

**FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”  
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**DIOGO HENRIQUE LIMA GONÇALVES DA SILVA**

**APLICATIVO MULTIPLATAFORMA DE ESCRITA COLABORATIVA  
EM TEMPO REAL**

**MARÍLIA  
2015**

**DIOGO HENRIQUE LIMA GONÇALVES DA SILVA**

**APLICATIVO MULTIPLATAFORMA DE ESCRITA COLABORATIVA  
EM TEMPO REAL**

Trabalho de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador  
Prof: Ms. Ricardo José Sabatine

**MARÍLIA  
2015**

SILVA, Diogo Henrique

**Aplicativo Multiplataforma de Escrita Colaborativa em Tempo Real/** Diogo Henrique Silva; orientador: Prof. Ms. Ricardo José Sabatine.  
Marília, SP: [s.n.], 2015.

84 folhas

Monografia (Bacharelado em Sistemas de Informação): Centro  
Universitário Eurípides de Marília.



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM  
MANTIDO PELA FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

---

Diogo Henrique Lima Gonçalves da Silva

Aplicativo Multiplataforma de Escrita Colaborativa em Tempo Real.

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Sistemas de Informação.

Nota: 10 ( DEZ )

Orientador: Ricardo José Sabatine Ricardo Sabatine

1º.Examinador: Renata Aparecida de Carvalho

Paschoal Renata Aparecida de Carvalho

2º.Examinador: Paulo Rogério de Mello Cardoso

Paulo Rogério de Mello Cardoso

Marília, 01 de dezembro de 2015.

## **AGRADECIMENTOS**

Agradeço a Deus,

Por ter me dado a sabedoria, a coragem e a força necessária para realizar esse sonho.

Aos meus Pais, Silvana e Paulo,

Por sempre estarem ao meu lado, sempre auxiliando em minhas decisões e acreditando na minha capacidade, torcendo por mim.

À minha companheira Marina,

Agradeço imensamente pelo companheirismo, por me fortalecer em todas as vezes que pensei em desistir, pelo apoio e dedicação ao me ajudar em etapas importantes do trabalho.

Aos meus sogros, Roselli e Waldomiro,

Por sempre me apoiarem, me passando dicas importantes e pela motivação necessário para realização desse sonho.

Aos meus amigos de sala,

Pela motivação, pela força e principalmente por sempre manterem o bom humor em momentos difíceis que passamos.

Ao meu Professor/Orientador Ricardo Sabatine,

Por ter acreditado na minha ideia, pela orientação deste trabalho, por seus ensinamentos, pela motivação e pelo conhecimento adquirido em sua disciplina do curso de sistemas de informação.

Aos professores do Curso de Sistemas de Informação,

Pela importante participação no conhecimento e habilidades adquiridos ao longo da graduação

Aos meus colegas de trabalho da empresa Interfocus,

Pela compreensão, apoio e por disponibilizar um tempo para que fosse possível finalizar este trabalho.

*"O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo.  
Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas  
admiráveis."*

*José de Alencar*

## RESUMO

O caminho para se tornar um escritor profissional ainda é uma tarefa muito burocrática, por conta disso alguns escritores internacionais estão buscando novas formas de se lançarem no mercado, uma delas é a escrita colaborativa. Sendo assim, o objetivo deste é o desenvolvimento de uma aplicação multiplataforma que contempla funcionalidades de escrita colaborativa, onde o diferencial consiste em como acontece a colaboração, pois tudo é em tempo real. O desenvolvimento se guiou pela metodologia “*Mobile First*”, a qual segue a tendência do mercado móvel, e busca atender todas as funcionalidade por igual entre os dispositivos, iniciando por telas menores. Como resultado, foi possível obter uma aplicação que pode ser utilizada nas plataformas iOS e Android, que conseguiu atender como uma solução à escrita de livros colaborativa, que possui disponibilidade imediata do conteúdo escrito em cada capítulo e um alto nível de interação entre os usuários. O projeto foi elaborado com base em pesquisas que permitiram obter conhecimento sobre novas tecnologias existentes no desenvolvimento móvel, além disso, foi realizado um aprofundamento em metodologias de trabalho colaborativo, as quais serviram de alicerce fundamental para a criação de um método de colaboração eficiente.

**Palavras-Chave:** Aplicação Móvel, Multiplataforma, Colaboração, Escrita, Livros, Tempo Real.

## **ABSTRACT**

The road to become a professional writer is still a bureaucratic task, because of this some international writers are looking for new ways to launch on the market, and one of them is the collaborative writing. Therefore, the main objective is the development of a multiplatform application that contains collaborative writing functionalities, where the differential consists on how collaboration happens, because everything is in real-time. The development guide was the Mobile First methodology, which follows the mobile market trending, and seeks to deliver all the functionalities between devices equally, starting with small screens ones. As result, it was possible to get an application that is available on iOS and Android platforms, which could attend as a solution to collaborative books writing that, have the availability of written content in each chapter and a high level of interaction between users. The project base was a research, that allowed obtaining knowledge about existing new technologies on the mobile development, furthermore, a deepening held in collaborative work methodologies, which served as a solid key for the creation of an efficient method of collaboration.

**Keywords:** Mobile Application, Multiplatform, Collaboration, Writing, Books, Real-Time.

## LISTA DE ILUSTRAÇÕES

|   |    |
|---|----|
| Figura 1. Estrutura Linear.....   | 19 |
| Figura 2. Estrutura Forma de Árvore.....  | 20 |
| Figura 3. Estrutura Network.....  | 21 |
| Figura 4. Modelo 3C de Colaboração.....   | 22 |
| Figura 5. Gráfico de Funcionalidade.....  | 24 |
| Figura 6. Fórmula para determinar IF.....   | 24 |
| Figura 7. Compartilhamento de um arquivo.....   | 29 |
| Figura 8. Website do Boston Globe visualizado em diversos dispositivos, um exemplo de um design responsivo..... | 36 |
| Figura 9. Aumento no trafego de dados da AT&T entre 2006 e 2009.....  | 37 |
| Figura 10. Arquitetura da Aplicação.....  | 44 |
| Figura 11 – Diagrama das Regras de Negócio da Aplicação.....  | 46 |
| Figura 12. Dispositivos entregues ao redor do mundo por seguimento, 2012 – 2017.....                            | 48 |
| Figura 13. Modelo 3C de colaboração na aplicação.....   | 50 |
| Figura 14. Diagrama do Método de Criação de Usuário.....  | 52 |
| Figura 15. Diagrama do Método de Login.....   | 54 |
| Figura 16. Diagrama dos Métodos de Gerência de Livros.....  | 57 |
| Figura 17. Pesquisa de livros.....  | 58 |
| Figura 18. Diagrama do Método de Chat.....  | 60 |
| Figura 19 – Diagrama de Requisições da Aplicação.....   | 61 |
| Figura 20. Cadastro de Usuario.....   | 64 |
| Figura 21. Login de Usuário.....  | 65 |
| Figura 22. Menu de Opções.....  | 66 |
| Figura 23. Lista de Livros do Usuário.....  | 67 |
| Figura 24. Criação de Livros.....   | 68 |
| Figura 25. Detalhes do Livro.....   | 69 |
| Figura 26. Edição Detalhes do Livro.....  | 70 |
| Figura 27. Criação de Capítulos.....  | 71 |
| Figura 28. Editor de Capítulo.....  | 72 |
| Figura 29. Chat do Capítulo.....  | 73 |

|                                    |    |
|------------------------------------|----|
| Figura 30. Pesquisa de Livros..... | 74 |
| Figura 31. Seguir Livro .....      | 75 |

## **LISTA DE TABELAS**

**Tabela 1. Resultado da avaliação da aplicação**

## LISTA DE ABREVIATURAS E SIGLAS

|              |  |
|--------------|--|
| API          | Application Programming Interface                                  |
| APP          | Application  |
| BPM          | Business Process Management  |
| CSS3         | Cascading Style Sheets 3   |
| CLI          | Command Line Interface   |
| DOCX         | Extensão de arquivos do Office Word                                |
| GPS          | Global Positioning System  |
| IOS          | Sistema operacional de dispositivos móveis da Apple                |
| HTML5        | HyperText Markup Language 5  |
| MODELO 3C    | Modelo de Comunicação, Coordenação e Cooperação                    |
| MOBILE FIRST | Metodologia de desenvolvimento que prioriza os dispositivos móveis |
| MVC          | Model - View - Controller ou Modelo – Visualização - Controle      |
| REST         | Representational State Transfer                                    |
| SDK          | Software Development Kit   |
| TI           | Tecnologia da Informação   |
| WEB          | Rede – retirado de World Wide Web                                  |

# SUMÁRIO

|  |    |
|--|----|
| INTRODUÇÃO .....   | 15 |
| 1 SISTEMAS COLABORATIVOS.....  | 18 |
| 1.1 Estruturas de Sistemas Colaborativos .....   | 19 |
| 1.1.1 Linear .....   | 19 |
| 1.1.2 Árvore .....   | 20 |
| 1.1.3 Network.....   | 21 |
| 1.2 Modelo de interação em Sistemas Colaborativos.....   | 21 |
| 1.2.1 Modelo 3C.....   | 22 |
| 1.3 Métricas de Qualidade.....   | 23 |
| 1.3.1 Medindo a Qualidade .....  | 23 |
| 1.4 Exemplos de Aplicações Colaborativas .....   | 25 |
| 1.4.1 Google Docs.....   | 25 |
| 1.4.2 Penflip .....  | 26 |
| 1.4.3 Wattpad .....  | 26 |
| 1.4.4 Waze.....  | 27 |
| 1.5 Considerações finais.....  | 27 |
| 2 TECNOLOGIA .....   | 28 |
| 2.1 Tecnologia para Sistemas Colaborativos .....   | 28 |
| 2.1.1 Técnicas de sistemas colaborativos .....   | 29 |
| 2.1.2 Ferramentas .....  | 32 |
| 2.2 Tecnologia para Desenvolvimento Multiplataforma.....                                       | 34 |
| 2.2.1 Principais Técnicas.....   | 35 |
| 2.2.2 Ferramentas disponíveis no mercado .....   | 38 |
| 2.3 Considerações Finais.....  | 41 |
| 3 DESENVOLVIMENTO DE UM APLICATIVO MULTIPLATAFORMA DE ESCRITA COLABORATIVA EM TEMPO REAL ..... | 42 |
| 3.1 Funcionalidades.....   | 43 |
| 3.2 Arquitetura .....  | 44 |
| 3.2.1 Modelagem das regras de negócio .....  | 46 |
| 3.2.2 Técnicas de implementação.....   | 47 |

|        |                                     |    |
|--------|-------------------------------------|----|
| 3.3    | Escrita de Livros Colaborativa..... | 49 |
| 3.4    | Ferramentas.....                    | 50 |
| 3.4.1  | Gestão de Usuários.....             | 51 |
| 3.4.2  | Gestão de Livros.....               | 55 |
| 3.4.3  | Escrita de Livro em Tempo real..... | 61 |
| 3.5    | Considerações Finais.....           | 62 |
| 4      | RESULTADOS.....                     | 64 |
| 4.1    | Aplicação Multiplataforma.....      | 64 |
| 4.1.1  | Cadastro de Usuário.....            | 64 |
| 4.1.2  | Login de Usuário.....               | 65 |
| 4.1.3  | Menu de Opções.....                 | 66 |
| 4.1.4  | Lista de Livros do usuário.....     | 67 |
| 4.1.5  | Criar Livro.....                    | 68 |
| 4.1.6  | Detalhes do Livro.....              | 69 |
| 4.1.7  | Editar Detalhes do Livro.....       | 70 |
| 4.1.8  | Criar capítulos do livro.....       | 71 |
| 4.1.9  | Escrever Capítulo.....              | 72 |
| 4.1.10 | Abrir Chat.....                     | 73 |
| 4.1.11 | Pesquisar Livro.....                | 74 |
| 4.1.12 | Seguir Livro.....                   | 75 |
| 4.2    | Análise dos Resultados.....         | 76 |
| 4.3    | Considerações Finais.....           | 79 |
| 5      | CONCLUSÃO.....                      | 81 |
| 5.1    | Trabalhos Futuros.....              | 82 |
|        | REFERÊNCIAS BIBLIOGRÁFICAS.....     | 83 |

## INTRODUÇÃO

Escrever um livro exige muita confiança no próprio trabalho e persistência por parte dos escritores, pois é necessário ter um bom conteúdo para que uma editora o avalie e possivelmente entre em contato com o autor para apoiá-lo na divulgação e publicação. Assim como ocorre com qualquer produto, os livros passam por especificações, revisões e são projetados desde o princípio. Entretanto, uma das maiores dificuldades em meio aos projetos é atingir os possíveis requisitos de um leitor.

Quando se fala de atingir os requisitos de um leitor é necessário lembrar que o primeiro leitor a avaliar um conteúdo que foi escrito é o próprio escritor, que por sua vez pode não possuir experiência necessária para verificar a autenticidade e valor da obra que está desenvolvendo, como faria um autor (KIEFER, 2010).

Segundo Hartmann, “Esta é uma incrível oportunidade para mim como autora a romper as barreiras na relação autor/leitor. Será incrível escrever sabendo que as pessoas estarão visualizando cada palavra, parágrafo e capítulo...” (HARTMANN, 2012). Assim como essa autora, outros escritores estão partindo para novas práticas de escrita, a fim de ultrapassar as barreiras entre escritor e leitor, para isso, utilizam uma das ferramentas disponibilizadas pelo Google. O Google Docs possui recursos de compartilhamento e sincronização em tempo real que são vistos como uma oportunidade de interagir com o público e tornar o desenvolvimento de um livro um trabalho colaborativo (GOOGLE DOCS, 2015).

Essa ferramenta possui recursos para edição de texto e pode ser utilizado em ambiente colaborativo utilizando opções de compartilhamento, entretanto, no contexto de desenvolvimento de livros pode conter diversas limitações que existem porque este recurso foi criado para ser utilizado apenas como um editor de textos, sem a finalidade específica de desenvolvimento de livros, portanto, não é um ambiente centralizado. Sendo assim, não é possível que o usuário realize pesquisas por livros e autores (GOOGLE DOCS, 2015).

Tais limitações vão além do mencionado, pois o que caracteriza um sistema colaborativo é a interação entre usuários que, deve ocorrer de maneira organizada, sem complexidade, e principalmente sem “*delay*” (espera) na atualização de conteúdo, algo que não é explorado pela ferramenta. Assim como ocorre em redes sociais, os usuários possuem uma liberdade para explorar todo tipo de assunto e interagir como querem, expressando suas opiniões e colaborando em assuntos específicos sem quaisquer formalidades.

Já a sincronização de textos em tempo real deve acontecer com o menor tempo de resposta possível, portanto, o desenvolvimento de uma aplicação multiplataforma com o princípio de centralizar as funcionalidades de escrita em tempo real voltadas ao desenvolvimento textual em um ambiente com recursos específicos, deve tanto elevar o nível de interação entre os usuários quanto diminuir a curva de produção de um livro (JARAMILLO et al, 2014).

Para o desenvolvimento da aplicação, foi utilizado como base a metodologia “*Mobile First*”, a qual determina que o desenvolvimento deve priorizar dispositivos com telas menores, com o propósito de entregar à todos, os recursos de forma igual (HARB et al, 2011). Sendo assim, as tecnologias de desenvolvimento foram definidas, onde o framework de desenvolvimento escolhido, o Ionic, que permitiu a criação de uma aplicação híbrida multiplataforma, em que a camada de visualização é web e as regras de negócio são nativas (IONIC, 2015).

O projeto teve como principal objetivo o desenvolvimento de uma aplicação multiplataforma, que proveu um ambiente voltado para escrita de livros colaborativa, onde permite que os usuários possam criar seus livros e interagir entre si através de um chat disponível em cada capítulo, este chat possui o intuito de aproximar as pessoas para aumentar o nível de colaboração na criação de conteúdo.

### **Objetivos Gerais**

O objetivo geral desse trabalho é desenvolver uma aplicação voltada ao público de escritores profissionais, pessoas que desejam se tornar escritores e também aqueles possuem o hábito da leitura. Esta aplicação permitirá que essas pessoas realizem o trabalho de escrever livros como uma atividade em grupo, onde o escritor fica responsável por desenvolver o conteúdo, os leitores que o acompanham por discutir assuntos sobre este conteúdo, e com isso, proporcionar uma rica troca de informações e feedbacks que o escritor precisa para alavancar sua carreira.

De modo a contribuir para ambos, escritor e leitor, já que o escritor pode contar com esses recursos para saber o que o leitor estará pensando sobre seu livro, e o leitor recebe em troca um conteúdo cada vez melhor.

## **Objetivos Específicos**

Os objetivos específicos são:

- Aproximar os escritores e leitores para que possam trabalhar em conjunto, para diminuir o tempo de feedback;
- Contribuir para a diminuição da burocracia que é lançar um livro, não sendo mais necessário que este seja aprovado por uma editora antes de ser lançado;
- Oferecer um espaço para que o público geral que possui o hábito de ler, possam também se expressar sobre o que estará lendo diretamente ao autor;
- Aumentar o alcance que um escritor iniciando pode ter, ao compartilhar a aplicação com outras pessoas;
- Introduzir uma maneira diferente de colaborar em um trabalho escrito em tempo real, sendo este organizado e atrativo por quem participa;

## **Organização do Trabalho**

A estrutura desta monografia é composta por cinco capítulos.

No capítulo 1 é feita uma revisão bibliográfica sobre sistemas colaborativos e soluções no mercado.

No capítulo 2 é feita uma análise das principais tecnologias para desenvolvimento multiplataforma e sincronização de texto em tempo real.

No capítulo 3 são apresentados os detalhes da implementação da aplicação multiplataforma e suas funcionalidades.

No capítulo 4 são apresentados os resultados obtidos.

No capítulo 5 é apresentada a conclusão do trabalho.

## 1 SISTEMAS COLABORATIVOS

Um sistema colaborativo constitui em um ambiente virtual específico, que possibilita o trabalho em grupo e interação social, um novo local para convivência humana. Sendo assim, conhecimentos técnicos sobre a área se tornam menos essenciais, já que é preciso entender as necessidades das pessoas, e identificar as características que os novos seres humanos “digitais” agora possuem (NICOLACI, 2012).

Essas características influenciam em como um sistema deve ser planejado e desenvolvido para permitir a colaboração entre os integrantes de um grupo. E essa interação deve ocorrer de forma natural, para isso não deve ser exigido ou forçado qualquer tipo de formalidade que se torne uma barreira aos usuários. Assim como ocorre em redes sociais, os usuários possuem uma liberdade para explorar todo tipo de assunto, interagir como quiserem e, expressar suas opiniões colaborando em assuntos específicos (NICOLACI, 2012).

Além de possibilitar a liberdade e interação entre os usuários, os sistemas colaborativos se tornaram uma ferramenta indispensável no desenvolvimento criativo em grupo. De acordo com Ivan e Ciurea, eles “... representam um novo domínio interdisciplinar, que é a intersecção entre as áreas de Tecnologia da Informação, Administração, Sociologia e etc.” (IVAN e CIUREA, 2009).

Como bom exemplo de aplicação para esse tipo de conceito no mercado, temos o Google Docs, uma aplicação web que conta com todos os recursos de edição que são conhecidos em ferramentas de edição como o Microsoft Word e Open Writer, sem a necessidade de ser instalado localmente. Essa aplicação acabou com a dependência de plataformas específicas para ser utilizada, além do mais, a Google incluiu recursos de compartilhamento e edição em grupo – o que a princípio não era prioridade – e atualmente se tornou um dos melhores recursos da ferramenta auxiliando na sua popularização (GOOGLE DOCS, 2015).

Assim como utilizado no desenvolvimento de aplicações como esta, diversas APIs (Application Programming Interface) que oferecem métodos prontos para adicionar recursos de escrita em tempo real são facilmente encontradas ao realizar uma pesquisa sobre esse tipo de solução. Porém, um dos que mais se destacam dentre tais tecnologias é o Firebase, já que possibilita uma utilização de conceitos de transmissão em tempo real através de uma API REST e persistência de dados em um banco de dados não relacional. Além da transmissão dos dados em tempo real, a utilização de webservices permite uma alta escalabilidade na

integração entre as aplicações e servidor (FIREBASE, 2015).

Assim, essa intersecção deve ser medida e, para isso existem métricas e indicadores com o propósito de avaliar a qualidade da colaboração e nível de interação em um sistema colaborativo. A partir de cada estrutura de sistemas colaborativos é possível definir métricas que buscarão avaliar se o desenvolvimento de um conteúdo está ou não ocorrendo de maneira colaborativa conforme suas definições.

## 1.1 Estruturas de Sistemas Colaborativos

Existem diversas maneiras de classificar um sistema colaborativo, a orientação de como ocorre a interação e transferência de informações é o que deve definir cada tipo de estrutura.

### 1.1.1 Linear

Conforme a Figura 1, em sistemas colaborativos lineares existe uma sequência de entradas e saídas que determinam como deve ocorrer a colaboração.

**Figura 1. Estrutura Linear**



**Fonte: IVAN e CIUREA, 2009**

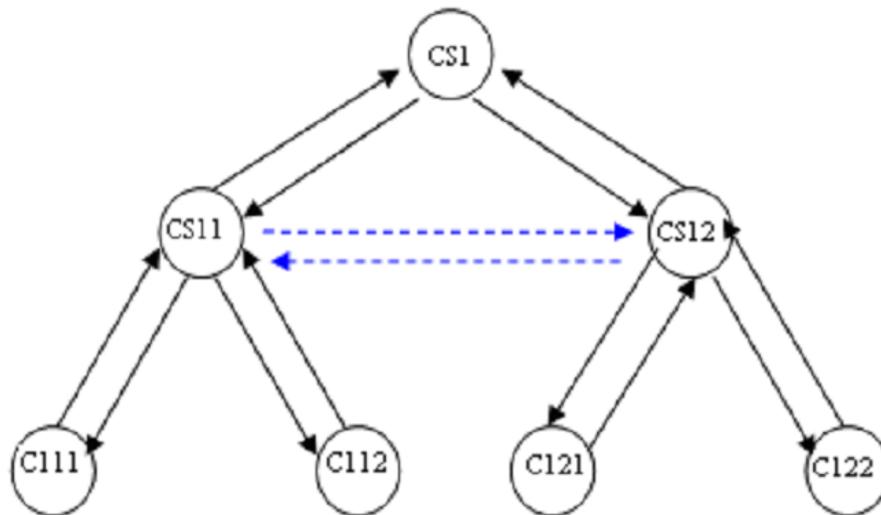
Em cada círculo existe um subsistema, as entradas são representadas por In e as saída On, seguindo de forma sequencial. Já os níveis intermediários são representados por k, que seriam os subsistemas envolvidos, ou seja, Ik e Ok.

Normalmente este tipo de sistema pode ser encontrado em meio acadêmico, pois são estruturados de forma linear e hierárquica, onde cada subsistema representa uma escola.

### 1.1.2 Árvore

Na Figura 2 está a estrutura em forma de árvore, onde as mensagens são transmitidas a partir do ponto CS1, e precisa passar por todos os nós até chegar ao limite à esquerda e direita, que no caso representa cada camada da hierarquia.

**Figura 2. Estrutura Forma de Árvore**



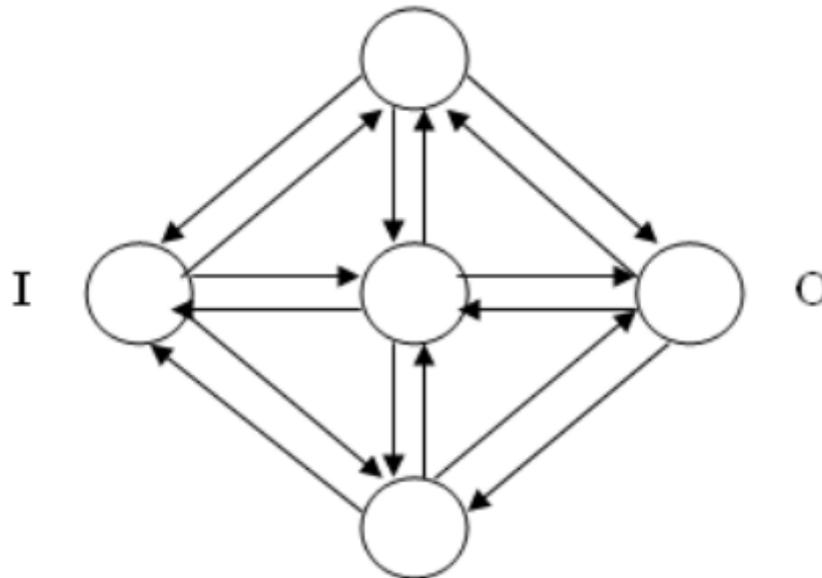
**Fonte: IVAN e CIUREA, 2009**

Normalmente é possível encontrar este tipo de estrutura colaborativa no gerenciamento de corporações, pois em todo processo existe uma cadeia hierárquica a seguir para que seja possível girar uma informação pelos setores da organização.

### 1.1.3 Network

Na estrutura de network todos os subsistemas se comunicam e participam da colaboração conforme é demonstrado na Figura 3.

**Figura 3. Estrutura Network**



**Fonte: IVAN e CIUREA, 2009**

Assim como ocorrem em redes sociais e sistemas que permitem a interação sem barreiras ou restrições, em uma network o trabalho será desenvolvido inteiramente colaborativo, pois todos os integrantes poderão interagir e opinar sem qualquer tipo de formalidade ou hierarquia.

## 1.2 Modelo de interação em Sistemas Colaborativos

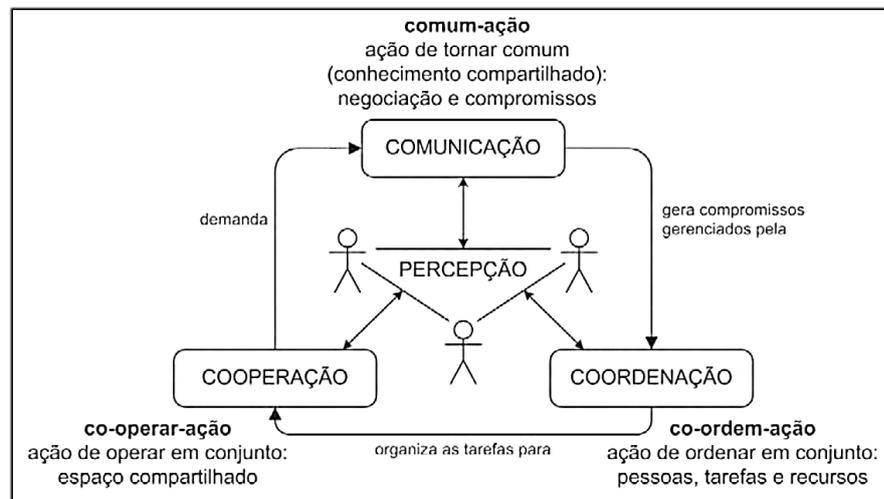
Um modelo de interação define um conjunto de dimensões para que de fato aconteça a colaboração em um trabalho específico, de forma produtiva e com melhoria contínua. E através deste, é possível definir quais atividades um sistema possuirá, pois nele também são relacionados os tipos de interação que podem atuar em conjunto e gerar um produto final melhor (FUKS et al, 2009).

O modelo padrão definido pelo autor, foi criado com a missão de identificar quais seriam as dimensões de um sistema colaborativo para que pudesse haver eficiência quando trabalham em conjunto e, com o Modelo 3C foi possível definir três tipos que serão especificados neste capítulo.

### 1.2.1 Modelo 3C

O Modelo 3C de colaboração certifica a colaboração em três dimensões: comunicação, coordenação e cooperação. Conforme esquematizado na Figura 4, as dimensões são divididas para que sejam analisadas em sua colaboração específica, porém, elas trabalham de forma conjunta para que então a colaboração aconteça (FUKS et al, 2009).

**Figura 4. Modelo 3C de Colaboração**



**Fonte:** FUKS et al, 2009

Essas dimensões possuem funções específicas que colaboram entre si para que o resultado final seja um conteúdo que passou por etapas dentro de um processo sequencial, conforme descrito abaixo:

- **Comunicação:** A comunicação é responsável pela troca de mensagens, conversação, e principalmente pela argumentação e negociação entre os integrantes (FUKS et al, 2009).

- **Coordenação:** A coordenação, é responsável pelo gerenciamento da interação entre as pessoas, não deixando que ocorra desperdício de tempo em conversas alheias, focando nas atividades principais e recursos (FUKS et al, 2009).
- **Cooperação:** A cooperação é caracterizada pela interação entre as pessoas dentro de um espaço onde estão trabalhando em conjunto, para a produção de conteúdo (FUKS et al, 2009).

### 1.3 Métricas de Qualidade

Para que seja possível medir a qualidade de sistemas colaborativos, é necessário entender quais as características que este deve possuir, pois essas definem padrões que auxiliarão em sua manutenção e futuras alterações desse sistema, além oferecer uma importante atenção para o nível de segurança das informações, são elas: complexidade, confiabilidade, manutenibilidade, funcionalidade, usabilidade e estabilidade (IVAN e CIUREA, 2009).

#### 1.3.1 Medindo a Qualidade

Para analisar a validade dessas características, deve-se verificar fatores como o tamanho da mensagem a ser transmitida e a rota que irá percorrer. Dessa forma, é possível descobrir se o sistema está cuidando para que esta mensagem não se perca no caminho ou até mesmo seja alterada. (IVAN e CIUREA, 2009)

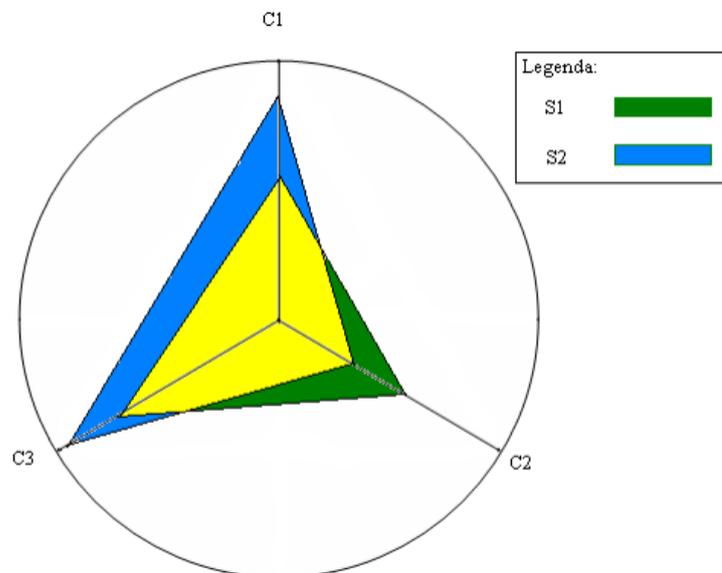
A confiabilidade é medida através da análise do número de problemas resolvidos pelo próprio sistema e o número total de problemas especificados.

A manutenibilidade é uma etapa importante no desenvolvimento de um software, pois é necessário levar em consideração o tempo em que será utilizado, já que em se tratando de sistemas colaborativos normalmente utilizados por mais de três anos, deve-se avaliar a necessidade de alterar e programar novas funcionalidades a longo prazo.

Já a complexidade trata-se de como são medidas as interdependências, recursos externos e entrada e saídas do sistema. Para isso é analisado o fluxo de informações de cada tipo separadamente, para que seja avaliada a carga de componentes para identificar o nível aceitável.

Conforme a Figura 5, onde C1 representa a complexidade, C2 a confiabilidade e C3 a manutenibilidade, e de acordo com os limites S1 e S2, é possível avaliar o nível funcionalidade de um sistema colaborativo. Os limites S1 e S2 são divididos entre mínimo e máximo para determinar a intersecção entre as características e chegar a um valor aceitável conforme estiver descrito nos testes do projeto.

**Figura 5. Gráfico de Funcionalidade**



**Fonte: IVAN e CIUREA, 2009**

A intersecção desse gráfico é obtida pelo resultado de IF, índice de funcionalidade, e a fórmula para chegar a tal índice pode ser representado através da Figura 6:

**Figura 6. Fórmula para determinar IF**

$$IF = \frac{\min \{S_1, S_2\}}{\max \{S_1, S_2\}}$$

**Fonte: IVAN e CIUREA, 2009**

## 1.4 Exemplos de Aplicações Colaborativas

Neste capítulo serão apresentados trabalhos relacionados que por sua vez possuem recursos para possibilitar o trabalho colaborativo e assim como proposto no projeto pretendem resolver os problemas da escrita colaborativa.

Dessa forma, o Google Docs possui recursos que permitem a edição e colaboração na escrita de textos em tempo real, entretanto, não é uma ferramenta totalmente dedicada ao desenvolvimento de livros. Assim, aplicações tais como o Penflip e Wattpad, podem ser consideradas como as que possuem mais recursos e um ambiente preparado para atender nesse contexto.

Portanto, serão observadas as informações disponibilizadas pelas empresas desenvolvedoras dessas aplicações, com o intuito de obter um estudo de mercado sobre o que já existe para atender esta necessidade.

### 1.4.1 Google Docs

O Google Docs é um conjunto de ferramentas desenvolvido pela Google, que possuem recursos que vão além desde um simples editor de textos, já que através do Docs é possível criar documentos em formato “docx”, utilizado pelo Word da Microsoft e editá-los sem a necessidade de instalação local. Além de também conta com todos os recursos que o de seu concorrente, a Google também incluiu opções de compartilhamento e edição por múltiplos usuário simultâneos, tornando o desenvolvimento de trabalhos colaborativos de forma simples (GOOGLEDOCS, 2015).

A princípio essa ferramenta não tinha um foco colaborativo, entretanto, acabou tornando-se popular em ambientes tanto profissionais quanto educacionais. Dessa forma, para o acesso total aos recursos, é necessário um e-mail cadastrado no Gmail que está dentro dos produtos que a empresa possui.

### **1.4.2 Penflip**

O Penflip é uma aplicação web para desenvolvimento textual online ou offline, que contém ferramentas de edição para auxiliar os usuários. Este também conta com recursos colaborativos, sendo que em um deles os usuários que estão lendo um texto podem deixar comentários em frases e palavras do texto, responder à questionários feitos pelo autor e até mesmo deixar comentários sobre capítulos (PENFLIP, 2015).

Por conseguinte, as alterações da ferramenta não são em tempo real, ou seja, um usuário somente consegue visualizar as alterações após recarregar a página. Outro detalhe está para a falta de centralização e afirmação sobre o tipo de conteúdo que pode ser produzido pela ferramenta, tratando qualquer conteúdo como um projeto.

### **1.4.3 Wattpad**

Esta aplicação foi desenvolvida com o propósito de prover um ambiente de desenvolvimento textual e leitura de conteúdo gratuito para qualquer pessoa. Para tal, a aplicação possui recursos de edição textual, leitura de conteúdo de outros usuários e interação entre os usuários através de trocas de mensagens (WATTPAD, 2015).

Entretanto a ferramenta não possui como proposta principal o trabalho colaborativo, e em tempo real, mas permite que os usuários comentem em cada capítulo escrito das histórias. Outro fator que se torna uma barreira para o desenvolvimento em grupo é a atualização de conteúdo em tempo real, pois os usuários poderão ler conteúdo desatualizado e classificando de forma antecipada algo que ainda estaria em desenvolvimento.

Por fim, não se trata de um ambiente focado no desenvolvimento de livros, e sim um lugar onde qualquer um poderia escrever uma história e ler conteúdo gratuito.

#### **1.4.4 Waze**

O Waze é um aplicativo de GPS, que a princípio começou como uma ideia de trazer algo mais leve com interface limpa para facilitar a navegação nos mapas. Mas, alguns recursos de colaboração passiva e ativa foram adicionados, tornando-o um GPS com interações sociais (WAZE, 2015).

Como ocorre em uma rede social, os usuários interagem entre si de duas maneiras, de forma ativa, ao cadastrar avisos sobre acidentes, policia, rotas bloqueadas e etc. Passiva, ao passar pelos avisos que foram adicionados por outros usuário, o app reconhece e valida a informação, a tornando verdadeira e popular.

### **1.5 Considerações finais**

Um sistema colaborativo precisa conter elementos sociais que possibilitam o desenvolvimento de conteúdo em grupo de maneira natural. Assim como utilizado em exemplos como o Google Docs e Penflip, o desenvolvimento textual em grupo se tornou popular e possível em qualquer contexto, algo que começou a chamar a atenção das empresas e principalmente das pessoas que desejam desenvolver e compartilhar seu trabalho.

No entanto, cada sistema colaborativo deve ser caracterizado por sua estrutura, já que alguns não utilizam seus recursos como meio, mas sim como forma de permitir que as informações circulem em seu sistema. E a melhor maneira de se obter uma boa colaboração em um desenvolvimento é quando não existem barreiras, ou processos hierárquicos para que isso ocorra, tudo deve ser interligado como em uma rede social.

Sobretudo, todo sistema deve ser avaliado e medido para que seja possível comprovar sua eficiência e segurança. Os sistemas colaborativos também entram nesse conjunto, pois por se tratar de um sistema que será utilizado por uma quantidade grande de usuários e por tempo indeterminado, é necessário que este tenha as características básicas: confiabilidade, manutenibilidade e complexidade.

## **2 TECNOLOGIA**

Como visto no capítulo anterior, os sistemas colaborativos possuem características específicas que o definem, e são justamente elas que possibilitam que ocorra o trabalho em conjunto. Ainda foram apresentados diversos casos no mercado de aplicações que possuem o mesmo foco, o trabalho conjunto no desenvolvimento textual.

Além de focar neste tipo de sistema, o projeto tem como objetivo principal criar uma aplicação que poderá ser utilizada em diferentes sistemas operacionais de dispositivos móveis, para diminuir a deficiência existente no mercado referente a escrita colaborativa.

Seguindo o objetivo de oferecer funcionalidades que irão aproximar mais as pessoas, surge a ideia de que tudo deve ocorrer em tempo real, ou seja, cada palavra escrita por um usuário já pode ser visualizada pelos demais, e para isso existe uma gama de tecnologias disponíveis que serão apresentadas neste capítulo.

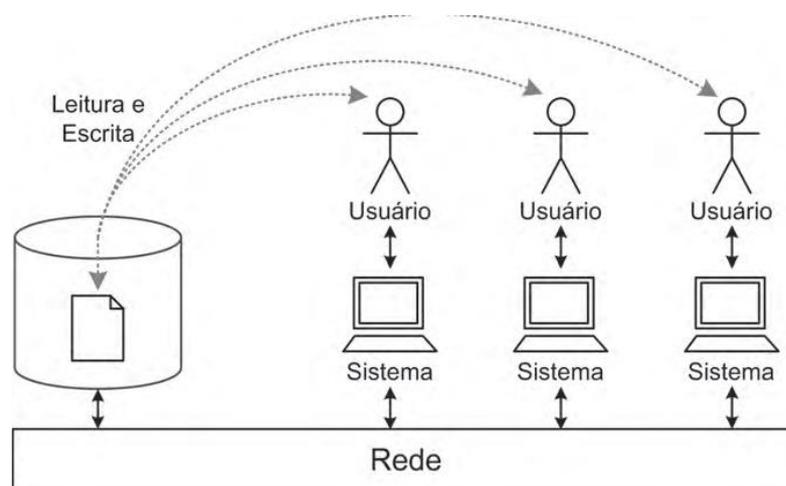
Por fim, a escolha de boas ferramentas e técnicas de desenvolvimento define a melhor forma de iniciar o desenvolvimento do projeto, facilitando assim o entendimento de técnicas que serão aprofundadas.

### **2.1 Tecnologia para Sistemas Colaborativos**

O desenvolvimento de um sistema colaborativo deve ser planejado, seguindo os requisitos essenciais para efetivamente garantir o seu funcionamento. Para tal propósito deve-se buscar conhecer as tecnologias disponíveis no mercado para utilizar a que melhor irá atender neste projeto (LIMA et al, 2009).

Conforme apresentado na Figura 7, é possível verificar o funcionamento de um sistema distribuído, onde vários usuários compartilham um mesmo arquivo ou dado.

**Figura 7. Compartilhamento de um arquivo**



**Fonte: LIMA et al, 2009**

De acordo com a autora, vários dos desafios estão relacionados ao ambiente computacional distribuído, formado pelos computadores utilizados pelos membros do grupo e toda a infraestrutura básica de comunicação entre as máquinas (LIMA et al, 2009). Portanto é de grande importância a contextualização de alguns conceitos sobre a comunicação entre cliente e servidor, para que assim seja analisado o formato mais adequado ao sistema colaborativo.

### **2.1.1 Técnicas de sistemas colaborativos**

As técnicas de sistemas colaborativos são várias dimensões de um processo de colaboração que quando ocorrem de forma conjunta, obedecendo o Modelo 3C, enriquecem o desenvolvimento de conteúdo, garantindo que o trabalho tenha um único foco e traga o melhor resultado (SANTAROSA E CRISTINA, 2009).

Com base no Modelo 3C, será apresentado um conjunto de conceitos sobre cada dimensão do modelo, tornando viável a obtenção de uma base técnica do que será implementado em cada funcionalidade da aplicação onde houver interação entre os usuários e, com isso refinar o conceito de escrita de livros colaborativa.

### **2.1.1.1 Comunicação**

A comunicação é o elemento principal de um trabalho conjunto, pois é nela que ocorre a troca de informações entre duas ou mais pessoas em um grupo. Então, toda troca de mensagem deve conter as informações codificadas no envio para ser decodificada no receptor, o qual tem a função de interpretar as mesmas

Além de mensagens em formato textual e verbal, há também outras formas linguísticas para comunicação, tais como a linguagem corporal e sinais. Sendo assim, é necessário a definição de um formato de linguagem ou protocolo compartilhado, de tal maneira que será interpretado por ambas as partes (SANTAROSA E CRISTINA, 2009).

Dessa forma, após definir o formato linguístico para que ocorra a comunicação entre as pessoas, também é preciso definir como será feita a interação, podendo ser através de: mensagens de texto em um comunicador aberto, comentários individuais, ou até mesmo avaliação do conteúdo através de notas e revisões.

### **2.1.1.2 Coordenação**

Na coordenação é fundamental que seja feito um planejamento após a definição do objetivo principal que será desenvolvido pelo grupo. O planejamento deve incluir de forma distribuída todas as tarefas que irão ocorrer durante o desenvolvimento de um conteúdo. Entretanto, o foco principal da coordenação é administrar e gerenciar as atividades que estão em execução, verificando sempre o conteúdo das mensagens que estão sendo trocadas para que haja conexão contínua com o assunto tratado (SANTAROSA E CRISTINA, 2009).

O grupo pode ser organizado em uma divisão de papéis, onde existirá alguém responsável por gerenciar as atividades e funções de cada integrante e outro para analisar as informações que estão sendo trocadas, verificando cada decisão tomada. As funções são definidas de forma dinâmica, podendo ser fixas ou temporárias, poderão ser empregadas conforme a necessidade de um contexto, mas principalmente de acordo com o tipo de interação que existir disponível no sistema em questão (SANTAROSA E CRISTINA, 2009).

Ainda existem três maneiras de executar a coordenação em um grupo:

- **Individualmente:** aqui cada integrante tem uma responsabilidade, e executa suas tarefas da melhor maneira possível, e por fim os resultados são somados para formar o conteúdo final (SANTAROSA E CRISTINA, 2009).
- **Repassando as tarefas:** nesta forma as tarefas são executadas de maneira conjunta entre os integrantes, onde sempre há necessidade de troca de ideias e os resultados são passados entre eles (SANTAROSA E CRISTINA, 2009).
- **Orquestrando:** em tal forma de organização todas as atividades são de forte dependência entre si, e precisam ser coordenadas para que todo o esforço conjunto esteja sincronizado e com isso chegar ao resultado esperado (SANTAROSA E CRISTINA, 2009).

### 2.1.1.3 Cooperação

A cooperação tem como principal objetivo o trabalho em grupo com o intuito de produzir conteúdo de forma colaborativa, e para isso é preciso ter um espaço que será compartilhado pelos integrantes, mesmo que este seja um recurso de comunicação de uma aplicação, o importante é que todos possam trabalhar em equipe (SANTAROSA E CRISTINA, 2009).

Sobre o espaço em que será compartilhado pelas pessoas, existem duas situações em que se difere como deve ocorrer a cooperação: primeiro, quando todos os participantes estão no mesmo local, ou seja, não existe atraso na interação e, segundo, quando os participantes estão fisicamente distantes e precisam de algum meio para estabelecer a interação (SANTAROSA E CRISTINA, 2009).

Portanto, no caso de uma aplicação colaborativa, os recursos de interação precisam ocorrer de forma síncrona para que as atividades em grupo ocorram fluidamente, e com isso as decisões sejam tomadas de maneira mais prática com o conhecimento de todos os interessados.

## 2.1.2 Ferramentas

Na composição da arquitetura de um sistema colaborativo pode ser comum a procura por soluções existentes no mercado que proporcionam uma experiência padronizada por uma comunidade ou empresa. Tendo em vista o prazo para desenvolvimento de tal tipo de aplicação, a escolha desse tipo de ferramenta torna possível a entrega do produto final com a qualidade e especificações propostas.

### 2.1.2.1 Etherpad

Lançado em novembro de 2008, o software Etherpad foi comprado pela Google em 2009 para ser incorporado às ferramentas da empresa, que em seguida o tornou open-source. Ao se tornar open-source atraiu a atenção de desenvolvedores que criaram suas próprias aplicações e ajudaram aperfeiçoar os recursos do Etherpad (ETHERPAD, 2015).

A proposta a princípio era oferecer um editor de textos colaborativo em tempo real, algo que proporcionaria facilidade na realização de trabalhos em grupo, escrita de artigos, compartilhamento de ideias, escrita de livros, e tudo isso funcionando direto do navegador. Qualquer pessoa pode utilizar a ferramenta para criar seus próprios documentos colaborativos, seja instanciando o serviço em um servidor local, ou através de outras aplicações que já utilizam o Etherpad.

Para incorpora-lo em outras aplicações, a empresa criou uma API bem documentada que oferece métodos para manipular todos os seus recursos, e foi desta forma que surgiram as empresas como Telecomix Pad, Framapad, Mozilla Pad, Titan pad, QikPad, etc. Estas empresas oferecem novas possibilidades de uso desses recursos, e muitas delas possuem planos gratuitos para empresas e público geral. (ETHERPAD, 2015)

Por fim, qualquer desenvolvedor que esteja interessado em contribuir com novas ideias, aprimoramento de funções, criação de novos métodos ou melhorias na documentação da API, pode acessar o projeto disponibilizado no Github, e lá todo código que foi adicionado ou modificado, deverá ser revisado e possivelmente lançado em próximas versões.

### 2.1.2.2 Google Drive API

O Google Drive Real-Time API foi desenvolvido para permitir que os desenvolvedores desenvolvam sistemas que faça o compartilhamento de dados em tempo real como se estivesse funcionando localmente. Porém, toda a sincronização ocorre somente visualmente na web, ou seja, quando existe uma modificação por um usuário, o modelo de dados é alterado automaticamente para todos os usuários que estiverem editando o documento (GOOGLE DRIVE, 2015).

Entretanto, existem algumas aplicações específicas para esse tipo de recurso, pois por não se tratar de uma sincronização de dados realizada do lado do servidor, mas sim do lado do cliente, acaba tornando limitado em alguns casos (GOOGLE DRIVE, 2015).

Aplicações que não necessitam que a persistência seja assíncrona, tais como editores de texto colaborativos, cadastros colaborativos (cada usuário escolhe uma opção), chat temporário, são exemplos de boas formas de utilizar os recursos da API.

Levando-se isso em consideração, aplicações que precisam que os dados sejam persistidos a todo momento e que tudo que foi alterado seja sincronizado diretamente do lado do servidor, ou seja, utilizando tecnologias de comunicação assíncrona, não são aplicáveis com esta API.

### 2.1.2.3 FireBase

De acordo com a documentação, o Firebase é uma plataforma baseada em conceito de nuvem, que fornece serviços de hospedagem de dados através de diversas integrações para o desenvolvimento de aplicações nativas ou Web. Tornou também possível a integração com aplicações *Web Mobile* que utilizam as diretrizes do AngularJS, entre outras, aplicações nativas para as plataformas Android e iOS (FIREBASE, 2015).

Os serviços de integração são fornecidos através de uma API que possui uma rica documentação para todos os recursos de sincronização entre os serviços e aplicações.

Exemplos de utilização do serviço:

- Aplicações que precisam que os dados sejam sincronizados em tempo real entre vários tipos de plataformas (HILTON et al, 2014).
- Ambiente web para desenvolvimento de software colaborativo (HILTON et al, 2014).

- Projetos em que seja necessário que o conteúdo mostrado esteja sincronizado em todas as aplicações (HILTON et al, 2014).

De acordo com Hilton (2014), “Como a aplicação funciona na nuvem, existe um potencial enorme para educação e colaboração à distância”. Sendo assim, o fato do serviço funcionar em nuvem se torna um diferencial no desenvolvimento de uma aplicação colaborativa independentemente da plataforma em que a aplicação será executada.

## 2.2 Tecnologia para Desenvolvimento Multiplataforma

Por conta da grande quantidade de tecnologias disponíveis no mercado, encontra-se necessário o estudo de cada uma para avaliação individual e enquadramento com os requisitos do projeto.

“[...]escrever um software especificamente para dispositivos móveis é algo trabalhoso, pois grande parte do trabalho precisa ser duplicado para que seja possível escrever aplicações nativas para desktops, Android, e iOS” (HILTON et al, 2014). Segundo o autor, o desenvolvimento nativo e específico por plataforma pode ser algo que necessita muito trabalho duro, com isso o processo se torna extenso.

Sendo assim, o intuito do desenvolvimento multiplataforma é ampliar o alcance da aplicação, portando a qualquer plataforma disponível sem restrições e com tempo de desenvolvimento menor o possível.

Ainda existem ferramentas disponíveis no mercado que possuem recursos e ambientes preparados para a reutilização de código e compilação para diversas plataformas, a essas ferramentas dá-se o nome de “*Cross-platform frameworks*”, e neste capítulo serão apresentadas algumas opções e técnicas para um melhor emprego de tal desenvolvimento neste projeto.

## 2.2.1 Principais Técnicas

O início de um projeto que pretende atender múltiplas plataformas precisa ser muito bem avaliado em alguns aspectos, para com isso poder direcionar o desenvolvimento a um ponto de partida que não terá implicações negativas em um futuro crescimento. Após tomar tal conhecimento, pode ser notável que tornem mais claros quais serão os passos a seguir, portanto, uma pesquisa sobre as principais técnicas de desenvolvimento multiplataforma servirá como guia para o andamento linear do projeto.

### 2.2.1.1 M.

Inicialmente não existiam maneiras para visualizar um conteúdo disponível para telas de computadores em dispositivos móveis, e para tornar isso possível foi criado o conceito Mdot ou M. como é conhecido no Brasil. Basicamente, todo o conteúdo apresentado é completamente diferente e reduzido para que o usuário consiga visualizar em telas pequenas de dispositivos moveis (MOHOROVICIC, 2013).

A princípio, essa técnica era uma ótima maneira de introduzir a navegação web em dispositivos móveis, já que na época ainda não era uma prioridade para o desenvolvimento web criar páginas com interface própria para esses dispositivos. Sendo assim, a técnica foi perdendo cada vez mais sua popularidade com o crescimento do mercado móvel.

### 2.2.1.2 RESS (*Responsive Web Design*)

O RESS ou Design Web Responsivo foi criado com o objetivo de adequar a forma como é consumido o conteúdo de uma página web em diversos dispositivos e formatos de tela. O conceito define que o dispositivo deve ser reconhecido conforme estiver definido no “*CSS3 Media Queries*”, dessa forma quando um dispositivo tiver uma tela menor, o mesmo automaticamente ajusta o conteúdo para ser apresentado e adaptado para seu uso (MOHOROVICIC, 2013).

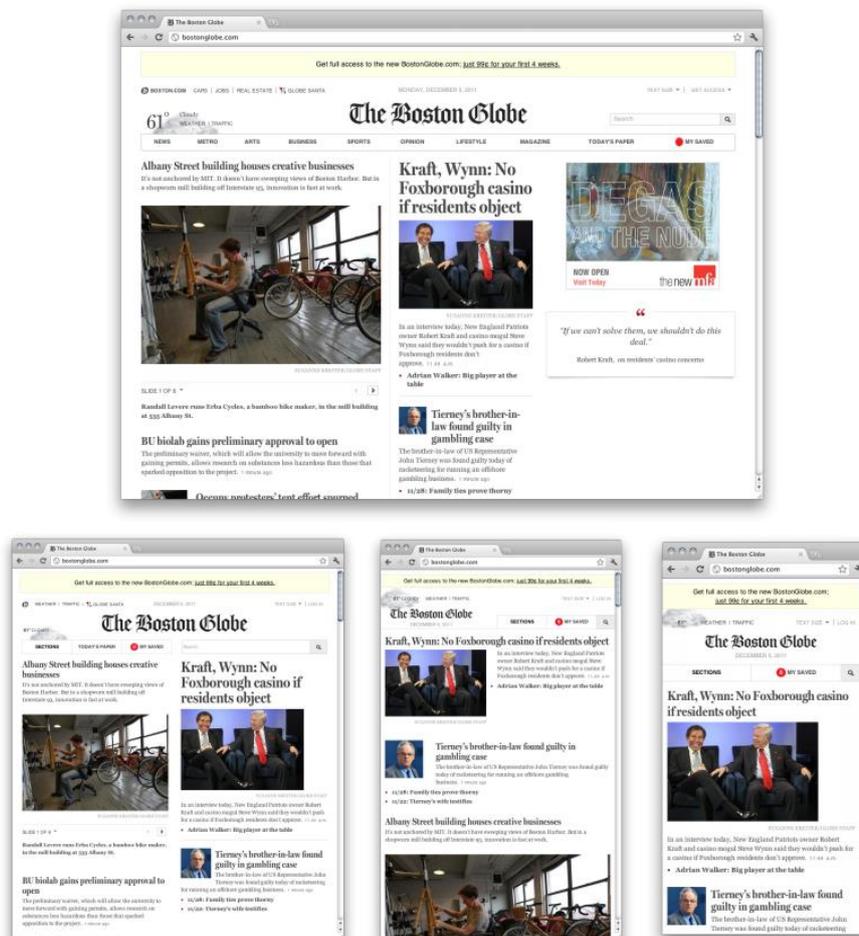
Os recursos mais importantes para uma página com design responsivo são:

- Uma “grid” flexível;
- Imagens com tamanho flexível;
- CSS3 Media Queries;

“Grid” e imagens flexíveis consistem em elementos que possuem características de redimensionar sua estrutura de maneira proporcional e sem perder a qualidade. No caso da “Grid” a área que estiver disponível para uso é preenchida pelo conteúdo da página conforme o tamanho da tela. Já no caso das imagens, conforme o tamanho da tela sua proporção será alterada sem perda de qualidade ou deformação da estrutura inicial (MOHOROVICIC, 2013).

Na F é apresentado um exemplo de como é o funcionamento da adaptação que esse tipo de design assume em diferentes dispositivos, telas e orientações.

**Figura 8. Website do Boston Globe visualizado em diversos dispositivos, um exemplo de um design responsivo**



Fonte: HARB et al, 2011.

### 2.2.1.3 Mobile First

Diferente do que acontece no design responsivo, tudo que for desenvolvido partindo pelo conceito do Mobile First, deverá priorizar primeiro os dispositivos móveis e então seguir para os dispositivos com telas maiores, tais como tablets e em seguida os desktops.

O Mobile First é uma nova forma de pensar que está seguindo uma tendência no mercado, o que aponta o crescimento relativo na utilização de dispositivos móveis. E ao utilizar esta técnica, o desenvolvedor conseguirá padronizar os recursos de uma aplicação ou página web em todas as plataformas, já que serão desenvolvidos com base em telas pequenas primeiramente (HARB et al, 2011).

Conforme apresentado na Figura 9, após o lançamento do dispositivo móvel da empresa Apple, o iPhone, houve um crescimento muito alto no consumo de dados móveis, algo que indicava que o mercado estava mudando, e precisava ainda mais de conteúdo adequado para o consumo em telas pequenas.

**Figura 9. Aumento no tráfego de dados da AT&T entre 2006 e 2009**



Fonte: HARB et al, 2011.

Logo, o propósito principal é basear todo o projeto na capacidade que um dispositivo móvel consegue comportar para que não sejam necessárias futuras adaptações ao ter que utilizar uma aplicação web feita para desktop que possuem telas maiores e podem comportar muitos componentes visuais, para a tela pequena de um celular.

## **2.2.2 Ferramentas disponíveis no mercado**

Assim como visto anteriormente, cada tecnologia possui uma particularidade que define como será o processo de desenvolvimento, que por sua vez pode ser algo demorado e requerer um maior aprofundamento, ou então ser simples e rápido para obter resultados satisfatórios. Seguindo esse caminho, analisar as ferramentas de desenvolvimento multiplataforma disponíveis no mercado se faz necessário para o total aproveitamento do tempo disponível, além do cumprimento com todas as funcionalidades propostas para a conclusão do projeto.

### **2.2.2.1 PhoneGAP**

O PhoneGap é um framework open source para desenvolvimento de aplicações híbridas multiplataforma. As aplicações são híbridas, pois são desenvolvidas usando tecnologias web como HTML5, CSS e JavaScript, em seguida todo o conteúdo é encapsulado em um pacote nativo específico para cada plataforma, entretanto, é possível desenvolver uma aplicações funcionais nos ambientes online ou off-line (PHONEGAP, 2015).

Existe a possibilidade de utilizar combinações de scripts para desenvolver uma interface, utilizando JQuery, MooTools e XUI. Mas toda sua lógica de programação e acesso à recursos do dispositivo deve ser feito utilizando a linguagem Javascript (PHONEGAP, 2015).

Este acesso aos recursos do dispositivo é provido através do PhoneGap API, que possui uma camada a camada de visualização e hardware. Para a utilização dos recursos a API disponibiliza uma biblioteca em Javascript para cada item do dispositivo (PHONEGAP, 2015).

Como exemplo do uso de tal ferramenta, pode-se mencionar sistemas bancários para gerenciamento à conta pessoal, aplicações de marketing e consulta a dados online. Em geral são aplicações que não utilizarão muitos recursos do dispositivo móvel para exercer suas funções e não precisam ser constantemente utilizadas sem conexão com a internet.

### 2.2.2.2 Appcelerator Framework

O Appcelerator é um framework que possui uma grade de ferramentas para auxiliar no desenvolvimento de aplicações nativas multiplataforma. Basicamente o funcionamento das aplicações criadas por este framework ocorrem da seguinte maneira, o código é todo escrito na linguagem JavaScript, e então é interpretado através do SDK Appcelerator, que integra na linguagem do dispositivo de destino.

Em tempo de execução, o código é avaliado através do interpretador JavaScript já existente na plataforma do dispositivo. Quando o aplicativo é aberto, o JavaScript é executado dentro da plataforma nativa, os códigos na linguagem são injetados como um pacote em tempo de execução. No caso o SDK Appcelerator trabalha como um intermediador entre os componentes e recursos próprios e o código em JS (APPCELERATOR, 2015).

O maior diferencial entre o PhoneGap e o Appcelerator, é que tudo acontece em tempo de execução sem a necessidade de utilizar componentes web para funcionar.

Vantagens:

- Reutilização de códigos para plataformas diferentes (Android, iOS, Windows Phone e HTML5);
- Tempo de produção das aplicações;
- Utilização de tecnologias WEB conhecidas;
- Não usa Browser para execução;
- Documentação da API;
- Comunidade Online;

Desvantagens:

- Não oferece acesso a todos os recursos dos dispositivos;
- Tempo de compilação;
- Tamanho da aplicação;
- Velocidade da aplicação;
- Dificuldade para iniciar o desenvolvimento muito avançada;
- Documentação pouco organizada;

### 2.2.2.3 IONIC

O Ionic foi criado em 2013 com o objetivo de criação de aplicações híbridas para dispositivos móveis, especificamente para as plataformas Android e IOS. Este framework está repleto de componentes e ferramentas, e estas são:

- **Apache Cordova:** é um conjunto de APIs que permitem que o desenvolvedor de aplicações móveis tenha acesso aos recursos nativos dos dispositivos como câmera e sensores de movimento apenas utilizando JavaScript (CORDOVA, 2015);
- **AngularJS:** é um framework javascript MVC que possui diversos módulos para a criação da camada de visualização Web e integração com a regra de negócio da aplicação (ANGULARJS, 2015);
- **Ionic CLI:** é um conjunto de linhas de comando para controlar a aplicação através do terminal, por lá é possível iniciar a aplicação, fazer um “Debug” e até mesmo iniciar novos projetos (IONIC, 2015);

O tempo de desenvolvimento utilizando este framework é relativamente curto, quando comparado com os demais apresentado pode se tornar uma ótima alternativa, e atualmente é vista como uma oportunidade para quem nunca desenvolveu aplicativos móveis, já que começar um projeto é extremamente fácil e a enorme quantidade de tutoriais online ajuda muito.

O fórum disponibilizado pela IONIC é muito rico em informação e colaboração por parte de milhares de desenvolvedores, que ajudam em cada dúvida, algo que conta muito para quem quer começar a trabalhar com o framework e não sabe como resolver problemas, e também para desenvolvedores experientes que precisam de novas ideias.

Um dos maiores incentivos para utilizar o IONIC são a documentação e tutoriais que a própria empresa disponibiliza para os usuários, todas as ferramentas e recursos estão documentados e exemplificados para que o usuário consiga aprender e solucionar problemas que possam ocorrer durante o desenvolvimento (IONIC, 2015).

## 2.3 Considerações Finais

No desenvolvimento de uma aplicação colaborativa existem diversas técnicas que podem servir como base para as funcionalidades que possuirão recursos colaborativos. Essas técnicas são baseadas no Modelo 3C, onde o processo de colaboração acontece com a união entre a comunicação, a coordenação e a colaboração, para que assim um trabalho colaborativo seja realizado de maneira mais produtiva e eficiente.

Além de técnicas para desenvolvimento colaborativo, uma aplicação também deve possuir recursos que facilite a interação entre as pessoas. Para isso, existem soluções no mercado que podem ser integradas em aplicações e serem utilizadas com pouco tempo de desenvolvimento, mas totalmente funcionais.

E para implementar tais recursos, o desenvolvimento da aplicação pode partir de um dispositivo desktop, priorizando primeiro telas maiores para então adaptador aos dispositivos móveis com telas menores, usando a técnica de design responsivo, ou então, partir de um dispositivo móvel priorizando as telas menores e a partir disso trazer a aplicação para dispositivos com telas menores, algo que segue as tendências do mercado.

Portanto, para atingir também outras plataformas móveis e possibilitar que mais pessoas consigam ter acesso à aplicação, existem ferramentas no mercado que visam a reutilização de código e exportação para múltiplas plataformas, de forma que economize no tempo de desenvolvimento sem deixar de obter uma aplicação completa e com recursos que uma aplicação nativa possuiria.

### **3 DESENVOLVIMENTO DE UM APLICATIVO MULTIPLATAFORMA DE ESCRITA COLABORATIVA EM TEMPO REAL**

O trabalho colaborativo é uma forma muito rica para criação de conteúdo, onde mais de uma mente está conectada e diferentes perspectivas de algo são apresentadas, dessa forma há menos risco de ocorrerem falhas que venham a atrapalhar atingir o objetivo proposto. Com este pensamento foram desenvolvidos os sistemas colaborativos, que são vistos como ferramentas fundamentais para proporcionar o trabalho em grupo em ambientes corporativos. Ao utilizar um sistema colaborativo a empresa deve ter em mente que o seu funcionamento depende da total liberdade de expressão de seus colaboradores, ou seja, não devem existir barreiras que possam inibir a criatividade dos participantes.

Esses sistemas possuem características específicas que definem como deve ocorrer a interação entre os usuários para que haja colaboração em uma tarefa específica, tais como a liberdade de cada participante ou até mesmo o nível de interação entre os mesmos, que mede o tempo de resposta entre cada interação, onde quanto mais próximo de “tempo real” maior será este nível.

No mercado existem soluções tanto corporativas quanto sociais para que exista uma interação colaborativa em diversas áreas, as mais conhecidas são as Redes Sociais, onde todo o conteúdo existente é desenvolvido pela comunidade através da interação entre milhões de pessoas. Além do uso social, existe também a colaboração em meio acadêmico, onde surgiram aplicações voltadas para o desenvolvimento de software em equipe, sendo que todo o código é editado em tempo real por todos os participantes, havendo assim um alto nível de colaboração e aumento da qualidade.

Através dessa análise foi possível enxergar o melhor caminho para o desenvolvimento da aplicação proposta no projeto, sendo esta uma solução para a escrita de livros de forma colaborativa voltada para o meio comum da profissão de escritor e buscando aumentar o nível de interação entre leitor e escritor, para que assim cresça também a produção de conteúdo com maior relevância ao seu público.

O objetivo principal deste projeto é o desenvolvimento de uma aplicação multiplataforma que proporcione tal interação e possua as funcionalidades que representam o núcleo da ideia que o define, ou seja, a escrita e colaboração em tempo real. Para isso, foram

utilizadas ferramentas que já possuem alguns desses recursos bem definidos e prontos para serem utilizados, e assim houve uma redução do tempo para obter a aplicação totalmente funcional.

### 3.1 Funcionalidades

As funcionalidades de uma aplicação devem representar o principal motivo para uma pessoa desejar utilizá-la, portanto, elas devem ser limitadas sobre quais serão as possibilidades de seu uso.

Seguindo os objetivos que foram apresentados, serão especificadas as funcionalidades que representam a ideia do projeto através de recursos que permitirão a escrita em tempo real de maneira colaborativa:

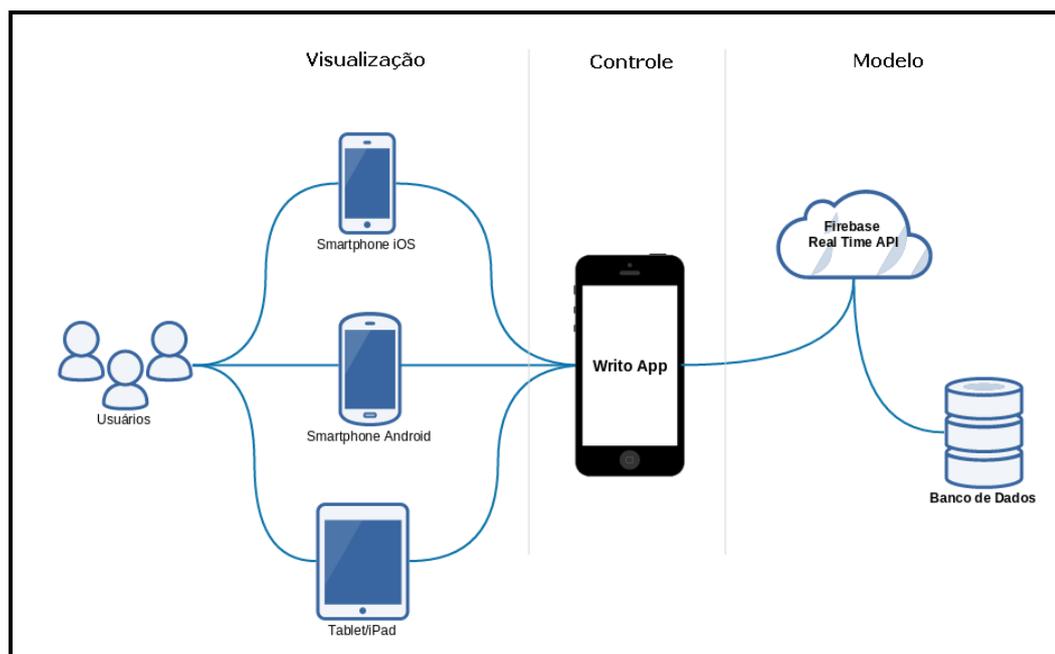
- **Cadastrar uma nova conta:** O novo usuário pode realizar um novo cadastro através da aplicação ao preencher os dados: Nome, E-mail e Senha;
- **Realizar Login:** Após o usuário cadastrar na aplicação, ele consegue realizar o login com seus dados persistidos na base de dados;
- **Criar um Livro:** O usuário cadastrado pode criar seus próprios livros em sua página principal, onde deve clicar em um botão “+” e digitar o nome do livro;
- **Adicionar detalhes sobre o Livro:** Ao clicar em um livro da lista, o usuário é levado para uma tela onde o mesmo deve incluir dados adicionais sobre o livro que está escrevendo, bastando apenas clicar no botão “Editar” no canto superior direito.
- **Criar novos Capítulos:** Ao criar um novo livro, o usuário pode criar um novo capítulo, clicando no botão “Novo Capítulo”, onde irá surgir um botão “Capítulo 1” e assim por diante.
- **Escrever Livro:** Com o novo capítulo criado, o usuário pode escrever seu livro normalmente, já que a aplicação o leva para um editor de textos, seja para começar um novo conteúdo ou continuar de onde parou.

- **Pesquisar Livros:** É possível pesquisar por novos livros para ler e colaborar através de um campo de pesquisa.
- **Seguir Livro:** Ao conhecer o conteúdo escrito por um autor, o usuário também consegue seguir o livro, que será adicionado a uma lista de livros que estão sendo seguidos.
- **Comentar Livro:** Além de acompanhar o conteúdo sendo escrito, o usuário consegue comentar e discutir com outros para obter novas ideias ou melhorar o que já foi escrito.

### 3.2 Arquitetura

A arquitetura de uma aplicação consiste em satisfazer todas as camadas que serão utilizadas para o funcionamento da mesma. Sendo assim, na Figura 10 estão alinhadas as tecnologias e como interagem para que um usuário consiga utilizar a aplicação.

**Figura 10. Arquitetura da Aplicação**



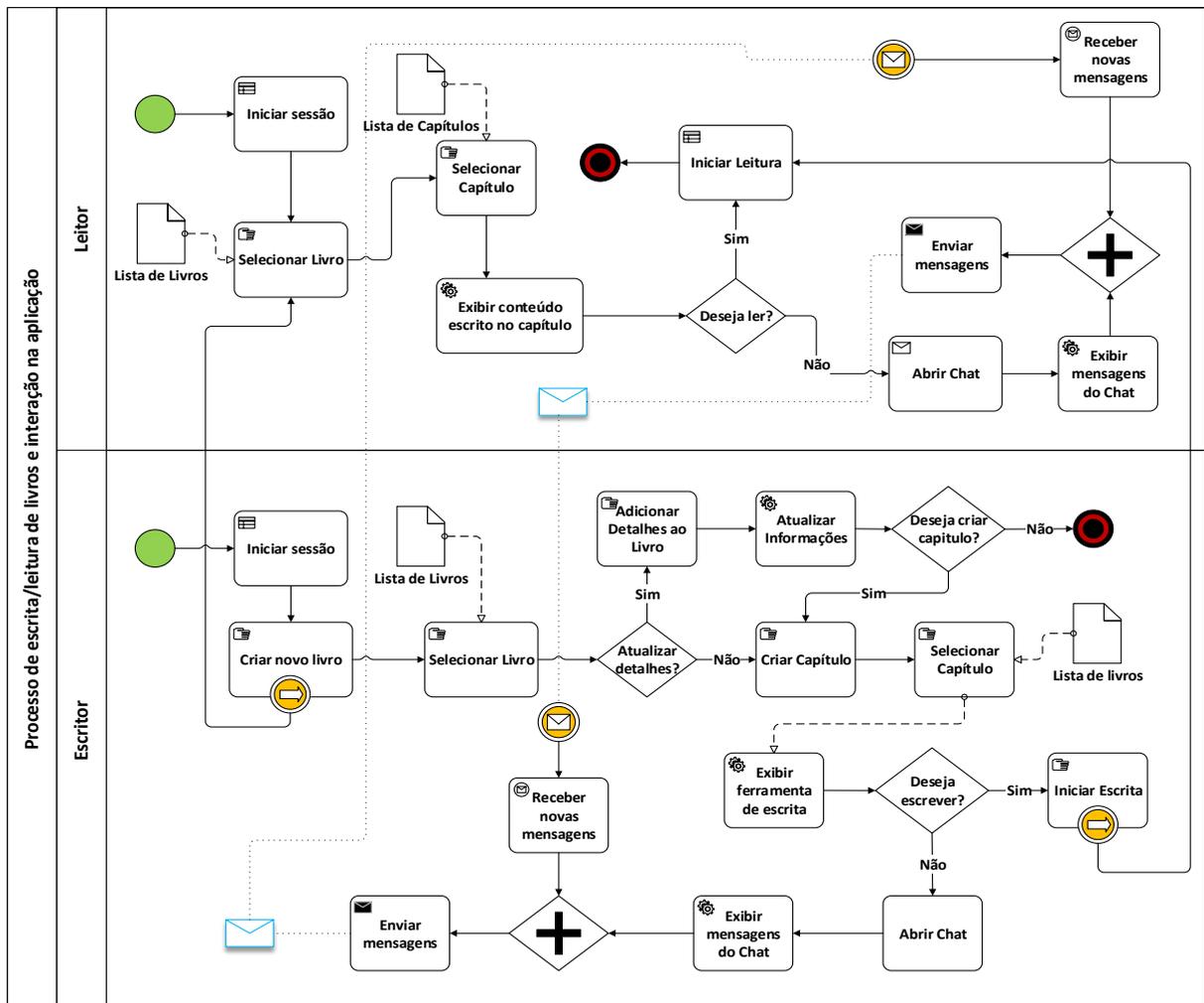
**Fonte: Desenvolvido pelo autor**

Para descrever seus componentes é preciso seguir o fluxo entre as camadas, que iniciam a partir da visualização, onde os usuários conseguem acessar e interagir a partir de qualquer dispositivo móvel já que o Ionic Framework torna possível o desenvolvimento multiplataforma. Já a camada de controle, representada pelo “Writo App”, é responsável pelos componentes de sincronização e validação onde está integrada a API Firebase para a persistência e sincronização dos dados em tempo real. Sendo assim, toda a camada de modelo é abstraída pelo serviço Firebase, que armazena e oferece recursos para a manipulação dos dados persistidos e recuperação dos dados através de Websockets em tempo real.

### 3.2.1 Modelagem das regras de negócio

A aplicação foi desenvolvida com base em uma visão estratégica de negócio, a qual poderá auxiliar em uma futura ampliação. Esta visão pode ser obtida a partir da elaboração de um diagrama de BPM, onde foram expostas todas as regras de negócio da aplicação, de acordo com a Figura 11 (BRAMBILLA et al, 2011).

**Figura 11 – Diagrama das Regras de Negócio da Aplicação**



**Fonte: Desenvolvido pelo Autor**

Com o diagrama, é possível identificar como ocorre cada processo na aplicação, desde as operações manuais que são realizadas pelo usuário, até os processos automáticos que são executados pela aplicação.

### 3.2.2 Técnicas de implementação

As técnicas de desenvolvimento existentes foram apresentadas com o intuito de identificar a trajetória do mercado referente ao desenvolvimento para múltiplas plataformas, dessa forma foram contempladas as que são habitualmente utilizadas e também as que estão deixando de existir por não mais atender ao mercado.

Para o desenvolvimento deste projeto foram analisadas as técnicas de desenvolvimento “*Responsive Web Design*” e “*Mobile First*”, onde a primeira visa a adaptação do que já existe para diferentes plataformas, e a segunda o desenvolvimento iniciando primeiramente para as plataformas móveis. As vantagens do desenvolvimento responsivo aparecem quando um conteúdo já existe e possui recursos que não foram desenvolvidos para serem executados em diferentes plataformas além do ambiente web, sendo assim, pode ser vantajosa à adaptação utilizando essa técnica. Entretanto, existem algumas desvantagens que tornam o design responsivo não ser uma boa alternativa para um projeto que se inicia.

As desvantagens estão presentes na quantidade de elementos que uma página Web para desktops pode suportar e que em dispositivos móveis não são suportados. E foi analisando esses pontos que tornou o “*Mobile First*” a melhor opção para o desenvolvimento da aplicação do projeto. Essa técnica tem como propósito principal priorizar o desenvolvimento mobile, e o objetivo é tornar possível atender todas as plataformas sem que haja problemas ou adaptações para diferentes tamanhos de telas onde o conteúdo será consumido.

O desenvolvimento priorizando os dispositivos móveis segue uma tendência conforme é demonstrado na Figura 12, onde até 2017 haverá até oito vezes mais celulares do que computadores pessoais, ou seja, todo o conteúdo disponibilizado será visualizado primeiro a partir desses dispositivos, algo que comprova a grande importância de investir em tal técnica de desenvolvimento.

**Figura 12. Dispositivos entregues ao redor do mundo por seguimento, 2012 – 2017**

| <b>Device Type</b>           | <b>2012</b>      | <b>2013</b>      | <b>2014</b>      | <b>2017</b>      |
|------------------------------|------------------|------------------|------------------|------------------|
| PC (Desk-Based and Notebook) | 341,263          | 315,229          | 302,315          | 271,612          |
| Ultramobile                  | 9,822            | 23,592           | 38,687           | 96,350           |
| Tablet                       | 116,113          | 197,202          | 265,731          | 467,951          |
| Mobile Phone                 | 1,746,176        | 1,875,774        | 1,949,722        | 2,128,871        |
| <b>Total</b>                 | <b>2,213,373</b> | <b>2,411,796</b> | <b>2,556,455</b> | <b>2,964,783</b> |

**Fonte: MOHOROVICIC, 2013**

A aplicação dos conceitos do “*Mobile First*” partiram da escolha do framework de desenvolvimento, o qual foi escolhido o Ionic, uma ferramenta que permite o desenvolver aplicações híbridas para múltiplas plataformas móveis, inicialmente para iOS e Android. Durante o desenvolvimento, foram criadas funcionalidades que podem ser acessados em qualquer tamanho de tela de dispositivo móvel, e também facilmente adequado à telas maiores.

Além disso, a utilização de tal framework, permite que uma página web utilize as mesmas regras de negócio que uma aplicação móvel, já que a camada de visualização é totalmente web. Com isso, faz valer a possibilidade de ampliação para aplicações web que serão utilizadas em desktops e notebooks, sem que haja necessidade fazer limitações nas funcionalidades que já são utilizadas na aplicação móvel.

Sendo assim, a utilização do “*Mobile First*” como técnica de desenvolvimento, foi essencial nas decisões de escolha das tecnologias que fizeram parte da aplicação, além de

proporcionar uma possibilidade de ampliação futura sem perder qualidade ou fazer limitações como aconteceria em outras técnicas.

### **3.3 Escrita de Livros Colaborativa**

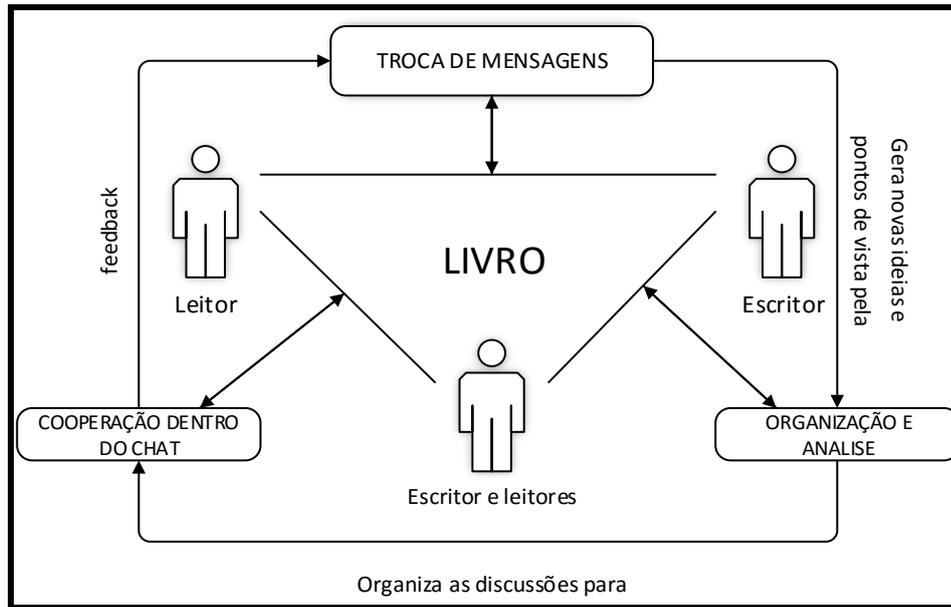
A escrita de livros tradicional para pessoas que não possuem experiência na área pode ser muito desafiadora e frustrante, isto ocorre porque a primeira interação entre o escritor e o leitor somente acontece quando o livro finalmente é disponibilizado oficialmente por alguma editora para o público. Dessa forma, o leitor pode não gostar do resultado final e isso pode levar a muitos lançamentos até que finalmente o escritor seja notado por ter encontrado aquilo que o leitor procure em um livro seu.

Para que seja possível diminuir a quase zero o tempo de resposta dos leitores, foi adotado na aplicação a partir do Modelo 3C de colaboração, uma estrutura de colaboração entre escritores e leitores.

Esta estrutura consiste da seguinte maneira (Figura 13):

1. O escritor escreve o capítulo de seu livro que em tempo real é persistido e disponibilizado ao público.
2. Os leitores acompanham o livro e compartilham suas ideias e sugestões através do chat disponível dentro de cada capítulo, tudo em tempo real.
3. O escritor participa e analisa as discussões com os leitores e as organiza para que seja mais produtivo e o assunto voltado ao livro/capítulo que está sendo escrito.

**Figura 13. Modelo 3C de colaboração na aplicação**



**Fonte: Elaborado pelo autor**

A vantagem de se utilizar o Modelo 3C é que a colaboração deve ocorrer de forma natural, já que o mesmo modelo já foi empregado em outras aplicações colaborativas e em redes sociais. Sendo assim, o conceito do Modelo 3C deve ser caracterizado como uma estrutura de sistema colaborativo network, onde não existem restrições, hierarquia e principalmente qualquer tipo de formalidade.

### 3.4 Ferramentas

As ferramentas de desenvolvimento tais como frameworks e bibliotecas que já possuem um aprofundamento maior em recursos que podem ser utilizados de maneira instantânea, certamente são ótimos para um ponto de partida com mais garantia, sendo assim possível implementar uma aplicação funcional de menor custo/tempo e maior nível de qualidade.

Dentre as ferramentas apresentadas durante a revisão bibliográfica as que mais se mostraram ser adequadas para o projeto foram: Ionic Framework, Firebase e suas bibliotecas Angularfire e também o Firepad para a sincronização de texto. As ferramentas escolhidas

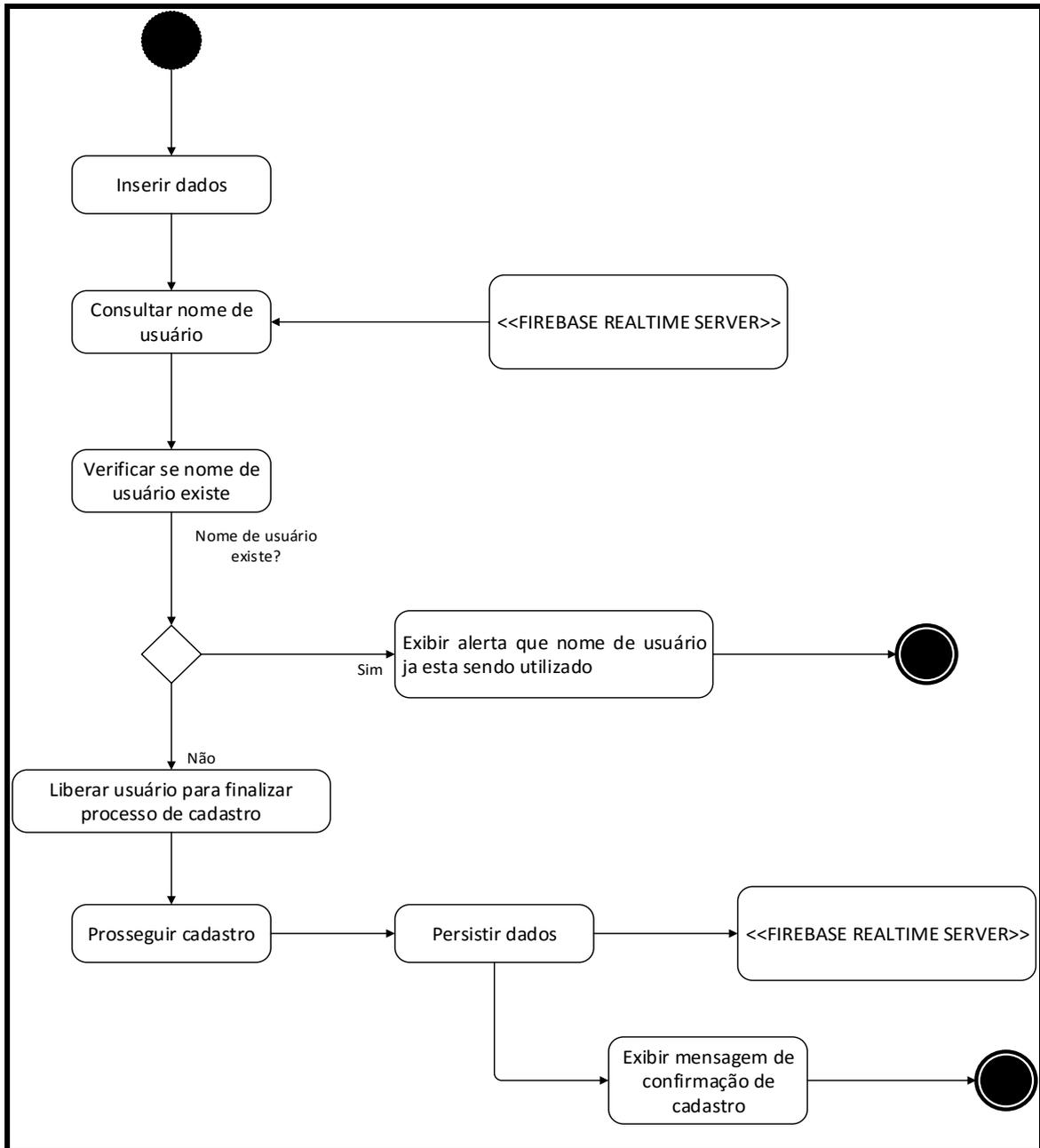
possuem um conjunto de funcionalidades e recursos que facilitam o desenvolvimento multiplataforma e principalmente que atendeu aos requisitos do projeto, o que inclui o comunicação assíncrona.

Logo, este capítulo determinou cada processo de desenvolvimento utilizando cada tecnologia para se obter a aplicação esperada, dessa forma, cada etapa determinou como foi possível implementar todos os recursos a serem utilizados pelo usuário e suas limitações.

### 3.4.1 Gestão de Usuários

- **Cadastro de Usuários:** Método utilizado para cadastrar novos usuários. Devem ser inseridos os dados de login e senha que o novo usuário desejar. Para verificar a existência de um possível cadastro existente com o mesmo nome inserido, o processo de cadastro é validado conforme a seguir (Figura 14):
  - a) O futuro usuário informa seu nome de login.
  - b) A aplicação faz comunicação com a base dados através de métodos disponibilizados pelo serviço Firebase através de uma API REST.
  - c) O nome que o usuário inseriu é comparado em tempo real para verificar a existência na base de dados;
  - d) Caso a verificação encontre o nome, é exibido um alerta que o nome já está sendo utilizado.
  - e) Caso a verificação não encontre o nome, o usuário é liberado para prosseguir com o cadastro.
  - f) O usuário deve clicar no botão disponível para prosseguir com o cadastro após a confirmação do login, e ter inserido uma senha.
  - g) Após prosseguir com o cadastro, os dados são persistidos no banco de dados a partir de uma conexão utilizando os métodos da API REST, e uma mensagem de confirmação de cadastro é exibida na tela.

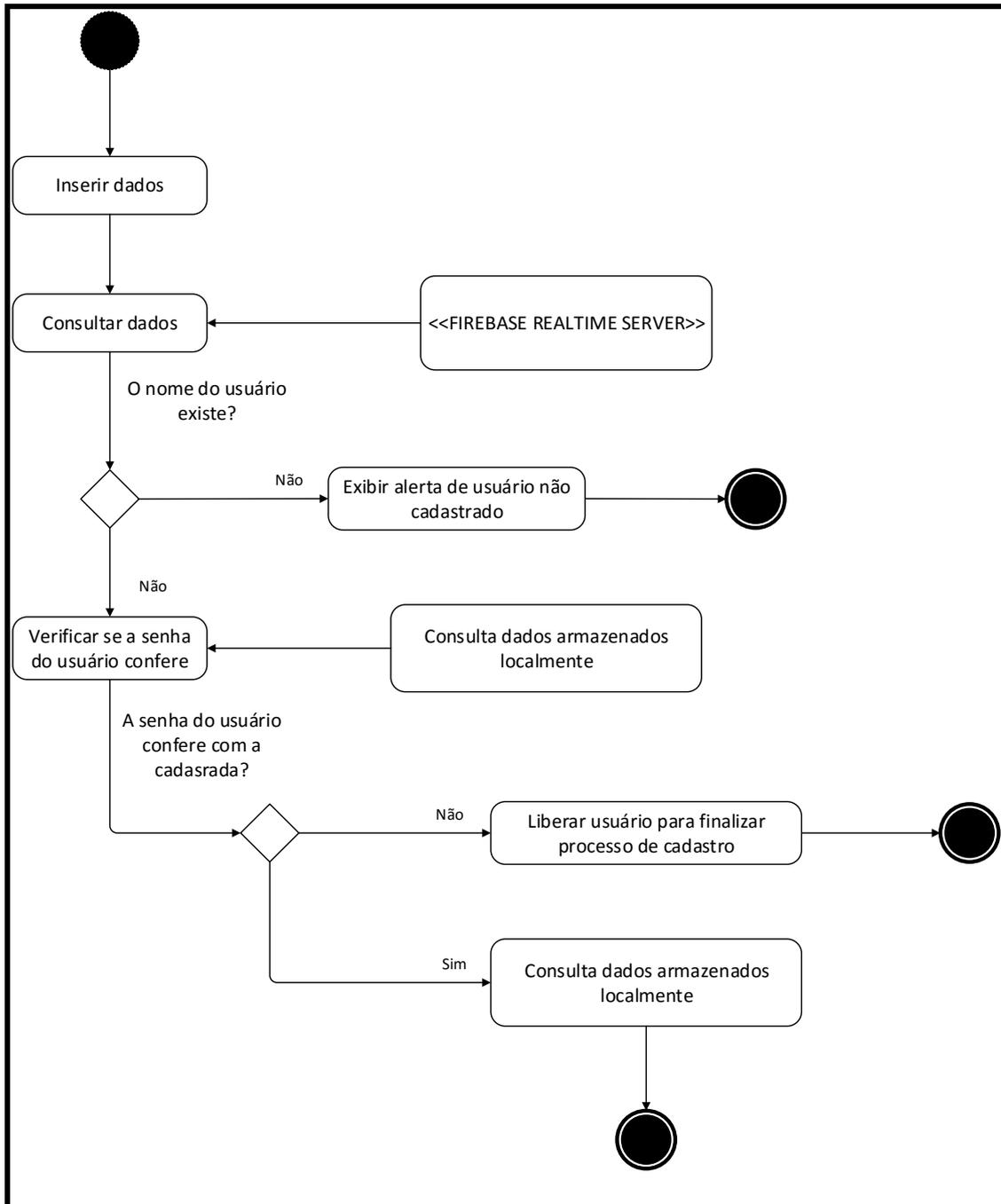
Figura 14. Diagrama do Método de Criação de Usuário



Fonte: Elaborado pelo autor

- **Fazer Login:** Método utilizado para acessar a aplicação. Deve ser inserido o login e a senha do usuário. Para validar os dados do usuário e garantir que o mesmo esteja cadastrado no sistema antes de liberar o acesso, o processo acontece conforme apresentado na Figura 15.
  - a) O usuário informa os dados de acesso.
  - b) Os dados do usuário são consultados na base de dados através de métodos disponibilizados pelo serviço Firebase através de uma API REST.
  - c) Caso o nome esteja cadastrado, a senha do usuário é comparada com os dados armazenados na DOM do Angular localmente.
  - d) Caso o nome não conferir, a aplicação exibe uma mensagem informando que o mesmo não está cadastrado no sistema.
  - e) Caso a senha conferir, o token do cadastro do usuário é enviado através de uma lista remota e libera o acesso à aplicação ao usuário.
  - f) Caso a senha não conferir, a aplicação exibe uma mensagem informando que a senha é inválida.

Figura 15. Diagrama do Método de Login



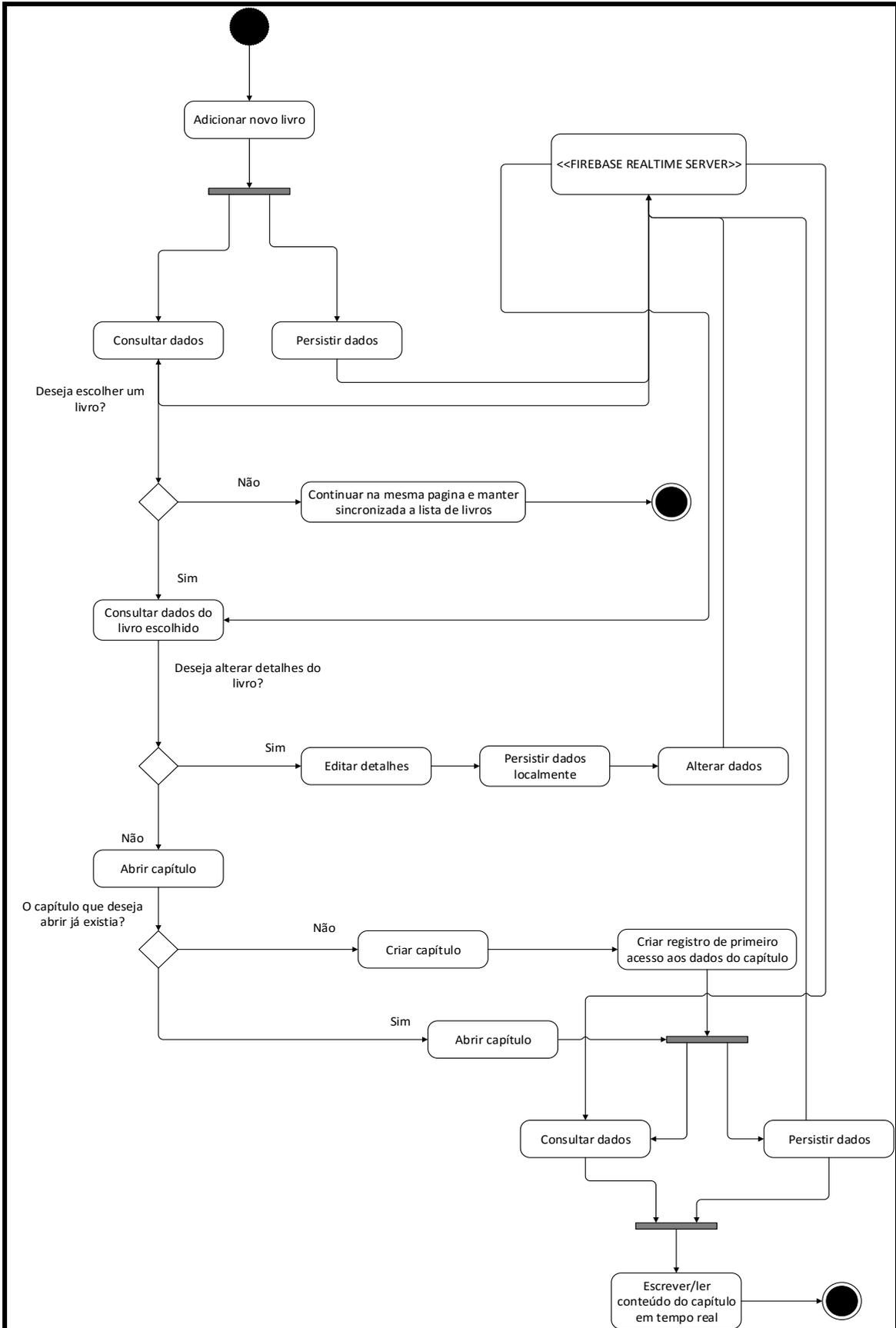
Fonte: elaborado pelo autor

### 3.4.2 Gestão de Livros

- **Criar Livro:** Este método é utilizado para criar novos livros. O usuário pode criar um novo livro, que é sincronizado em todos os dispositivos em tempo real, conforme a seguir (Figura 16)
  - a) O usuário clica no botão com o símbolo “+” no canto superior direito para criar um novo livro em sua lista.
  - b) Uma caixa de texto solicita o nome do livro a ser criado e o usuário deve informar e clicar em OK para confirmar.
  - c) Após confirmar, o método que adiciona o livro recebe o nome informado e utilizando um método de comunicação com a base de dados, o mesmo é persistido.
  - d) O livro é sincronizado na lista de livros que o usuário possui utilizando os métodos da API REST.
  
- **Editar Detalhes:** Este método é utilizado para adicionar detalhes ao livro adicionado pelo usuário. Após ter criado o livro, o usuário o seleciona para então adicionar detalhes como: título, tema e sinopse. O processo ocorre conforme apresentado abaixo (Figura 16):
  - a) O usuário seleciona o mesmo e clica no botão “editar” no canto superior direito.
  - b) O aplicativo abre uma página com campos referente aos detalhes do livro, com dados existentes para serem alterados.
  - c) O usuário insere as informações que serão atualizadas e clica em “Salvar Alterações”.
  - d) Um método que recebe localmente os dados atualiza as informações no banco de dados utilizando métodos disponibilizados pelo serviço Firebase através de uma API REST.
  - e) O aplicativos consulta os dados do livro selecionado e apresenta na tela utilizando os métodos da API REST.

- **Criar Capítulo:** Nessa funcionalidade é possível criar capítulos para o livro. Dentro do livro selecionado, é possível criar capítulos onde o mesmo pode inserir conteúdo ao mesmo. O processo ocorre conforme apresentado abaixo (Figura 16):
  - a) O usuário clica no botão “Adicionar Capítulo” para criar um novo capítulo.
  - b) O método responsável por adicionar o capítulo cria um número sequencial que representa o capítulo dentro do livro no serviço de persistência Firebase.
  - c) Para cada capítulo criado, um novo botão deverá aparecer na lista de capítulos do livro, utilizando os métodos de consulta da API REST.
  
- **Editar/Visualizar Conteúdo do Capítulo:** Esta funcionalidade é responsável pela edição e visualização do conteúdo escrito em um capítulo. Ao abrir um capítulo, é possível o usuário escrever e editar o conteúdo do mesmo, onde os usuários que não são proprietários do livro podem apenas visualiza-lo. O processo deve acontecer conforme a seguir (Figura 16):
  - a) Ao clicar em um capítulo, o método responsável por consulta e persistência de texto busca o conteúdo escrito pelo usuário e posiciona o ponteiro onde o mesmo parou da última vez.
  - b) O conteúdo escrito pelo usuário é persistido em tempo real, para isto um canal de comunicação bidirecional é aberto, o qual é responsável por transmitir as requisições do método da API REST que verifica as alterações no conteúdo escrito e o atualiza no registro do Firebase.

Figura 16. Diagrama dos Métodos de Gerência de Livros

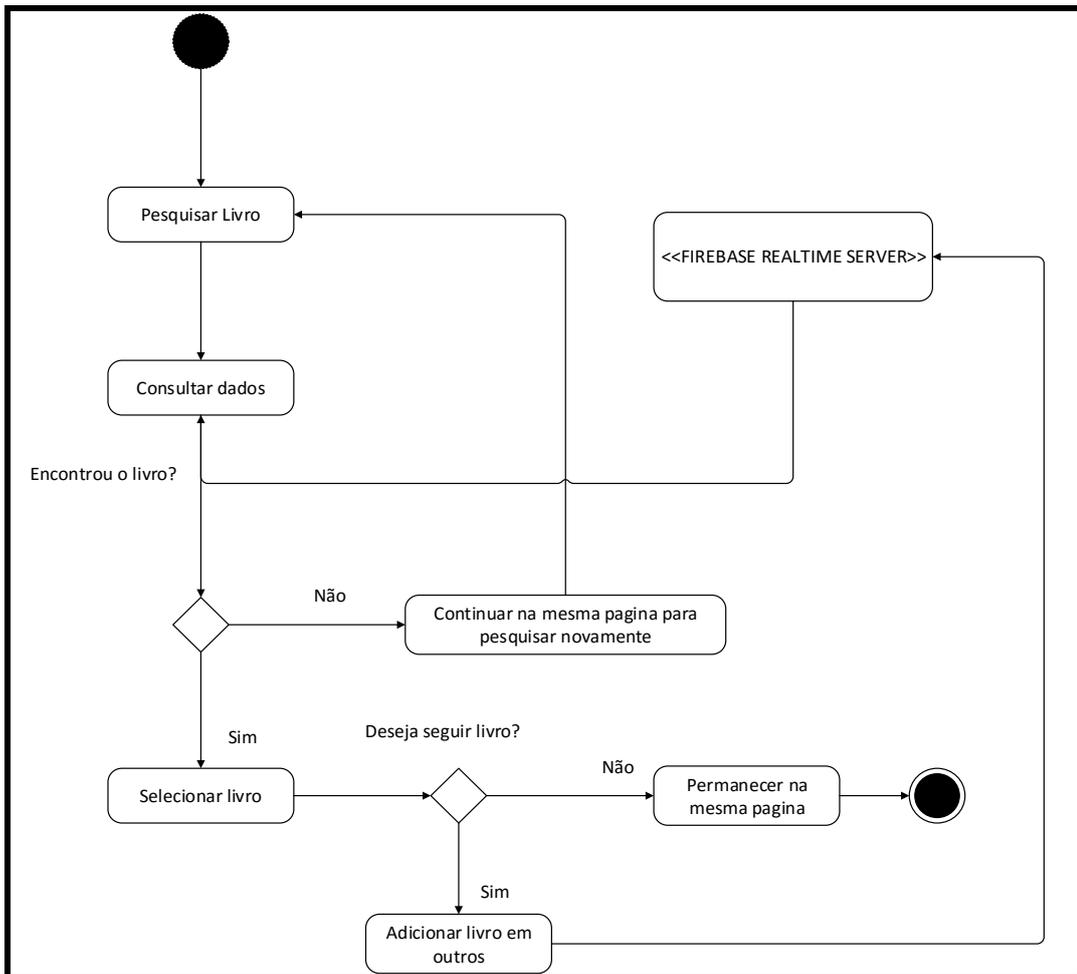


Fonte: Desenvolvido pelo autor

- **Pesquisar Livros:** Este método é utilizado para pesquisar outros livros e segui-los. Esta funcionalidade permite que o usuário pesquise novos livros para que possam acompanhar e adiciona-los a uma lista. O processo de pesquisa e adicionar livros a lista ocorre da seguinte forma (Figura 17):

- O usuário abre o menu de opções e seleciona a opção pesquisar.
- Digita o nome do livro desejado.
- Seleciona um livro.
- Caso o usuário decidir acompanhar o livro escolhido, ele poderá clicar em seguir livro, então o livro será adicionado à lista de outros livros.
- Caso o mesmo não tenha interesse em seguir, ele permanecerá na mesma tela.

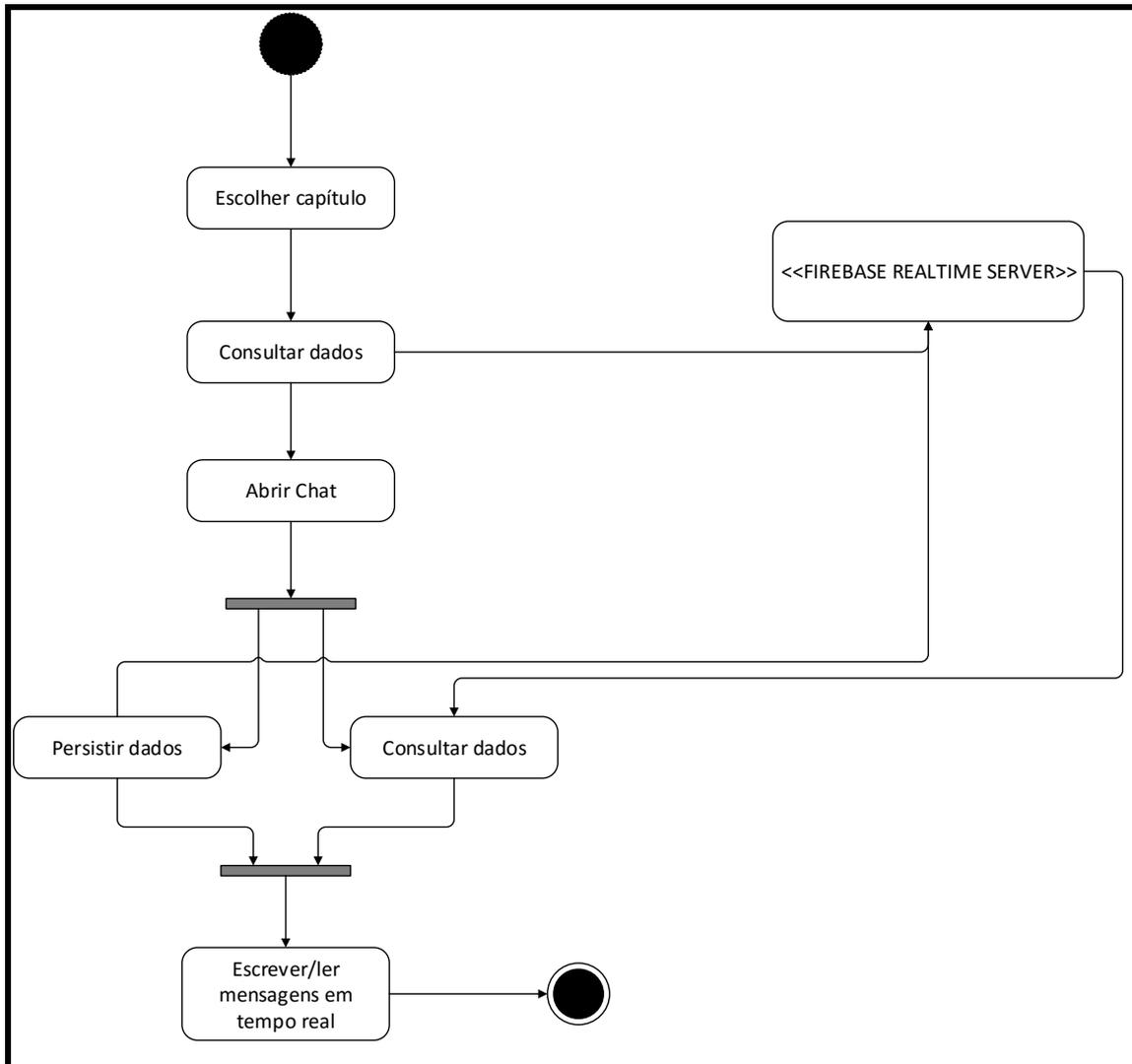
**Figura 17. Pesquisa de livros**



**Fonte: Desenvolvido pelo autor**

- **Abrir Chat:** Este método é utilizado para iniciar o chat do capítulo. Esta funcionalidade dá início a um chat onde os usuários podem discutir sobre o capítulo que estão lendo e passam informações que são analisadas pelo autor do livro. O processo para abrir o chat e interagir ocorre conforme descrito abaixo (Figura 18):
  - f) O usuário abre um capítulo e clica no botão “Chat”.
  - g) Uma janela é aberta e o método da API REST realiza uma consulta para buscar o conteúdo escrito por outros usuários.
  - h) Um canal de comunicação bidirecional é aberto para receber novas conversas.
  - i) O usuário pode utilizar o campo localizado na parte inferior da tela para escrever, e clicar no botão “Enviar” para enviar o que foi escrito.
  - j) O conteúdo é recebido pelo método da API REST responsável por persistir os dados ao chat.

Figura 18. Diagrama do Método de Chat



Fonte: Desenvolvido pelo autor

### 3.4.3 Escrita de Livro em Tempo real

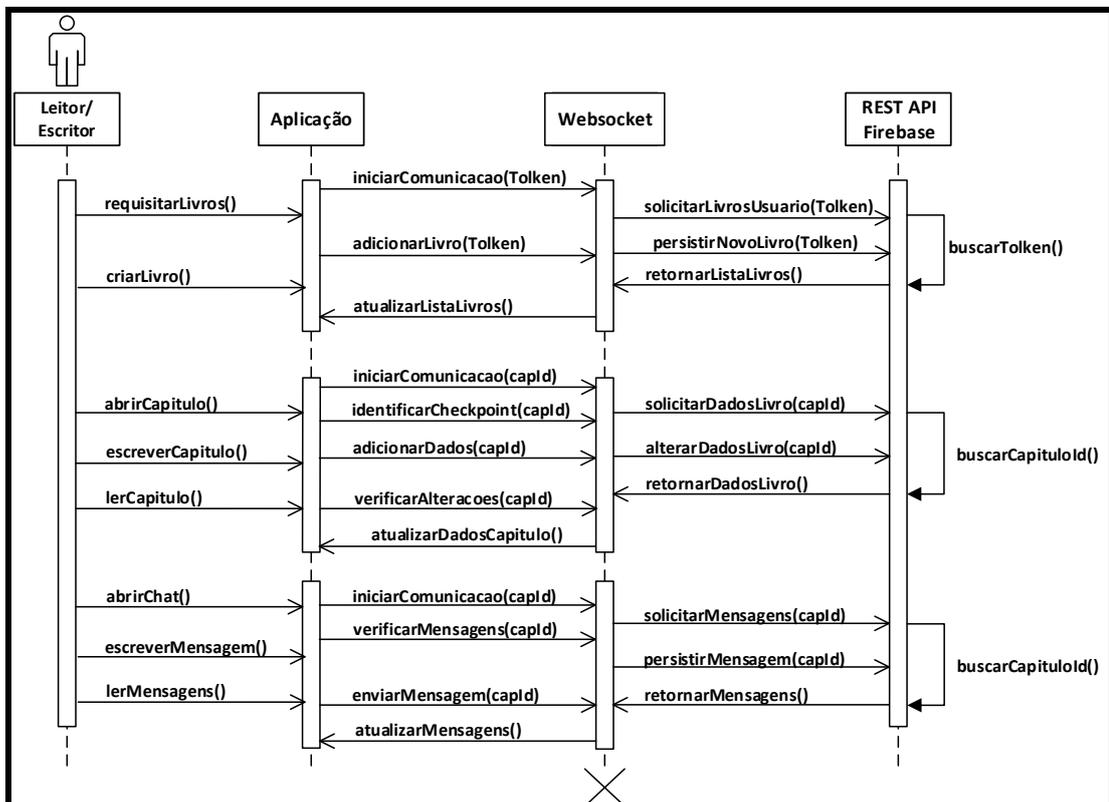
A sincronização em tempo real é um dos recursos que eleva o nível interação entre os usuários, e com isso, aproxima os leitores dos escritores, trazendo os para dentro do projeto de escrita do livro que estão lendo e interagindo.

A princípio todas as requisições são sincronizadas em tempo real através de canais bidirecionais chamados “WebSockets”, os principais recursos são: criação de novos livros, criação de capítulos, escrita, leitura, chat e recuperação das informações que são exibidas na tela. Esses canais são responsáveis por levar cada requisição que a aplicação faz ao mesmo tempo que receber dados que foram solicitados.

Uma API REST oferecida pelo serviço Firebase, permite controlar a persistência e recuperação de dados de maneira muito ágil, e por se tratar de um banco não relacional que possui recursos para atender múltiplas requisições em tempo real, o serviço aliado aos canais bidirecionais permitiram a implementação dos recursos de escrita da aplicação.

Para exemplificar como ocorre a comunicação e troca de requisições, foi elaborado um diagrama de sequência, conforme Figura 19 (MILICEV, 2009).

**Figura 19 – Diagrama de Requisições da Aplicação**



Fonte: Desenvolvido pelo autor

O uso de ferramentas como o Firebase e a API Firepad, foram essenciais na implementação dos mecanismos de sincronização para o desenvolvimento das principais funcionalidades da aplicação, e dessa forma, o foco principal passou a ser a apenas o conhecimento de como deveria ser implementado e utilizado cada método disponível para consumir tais recursos.

### 3.5 Considerações Finais

A aplicação possui um conjunto de funcionalidades que adicionam e enriquecem a utilização da mesma, de forma que seja fácil criar livros, adicionar detalhes ao livro, criar capítulos personalizados, escrever e formatar o conteúdo de cada capítulo e interagir com outros usuários através do chat dentro de cada capítulo.

Já a arquitetura, foi possível verificar como funciona cada camada da aplicação, sendo ela dividida entre visualização, controle e modelo, onde a primeira pode ser utilizada a partir de dispositivos móveis que possuem telas e sistemas operacionais distintos, a segunda é responsável pelas regras de negócio, ou seja, onde todas as requisições são gerenciadas e passadas para a próxima camada, a qual recebe e atende cada requisição de dados, persistindo e realizando alterações.

O desenvolvimento foi guiado pela técnica “*Mobile first*”, o qual prioriza iniciar a partir de dispositivos móveis, para garantir que os recursos implementados na aplicação possam estar disponíveis em todos os dispositivos que possuem telas menores, e serem adaptados para maiores sem que haja necessidade de limitar os recursos já implementados.

Sobre a escrita colaborativa, foi elaborado com base no Modelo 3C de colaboração, um método padronizado de como poderá acontecer o trabalho conjunto aliado à escrita de livros, para isso, são distribuídos papéis para cada usuário, sendo o escritor e o leitor responsáveis pela comunicação, em seguida o escritor por coordenar as discussões, e por último o leitor por colaborar com o escritor, passando seu feedback sobre o conteúdo.

As ferramentas escolhidas para implementação foram baseadas na técnica de desenvolvimento, e por priorizar os dispositivos móveis, o Ionic Framework foi elegido como a melhor ferramenta por ser possível desenvolver para múltiplas plataformas e oferecer muito conteúdo que ajudou na resolução de cada desafio. Além dessa ferramenta, também foi escolhido o Firebase, um serviço para persistência e recuperação de dados em tempo real, o

qual oferece uma API que ajudou na implementação dos recursos colaborativos em tempo real.

Por fim, foi especificado como ocorrem cada requisição em tempo real, sendo essas sincronizadas através de canais bidirecionais chamados “*Websockets*”, os quais são utilizados para chamar métodos da API Rest fornecida pelo serviço Firebase, que recebe e atende a todas as requisições em tempo real.

## 4 RESULTADOS

O resultado desse trabalho foi uma aplicação móvel multiplataforma, que pode ser acessada nas plataformas Android e iOS. Como resultado, será apresentado o uso da aplicação em imagens que exemplificam cada funcionalidade.

### 4.1 Aplicação Multiplataforma

A aplicação foi desenvolvida uma única vez para ambas as plataformas, sendo assim, em cada imagem exibe como a interface se comporta no iOS e ao lado no Android, imagens que foram obtidas a partir de um serviço do framework que permite a visualização pelo navegador.

#### 4.1.1 Cadastro de Usuário

Na Figura 20 ilustra a tela de cadastro de usuários da aplicação que permite o registro de novos usuários, assim como escolher fazer o login caso o mesmo já possua cadastro.

**Figura 20. Cadastro de Usuario**

| IOS                           | Android                       |
|-------------------------------|-------------------------------|
| <b>Cadastro</b>               | <b>Cadastro</b>               |
| Seu Nome                      | Seu Nome                      |
| Login                         | Login                         |
| Senha                         | Senha                         |
| <b>Cadastrar</b>              | <b>Cadastrar</b>              |
| <b>Ou se tiver cadastrado</b> | <b>Ou se tiver cadastrado</b> |
| <b>Log in</b>                 | <b>Log in</b>                 |
|                               |                               |

### 4.1.2 Login de Usuário

Na Figura 21 ilustra o login da aplicação, que permite o acesso a aplicação mediante a autenticação dos dados de usuário e senha.

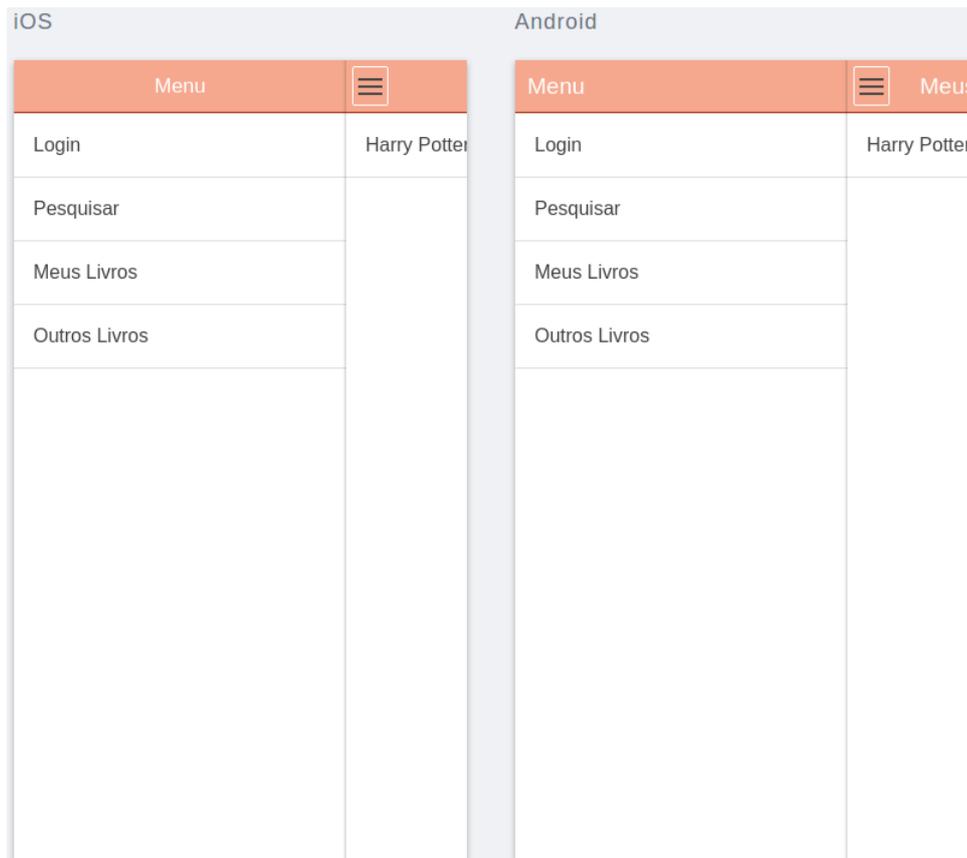
**Figura 21. Login de Usuário**

The figure displays two mobile application login screens side-by-side, labeled 'iOS' and 'Android'. Both screens have a light gray header bar. The iOS screen has 'Login' on the left and 'Fechar' on the right. The Android screen has 'Login' on the left and 'Fechar' on the right. Below the header, both screens have two text input fields: the first is labeled 'Login' and the second is labeled 'Senha'. Below the input fields is a prominent red button with the text 'Log in' in white. The bottom half of each screen is a large, empty white rectangular area.

### 4.1.3 Menu de Opções

Na Figura 22 é apresentado o menu de opções que o usuário possui acesso após realizar login, para exibi-lo basta apenas clicar no botão superior esquerdo da tela.

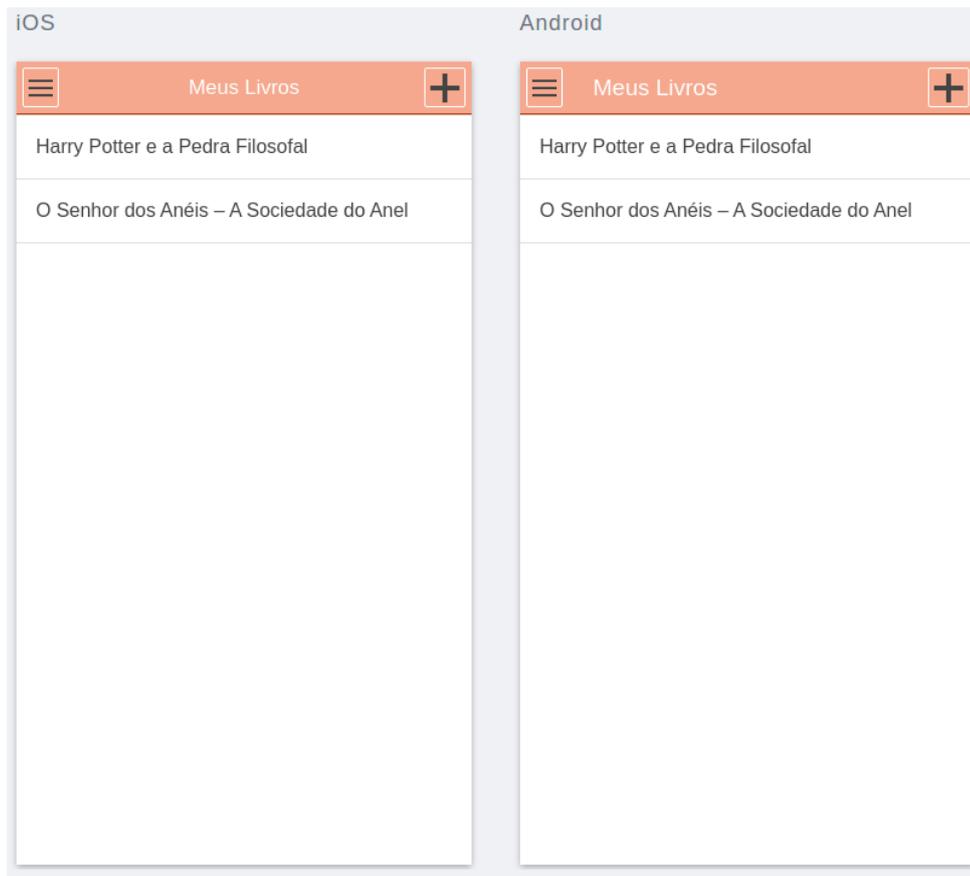
**Figura 22. Menu de Opções**



#### 4.1.4 Lista de Livros do usuário

Na Figura 23 ilustra a lista de livros do usuário, que é carregada após o início da sessão e permanece sincronizada enquanto aberta.

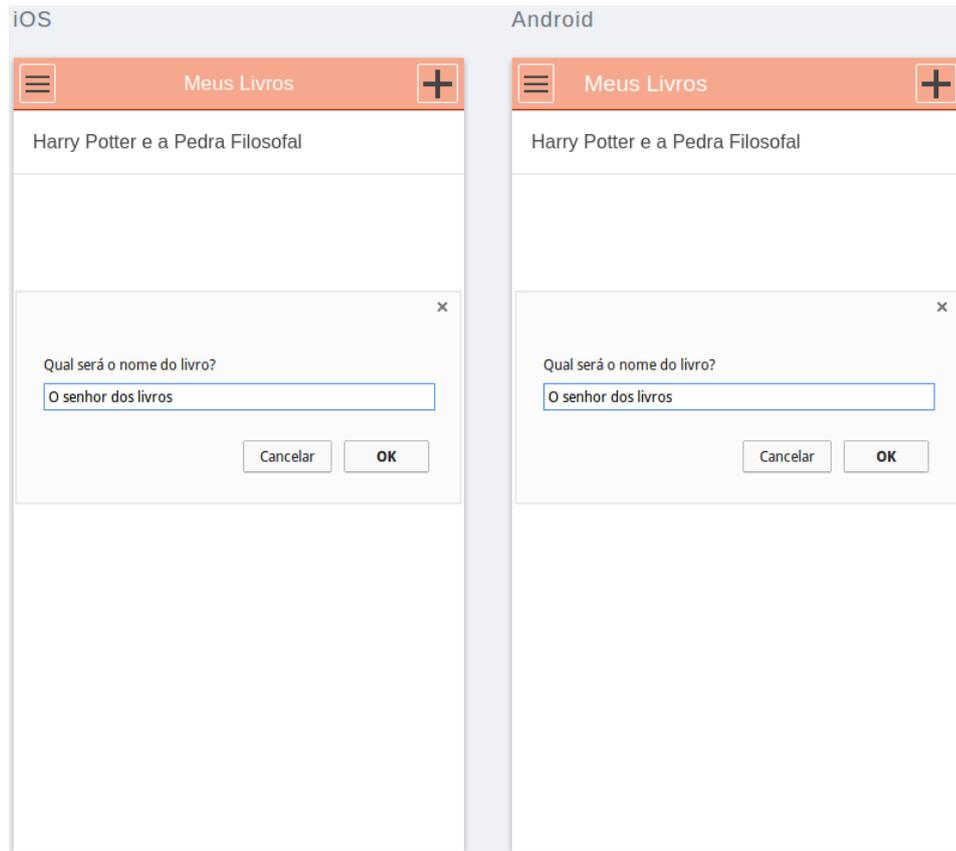
**Figura 23. Lista de Livros do Usuário**



### 4.1.5 Criar Livro

Na Figura 24 ilustra a criação de novos livros, que pode ser iniciada ao clicar no botão localizado no canto direito da tela, então a aplicação solicitará um nome.

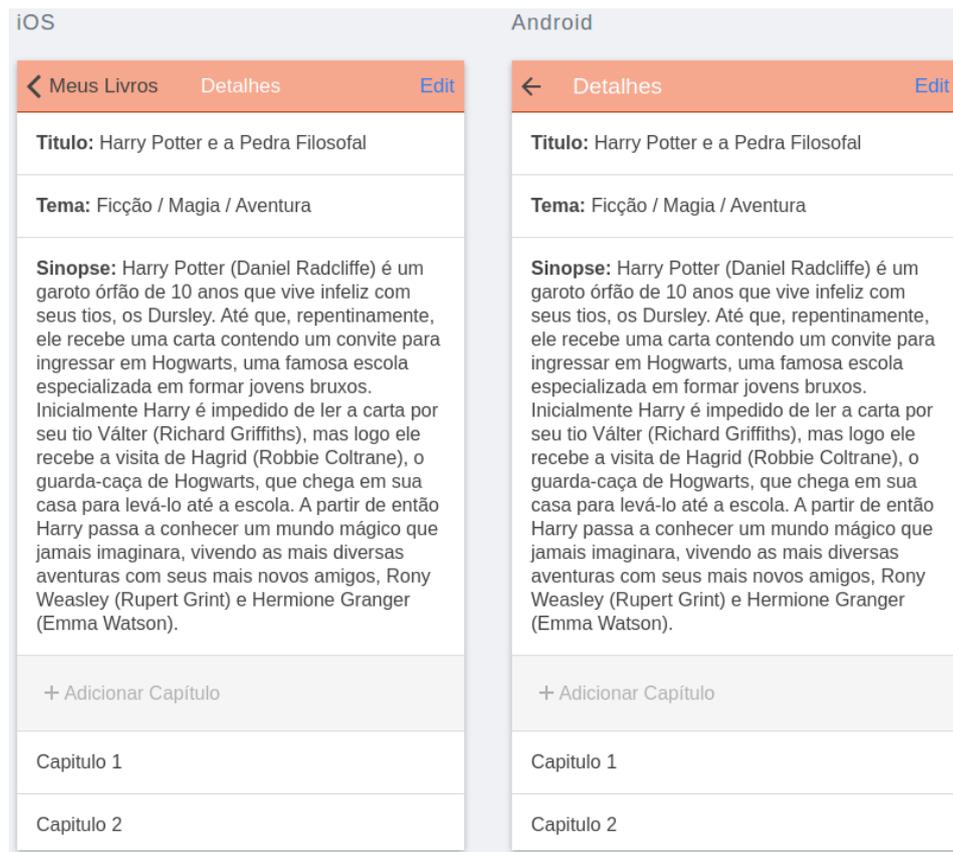
**Figura 24. Criação de Livros**



### 4.1.6 Detalhes do Livro

Na Figura 25 ilustra a tela de detalhes do livro, esses detalhes podem a desejo do usuário.

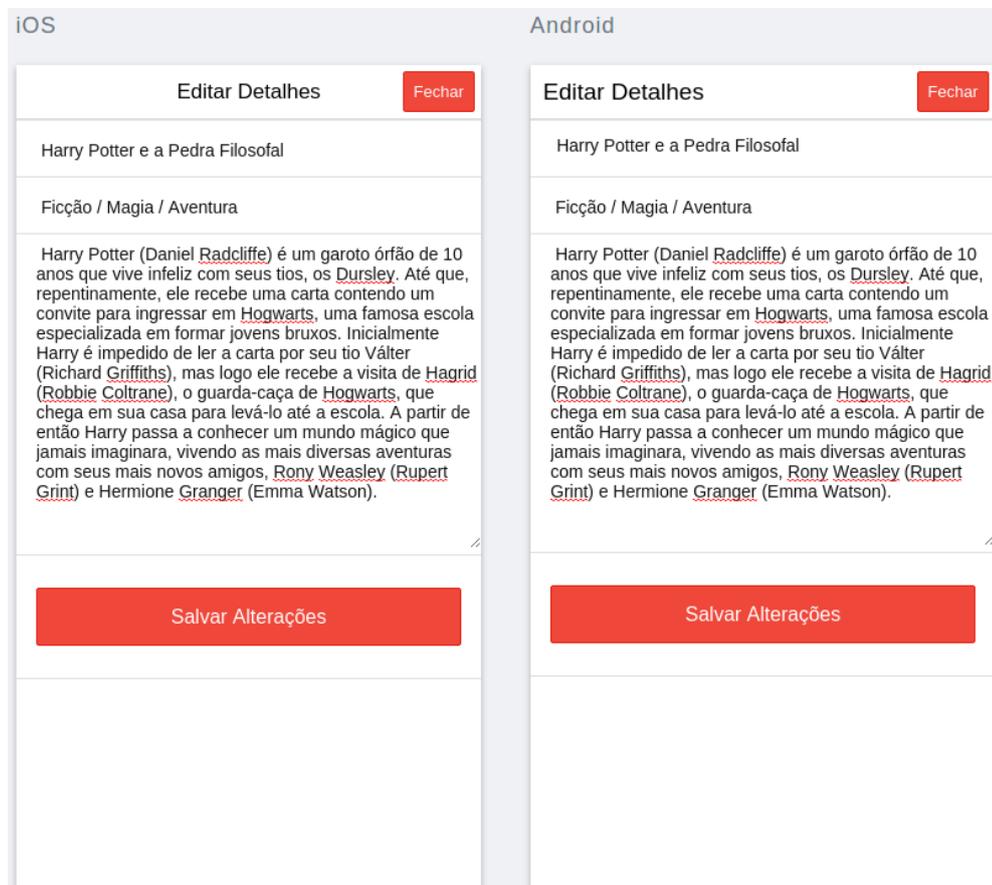
**Figura 25. Detalhes do Livro**



### 4.1.7 Editar Detalhes do Livro

Na Figura 26 ilustra o funcionamento da edição de detalhes do livro que foi criado, dessa forma o usuário poderá alterar as informações que inseriu na criação e adicionar outras informações

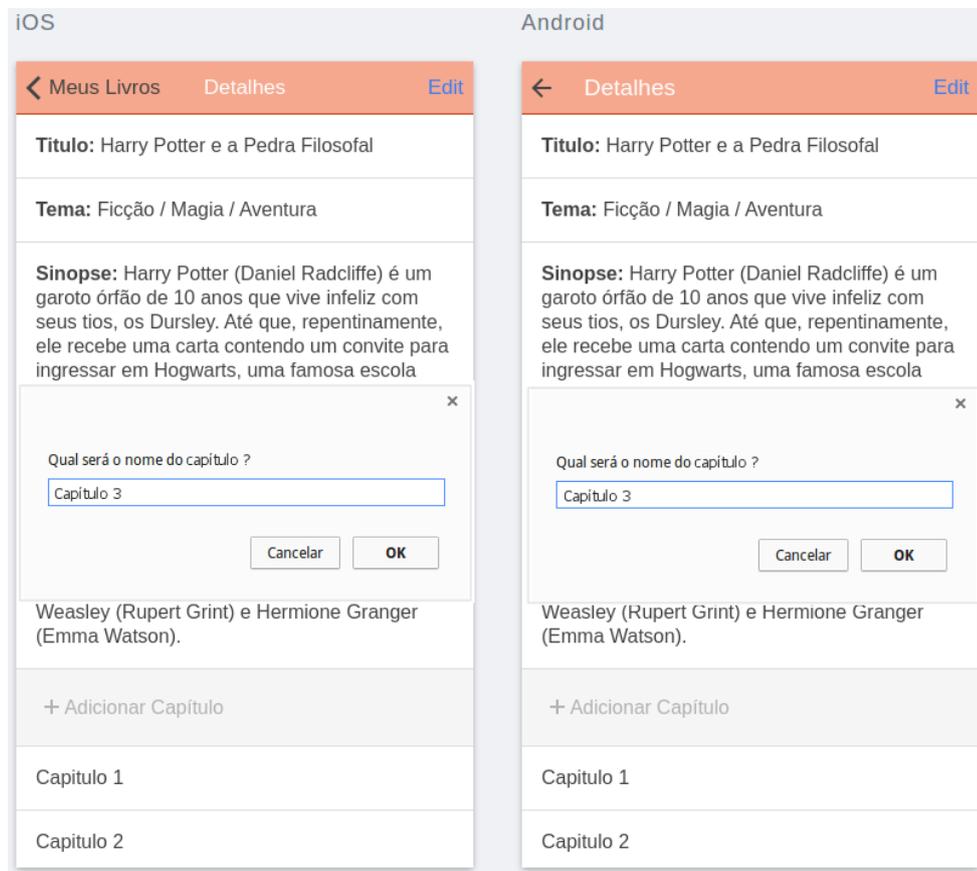
**Figura 26. Edição Detalhes do Livro**



### 4.1.8 Criar capítulos do livro

Na Figura 27 ilustra o funcionamento da criação de capítulos de um livro, onde o usuário deve clicar no botão “adicionar capítulo” para que a aplicação solicite um nome para capítulo.

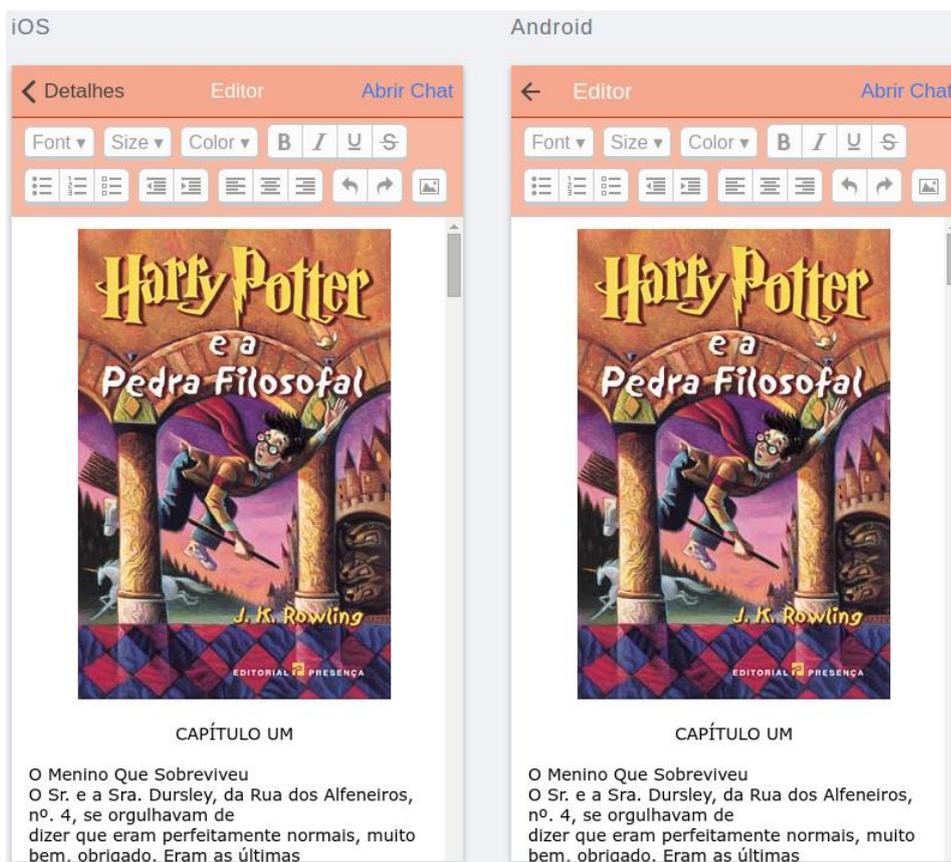
**Figura 27. Criação de Capítulos**



### 4.1.9 Escrever Capítulo

Na Figura 28 exibe o editor de capítulo, onde o usuário dono do livro pode editar o conteúdo de seus capítulos que será exibido aos leitores.

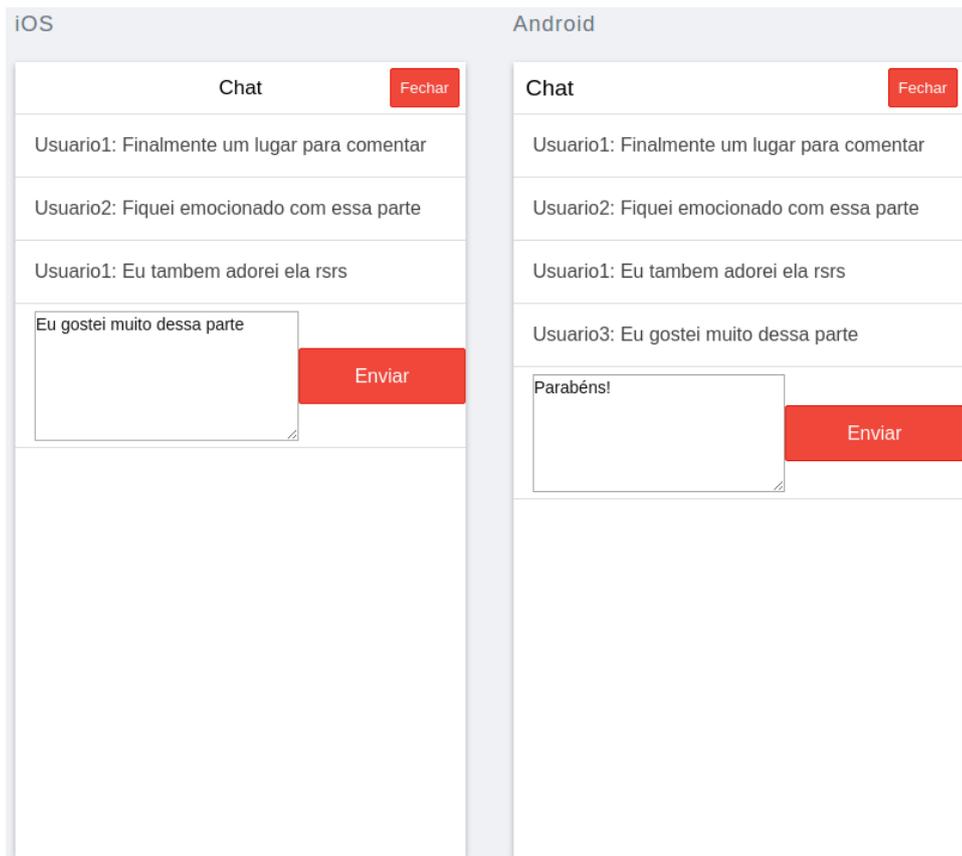
**Figura 28. Editor de Capítulo**



#### 4.1.10 Abrir Chat

Como é possível observar na Figura 29, a aplicação permite que o usuário que estiver escrevendo ou lendo um capítulo acesse o chat para interagir com outras pessoas.

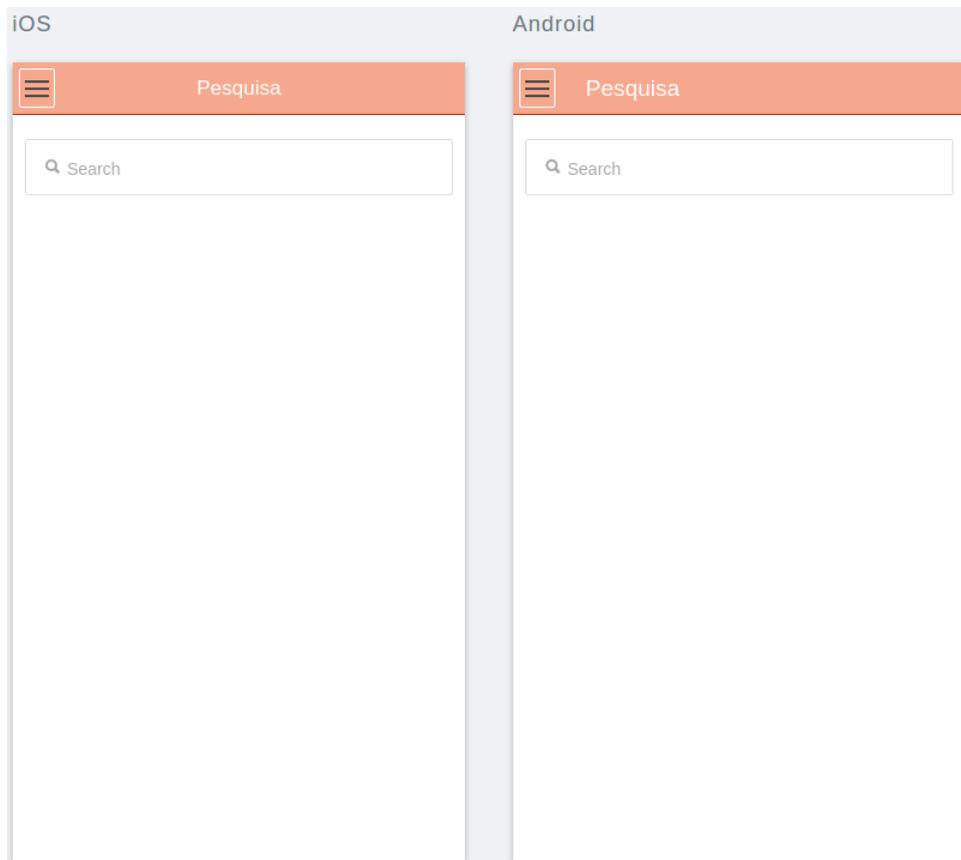
**Figura 29. Chat do Capítulo**



### 4.1.11 Pesquisar Livro

Na Figura 30 é apresentado a opção de pesquisa por novos livros, onde o usuário pode conhecer novos livros.

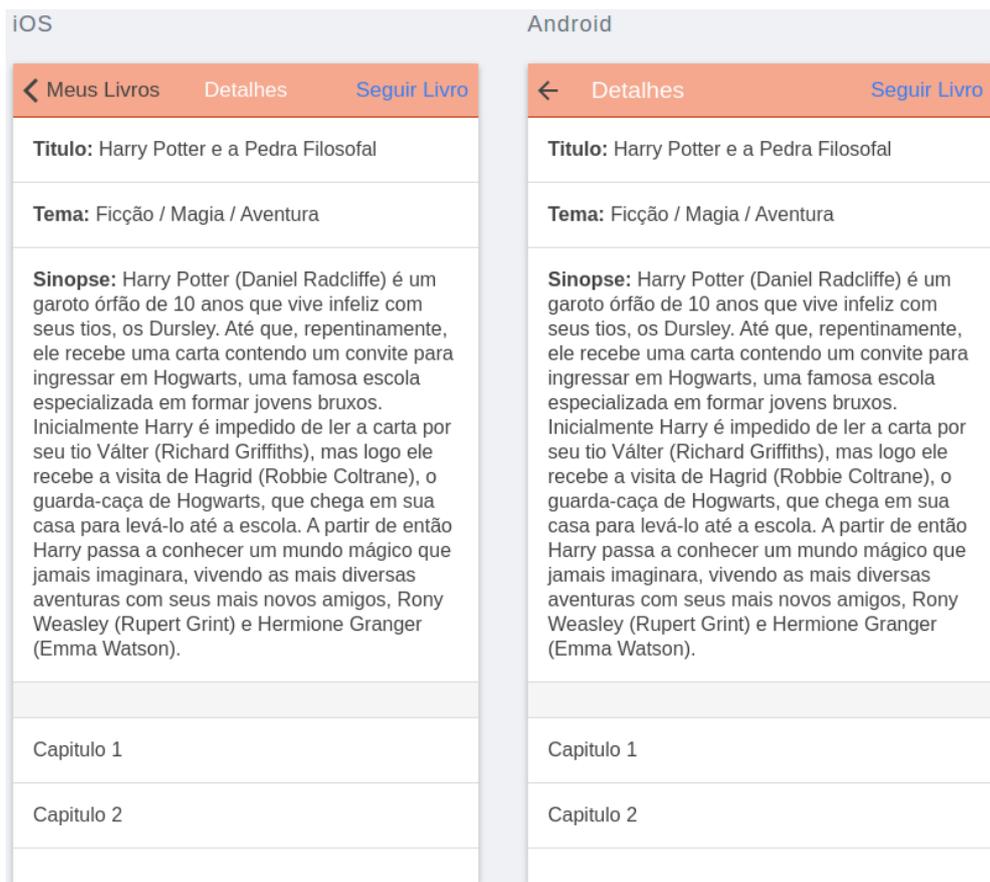
**Figura 30. Pesquisa de Livros**



#### 4.1.12 Seguir Livro

Conforme é apresentado na Figura 31, a aplicação permite que o usuário siga um livro de outro usuário, a partir disso o livro será incluído na opção “Outros Livros” do menu.

**Figura 31. Seguir Livro**



## 4.2 Análise dos Resultados

Para analisar os resultados desse trabalho, foi realizado uma verificação de cada recurso colaborativo da aplicação, onde o objetivo principal era avaliar o tempo de cada requisição, e com isso obter um nível de qualidade da colaboração na aplicação.

Uma aplicação colaborativa deve possuir recursos que possibilitam o trabalho entre vários usuários sem aumentar o tempo gasto em cada atividade em relação ao que levaria caso tivesse sido realizado individualmente. Sendo assim, é necessário avaliar o tempo utilizado em cada recurso colaborativo, e com isso obter um ponto de referência na utilização de tais recursos (GOLDMAN et al, 2011).

Na análise da aplicação, a qualidade da colaboração foi avaliada a partir da medição do tempo de resposta para cada requisição realizada por recursos colaborativos, tais como a escrita em tempo real e a troca de mensagens através do chat. Com isso foi possível afirmar a possibilidade de haver sinergia em um livro escrito em tempo real e colaborado através de troca de mensagens.

A medição desses dados foram obtidos utilizando as ferramentas de desenvolvimento disponíveis no navegador Google Chrome, assim como também presentes no Firefox, para isso, foram analisadas as requisições dos recursos colaborativos acompanhando os canais “*Websockets*” para obter o tempo gasto em cada um. Sendo assim, esses dados foram adequados a um conjunto de critérios para avaliação do tempo em cada situação.

O processo para obter tais dados ocorre conforme os passos apresentados a seguir:

- a) Abrir tela desejada;
- b) Clicar com o botão direito no componente desejado;
- c) Selecionar a opção "Inspecionar elemento";
- d) Clicar na aba "Network";
- e) Selecionar o filtro "WS";
- f) Escolher um dos canais listados;
- g) Selecionar a aba "Frames";
- h) Observar que a cada alteração no conteúdo, o canal envia uma requisição de persistência (marcado em verde) e em seguida uma requisição dos mesmos dados que foram persistidos;

- i) Utilizar os dados da coluna “Time” para aplicar os critérios de avaliação;

Os critérios a seguir foram utilizados para classificar o tempo gasto em cada requisição referente a um tipo de análise conforme especificados a seguir:

1. **Complexidade:** São analisados os recursos externos de entrada e saída do sistema para obter a carga dos componentes individualmente.
2. **Confiabilidade:** É analisada através do número de problemas resolvidos pelo próprio sistema e o número total de problemas específicos.
3. **Eficiência:** São analisadas as características que definem a capacidade da aplicação em relação a performance.
4. **Funcionalidade:** É analisado as funcionalidades para definir a capacidade de recuperação, inclusão e alteração das informações.

Na escrita de livros e troca de mensagens, não deve haver um tempo de “*delay*” muito alto por se tratar de requisições em tempo real, portanto, é absolutamente crítico se este não atualizar as informações que foram alteradas em menos de 1000 milésimos de segundo. O motivo de ter definido 1000 milésimos de segundo como atraso é porque, quando um dado é transmitido com um “*delay*” maior que esse tempo, ele acaba sendo cortado e transmitido em pacotes por segundo, classificando como atraso de resposta, e para ser considerado tempo real.

Os critérios da “Escrita de Livro” foram avaliados da seguinte maneira:

- Complexidade: Foram somados o tempo de escrita e o tempo para o conteúdo ser visualização pelo segundo usuário;
- Confiabilidade: Para simular uma recuperação de erro, 16384 caracteres foram apagados e colados novamente múltiplas vezes até tentar exceder as requisições suportadas;
- Eficiência: Foi testado o tempo que leva para carregar todo o conteúdo de um capítulo com 16384 caracteres;

- Funcionalidade: Foi testado tempo que leva para digitar, apagar e ser visualizado uma letra pelo segundo usuário;

Os critérios da “Troca de mensagens no Chat” foram avaliados da seguinte maneira:

- Complexidade: Foram somados o tempo de envio e o de recebimento pelo segundo usuário;
- Confiabilidade: Para simular uma recuperação de erro, uma mensagem de 16384 caracteres foi enviada, pela análise do WebSocket, a mensagem foi quebrada em vários pacotes para que não houvesse um problema maior;
- Eficiência: Foi testado o tempo que leva para carregar 10 mensagens recebidas;
- Funcionalidade: Foi testado tempo que leva para uma mensagem ser recebida pelo segundo usuário;

Para avaliação da aplicação, utilizando os critérios e metodologia citados, a aplicação foi testada em uma simulação no próprio navegador em modo de desenvolvimento, e os resultados podem ser observados na Tabela 1.

**Tabela 1. Resultado da avaliação da aplicação**

| <b>Recurso</b>             | Complexidade | Confiabilidade | Eficiência | Funcionalidade |
|----------------------------|--------------|----------------|------------|----------------|
| Escrita de Livro           | 1012ms       | 742ms          | 659ms      | 188ms          |
| Troca de Mensagens no Chat | 1149ms       | 3076ms         | 1015ms     | 235ms          |

### 1) Quais foram os pontos negativos da solução e problema no desenvolvimento.

Houveram pontos negativos referente a como a interação teve que ser implementada, já que a princípio a ideia era uma sessão de comentários, assim como encontrada em outras aplicações sociais, mas devido à complexidade que há em tal implementação utilizando o serviço Firebase, foi decidido que seria melhor implementarmos um chat com mensagens armazenadas por capítulo, o qual possui mensagens que são armazenadas por capítulo.

Outro ponto negativo, foi a interface da aplicação ser web, ou seja, necessita que o usuário esteja sempre conectado à internet para que consiga visualizá-la. Porém, foi identificado que a partir da primeira vez em que a aplicação é aberta, tudo permanece disponível off-line, e o conteúdo escrito é atualizado quando o dispositivo se conecta novamente à rede.

Por fim, o único problema encontrado no desenvolvimento, foi a implementação utilizando os métodos de sincronização em tempo real que o Firebase oferece, pois alguns métodos tiveram que ser adaptados ao AngularJS que foi adaptado pelo Ionic framework.

## **2) Quais foram as lições aprendidas.**

Uma das principais lições foi o reconhecimento da capacidade de superação, de ir além dos limites que jamais teria conhecido se não tivesse passado por tal desafio, já que em muitos momentos foi necessário exceder essa capacidade, passando noites em claro para incorporar pensamentos e pesquisas em texto acadêmico.

Além do excelente ganho que é conhecer seus limites, com as pesquisas e o desenvolvimento da aplicação foi possível obter um nível de aprendizagem que vai muito além das salas de aula, aliado a necessidade de ter um conhecimento técnico prévio, que já havia sido transmitido pelos professores durante a graduação.

Profissionalmente, estar em meio a pesquisas de novas tecnologias foi uma grande oportunidade para encontrar um caminho no mercado de tecnologia a ser seguido, sendo este o desenvolvimento para dispositivos móveis, e com isso foram encontradas formas de se destacar ao escolher ferramentas que estão em alta neste meio.

## **4.3 Considerações Finais**

Como resultado, foi obtido uma aplicação multiplataforma, sendo ela disponibilizada para as plataformas Android e iOS, e que contemplou os objetivos do projeto. E, a partir das imagens, foram apresentados todos os recursos disponíveis na aplicação, demonstrando o e indicando como funcionam.

Para avaliar se o nível de colaboração poderá ser eficiente em relação ao tempo

necessário para executar funcionalidades colaborativas como a escrita de livros em tempo real e a troca de mensagens via chat, foi realizado uma análise das requisições utilizando critérios que buscam apontar se o nível de qualidade é satisfatório.

Dentre os pontos negativos da solução, estão a forma como foi implementada a interação na aplicação e a aplicação possui a camada de visualização web, ambas foram contornadas a um ponto que não houvesse tanta influência no resultado final da aplicação. Além disso, houve um problema que impactou na forma que foi desenvolvido a funcionalidade de escrita, onde a sincronização e visualização precisou ser adaptado ao AngularJS oferecido pelo Ionic Framework.

Por fim, este capítulo apresentou todos os resultados que foram obtidos com o desenvolvimento da aplicação, os quais passaram por uma análise onde foi obtido o tempo das requisições dentro de critérios que buscam avaliar a qualidade da aplicação. Então, foram apontados os pontos negativos da solução obtida e o problema ocorrido durante o desenvolvimento.

## 5 CONCLUSÃO

Para esse projeto, o desenvolvimento foi composto por uma aplicação móvel multiplataforma, atendendo as plataformas Android e iOS, que faz integração com a base de dados através de uma API Rest oferecido pelo serviço Firebase.

Com este estudo, foi possível ter conhecimento sobre as novas tecnologias que existem para desenvolvimento móvel, e como o mercado tecnológico foi influenciado seguindo uma tendência através do aumento de dispositivos móveis.

Em relação à implementação da aplicação multiplataforma utilizando o framework Ionic, foi identificado no aprendizado da linguagem Javascript que essa possui diversas variações para usos específicos tais como o AngularJS, que foi utilizado tanto na visualização quando nas regras de negócio, existe sempre um universo de informações que são concentradas em fóruns internacionais e vídeo aulas em inglês já que o material é escasso em português.

Também foi possível notar que existem aplicações móveis no mercado que também são voltadas para a escrita de conteúdo online, assim como livros e artigos, tornando assim a existência de um novo nicho dentro do mercado de aplicações móveis.

Já em relação ao serviço de persistência em tempo real, o Firebase, foi possível notar que outras empresas e desenvolvedores também estão aderindo a este tipo de serviço, já que a pouco tempo a Google fechou parceria com este serviço, para fim de promover e demonstrar novas utilizações do mesmo.

Por fim, houveram momentos em que foi necessário discutir maneiras de resolver possíveis problemas de indisponibilidade de internet, já que a aplicação possui a camada de visualização web, os problemas precisaram ser contornados através de persistência local, até que a aplicação volte a se conectar e então sincronize as informações.

Com tudo o que foi descrito, conclui-se que é possível criar uma aplicação multiplataforma de escrita colaborativa de livros para que novos escritores tenham uma alternativa que irá alavancar sua carreira como escritor de forma que ele possa adquirir experiências interagindo com outros usuários ao disponibilizar seu conteúdo publicamente e conversar através de um chat.

## 5.1 Trabalhos Futuros

A aplicação desenvolvida possibilita que uma pessoa seja um escritor mesmo sem uma editora e consiga um feedback imediato, além de possibilitar a interação e colaboração entre os escritores e leitores, mas este trabalho poderá evoluir recebendo novas funcionalidades deixando o sistema mais completo.

Das novas funcionalidades, foi possível identificar:

- **Permissões de acesso:** A aplicação foi desenvolvida para que um usuário possa escrever e ler livros de outros usuários. O próximo passo é implementar permissões que irão restringir que um usuário não altere ou visualize controles de alteração que o proprietário possui.
- **Integração com redes sociais:** Seguindo uma tendência do mercado, a integração da aplicação com redes sociais é um passo adiante na popularização do conteúdo desenvolvido pelos usuários, tornando possível compartilhar livros, cadastrar através de um perfil da rede social na aplicação, entre outros.
- **Melhorias na interface:** Adicionar uma introdução, que terá função de instruir os usuários na primeira utilização, além de realizar estudos de personas, para identificar novas maneiras de tornar a aplicação simples e fácil de ser utilizada.
- **Mensagens de Push:** Adicionar recursos que irão notificar os usuários que seguiram um livro, quando o mesmo estiver sendo escrito, para que assim possa se juntar ao grupo de discussões e colaborar.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANGULARJS. Documentação do AngularJS API. Disponível em: <<https://docs.angularjs.org/api>>. Acessado em 06 de Junho de 2015.

APPCCELERATOR. Documentação do Appcelerator Platform. Disponível em: <<https://docs.appcelerator.com/>>. Acessado em 06 de Junho de 2015.

BRAMBILLA, M et al; BPMN and Design Patterns for Engineering Social BPM Solutions. In: Daniel, F; Barkoui, K; Dustdar, S. Business Process Management Workshops: BPM 2011 International Workshops, 2011. p 270 - 281.

CORDOVA. Documentação do Cordova. Disponível em: <<https://cordova.apache.org/docs/en/latest/guide/overview/>>. Acessado em 06 de Junho de 2015.

ETHERPAD. Documentação da Etherpad API. Disponível em: <<http://etherpad.org/doc/v1.5.7/>>. Acessado em 06 de Junho de 2015.

FIREBASE. Documentação do Serviço Firebase para Persistência de dados e Sincronização em Tempo Real. Disponível em: <<https://www.firebase.com/docs/>>. Acessado em 06 de Junho de 2015.

FUKS, H et al; Teorias e Modelos de Colaboração. In: PIMENTEL, M; FUKS, H. (Org.). Sistemas Colaborativos. Rio de Janeiro: Campus, 2012. p 24- 26.

GOOGLE DOCS. Disponível em:<<https://www.google.com/docs/about/>>. Acessado em 25 de maio de 2015.

GOOGLE DRIVE. Documentação da Google Drive Realtime API. Disponível em: <<https://developers.google.com/google-apps/realtime/>>. Acessado em 06 de Junho de 2015.

HARB, E. et al. Responsive Web Design. N. 2011, <http://courses.iicm.tugraz.at/iaweb/surveys/ws2011/g3-survey-resp-web-design.pdf>. Acessado em: 12 de setembro de 2015.

HARTMANN, S. “The Naked Writer. Disponível em: <http://silviahartmann.com/live>”. Tradução de Diogo Henrique L. G. Da Silva. Acessado em: 28 de fev de 2015.

HILTON, W; LOFARO, et. al. A lightweight, cross-platform, multiuser robot visualization using the cloud. IEEE, Chicago, p. 1-6, 2014. Tradução de Diogo Henrique L. G. Da Silva.

IONIC. Documentação do Ionic Framework. Disponível em: <http://ionicframework.com/docs/>. Acessado em 06 de Junho de 2015.

IVAN, I; CIUREA, C. Quality Characteristics of Collaborative Systems. IEEE, Bucharest, p. 1-5, 2009.

JARAMILO, D.; NYGUYEN, V.; NEWHOOK, R. (2014) “Real-time experience techniques for collaborative tools on mobile”. SOUTHEASTCON 2014, p.1-6, Lexington, Kentucky. Tradução de Diogo Henrique L. G. Da Silva.

KIEFER, C. Para Ser Escritor. São Paulo: Leya, 2010. 115 p.

LIMA, R et al; Arquiteturas Distribuídas para Sistemas Colaborativos. In: PIMENTEL, M; FUKS, H. (Org.). Sistemas Colaborativos. Rio de Janeiro: Campus, 2012. p 339.

MILICEV, D; Chapter 15: Collaborations and Interactions. In: Milicev, D. Model-Driven Development with Executable UML: Wrox, 2009. p 519.

MOHOROVICIC, S. Implementing Responsive Web Design for Enhanced Web Presence. MIPRO 2013, Opatija, Croatia, p. 1-5, 2013.

NICOLACI-DA-COSTA, A et al; Sistemas Colaborativos Para uma Nova Sociedade e Um Novo Ser Humano. In: PIMENTEL, M; FUKS, H. (Org.). Sistemas Colaborativos. Rio de Janeiro: Campus, 2012. p 3 - 15.

PENFLIP. Nossa História. Disponível em:<<https://www.penflip.com>>. Acessado em 25 de maio de 2015.

PHONEGAP. Documentação do Adobe Phonegap. Disponível em: <<http://docs.phonegap.com/>>. Acessado em 12 de setembro de 2015.

SANTAROSA, A; CRISTINA, A; Ontologia de Colaboração. In: PIMENTEL, M; FUKS, H. (Org.). Sistemas Colaborativos. Rio de Janeiro: Campus, 2012. p 36 - 45.

WATTPAD. Quem Somos. Disponível em:<<https://www.wattpad.com>>. Acessado em 25 de maio de 2015.

WAZE. Sobre Nós. Disponível em: <<https://www.waze.com>>. Acessado em 25 de maio de 2015.