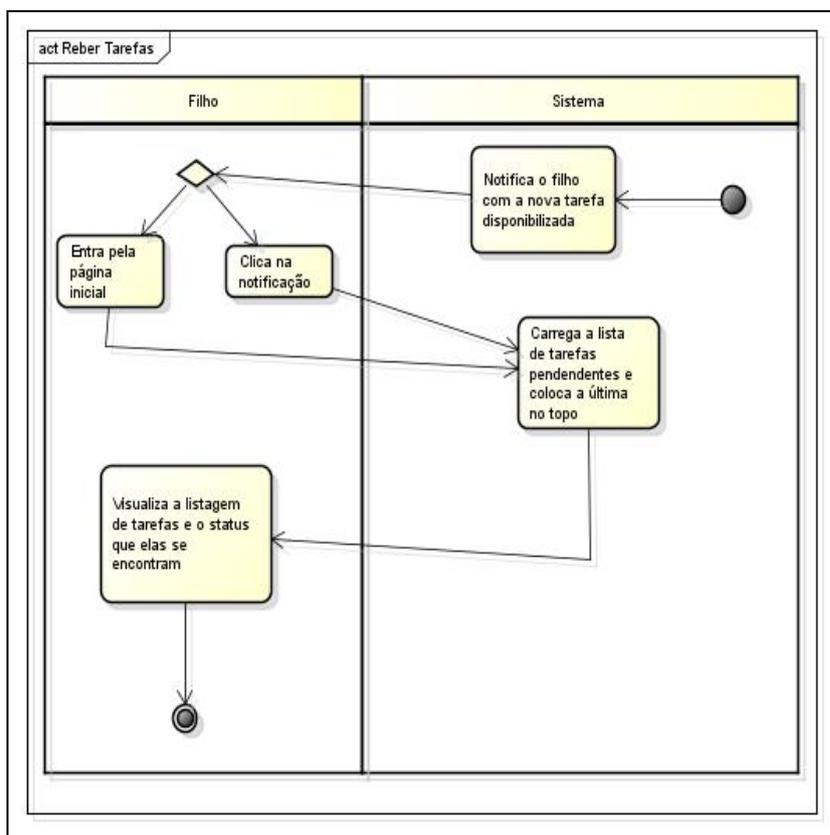


Fonte: Do autor

Este diagrama demonstra como o responsável irá validar as tarefas quando os filhos marcarem-as como realizadas. Inicialmente, o aplicativo envia uma notificação para o responsável informando-o sobre o novo estado da tarefa. Posteriormente, o responsável deverá verificar visualmente se, de fato, a tarefa foi efetuada. Caso a atividade tenha sido realizada dentro do período de tempo pré-definido, o bônus gerado ao filho é duplicado. Se caso seja uma simples tarefa que apenas vincula uma tarefa ao filho, o responsável gera a bonificação padrão. Senão, caso a mesma não tenha sido feita, o filho não recebe as recompensas. Após esta verificação, o sistema notifica o dispositivo do filho com a resposta da validação da tarefa por parte do responsável, e então atualiza o status da lista de tarefas do dispositivo do filho.

O próximo diagrama ilustrado nesta seção, trata-se do fluxo de atividades que representa o caso de uso receber tarefas que está atrelado ao filho. Este diagrama é demonstrado na Figura 4.

Figura 4: Diagrama de Atividade Receber Tarefa

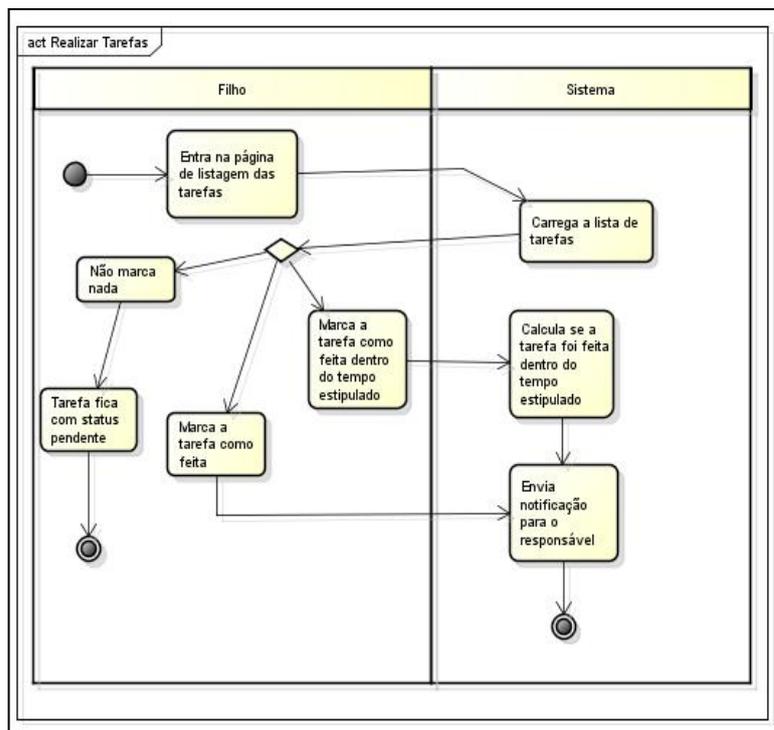


Fonte: Do autor

Este diagrama representa como será arquitetado na modelagem do aplicativo o fluxo de atividades para que o filho receba tarefas. Primeiramente, o dispositivo do filho receberá uma notificação que informa a nova tarefa que seu pai/mãe criou para que ele faça. Então, o filho poderá carregá-la tanto clicando na notificação que recebeu, quanto acessando-a através do componente que estará presente na página inicial do aplicativo. Ambas as formas de acesso vão direcioná-lo para a página que carrega a listagem de tarefas que estão pendentes no dispositivo do filho. Após ser carregada a listagem, o filho poderá visualizar as suas tarefas e o status que elas se encontram, colocando as últimas tarefas disponibilizadas no topo da lista.

Por fim, o diagrama de atividades da Figura 5 representa como será estruturado o processo de funcionamento do caso de uso *Realizar Tarefas* que demonstra como os filhos poderão modificar o *status* da tarefa para realizada, e dessa forma notificar seu responsável que realizaram uma tarefa.

Figura 5: Diagrama de Atividade Realizar Tarefa



Fonte: Do autor

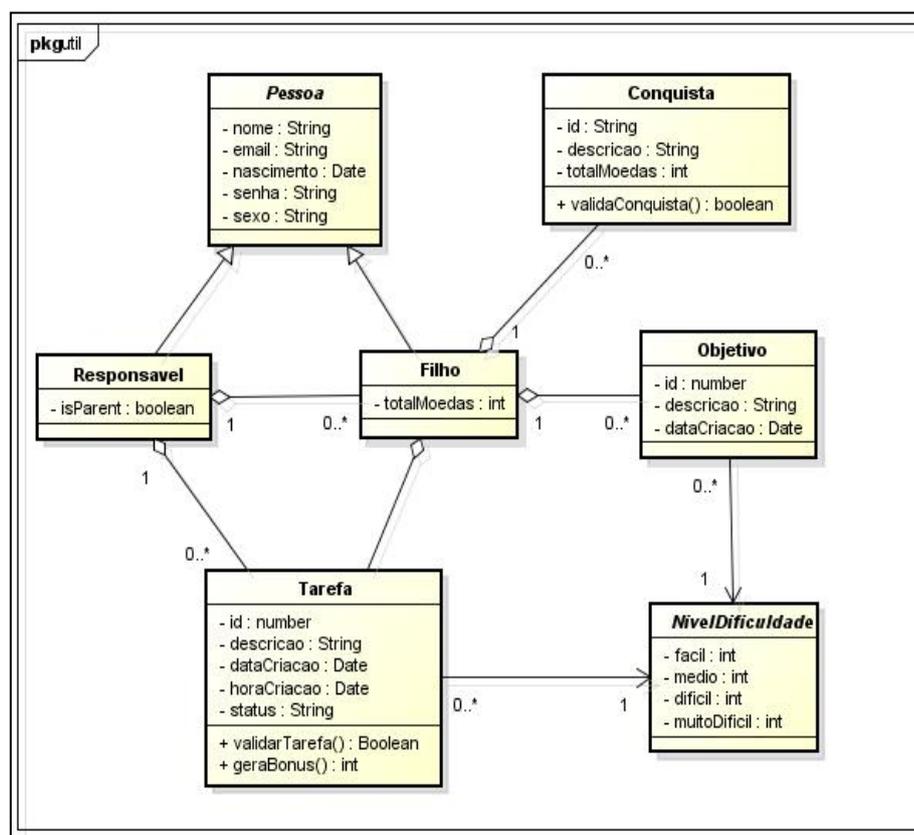
Este diagrama representa o fluxo de atividades que apresenta a forma de como o filho irá notificar o responsável que realizou uma tarefa. Inicialmente, o filho acessa o componente que vai direcioná-lo para a sua lista de tarefas. Depois o sistema carrega a sua listagem de tarefas, e então o filho pode marcá-la como tarefa realizada ou não. Se ele não marcar nada, então a tarefa ficará com o status de pendente. Caso ele marque a tarefa como realizada como dentro do intervalo de tempo estipulado, então o sistema irá calcular se realmente está dentro do período de tempo solicitado e enviar a notificação para o responsável com esta verificação já validada. E caso seja uma tarefa que não tenha intervalo de tempo pré-definido e o filho marcá-la como feita, o sistema envia a notificação para o dispositivo do responsável para que este faça a validação da tarefa antes de gerar o bônus pela tarefa realizada para o filho.

Portanto, dito como será feita a modelagem das principais funcionalidades deste aplicativo através dos diagramas de atividades, na próxima seção será demonstrado o diagrama de classe que será utilizado para o desenvolvimento do protótipo do aplicativo.

4.2.4 Diagrama de Classes

Através do diagrama de classes é possível descrever toda a estrutura do sistema, bem com as classes, atributos, métodos e o relacionamento entre esses componentes. A Figura 6 retrata o diagrama de classes que será utilizado no desenvolvimento do protótipo do aplicativo.

Figura 6: Diagrama de Classes



Fonte: Do autor

A classe *Pessoa* é uma classe abstrata, logo não poderá ser instanciada, e serve apenas para compartilhar os dados que os responsáveis e o filhos têm em comum.

A classe *Responsável* armazena os dados que herda de *Pessoa*. Além disso, terá um atributo para setar que o usuário é um responsável, e terá uma lista de filhos e lista de tarefas.

Já a classe *Filho* armazena as informações que ela herda da classe *Pessoa*, e tem uma lista de tarefas, uma lista de objetivos e uma lista de conquistas.

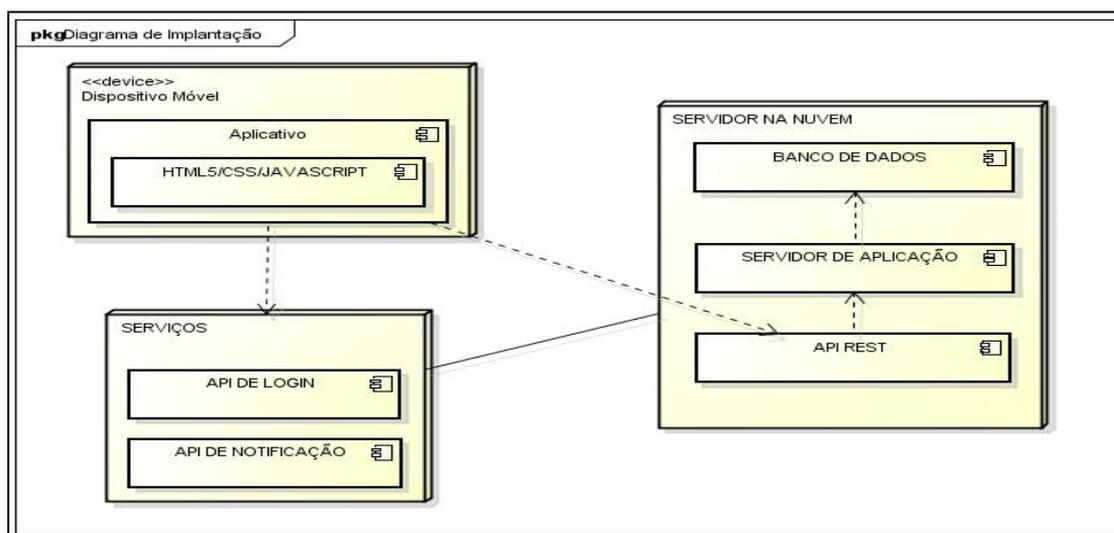
A classe *Tarefa* terá os dados das tarefas e estará associada a um *Filho*, um Responsável, bem como ao nível de dificuldade que representa.

As classes *Conquista* e *Objetivo* estão associadas a um objeto da classe *Filho*. E a classe *NivelDificuldade* estará associada a uma tarefa, e ela também será uma classe abstrata.

4.2.5 Diagrama de Implantação

Os diagramas de implantação demonstram como ocorre o relacionamento entre os componentes de *software* e *hardware* da aplicação. A Figura 7 explana como ocorrerá este relacionamento no aplicativo.

Figura 7: Diagrama de Implantação



Fonte: Do autor

O diagrama acima ilustra que o aplicativo será executado no dispositivo móvel do cliente, e a aplicação será executada utilizando HTML, CSS e JavaScript. O aplicativo vai realizar conexões HTTP com a API REST do servidor na nuvem, e esta será responsável por efetuar a comunicação entre o servidor de aplicação e o banco de dados Firebase. Por fim, o aplicativo também se conecta com os serviços da API de *login* e a API de notificação, sendo que ambos os serviços estão associados ao servidor na nuvem, pois fazem parte do ecossistema dele.

4.3 AVALIAÇÃO DO APLICATIVO

Embora a ideia central do projeto seja o desenvolvimento de um aplicativo, o método de avaliação do projeto delimitou-se a desenvolver um protótipo da aplicação móvel, ao qual contemplará a maioria das funcionalidades supramencionadas neste capítulo. Posteriormente ele será apresentado funcionando para usuários que possuam algum filho que esteja dentro da faixa etária infantojuvenil, e estes em seguida responderão um formulário de avaliação do protótipo para que se tenha uma apreciação dos usuários mediante a uma avaliação externa.

Os dados de avaliação do sistema serão obtidos por meio deste questionário, conforme disponibilizado no Apêndice B.

5 DESENVOLVIMENTO DO PROTÓTIPO DA APLICAÇÃO

Neste capítulo são descritos quais objetivos atingiu-se através do desenvolvimento do protótipo, além de relatar como ocorreu a configuração do ambiente e como foi configurado a estrutura da plataforma BaaS do Firebase. Ademais também descreve-se o como foi realizado o desenvolvimento do *frontend* do aplicativo e como foi realizada a integralização do *frontend* com os serviços do Firebase, além de analisar os resultados dos testes preliminares que foram realizados no sistema móvel.

5.1 OBJETIVO DO PROTÓTIPO

Através do protótipo desenvolvido, buscou-se implementar a maioria das funcionalidades que foram idealizadas a partir da modelagem da aplicação de forma prática, com o intuito de que pudesse ser disponibilizado a primeira versão da aplicação móvel para que os usuários aos quais o aplicativo foi apresentado pudessem avaliá-la funcionando, sobretudo, implantando os recursos que foram ilustrados por meio do diagrama de casos de uso da Figura 1.

Em decorrência disso foi necessário estruturar toda a arquitetura do sistema de forma que cada camada da aplicação contivesse de forma bem clara sua responsabilidade perante funcionamento holístico do sistema. Dessa maneira utilizou-se uma camada modelo onde foram criadas interfaces para que fossem mapeados todos os atributos necessários para que houvesse a persistência correta das informações na base de dados, uma camada de serviço que provê os métodos que realizam a comunicação entre o sistema e a plataforma BaaS do Firebase, fornecendo as operações CRUD do sistema para que haja a manipulação dos dados no banco NoSQL da plataforma.

Ademais também foi necessário criar a camada de visual do aplicativo a qual foi elaborada utilizando o desenvolvimento baseado em componentes disponibilizado através do *framework* Ionic, onde cada componente contém toda a estrutura HTML, CSS e JS atrelado a ele, facilitando as manutenções e reaproveitamento de código na aplicação, e a partir da interface visual do aplicativo o usuário pode aproveitar de todas as funcionalidades que o sistema fornece.

Portanto, após o desenvolvimento do protótipo se obteve a primeira amostragem do funcionamento do aplicativo, onde se pôde realizar os primeiros testes unitários de cada funcionalidade implementada no sistema durante o desenvolvimento, bem como os teste de integração entre os diversos componentes que fazem parte do sistema após o protótipo ter sido finalizado, e dessa forma, detectar as lacunas que podem ser preenchidas como melhorias futuras da aplicação. Além disso, buscou-se desenvolver uma interface visual que permita uma fácil usabilidade do sistema por parte dos usuários, tendo em vista que uma parte do sistema é direcionado ao público infantojuvenil, logo foi preciso elaborar componentes visuais que foram desenvolvidos sendo priorizada a usabilidade da aplicação, para cada funcionalidade existente no sistema que tem como público-alvo estes usuários, e isto foi facilitado devido a utilização do *framework* Ionic.

5.2 FUNCIONALIDADES IMPLEMENTADAS

O protótipo do aplicativo desenvolvido contemplou a maioria das funcionalidades que previamente foram idealizadas a partir de quando foi definido como seria o fluxo de funcionamento ideal do sistema, onde foi descrito na Seção 4.1 do capítulo anterior. A seguir são destacadas as principais funcionalidades implementadas no sistema.

5.2.1 Autenticação dos usuários

Este recurso foi implementado utilizando-se do serviços Authentication que fazem parte da plataforma do Firebase, onde este recurso realiza o gerenciamento dos usuários que estão cadastrados no aplicativo mediante ao *e-mail* e senha que foram registrados inicialmente, onde cada respectivo usuário terá acesso a sua respectiva área restrita, válido tanto para os pais, quanto para os filhos.

5.2.2 Persistência dos dados

A persistência das informações cadastradas na aplicação é intermediada pelo banco de dados Realtime Database do Firebase, onde é realizado a sincronização instantânea e os dados são atualizados em tempo real após cada

operação CRUD realizado pelos usuários do sistema, inclusive funcionando quando o aplicativo está desconectado da rede, pois os dados são atualizados após ocorrer o restabelecimento da conexão.

5.2.3 Usabilidade da aplicação

A Experiência do usuário também foi algo planejado durante todo o processo de desenvolvimento do protótipo, pois como é uma aplicação direcionada aos jovens, logo necessitou-se enfatizar como ocorreria as questões relacionadas às interações do usuário com as funcionalidades do sistema. Então buscou-se desenvolver interfaces de usuário bem intuitivas, a fim de facilitar as ações dos usuários no sistema móvel.

5.2.4 Mecanismo de validação do aplicativo

Ressalta-se principalmente o mecanismo de validação implementado na aplicação, feito tanto para a validação das tarefas para que fosse gerada a bonificação destinada ao filho somente quando estas, de fato, tenham sido realizadas; e também para validação dos objetivos do filho por parte dos pais.

Dessa forma, possibilitou-se viabilizar através do aplicativo uma maneira de os pais estarem plenamente conscientes dos reais anseios por parte de seus filhos, e tendo ciência disso, possam fomentar de forma progressiva e contínua atividades lúdicas para que estimulem seus filhos a realizar as tarefas aos quais irão aproximá-los cada vez mais dos seus objetivos.

5.3 CONFIGURAÇÃO DO AMBIENTE

O aplicativo foi desenvolvido através do editor de código-fonte Visual Studio Code na versão 1.27.2 no sistema operacional Windows 10, ambos da Microsoft. Para o desenvolvimento do frontend da aplicação utilizou-se o Ionic na sua versão 3.9.2, e todo o processo de instalação e configuração inicial do projeto fora realizado seguindo o passo-a-passo que a documentação do *framework* disponibiliza através do seu site oficial, em conjunto com o Ionic está o *framework* Javascript AngularJS que foi utilizado na versão 5.2.11. Ademais também utilizou-se o Node JS na versão

8.9.4 que foi utilizado como gerenciador de dependências de pacotes e como servidor para execução da aplicação durante o seu processo de desenvolvimento.

Para que a aplicação funcionasse da forma correta, utilizou o NPM como gerenciador de pacotes, ferramenta que já vem acoplada ao ecossistema do Node JS, sendo utilizado para armazenar e gerenciar as dependências da aplicação, sendo que esse controle é feito utilizando como referência o arquivo *package.json* que contém todas os pacotes necessários para o funcionamento do sistema. Destaca-se que um dos pacotes utilizados como dependência da aplicação foi o *AngularFire2* que provê métodos para realizar a comunicação entre o sistema e o Firebase. Este foi instalado de forma separada, através do NPM e ficou atrelado às dependências do aplicativo juntamente com as outras dependências que já haviam sido criadas a partir da instalação e configuração do Ionic.

Os pacotes ficam armazenados em um diretório chamado *node_modules* e nele contém todos os módulos que são utilizados de alguma forma como dependência da aplicação. A Figura 8 ilustra como ficou estruturada as dependências do projeto no arquivo *package.json*.

Figura 8: Dependências do aplicativo

```

1 package.json x
2 {
3   "name": "app-firebase",
4   "version": "0.0.1",
5   "author": "Ionic Framework",
6   "homepage": "http://ionicframework.com/",
7   "private": true,
8   "scripts": {
9     "start": "ionic-app-scripts serve",
10    "clean": "ionic-app-scripts clean",
11    "build": "ionic-app-scripts build",
12    "lint": "ionic-app-scripts lint"
13  },
14  "dependencies": {
15    "@angular/animations": "5.2.11",
16    "@angular/common": "5.2.11",
17    "@angular/compiler": "5.2.11",
18    "@angular/compiler-cli": "5.2.11",
19    "@angular/core": "5.2.11",
20    "@angular/forms": "5.2.11",
21    "@angular/http": "5.2.11",
22    "@angular/platform-browser": "5.2.11",
23    "@angular/platform-browser-dynamic": "5.2.11",
24    "@ionic-native/core": "~4.11.0",
25    "@ionic-native/fcm": "^4.16.0",
26    "@ionic-native/firebase": "^4.16.0",
27    "@ionic-native/push": "^4.16.0",
28    "@ionic-native/splash-screen": "~4.11.0",
29    "@ionic-native/status-bar": "~4.11.0",
30    "@ionic/storage": "2.1.3",
31    "angularfire2": "^5.0.0-rc.3",
32    "cordova-android": "7.0.0",
33    "cordova-plugin-device": "^2.0.2",
34    "cordova-plugin-fcm-with-dependency-updated": "^2.2.5",
35    "cordova-plugin-ionic-keyboard": "^2.1.2",
36    "cordova-plugin-ionic-webview": "^2.1.3",
37    "cordova-plugin-splashscreen": "^5.0.2",
38    "cordova-plugin-statusbar": "^2.4.2",
39    "cordova-plugin-whitelist": "^1.3.3",
40    "firebase": "^4.8.0",
41    "ionic-angular": "3.9.2",
42    "ionicons": "3.0.0",
43    "rxjs": "^5.5.2",
44    "sw-toolbox": "3.6.0",
45    "zone.js": "0.8.26"
46  },
47  "devDependencies": {
48    "@ionic/app-scripts": "3.1.11",
49    "typescript": "~2.6.2"
50  }
51 }

```

Fonte: Do Autor

O *backend* da aplicação ficou sob controle da plataforma BaaS do Google, para utilizá-lo basta ter uma conta cadastrada no Google e acessar o painel do Firebase, depois clicar em adicionar projeto, então adicionar o nome do projeto, o país/região que o projeto vai ser criado e concordar com os termos de uso da plataforma. Inicialmente, o projeto é criado no plano Spark que é plano gratuito de utilização da plataforma. Vale ressaltar que existem limitações nesse plano, porém ele atende todas as demandas necessárias para o funcionamento básico do protótipo, caso seja preciso fazer um upgrade do plano para obter melhores recursos, pode ser feito a qualquer momento, entretanto é preciso pagar uma taxa mensal que varia conforme o plano escolhido.

Posteriormente, após ser criado o projeto obteve-se acesso a todas as funcionalidades providas pela plataforma, contudo, foi necessário adicionar o

Firestore ao aplicativo, e esta opção já é disponibilizada logo após acessar o painel administrativo do Firebase, no caso do aplicativo desenvolvido foi adicionado as configurações de um aplicativo web ao projeto, tendo em vista que Ionic roda sobre o *webview* do sistema operacional do dispositivo móvel, ele pode se encaixar nesta categoria. Para adicionar o Firebase ao aplicativo foi criado um arquivo separado que exporta uma constante com as configurações que foram fornecidas pelo Firebase para que ocorra a comunicação entre o BaaS e o aplicativo, a Figura 9 demonstra a estrutura do objeto com as configurações (ocultada a *apiKey* por questões de segurança):

Figura 9: Configurações do Firebase

```
TS app.firebase.config.ts x
1  export const FIREBASE_CONFIG = {
2    apiKey: "AIzaSyA...",
3    authDomain: "app-tcc-a725c.firebaseio.com",
4    databaseURL: "https://app-tcc-a725c.firebaseio.com",
5    projectId: "app-tcc-a725c",
6    storageBucket: "app-tcc-a725c.appspot.com",
7    messagingSenderId: "579919920351"
8  };
```

Fonte: Do Autor

Em seguida, essa constante foi importada no arquivo *app.module.ts* que é o arquivo que inicializa o aplicativo Ionic, carregando assim todos os módulos fundamentais para o funcionamento correto do sistema. Feito isso, obteve-se as configurações necessárias para que haja comunicação entre a aplicação e o Firebase. A Figura 10 ilustra como ocorre essa inicialização dos serviços do Firebase no aplicativo.

Figura 10: Inicialização do Firebase no APP

```
@NgModule({
  declarations: [
    MyApp,
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
    AngularFireModule.initializeApp(FIREBASE_CONFIG),
    AngularFireAuthModule,
    AngularFireDatabaseModule
  ],
})
```

Fonte: Do Autor

5.4 ESTRUTURA DO BAAS

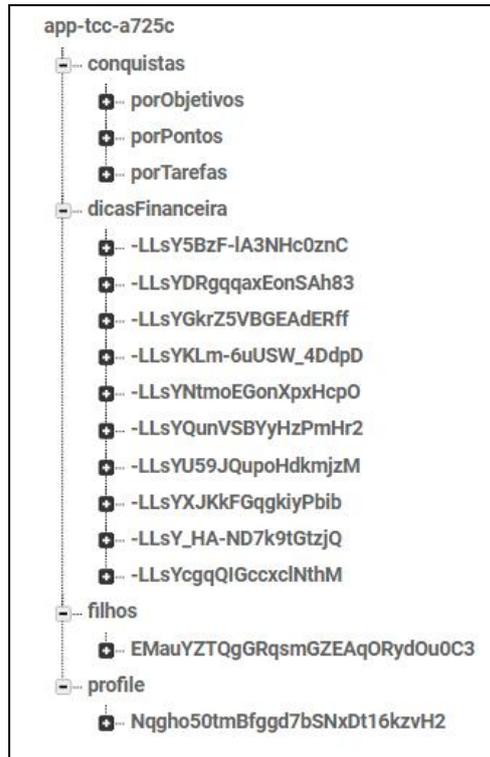
O projeto criado no Firebase já viabiliza diversos serviços que podem ser incorporados ao aplicativo. Estes serviços já foram descritos ao longo da Seção 2.6.1.2 no capítulo do referencial teórico, contudo, para o desenvolvimento do protótipo delimitou-se à utilização de apenas dois serviços que fazem parte do ecossistema do Firebase, sendo eles: o Authentication e o Realtime Database.

O Authentication foi utilizado para realizar o controle de acesso aos usuários a sua respectiva área restrita do aplicativo através da tela de login. Ele facilita este controle, pois quando é feito um novo cadastro de usuário no sistema, além de ocorrer a persistência das informações na base de dados, também é vinculado o mesmo *e-mail* e senha informados no formulário aos serviços do Authentication, ao qual este fica completamente segregado do banco, logo o *e-mail* e senha persistidos no Authentication viabilizam o mecanismo de autenticação no sistema e controle de sessão dos usuários no aplicativo.

Ademais, quando o usuário estiver logado ele também pode alterar suas informações através do formulário de edição, e a nova senha e *e-mail* cadastrados serão atualizados no Authentication e o usuário poderá acessar o aplicativo com suas novas credenciais registradas. Caso o usuário exclua seu perfil, suas informações também serão apagadas do serviço de autenticação do Firebase e não terão mais acesso à aplicação. Destaca-se que tais recursos foram implementados tanto para os pais, quanto para os filhos.

O Realtime Database é o banco de dados NOSQL do Firebase onde os dados são armazenados no formato JSON e quando é realizado alguma operação nele, os dados são atualizados em tempo real com todos os clientes que estão compartilhando a mesma instância da base de dados. Por ser uma base de dados não estruturada, é extremamente flexível e os dados persistidos podem ser tratados de diversas formas e com alta performance. No aplicativo, utilizou-se da árvore JSON fornecida pelo banco para criar os *nodes* da árvore com cada registro contendo uma chave associada. A Figura 11 exemplifica como se estruturou a árvore JSON para manipular os metadados da árvore:

Figura 11: Estrutura Realtime Database



Fonte: Do Autor

Nesta estrutura existem quatro *nodes* principais, sendo eles: "*conquistas*", que contém um aninhamento para que separe-se as conquistas por objetivo, por pontos e por tarefas realizadas; outro *node* "*dicasFinanceiras*", que armazenam as dicas sobre finanças que os usuários recebem de forma aleatória no aplicativo quando clicam no botão que indica esta funcionalidade na página inicial; outro *node* "*filhos*", que armazenam todas as propriedades referentes aos filhos, onde cada filho contém um aninhamento de dados que armazenam os objetivos e as tarefas recebidas específicas de cada um; e o *node* "*profile*" que armazena os metadados referentes aos pais, e cada objeto associado a ele contém dados aninhados que fazem o relacionamento com os dados dos seus filhos cadastrados, assim como aninhou-se as tarefas cadastradas pelo responsável no sistema que são enviadas aos filhos.

Salienta-se que o *hash* existente nos *nodes* aninhados acima, representam o UID de cada objeto e servem como um código identificador único de cada metadado, simular a uma chave-primária que existem nos bancos relacionais convencionais. Salienta-se que foi comumente utilizado aninhamento dos dados para que fosse facilitado o acesso às propriedades referentes a cada usuário cadastrado no sistema,

explorando assim a flexibilidade que o Realtime Database provê na sua plataforma, o que certamente seria dificultado se fosse utilizado um banco de dados relacional.

5.5 DESENVOLVIMENTO FRONT-END

O *frontend* do aplicativo fora desenvolvido utilizando o Ionic, AngularJS e Apache Cordova. Onde o Cordova realiza a comunicação com os recursos dos dispositivos móveis através de plugins, que executará a aplicação sobre a *WebView* do dispositivo, podendo ser implantado tanto em Sistemas Android, quanto IOS, mantendo apenas uma base de código composta por HTML, CSS e JavaScript. Já a combinação entre os *frameworks* AngularJS e o Ionic, possibilitam desenvolver uma aplicação baseada em componentes, sendo que cada componente é um fragmento de cada funcionalidade do sistema e contém uma finalidade específica, sendo compostos por um template HTML, CSS, e uma classe que gerencia as propriedades e comportamentos do sistema.

No aplicativo desenvolvido, cada tela existente no sistema corresponde a um componente, e isso facilitou a reutilização de código entre telas que compartilham os mesmos atributos, como a tela de cadastro por exemplo, que é utilizada tanto para realizar o cadastro, quanto para efetuar a atualização dos dados do usuário. A estrutura de cada página do aplicativo contém um código HTML que possui as marcações das tags fornecidas pelo *framework* Ionic, sendo que estas já possuem sua devida estilização CSS, e foram incorporadas seguindo a documentação do *framework*. A Figura 12 demonstra como ocorre a marcação HTML da tela de login.

Figura 12: Exemplo de código HTML utilizando componentes do Ionic

```

1 <ion-header>
2   <ion-navbar color="dark">
3     <ion-title text-center text-uppercase>Login</ion-title>
4   </ion-navbar>
5 </ion-header>
6 <ion-content>
7   <ion-row padding-right>
8     <ion-item>
9       <ion-label floating>E-mail</ion-label>
10      <ion-input type="text" [(ngModel)]="user.email"></ion-input>
11    </ion-item>
12    <ion-item>
13      <ion-label floating>Senha</ion-label>
14      <ion-input type="password" [(ngModel)]="user.senha"></ion-input>
15    </ion-item>
16  </ion-row>
17  <ion-row padding>
18    <button ion-button block (click)="login(user)" color='dark'>
19      Login
20    </button>
21    <p>Não é cadastrado? Faça o seu cadastro clicando
22      <a (click)="register()" style="font-weight: 600; font-size: 1.2em"> aqui</a>.
23    </p>
24  </ion-row>
25 </ion-content>

```

Fonte: Do Autor

Os eventos que ocorrem na aplicação ficam sob responsabilidade do arquivo TypeScript de cada componente que contém a classe que realiza o gerenciamento dos comportamentos do sistema. Este arquivo se trata de um super conjunto do JavaScript aos quais pode-se utilizar algumas funcionalidades como tipagem de variáveis, métodos acessores nas classes dos componentes para realizar o encapsulamento das propriedades, dentre outros recursos e, no final quando o aplicativo é compilado, ocorre o *transpile* do código TypeScript para o JavaScript compatível que será executado na aplicação. Ademais, nesse arquivo TypeScript do componente pode-se utilizar todos os componentes básicos de uma aplicação feita com Angular, como: módulos, *templates*, metadata, *data binding*, diretivas, serviços e injeção de dependências.

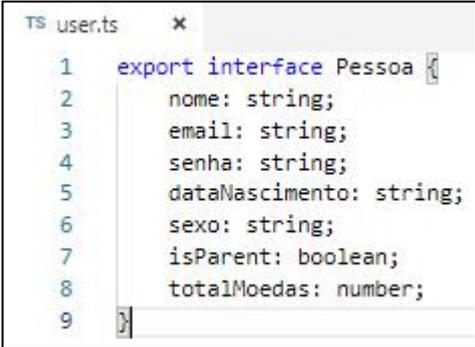
Além disso, as rotas de navegação da aplicação foram realizadas utilizando o *decorator IonicPage* disponibilizado pelo Ionic, possibilitando criar *links* diretos para cada página da aplicação, sem necessariamente precisar carregá-lo no módulo principal do aplicativo no arquivo *app.module.ts*, porém cada página deve ter seu respectivo arquivo *module* associado para que a navegação ocorra de forma adequada.

5.5.1 INTEGRAÇÃO DO FRONT-END COM O FIREBASE

A integração entre os serviços disponibilizados pelo Firebase e o *frontend* do aplicativo ocorreu utilizando-se da biblioteca *AngularFire2*, biblioteca que disponibiliza métodos simplificados para comunicação entre o *frontend* e o *backend* da aplicação (Seção 5.3). Os módulos foram importados no *app.module.ts* para que todos os componentes do sistema tenham acesso aos métodos fornecidos pela biblioteca. Feito isso, habilitou-se toda a estrutura necessária para que o aplicativo se comunique com o *backend* da aplicação de forma adequada.

Para que ocorresse uma padronização na codificação de desenvolvimento do aplicativo, e assim utilizar-se de boas práticas preconizadas na área de desenvolvimento de sistemas, sendo assim possibilitado um melhor reaproveitamento de código, além de otimizar manutenibilidade do sistema, criou-se uma camada de modelo que contém interfaces que mapeiam todos os atributos necessários para que ocorra a persistência dos dados no *backend* da aplicação. A Figura 13 exemplifica como ocorreu o mapeamento dos atributos que são persistidos no Realtime Database para armazenar as informações dos usuários, sendo válido tanto para os dados dos pais, quanto para as informações dos filhos.

Figura 13: Exemplo da interface modelo de Pessoa



```
TS user.ts x
1  export interface Pessoa {
2      nome: string;
3      email: string;
4      senha: string;
5      dataNascimento: string;
6      sexo: string;
7      isParent: boolean;
8      totalMoedas: number;
9  }
```

Fonte: Do Autor

Além disso, também foi criada uma camada de serviços para concentrar as regras de negócio do aplicativo e assim encapsular o código responsável por realizar as manutenções CRUD de cada objeto que é persistido na base de dados do Firebase. Nesta camada utilizou-se os métodos disponibilizados pelo *AngularFire2* para que ocorresse com sucesso as manipulações dos dados a partir das interações

do usuário com o sistema. Posteriormente, o serviço criado foi declarado nos *providers* do `app.module.ts` para que pudesse ser reutilizado por qualquer componente do sistema ao qual ele se relaciona. A Figura 14 ilustra como é feito o encapsulamento da classe que provê os serviços para que ocorram as manutenções das tarefas que os pais efetuam no aplicativo.

Figura 14: Classe que encapsula as manutenções das tarefas

```

TS tasks.ts  x
1  import { Injectable } from '@angular/core';
2  import { AngularFireDatabase } from 'angularfire2/database';
3
4  @Injectable()
5  export class TasksProvider {
6
7      constructor(private db: AngularFireDatabase) { }
8
9      getAll(key: string) {
10         return this.db.list(`profile/${key}/tasks`).snapshotChanges()
11             .map(changes => {
12                 return changes.map(c => ({ key: c.payload.key, ...c.payload.val() }));
13             });
14     }
15
16     save(task: any, key: string) {
17         return new Promise((resolve, reject) => {
18             if (task.key) {
19                 this.db.list(`profile/${key}/tasks`).update(task.key, { descricao: task.descricao })
20                     .then(() => resolve()).catch((e) => reject(e));
21             } else {
22                 this.db.list(`profile/${key}/tasks`).push({ descricao: task.descricao })
23                     .then(() => resolve());
24             }
25         });
26     }
27
28     remove(key: string, keyUser: string) {
29         return this.db.list(`profile/${keyUser}/tasks`).remove(key);
30     }
31 }

```

Fonte: Do Autor

Por fim, destaca-se que a classe controladora de cada componente da aplicação realiza a interligação entre as ações dos usuários com a camada de serviço de cada respectivo componente, o que acarreta na persistência, alteração, listagem ou exclusão no Realtime Database do Firebase. A Figura 15 demonstra o método que é chamado após o responsável cadastrar ou atualizar uma tarefa por meio de um formulário, e como ocorre comunicação com o *provider* que realiza as operações de persistência e atualização do respectivo componente.

Figura 15: Método de comunicação com a classe service de tarefas

```
onSubmit() {  
  if (this.form.valid) {  
    this.provider.save(this.form.value, this.user.key)  
      .then(() => {  
        this.toast.create({  
          message: 'Tarefa salva com sucesso.',  
          duration: 3000  
        }).present();  
        this.navCtrl.setRoot('TaskListPage');  
      }).catch((e) => {  
        this.toast.create({  
          message: 'Erro ao salvar a tarefa.',  
          duration: 3000  
        }).present();  
      })  
    }  
  }  
}
```

Fonte: Do Autor

5.6 CONSIDERAÇÕES FINAIS SOBRE O DESENVOLVIMENTO

Durante o processo de desenvolvimento do aplicativo ocorreram algumas dificuldades, o que é até corriqueiro em tratando-se da área de desenvolvimento deste tipo de tecnologia, mas nada que fosse algo extremamente complexo ao qual impediria o seguimento do processo de desenvolvimento da aplicação.

Dentre alguns aspectos que enquadram-se neste contexto, pode-se destacar a questão de versões das dependências do aplicativo, pois iniciou-se o desenvolvimento da aplicação utilizando uma versão mais antiga da biblioteca *AngularFire2*, ao qual já encontrava-se mais estabilizada e com mais informações em relação a correção de erros que ocorriam na respectiva versão que estava sendo utilizada, porém em uma determinada fase do desenvolvimento do protótipo esta versão já não estava contemplando algumas funcionalidades que seriam utilizadas para desenvolver o sistema móvel, logo realizou-se uma tentativa de atualização desta dependência da aplicação, mas este experimento acabou sendo infrutífero, pois quando conclui-se a atualização a aplicação simplesmente não funcionou mais conforme o esperado e muitos erros ocorreram.

Após ser realizado uma pesquisa sobre o porquê de tais erros estarem acontecendo, chegou-se a informações que descreviam que estes problemas ocorrem devido a incompatibilidades entre versões de algumas dependências que

estavam sendo utilizadas pelo *framework* AngularJS, como o RXJS, e a versão atualizada da biblioteca do *AngularFire2*. Portanto, optou-se por manter a versão que já estava sendo utilizada inicialmente no projeto, e assim abster algumas funcionalidades secundárias que seriam implementadas no protótipo.

Além disso, reitera-se que os objetivos delimitados para este protótipo só não foram atingidos em sua plenitude, pois infelizmente acabou não sendo implementado o sistema de notificações, ao qual seria o método disponibilizado para que tanto os pais, quanto os filhos, estivessem constantemente atualizados sobre o status de cada tarefa integradas entre eles no aplicativo. Este recursos seriam providos pelo Google Cloud Functions, sendo que este iria enviar notificações a partir do backend quando detectasse alterações na base de dados do Realtime Database de cada usuário específico. Além disso, o Firebase Cloud Messaging iria rodar em segundo plano, detectando e mostrando as notificações para os usuários no seu dispositivo.

Vale ressaltar, que foi tentado diversas vezes incorporar estas funcionalidades ao aplicativo, entretanto não obteve-se êxito na sua implementação, pois provavelmente também ocorreu alguma incompatibilidade entre as bibliotecas que interligam os métodos providos pelo Firebase aos que estavam sendo utilizadas pelo aplicativo, logo para que se conseguisse cumprir o cronograma estipulado para este projeto, optou-se por abster estas funcionalidades da aplicação.

Portanto, destaca-se que apesar destas dificuldades encontradas durante o processo de desenvolvimento da aplicação, de modo geral pode-se destacar os significativos aprendizados obtidos sobre o funcionamento das ferramentas utilizadas no projeto, como o Ionic, AngularJS, Firebase, e o funcionamento do ecossistema destas ferramentas como um todo. Pois, embora se possuísse algum conhecimento sobre como era realizado o desenvolvimento de aplicações com este tipo de ferramentas e também já se tivesse desenvolvido algumas pequenas aplicações de exemplo para aprender mais sobre o funcionamento delas, ao desenvolver, de fato, uma aplicação mais robusta e com diferentes funcionalidades, pode-se solidificar os conhecimentos sobre o mecanismo de funcionamento inerentes a aplicações desenvolvidas com estas tecnologias.

Além disso, salienta-se que por se tratar de um protótipo, foi priorizado a implantação das funcionalidades que compõe o embasamento central da aplicação, em detrimento a funcionalidades de secundárias.

6 RESULTADOS

Neste capítulo serão apresentadas os resultados obtidos a partir dos testes do protótipo que foram realizados em um dispositivo móvel após o término do processo de desenvolvimento, sendo que as funcionalidades testadas estão em consonância com metodologia referida no Capítulo 4.

6.1 TESTES EM UM DISPOSITIVO MÓVEL

Após terminar a codificação do protótipo desenvolvido, iniciou-se o processo de testes da aplicação. Embora o aplicativo desenvolvido com Ionic possa ser executado de forma híbrida, ou seja, tanto no sistema operacional IOS, quanto no Android, optou-se pela execução dos testes em um dispositivo Android na versão 6.0.1, tendo em vista que o Android SDK já encontrava-se instalado e configurado na máquina onde o aplicativo fora codificado, logo não seria necessário fazer todo esse processo novamente e o ambiente já estaria disponível com todas as bibliotecas e ferramentas fundamentais para realizar a implantação, testes e *debug* no sistema Android.

Inicialmente, necessitou-se adicionar a plataforma Android ao projeto através do comando *"ionic cordova add platform android"* disponibilizado pela ferramenta de linha de comando do Ionic. Feito isso, executou-se o *deploy* no dispositivo que estava conectado na máquina através do comando *"ionic cordova run android"*, e então todo o processo de implantação foi executado e logo após o término o aplicativo já encontrava-se instalado no dispositivo. Por conseguinte, introduziu-se os testes do aplicativo no respectivo dispositivo, salienta-se que tais testes ocorreram com dados meramente fictícios, tendo em vista apenas verificar se as funcionalidades implementados no protótipo funcionavam de maneira adequada.

Então, acessou-se o aplicativo e a tela de *login* foi apresentada. Logo em seguida clicou-se na opção para cadastrar um novo usuário e ali foi registrado um usuário fictício, informando os dados: nome, *e-mail*, senha e data de nascimento; e o usuário foi cadastrado no banco. Depois, retornou-se à tela de login e foram preenchidos os campos *e-mail* e senha com as credenciais cadastradas anteriormente, então ao clicar no botão de login, o usuário pode acessar a área restrita destinada aos pais, com sua autenticação realizada conforme o esperado

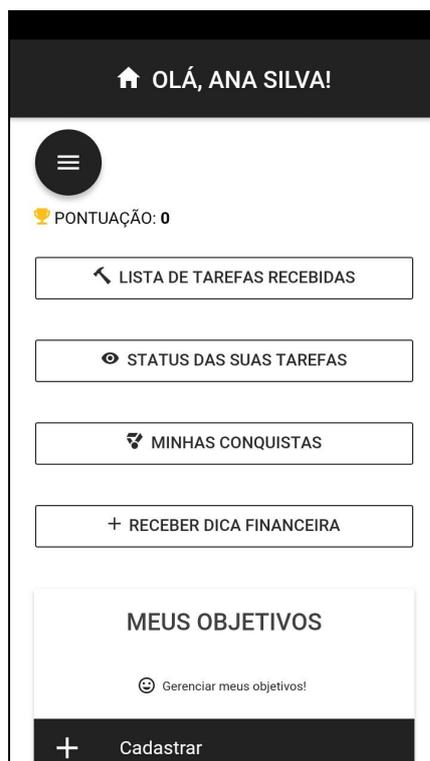
pelo serviço Authentication do Firebase. A tela inicial do aplicativo, pode ser vista na Figura 16:

Figura 16: Tela inicial da área de acesso dos pais



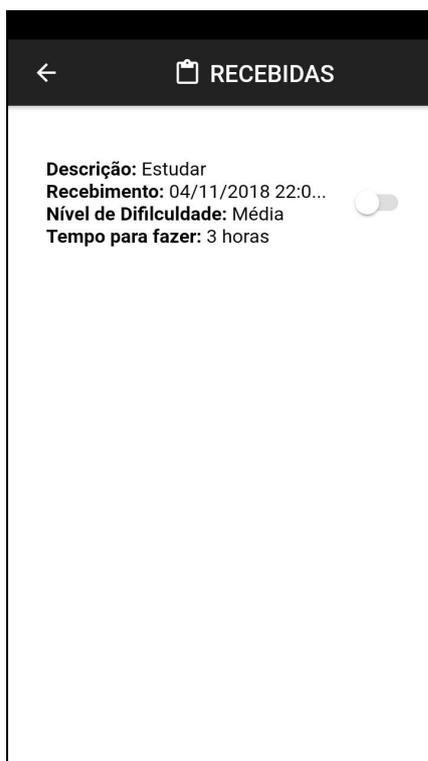
Fonte: Do Autor

Depois, foram realizadas as primeiras atividades, onde inicialmente cadastrou-se algumas tarefas, que já foram prontamente listadas, e depois se realizou operações de exclusão e atualização que também funcionaram de acordo com o que havia sido idealizado. Ressalta-se que tais operações foram realizadas com o console administrativo do Firebase aberto, e verificou-se que, de fato, os dados são atualizados instantaneamente na base de dados. Posteriormente, foi simulado o cadastro de uma filha por parte do responsável no sistema, e as informações foram persistidas. Logo, em seguida realizou-se o *logoff* para retornar a tela de *login*, e assim poder testar se a filha cadastrada anteriormente teria acesso a sua respectiva área de acesso permitida, então foi informada as credenciais de *login* do filha, pode-se verificar-se que tal ação ocorreu de maneira adequada. A tela inicial da área destinada a filha está ilustrada na Figura 17.

Figura 17: Tela inicial da área restrita aos filhos

Fonte: Do Autor

Verificado o funcionamento de tais funcionalidades, então foi feito novamente o *logoff* na área restrita da filha, sendo feita a autenticação novamente com os dados do pai, para que então pudesse ser testado o envio de tarefas. Então, clicado no botão "Enviar Tarefa" foi aberto o formulário que permite o envio da atividade para o filha, preenchendo-o com dados que já estavam pré-cadastrados, além de delimitar um tempo para que a execução da tarefa fosse feita, salienta-se que tal recurso é opcional. A tarefa foi corretamente enviada para a filha. Em seguida, repetiu-se o processo de *logoff* pai e autenticação da filha, e já na área da filha foi clicado no botão "Lista de Tarefas Recebidas", ao qual a tarefa já encontrava-se listada e com o *status* para ser feita. Tal estado está ilustrado na Figura 18.

Figura 18: Tarefa recebida pelo filho

Fonte: Do Autor

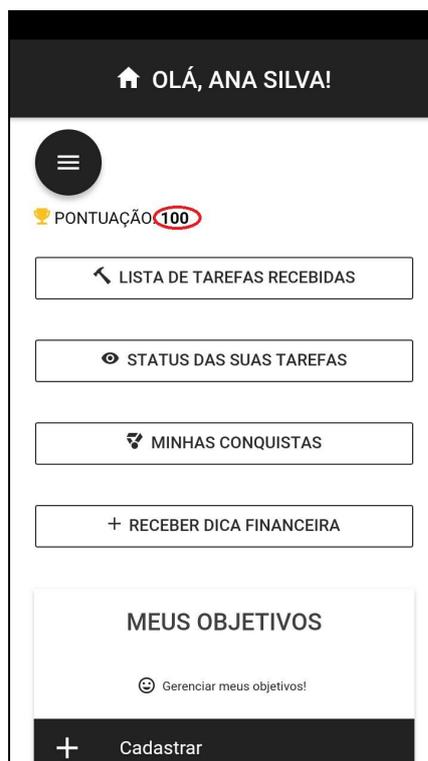
Logo depois, a tarefa foi marcada como realizada para que o responsável pudesse avaliá-la. Novamente, realizou o processo de *logoff* filha e *login* pai, e então acessou a listagem que mostra o *status* da tarefa do filha, e pode-se verificar que a tarefa já encontrava-se pronta para ser validada. O que pode ser visto através da Figura 19.

Figura 19: Validação da tarefa pelo pai após ter sido marcada pela filha

The screenshot shows a mobile application interface for managing tasks. At the top, there is a dark header with a back arrow on the left and the name 'ANA SILVA' in the center. Below the header is a white bar with the text 'GERENCIAR TAREFAS ENVIADAS'. The main content area is titled 'STATUS DAS TAREFAS' and contains the following information: 'Tarefa: Estudar', 'Envio: 04/11/2018 22:13:45', 'Confirmação: 04/11/2018 22:14:18', 'Nível: Média - 50 pontos', 'Tempo para fazer: 3 horas', and 'Status: Em avaliação' with a green checkmark. Below this information is a section titled 'Tarefa feita?' with two radio button options: 'Sim' and 'Não'. At the bottom of the screen is another white bar with the text 'GERENCIAR OBJETIVOS'.

Fonte: Do Autor

Depois, confirmou-se que a tarefa foi feita clicando na opção sim, e o status da tarefa já foi atualizado em tempo real, e conseqüentemente gerou a respectiva bonificação da filha, e quando novamente foi acessada a área destinada a filha, verificou-se que pontuação já encontrava-se atualizada e a filha recebeu pontuação duplicada por realizar a tarefa dentro do tempo pré-definido, tendo em vista que a tarefa valia 50 pontos e a pontuação ilustrada na tela inicial da filha encontrava-se em 100 pontos. Conforme demonstra a figura 20, logo a seguir:

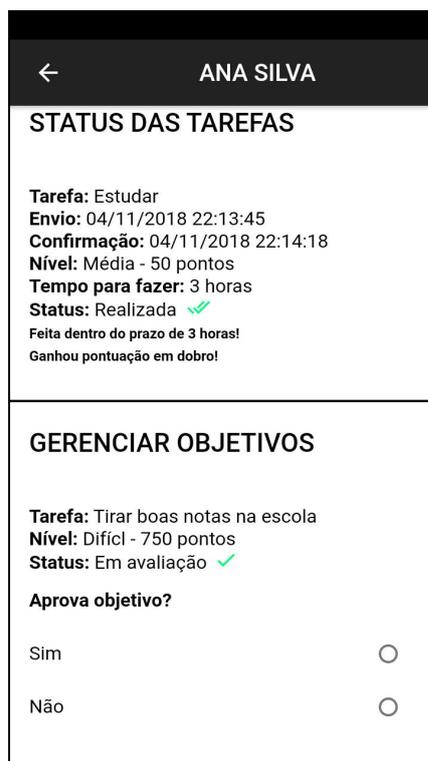
Figura 20: Pontuação do filho atualizada após ser gerada a bonificação

Fonte: Do Autor

Feito isso, o próximo teste efetuado foi em relação à funcionalidade de cadastro de objetivos na área de acesso referente a filha. Inicialmente, foi preciso cadastrar o objetivo onde foi informado a sua descrição e o nível de dificuldade que a filha imaginava que esta meta iria possuir. Contudo, quando a filha cadastra um novo objetivo, ele não é automaticamente confirmado no aplicativo e o mesmo fica no status “Em avaliação”, pois é efetuado o mesmo processo de validação que ocorre com a questão relacionada a verificação das tarefas.

Posteriormente, realizou-se a autenticação por parte do pai para que ele pudesse validar se o objetivo cadastrado pela filha é uma meta plausível de ser atingida por esta. Embora, se trate de uma questão apenas com a finalidade de mediar o nível de dificuldade da meta estabelecida pelo seu filho, este mecanismo é uma maneira de os pais estarem sempre conscientes dos anseios dos filhos, e dessa forma possam ajudá-los disponibilizando tarefas que irão induzi-los a ir realizando atividades de forma disciplinada e contínua, e assim viabilizar uma forma de aproximá-los cada vez mais de sua metas. O mecanismo de validação do objetivo mediado pelo responsável pode ser visualizado na Figura 21.

Figura 21: Validação do objetivo realizado pelo pai



Fonte: Do Autor

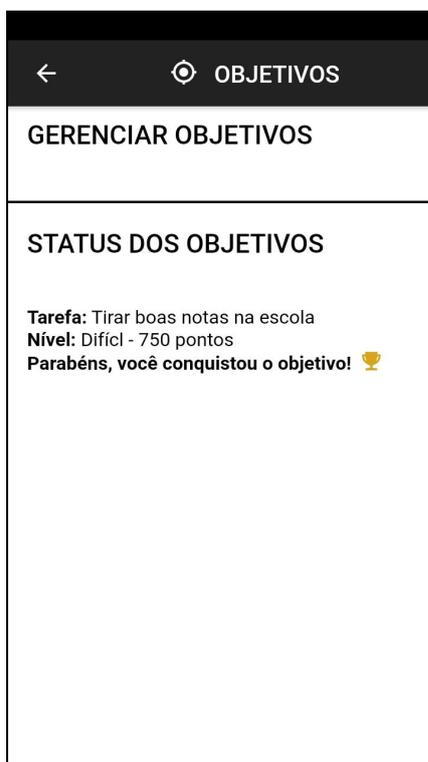
Em seguida, o objetivo cadastrado foi aprovado pelo pai e instantaneamente a listagem de objetivos da filha foi atualizada, onde o status do objetivo já encontrava-se “Em progresso” e com o botão de “liberação” da conquista bloqueado, sendo necessário que a filha realizasse tarefas até que ele fosse destravado. Para testar a funcionalidade de liberação de conquistas, criou-se várias tarefas fictícias para que pudesse ser atingido a pontuação que habilitaria o desbloqueio, repetindo o mesmo processo descrito anteriormente de cadastrar tarefas, o seu nível de dificuldade e um tempo pré-definido para a realização.

Um aspecto interessante que ocorre durante o processo da liberação de conquista é que enquanto o filho vai realizando as tarefas, a listagem com os status dos objetivos oferece um recurso que ilustra quantos pontos são precisos para que o filho atinja a sua meta, sendo este atualizado constantemente após cada tarefa validada com sucesso pelo responsável.

Dito isso, após ser obtida a pontuação que habilitou a liberação do objetivo para a filha, em seguida foi desbloqueado o botão para que pudesse ser efetuado o teste desta funcionalidade do aplicativo, onde foi clicado no botão e a conquista do objetivo foi liberada conforme o esperado. A Figura 22 ilustra a listagem dos

objetivos do painel da filha com status atualizado e com a sua liberação já processada.

Figura 22: Objetivo conquistado após filha atingir a pontuação



Fonte: Do Autor

Após concluir os testes relacionados a conquistas de objetivos do aplicativo, prosseguiu-se com o processo de experimentação do sistema ao qual foi dado ênfase na questão de liberação de conquistas do aplicativo, sendo este o mecanismo utilizado para que fosse implantado os métodos de gamificação que foram idealizados para que houvesse o maior motivação dos filhos para com a realização das tarefas. Para tanto, realizou-se o processo de cadastro, envio, recebimento e validação de diversas tarefas até que fosse habilitada a liberação das conquistas que estavam categorizadas em três diferentes áreas, sendo elas: por tarefas, por pontos e por objetivos; concluído o procedimento, foi acessado a página inicial destinada a filha e clicou-se no botão “Minhas Conquistas” para que pudesse ser acessada a página que contém os troféus que simbolizam a premiação pelo empenho na realização das tarefas por parte dos filhos. A tela do aplicativo que demonstra os troféus conquistados pelo filho no sistema pode ser vista na Figura 23.

Figura 23: Página que ilustra as conquistas atingidas pela filha

Fonte: Do Autor

Por fim, a última funcionalidade explorada durante a sequência de testes foi a questão relacionada ao recebimento de dicas financeiras, onde estas já estavam previamente cadastradas no banco de dados do Firebase e são apresentadas de modo aleatório para os usuários quando é clicado no botão da página inicial do aplicativo que contém este recurso, sendo este viabilizado tanto para os pais quanto para os filhos. Logo, acessou-se a área de acesso restrito de ambos os usuários criados anteriormente, e clicou-se no botão “Receber Dicas Financeiras” ao qual foi apresentado na tela uma dica de educação financeira de forma randômica, sendo que esta já estava previamente cadastrada no banco de dados. A Figura 24 exemplifica a dica de educação que foi apresentada ao usuário após ser clicado no botão que dispara esta ação no sistema.

Figura 24: Dica de Educação Financeira recebida pelo usuário

Fonte: Do Autor

Portanto, ao concluir o processo de testes acredita-se que foi possível atingir o objetivo primaz que era idealizado para esta aplicação, pois praticamente todos os recursos descritos na metodologia puderam ser implementados no sistema, exceto a questão relacionada ao envio de notificações. Salieta-se que todas estas funcionalidades já haviam sido testadas de forma unitária durante o período de desenvolvimento, contudo elas ainda não tinham sido testadas de forma integralizadas e em um dispositivo móvel, logo pode-se verificar que o sistema como um todo funcionou conforme havia sido planejado desde a fase modelagem da aplicação, sendo assim tornou-se possível apresentá-lo aos usuários para que pudessem avaliar se os recursos implementados no aplicativo são relevantes para que os filhos aprendam de maneira prática conceitos de educação financeira e se a médio/longo prazo isto acarretaria em melhorias na vida deles.

7 AVALIAÇÃO EXTERNA DO APLICATIVO

Para obter um *feedback* do sistema a partir da perspectiva dos usuários, o aplicativo foi apresentado em funcionamento para 8 pessoas que têm algum filho que estão na faixa etária dos 7 aos 18 anos, onde demonstrou-se todas as funcionalidades da aplicação para os possíveis usuários realizando o mesmo processo que foi descrito no Capítulo 6. Em seguida, disponibilizou-se um questionário para estes usuários através da ferramenta Google Forms, contendo as 6 perguntas listadas a seguir:

• Q1 - Você utilizaria este aplicativo para estimular conceitos de educação financeira para seu filho?

Sim Não

• Q2 - Você acha que mecanismo de bonificação e conquistas implementado no aplicativo aumentaria o engajamento do seu filho na realização de atividades cotidianas?

Pouco Razoável Indiferente Muito

• Q3 - Ao ter conhecimento dos objetivos do seu filho, você disponibilizaria através do aplicativo tarefas que irão aproximá-lo de suas metas?

Sim Não

• Q4 - Você acha que seu filho terá uma percepção melhorada em relação ao dinheiro ao verificar de forma prática como é trabalhoso para conquistá-lo?

Pouco Razoável Indiferente Muito

• Q5 - Você acha que seu filho melhorará o discernimento para lidar com as questões monetárias na vida adulta após praticar desde cedo hábitos inerentes a educação financeira?

Pouco Razoável Indiferente Muito

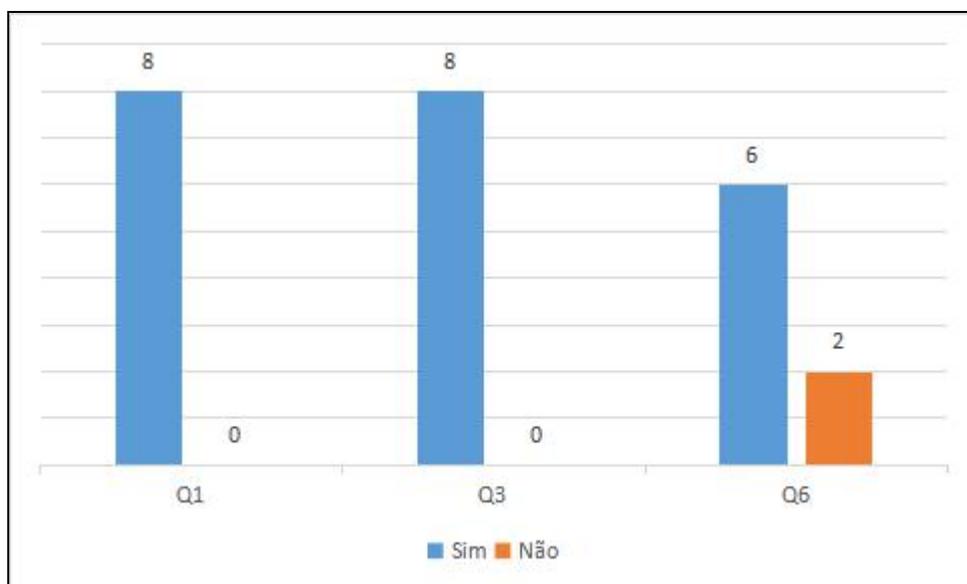
• Q6 - Você acha que a introdução do aplicativo com seu filho propiciaria um diálogo sobre finanças em casa?

() Sim () Não

Posteriormente, foram coletados os dados para que fosse possível analisá-los de maneira quantitativa e verificar se os usuários ratificaram a ideia de utilizar o aplicativo para apresentar conceitos de educação financeira para seus filhos utilizando métodos de gamificação.

Os resultados obtidos a partir das questões dicotômicas Q1, Q3 e Q6 são ilustrados de forma gráfica na Figura 25.

Figura 25: Gráfico de resultados das questões dicotômicas

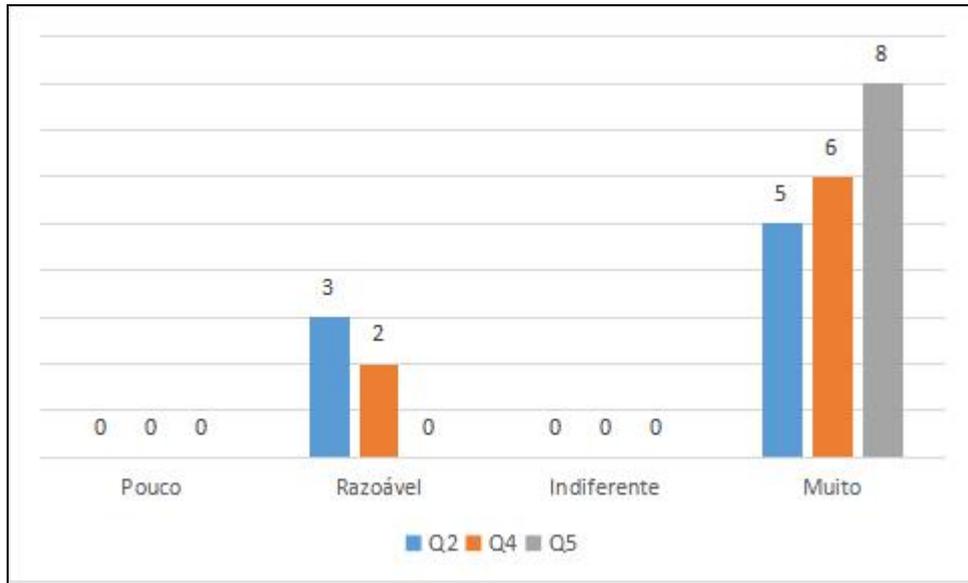


Fonte: Do Autor

Em relação à primeira questão (Q1), 100% dos usuários disseram que utilizariam o aplicativo para estimular a educação financeira na vida de seus filhos. O mesmo resultado foi obtido na terceira questão (Q3), onde os pais confirmaram que ao saber dos objetivos do seu filho iriam disponibilizar através do aplicativo tarefas que auxiliassem neste processo. Em relação a sexta questão (Q6), 75% dos pais responderam que a utilização do aplicativo iria propiciar um diálogo sobre finanças em casa com seus filhos, e 25% dos pais responderam que o aplicativo não propiciaria um diálogo sobre o assunto com seus filhos.

Com relação aos resultados obtidos através das questões Q2, Q4 e Q5 de escala avaliativa os dados são demonstrados no gráfico da Figura 26.

Figura 26: Gráfico de resultados das questões de escala avaliativa



Fonte: Do Autor

Em relação à questão dois (Q2), 62,5% dos pais responderam que o mecanismo de bonificação e conquistas do aplicativo aumentaria muito o engajamento do seu filho na realização das atividades cotidianas, e 37,5% acreditam que o engajamento aumentaria de forma razoável. Já em relação à terceira questão, 75% dos pais acreditam que seu filho teria uma percepção muito melhor em relação ao dinheiro ao verificar de forma prática como é trabalhoso para adquiri-lo, e 25% dos responderam que a percepção de seu filho em relação às finanças aumentaria de forma razoável. Em relação à questão cinco (Q5), todos os pais acreditam que seu filho terá um discernimento muito melhor para lidar com questões financeiras na vida, se praticarem desde cedo hábitos relacionados a educação financeira.

Embora a população participante da pesquisa não tenha sido muito grande, pode-se verificar através dos dados coletados que de modo geral os pais analisaram de forma positiva a ideia proposta pelo protótipo desenvolvido.

8 CONSIDERAÇÕES FINAIS

O protótipo do aplicativo desenvolvido acabou contemplando a maioria das funcionalidades que foram engendradas a partir da modelagem apresentada na metodologia proposta para o projeto. Ademais, destaca-se que as funcionalidades implementadas no aplicativo foram estruturadas para que tenham correlação com as ideias preconizada através da metodologia DSOP abordada no capítulo de referencial teórico do projeto, pois assim criou-se um método análogo ao qual os jovens podem exercer de maneira prática uma técnica já existente para aprender hábitos inerentes a educação financeira por meio de um conjunto de ações objetivas.

Além disso, o sistema de conquistas e bonificações implementado no aplicativo é um mecanismo de gamificação, utilizado para que os filhos não enxerguem a realização de tarefas como algo maçante, mas sim como um meio que será utilizado para que estes possam atingir seus objetivos, e assim mantenham-se empenhados em realizar as atividades para que estejam cada vez mais perto de lograr seus anseios, aproveitando-se de uma técnica que possivelmente já tenham conhecimento, pois é utilizada nos mecanismos dos jogos, porém aplicando-a nas suas vidas cotidianas para que os auxilie na realização de suas metas de vida.

Dessa forma, pôde-se viabilizar aos usuários um mecanismo para abordar a educação financeira no âmbito familiar, utilizando-se da tecnologia presente nos dispositivos móveis, pois é um recurso que já se tornou trivial no cotidiano das famílias brasileiras. Dessa forma, acredita-se que este mecanismo pode ser utilizado como um artifício complementar no processo educativo dos jovens em relação à área financeira, pois utilizando-se de gamificação pode-se abordar um tema que ainda precisa ser melhor explorado em nosso país de forma lúdica e instigante. Assim, os jovens podem ir fixando conceitos de responsabilidade financeira e os aprendizados obtidos durante a juventude irão perpetuar pelo restante de toda a sua vida adulta.

Por certo, os resultados da avaliação externa apresentados anteriormente demonstram que a ideia proposta pelo aplicativo é um mecanismo plausível para ensinar métodos da educação financeira utilizando dispositivos móveis junto com mecanismos de gamificação. Sobretudo, quando verificou-se que os pais aprovaram a proposta do projeto ao responder de forma positiva a maioria dos questionamentos.

Logo, entende-se que de modo geral os propósitos idealizados para este trabalho foram atingidos.

Com relação aos trabalhos relacionados citados no projeto, pode-se salientar a questão do sistema ser um aplicativo híbrido, podendo ser executado tanto em dispositivos Android, quanto em aparelhos IOS. Ademais, a questão dos filhos poderem delimitar objetivos, com a mediação realizada pelos pais, e assim poderem executar tarefas a fim de que possam conquistá-los no aplicativo, também é um aspecto relevante para ser destacado, pois é um mecanismo implementado para que os pais possam participar de forma ativa e ajudem os filhos na busca de suas metas, viabilizando através do sistema tarefas que auxiliem-os neste processo.

Apesar do aplicativo ter recursos similares conforme os trabalhos relacionados citados, principalmente com relação ao aplicativo Trato desenvolvido pelo Banco do Brasil, ressalta-se que é mais um projeto que pretende trazer à tona a conceitos de educação financeira com ênfase no público infantojuvenil, o que por si só é algo significativo, tendo em vista que trata-se de um tema que ainda é pouco debatido em tenra idade, mas que é vital conhecê-lo na vida adulta para que se tenha uma melhor qualidade de vida.

Em relação ao desenvolvimento destaca-se que a ênfase do projeto não foi nas tecnologias utilizadas, mas sim no embasamento da ideia de disseminar a educação financeira para os jovens por meio de dispositivos móveis. Para tanto, optou-se por utilizar tecnologias as quais já possuía alguma noção de seu funcionamento e pelo interesse em solidificar os conhecimentos sobre o desenvolvimento de aplicativos com estas, logo o processo de desenvolvimento foi extremamente produtivo pois se pode aprender mais sobre o funcionamento do Ionic, AngularJS e Firebase; tendo em vista que a medida que os problemas iam surgindo durante a etapa de desenvolvimento, necessitava-se ler a documentação, buscar informações sobre os *bugs* que ocorriam, e assim sedimentar os conhecimentos sobre estas tecnologias.

Frisa-se que em melhorias futuras do aplicativo é possível incorporar algumas novas funcionalidades, como a questão de vincular um serviço bancário para que os pais possam depositar uma mesada para o filho após ele atingir um determinado número de tarefas realizadas ou de conquistas no mês. Ademais, pode-se agregar algum serviço relacionado a questões de investimentos, onde o filho poderia aplicar seus pontos conquistados para que eles aumentem de forma progressiva com o decorrer do tempo e enquanto eles permanecessem empregados neste serviço, e

assim poder ir familiarizando os jovens com estes aspectos mais técnicos que abrangem a área relacionadas às finanças e que são muito importantes de serem praticados para obter uma vida financeira mais saudável. Este ponto abre um potencial para o desenvolvimento de um produto com viabilidade econômica para tornar-se comercial, o que se considera muito positivo.

Enfim, o protótipo foi desenvolvido com o intuito de que ele tenha grande potencial para crescer mediante ao incremento de novas funcionalidades que fomentam cada vez mais o tema da educação financeira na vida dos jovens, podendo assim arraigar de maneira progressiva a ideia de que desenvolver os hábitos comportamentais corretos ao longo do tempo mediante a um conjunto de ações, possibilitam à qualquer pessoa atingir suas metas e objetivos de vida. Logo, ir fixando esta mentalidade de maneira gradativa entre os jovens é uma possível forma de prepará-los para que futuramente se tornem cidadãos que impactem positivamente em diversas âmbitos da sua vida, podendo ser na esfera social, profissional, familiar, educacional e, sobretudo, financeira.

9 REFERÊNCIAS

ANDROID. Android. **Conheça o Android Studio**. 2018. Disponível em: <<https://developer.android.com/studio/intro/>>. Acesso em: 13 maio 2018.

APPLE. Apple. **Xcode - IDE - Apple Developer**. 2018. Disponível em: <<https://developer.apple.com/xcode/ide/>>. Acesso em: 13 maio 2018.

BARAZZETTI, Igor. **Qual a diferença entre site responsivo, site mobile (web app) e aplicativo para celular (app)?**. 2015. Disponível em: <<https://hitdigital.com.br/blog/qual-a-diferenca-entre-site-responsivo-site-mobile-web-app-e-aplicativo-para-celular-app-73>>. Acesso em: 13 maio 2018.

BB. BB. **BB prepara aplicativo de educação financeira para crianças**. 2016. Disponível em: <<http://www.bb.com.br/portallbb/page118,3360,3367,1,0,1,0.bb?codigoNoticia=52565>>. Acesso em: 10 junho 2018.

BELIZÁRIO, Jefferson. **Acabou! Microsoft não vende mais aparelhos com Windows Phone**. 2018. Disponível em: <<https://www.tudocelular.com/curiosidade/noticias/n123502/microsoft-nao-vende-mais-windows-phone-loja.html>>. Acesso em: 13 maio 2018.

CNC. PEIC. **Percentual de famílias com contas em atraso recua pelo quarto mês consecutivo em janeiro de 2018**. 2018. Disponível em: <http://cnc.org.br/sites/default/files/arquivos/analise_peic_janeiro_2018_2.pdf>. Acesso em: 22 abril 2018.

CORDOVA. Cordova. **Overview**. 2018. Disponível em: <<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>>. Acesso em: 20 maio 2018.

DE SOUZA, DÉBORA PATRICIA. **A importância da educação financeira infantil**. Belo Horizonte, 2012.

DOMINGOS, Reinaldo. **Terapia financeira**: realize seus sonhos com educação financeira. Editora DSOP, 2013.

FACEBOOK. React Native. **React Native · A framework for building native apps using React**. Disponível em: <<https://facebook.github.io/react-native/>>. Acesso em: 27 maio 2018.

FIREBASE. Firebase. **Produtos**. 2018. Disponível em: <<https://firebase.google.com/products/?hl=pt-br#grow-features>>. Acesso em: 27 maio 2018.

GUEDES, Thiago. **Crie aplicações com Angular**: O novo framework do Google. [S.l]: Casa do Código, 2017.

IONICFRAMEWORK. IONICFRAMEWORK. **Core Concepts**. 2018. Disponível em: <<https://ionicframework.com/docs/intro/concepts/>>. Acesso em: 27 maio 2018.

JUSTEN, Willian. **Como fazer seu site funcionar offline com PWA**. 2018. Disponível em: <<https://willianjusten.com.br/como-fazer-seu-site-funcionar-offline-com-pwa/>>. Acesso em: 13 maio 2018.

KASSARDJIAN, ACC. **Educação Financeira Infantil**. São Paulo, 2013.

LEITE, B. H.; SOUZA, A. N. **Gamifica Ação: A gamificação aplicada ao contexto social**. 2016. 22 p. Artigo (Tecnólogo em Sistemas para Internet) - Instituto Federal Sul-rio-grandense, Passo Fundo, 2016.

LIMA, Matheus. **INTRODUÇÃO AOS PROGRESSIVE WEB APPS**. 2017. Disponível em: <<https://tableless.com.br/introducao-aos-progressive-web-apps/>>. Acesso em: 13 maio 2018.

MEIRELLES, F. S. **29ª Pesquisa Anual do Uso de TI**. 2018. Disponível em: <<http://eaesp.fgv.br/sites/eaesp.fgv.br/files/pesti2018gvciappt.pdf>>. Acesso em: 22 abril 2018.

MICROSOFT. Xamarin. **Documentação do Xamarin - Xamarin**. 2018. Disponível em: <<https://docs.microsoft.com/pt-br/xamarin/>>. Acesso em: 27 maio 2018.

MILLER, Victor. **Aplicativo para educação financeira das crianças ganha prêmio Hackaton do Banco Original**. 2017. Disponível em: <<https://www.tudocelular.com/android/noticias/n115227/aplicativo-educacao-financeira.html>>. Acesso em: 10 junho 2018.

MINIJOB. miniJOB. **MiniJOB**. 2016. Disponível em: <<http://www.minijobapp.com.br/>>. Acesso em: 10 junho 2018.

NODEBR. NODEBR. **O que é a NPM do Node.JS?**. 2016. Disponível em: <<http://nodebr.com/o-que-e-a-npm-do-nodejs/>>. Acesso em: 27 maio 2018.

NODEJS. Node.js. **Node.js**. 2018. Disponível em: <<https://nodejs.org/en/>>. Acesso em: 27 maio 2018.

OCDE – **ORGANIZAÇÃO PARA A COOPERAÇÃO E DESENVOLVIMENTO ECONÔMICO**. Improving Financial Literacy: Analysis of issues and policies. Paris, 2005. 181 p.

PRADO, Ana Raquel Mechlin. **O endividamento dos brasileiros e a falta de educação financeira**. 2016. Disponível em: <<http://www.investmentosenoticias.com.br/noticias/negocios/o-endividamento-dos-brasileiros-e-a-falta-de-educacao-financeira>>. Acesso em: 22 abril 2018.

SCOLARI, Lidinara Castelli et al. **Educação financeira: análise de uma proposta desenvolvida no 7º ano no ensino fundamental**. 2014.

SCUDERO, ERICK. **Desenvolvimento mobile multiplataforma ou nativo, qual é melhor?**. 2017. Disponível em: <<https://becode.com.br/desenvolvimento-mobile-multiplataforma-ou-nativo/>>. Acesso em: 13 maio 2018.

SILVA, André. **Aplicações mobile: Web ou Nativa?**. 2012. Disponível em: <<http://blog.caelum.com.br/aplicacoes-mobile-web-ou-nativa/>>. Acesso em: 13 maio 2018.

TEIXEIRA, Fabricio. **O que são Progressive Web Apps?**. 2016. Disponível em: <<https://brasil.uxdesign.cc/o-que-são-progressive-web-apps-86e1b5306051>>. Acesso em: 13 maio 2018.

TOTH, Renato Molina. Abordagem NoSQL—uma real alternativa. **Sorocaba, São Paulo, Brasil: Abril**, v. 13, 2011.

VIANNA, Ysmar et al. **Gamification, Inc:** como reinventar empresas a partir de jogos. Rio de Janeiro: MJV, 2013.

APÊNDICES

Apêndice A - Descrição dos Principais Casos de Uso

Quadro 1: Caso de Uso Designar Tarefas

Descrição do Caso de Uso	Designar Tarefas
Ator Principal	Responsável
Resumo	Este caso de uso visa descrever como é realizado a designação das tarefas para o filho pelo responsável
Pré-Condição	O responsável deverá ter realizado login e já ter cadastrado o(s) filho(s) e a(s) tarefa(s).
Pós condição	Enviar notificação para o dispositivo do filho
Fluxo Principal	
Ações do Ator	Ações do Sistema
Responsável clica no ícone da página inicial do app que abre a tela para associar uma tarefa ao filho.	
	Abre a tela e carrega a listagem com as tarefas e os filhos já cadastrados no sistema..
Responsável seleciona o filho e a tarefa e clica no botão para habilitar a tarefa ao filho.	
	Recebe os dados enviados pelo responsável, salva e gera uma notificação para o dispositivo do filho com a nova tarefa que foi solicitada.
	Mantém a tarefa com o status pendente até que o filho marque-a como realizada.
Fluxo Secundário	
Ações do Ator	Ações do Sistema

Responsável delimita um período de tempo para que a tarefa seja efetuada.	
	Recebe os dados enviados, salva e gera uma notificação para o dispositivo do filho com a nova tarefa que foi solicitada e o período de tempo que ele tem para fazê-la..
	Mantém a tarefa com o status pendente até que o filho marque-a como realizada dentro do período pré-determinado, caso o tempo expire a tarefa será extinta.

Fonte: Do autor

Quadro 2: Caso de Uso Validar Tarefas

Descrição do Caso de Uso	Validar Tarefas
Ator Principal	Responsável
Resumo	Este caso de uso visa descrever como é realizado a validação das tarefas pelo responsável
Pré-Condição	O responsável deverá estar logado e já ter disponibilizado ao filho uma tarefa.
Pós condição	Gerar bonificação para o filho.
Fluxo Principal	
Ações do Ator	Ações do Sistema
	Notifica o responsável que o filho marcou a tarefa como realizada.
Verificar visualmente se o filho realmente realizou a tarefa.	
Caso realmente tenha sido feito a tarefa, marca a opção para gerar recompensa. Caso contrário, a tarefa torna-se inválida e não gera bonificação.	

	Recebe a opção selecionada pelo usuário e modifica o status da tarefa conforme a resposta do responsável.
	Envia uma notificação para o dispositivo do filho e conforme a resposta gera ou não o bônus de acordo com o que foi solicitado.
	Atualiza o status da atividade na listagem de tarefas do dispositivo do filho.
Fluxo Secundário	
Ações do Ator	Ações do Sistema
	Envia notificação que o filho marcou ou não a tarefa dentro do tempo estipulado.
Verificar visualmente se o filho realmente cumpriu a tarefa que foi determinada dentro do período de tempo pré-definido.	
Informa a resposta para o sistema.	
	Gera ou não a bonificação duplicada para o filho conforme a resposta enviada pelo seu responsável.
	Atualiza o status da atividade na listagem de tarefas do dispositivo do filho.

Fonte: Do autor

Quadro 3: Caso de Uso Receber Tarefas

Descrição do Caso de Uso	Receber Tarefas
Ator Principal	Filho
Resumo	Este caso de uso visa descrever como é armazenado as tarefas enviadas pelo responsável para o dispositivo do filho.

Pré-Condição	O filho deve estar logado no aplicativo.
Pós condição	
Fluxo Principal	
Ações do Ator	Ações do Sistema
	Envia notificação para o dispositivo do filho informando que ele tem uma nova tarefa a cumprir enviada pelo responsável.
Clica no ícone da notificação que redireciona para a tela que armazena a listagem de tarefas recebidas.	
	Carrega a lista de tarefas enviadas pelo responsável para o filho.
Visualizar a listagem das tarefas de acordo com o status que ela se encontra.	
Fluxo Secundário	
Ações do Ator	Ações do Sistema
Acessa a página inicial e seleciona um ícone com a lista de tarefas.	
	Carrega a lista de tarefas enviadas pelo responsável para o filho.
Visualizar a listagem das tarefas de acordo com status que ela se encontra.	

Fonte: Do autor

Quadro 4: Caso de Uso Realizar Tarefas

Descrição do Caso de Uso	Realizar Tarefas
--------------------------	------------------

Ator Principal	Filho
Resumo	Este caso de uso visa descrever como que filho modifica o status da tarefa para realizada e notifica o seu responsável.
Pré-Condição	O filho deverá ter realizado login no aplicativo., e ter atividades armazenadas listagem de tarefas pendentes.
Pós condição	Enviar notificação da resposta para o dispositivo do responsável.
Fluxo Principal	
Ações do Ator	Ações do Sistema
Acessa a página que armazena a listagem de tarefas pendentes.	
	Carrega a lista das tarefas delegadas pelo responsável para o filho.
Seleciona a tarefa e seta o componente toggle para verdadeiro (por padrão, ele estará desativado).	
	Envia notificação para o responsável informando a modificação do status da tarefa por parte do filho.
Fluxo Secundário	
Ações do Ator	Ações do Sistema
Marca a tarefa como feita dentro do período de tempo pré-definido.	
	Calcula se a tarefa está enquadrada dentro do período de tempo estipulado pelo responsável.
	Envia notificação para o responsável informando a modificação do status da tarefa por parte do filho.

Fonte: Do autor

Apêndice B - Questionário de avaliação do aplicativo

Q1 - Você utilizaria este aplicativo para estimular conceitos de educação financeira para seu filho?

Sim Não

Q2 - Você acha que mecanismo de bonificação e conquistas implementado no aplicativo aumentaria o engajamento do seu filho na realização de atividades cotidianas?

Pouco Razoável Indiferente Muito

Q3 - Ao ter conhecimento dos objetivos do seu filho, você disponibilizaria através do aplicativo tarefas que irão aproximá-lo de suas metas?

Sim Não

Q4 - Você acha que seu filho terá uma percepção melhorada em relação ao dinheiro ao verificar de forma prática como é trabalhoso para conquistá-lo?

Pouco Razoável Indiferente Muito

Q5 - Você acha que seu filho melhorará o discernimento para lidar com as questões monetárias na vida adulta após praticar desde cedo hábitos inerentes a educação financeira?

Pouco Razoável Indiferente Muito

Q6 - Você acha que a introdução do aplicativo com seu filho propiciaria um diálogo sobre finanças em casa?

Sim Não