

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

MY MUSIC - VERSÃO 1.1: UMA EXTENSÃO
COLABORATIVA DO APLICATIVO DE COMPOSIÇÃO
MUSICAL PARA LEIGOS

RENATO DAROSCI

BLUMENAU
2015

2015/2-20

RENATO DAROSCI

**MY MUSIC - VERSÃO 1.1: UMA EXTENSÃO
COLABORATIVA DO APLICATIVO DE COMPOSIÇÃO
MUSICAL PARA LEIGOS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Aurélio Faustino Hoppe, Mestre - Orientador

**BLUMENAU
2015**

2015/2-20

**MY MUSIC - VERSÃO 1.1: UMA EXTENSÃO
COLABORATIVA DO APLICATIVO DE COMPOSIÇÃO
MUSICAL PARA LEIGOS**

Por

RENATO DAROSCI

Trabalho de Conclusão de Curso aprovado
para obtenção dos créditos na disciplina de
Trabalho de Conclusão de Curso II pela banca
examinadora formada por:

Presidente:

Prof. Aurélio Faustino Hoppe, Mestre – Orientador, FURB

Membro:

Prof. Marcel Hugo, Mestre – FURB

Membro:

Prof. Paulo Fernando da Silva, Mestre – FURB

Blumenau, 11 de dezembro de 2015

Dedico este trabalho a todos aqueles que de alguma forma estiveram e estão próximos de mim, fazendo esta vida valer cada vez mais a pena.

AGRADECIMENTOS

A Deus, por me conceder a saúde, força e fé necessárias para superar as dificuldades.

À minha família por todo carinho dedicado e por sempre ter ensinado os valores e a garra necessária para superar desafios. Agradecendo em especial meu pai Valter Darosci e minha mãe Ivone Roberti Darosci, por sempre me incentivarem a seguir firme na jornada sem me desviar do caminho, sendo sempre uma pessoa de bom coração.

À minha amada esposa Natália Stephanie Busnardo, por todo o apoio e incentivo diário, além de toda paciência necessária para me suportar nos momentos difíceis. Sua presença foi de grande importância nesta jornada.

Aos meus irmãos Patrícia e Guilherme Darosci, por sermos os irmãos que somos, e que independente da distância estamos sempre unidos e levando um ao outro no coração com carinho.

Aos meus amigos, por entenderem a minha ausência nos últimos tempos e agradecer em especial às palavras de incentivo.

Ao meu orientador, Professor Aurélio Faustino Hoppe, pela oportunidade de realização deste trabalho e por sua orientação, apoio e confiança.

É fácil alcançar a imortalidade, basta fazer
uma coisa notável.

Dimebag Darrell

RESUMO

Este trabalho dedica-se à manutenção evolutiva realizada no aplicativo de composição musical para leigos, desenvolvido por Weitgenant (2014), para dispositivos da plataforma Android. Esta manutenção propõe melhorias quanto aos mecanismos de colaboração, através da criação de ambientes distintos de uso público e privado, listas de composições para publicação de composições, lista de versões permitindo a visualização e resgate das mesmas, lista de discussão e lista de colaboradores. Os testes realizados demonstraram que a aplicação atendeu aos requisitos propostos e mesmo apresentando algumas ressalvas por parte dos usuários que realizaram os testes, a aplicação se mostrou interessante e propõe uma experiência agradável.

Palavras-chave: Composição musical. CSCW. Cooperação. Android. IHC. *MyMusic*.

ABSTRACT

This work is dedicated to evolutionary maintenance performed on music composition application for lay people, developed by Weitgenant (2014), for the Android platform devices. This maintenance proposes improvements regarding collaboration mechanisms, by creating different environments of public and private use, compositions of lists for publication compositions versions list allowing you to view and redeem the same, mailing list and list of collaborators. The tests carried out showed that the application met the proposed requirements and even with some reservations on the part of users who performed the tests, the application proved interesting and offers a pleasant experience.

Key-words: Musical composition. CSCW. Cooperation. Android. IHC. MyMusic.

LISTA DE FIGURAS

Figura 1 - Interface para criação de composições musicais	20
Figura 2 - Painel principal do musicLine	22
Figura 3 - Tela principal Sound Club	23
Figura 4 - Tela principal do Best Composer	24
Figura 5 - Tela principal do Sibelius	25
Figura 6 – Diagrama de Casos de Uso	29
Figura 7 - Diagrama de classes do aplicativo	31
Figura 8 – Classe da tela de Versões	32
Figura 9 - Classe da tela de Editor	32
Figura 10 - Classes da tela de Convites	33
Figura 11 - Classes da tela do Fórum	34
Figura 12 – Diagrama de atividades macro do aplicativo	35
Figura 13 – Lista pública	47
Figura 14 – Reproduzindo áudio da composição	48
Figura 15 – Visualização da estrutura da composição	48
Figura 16 – Tela de <i>Login</i>	49
Figura 17 – Tela do cadastro de usuário	49
Figura 18 – Lista privada	50
Figura 19 – Tela de versões da composição	51
Figura 20 – Tela da área de edição	51
Figura 21 – Indicação de composição publicada	52
Figura 22 – Lista de usuários para convite	52
Figura 23 – Adicionar e-mail de convite	53
Figura 24 – Convidado inserido na lista	53
Figura 25 – Fórum de discussões	54
Figura 26 – Tela para inclusão de comentários	54

LISTA DE QUADROS

Quadro 1 – Características dos trabalhos correlatos.....	26
Quadro 2 - Leitura dos arquivos do servidor.....	37
Quadro 3 – Realizando o início dos <i>downloads</i>	38
Quadro 4 – <i>Download</i> das composições.....	38
Quadro 5 – Convertendo o texto do arquivo.....	39
Quadro 6 – Verifica acesso à composição.....	39
Quadro 7 – Recepção de parâmetros pelo editor.....	40
Quadro 8 – Método de <i>upload</i> dos arquivos.....	42
Quadro 9 – Selecionar os usuários para convite.....	43
Quadro 10 – Método para envio de e-mails.....	44
Quadro 11 – Carregar a lista de postagens do fórum.....	45
Quadro 12 – Adicionar postagens ao Fórum.....	45
Quadro 13 – Publicar a composição.....	46
Quadro 14 – Resultados do questionário de perfil dos usuários.....	56
Quadro 15 – Resultados do questionário de usabilidade.....	57
Quadro 16 - Comparação com os trabalhos correlatos.....	58
Quadro 17 – Comparando a evolução do trabalho.....	59
Quadro 18 - UC01 - Listar composições.....	65
Quadro 19 - UC02 - Baixar composições.....	66
Quadro 20 - UC03 - Fazer Login/Logout.....	66
Quadro 21 - UC04 - Cadastrar usuário.....	67
Quadro 22 - UC05 - Registrar Curtir.....	67
Quadro 23 - UC06 - Editar composição.....	68
Quadro 24 - UC07 - Salvar composições.....	68
Quadro 25 - UC08 - Gerar versão.....	69
Quadro 26 - UC09 - Enviar composição.....	69
Quadro 27 - UC10 - Convidar usuários.....	69
Quadro 28 - UC11 - Manter Fórum.....	70
Quadro 29 - UC12 - Publicar composição.....	70
Quadro 30 – Questionário de perfil do usuário.....	71

Quadro 31 – Instruções iniciais	72
Quadro 32 – Roteiro para avaliação	72

LISTA DE ABREVIATURAS E SIGLAS

ADT - *Android Developer Tools*

API – *Application Programming Interface*

AVD - *Android Virtual Device*

BPM – Batidas por minuto

CSCW - *Computer Supported Cooperative Work*

GED - Gerenciamento Eletrônico de Documentos

GUI - *Graphical User Interface*

IHC – Interface Humano-Computador

RF – Requisito Funcional

RNF – Requisito Não Funcional

SDK - *Software Development Kit*

UML - *Unified Modeling Language*

WAV – *Waveform Audio Format*

WYSIWIS - *What You See Is What I See*

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVOS.....	15
1.2 ESTRUTURA.....	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 COMPOSIÇÃO MUSICAL.....	16
2.2 CSCW.....	17
2.2.1 Groupware.....	17
2.2.2 Requisitos gerais do CSCW	18
2.3 IHC.....	18
2.4 APLICATIVO PARA CRIAÇÃO DE COMPOSIÇÕES MUSICAIS.....	20
2.5 TRABALHOS CORRELATOS	21
2.5.1 musicLine.....	21
2.5.2 Sound Club.....	22
2.5.3 Best Composer.....	23
2.5.4 Sibelius.....	24
2.5.5 Comparação entre os trabalhos correlatos.....	25
3 DESENVOLVIMENTO DO APLICATIVO	28
3.1 REQUISITOS	28
3.2 ESPECIFICAÇÃO.....	29
3.2.1 Diagrama de casos de uso	29
3.2.2 Diagrama de Classes.....	31
3.2.3 Diagrama de Atividades.....	34
3.3 IMPLEMENTAÇÃO	36
3.3.1 Técnicas e ferramentas utilizadas	36
3.3.2 Etapas da implementação.....	37
3.3.3 Operacionalidade da implementação	46
3.4 RESULTADOS E DISCUSSÕES	55
3.4.1 Experimento de usabilidade	55
3.4.2 Comparação com trabalhos correlatos.....	58
3.4.3 Alterações realizadas na aplicação	59
3.4.4 Aplicando requisitos do CSCW.....	60

4 CONCLUSÕES	61
4.1 EXTENSÕES	62
APÊNDICE A – DETALHAMENTO DOS CASOS DE USO	65
APÊNDICE B – QUESTIONÁRIO DE USABILIDADE	71

1 INTRODUÇÃO

A música existe e sempre existiu como uma produção cultural, pois estudos científicos mostram que a música era parte integrante do cotidiano já das tribos primitivas, e continua presente no cotidiano da sociedade. Acredita-se que a música tenha surgido há 50.000 anos, sendo as primeiras manifestações no continente africano e posteriormente expandindo-se com o dispersar da raça humana pelo planeta (MARÇAL, 2011).

Atualmente, a música encontra-se acessível a todos, numa escala incrível em comparação a uma pessoa do século passado por exemplo. Com a ascensão tecnológica e dos meios de comunicação, as pessoas têm acesso a todo tipo de gênero e conhecimento musical.

Caiado (2015) coloca que muitos pesquisadores descrevem a música como importante ferramenta no desenvolvimento da mente humana, raciocínio e concentração, principalmente se a educação musical vier desde cedo, quando criança. Conforme Severo (2013), “[...] Platão, em 300 a.C. já dizia que “a música é um instrumento educacional mais potente do que qualquer outro” [...]”. Faria (2001) define que a música é um importante fator na aprendizagem, pois a criança desde pequena já ouve música, a qual muitas vezes é cantada pela mãe ao dormir, conhecida como ‘cantiga de ninar’.

A revolução tecnológica que vem ocorrendo com o passar das décadas, vem trazendo cada vez mais novidades à sociedade, e no que diz respeito ao ramo musical, isto não tem sido diferente. Atualmente há uma infinidade de instrumentos musicais eletrônicos, que dentre as diversas funcionalidades disponíveis, oferecem em muitos casos, recursos de gravação, edição de áudio e até simulações de outros instrumentos, possibilitando que a criação musical ocorra cada vez de forma mais fácil e mais dinâmica.

Toda esta onda tecnológica possibilitou que as pessoas se aproximassem mais da música. Hoje, facilmente encontramos aplicações voltadas à criação e composição musical, tanto em computadores, em dispositivos móveis e/ou na própria rede.

É com esse pensamento que se busca neste trabalho elencar conceitos que agregados a recursos computacionais, servirão de base para a complementação do aplicativo de composição musical desenvolvido por Weitgenant (2014), buscando viabilizar sua utilização em um contexto colaborativo de desenvolvimento. Construir um ambiente onde além da criação musical o usuário possa disponibilizar suas composições para outros ouvirem ou até mesmo participarem do processo criativo, a fim de que a intimidade com a música seja aprimorada gradativamente e que através do trabalho colaborativo, os usuários possam trocar experiências, mesclar conhecimentos e desenvolver a percepção musical.

1.1 OBJETIVOS

O objetivo deste trabalho é dar continuação ao trabalho de Weitgenant (2014), acrescentando a funcionalidade de um ambiente colaborativo, onde leigos consigam criar e compartilhar composições.

Os objetivos específicos do trabalho são:

- a) disponibilizar uma aplicação com ambientes distintos, tratados respectivamente como ambiente público e ambiente privado, destinados à área de publicação dos protótipos desenvolvidos e à área de edição e desenvolvimento privado aos usuários do aplicativo;
- b) disponibilizar o controle de versionamento, responsável por gerir o ambiente de desenvolvimento colaborativo, tratando as alterações efetuadas em cada protótipo de composição, além de permitir que versões anteriores sejam recuperadas para serem manipuladas ou até mesmo mescladas em outras versões;
- c) disponibilizar uma lista de usuários, onde poderão ser adicionados compositores a uma determinada composição, enviando convites aos mesmos para que possam colaborar no desenvolvimento desta;
- d) disponibilizar uma lista de discussão, onde serão listados os detalhes sobre as alterações realizadas nas respectivas versões das composições, tendo como intuito não só o registro das alterações, mas também uma área de troca de opiniões e debates sobre cada composição.

1.2 ESTRUTURA

O trabalho apresentado está subdividido em quatro capítulos, sendo este primeiro capítulo dedicado a apresentar uma breve introdução ao tema, os objetivos do trabalho e a estrutura como é apresentado.

O segundo capítulo contém a fundamentação teórica necessária para um melhor entendimento do assunto e da solução implementada no aplicativo.

O terceiro capítulo apresenta o processo de especificação do aplicativo, onde são apresentados os diagramas de caso de uso, de atividades e de classes. Também são expostos detalhes da implementação e a operacionalidade do aplicativo do ponto de vista do usuário.

O último capítulo, por sua vez, aborda as conclusões a respeito dos resultados obtidos, além de sugestões para serem continuadas em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados diversos conceitos relacionados ao desenvolvimento da aplicação. Dividido em seções distintas, contextualizam-se composição musical, seguindo com a apresentação de conceitos de *Computer Supported Cooperative Work* (CSCW) e Interação Humano-Computador (IHC) aplicado ao desenvolvimento de aplicações para dispositivos móveis.

Após a contextualização dos conceitos, será apresentado o aplicativo de composição musical para leigos iniciado por Alvarenga (2013) e aprimorado por Weitgenant (2014), seguido pela apresentação de quatro trabalhos correlatos e algumas de suas características.

2.1 COMPOSIÇÃO MUSICAL

Segundo Gaigher (2012), nos dias de hoje, a música é um objeto largamente consumido por todos, porém, este fato não se deve ao crescimento do interesse das pessoas em produzir e ouvir músicas, mas sim ao fato de as pessoas terem mais acessos aos meios de produção, distribuição e reprodução musical. George Martin, músico formado com especialização em composição, arranjo e orquestração, diz que:

De todas as artes, a música é a mais sublime. Toca o coração de cada ser humano. Mesmo aqueles raros indivíduos que bancam os surdos, em algum momento de suas vidas são movidos pela música, de um jeito ou de outro. (MARTIN, 2002, p.XI).

Segundo Med (1996), define-se música pela arte de combinar os sons de forma simultânea e sucessiva, com ordem, equilíbrio e proporção dentro de um determinado tempo. Azevedo (1982) subdivide a música em três elementos fundamentais:

- a) harmonia: conjunto de sons dispostos de forma simultânea, ou seja, tocados ao mesmo tempo;
- b) melodia: é uma sucessão coerente entre sons e silêncios;
- c) ritmo: é a ordem e a proporção em que os sons da harmonia e da melodia são dispostos.

A composição musical acontece através do processo de organização de ideias musicais para a elaboração de uma peça, desde uma improvisação feita por uma criança ao xilofone, com total liberdade e espontaneidade ou ainda uma obra concebida dentro de regras e princípios (SWANWICK; FRANÇA, 2002, p. 9).

2.2 CSCW

CSCW é a abreviatura adotada para o termo *Computer Supported Cooperative Work*, em português “trabalho colaborativo auxiliado por computador” e, segundo Castilho (2008), trata-se de uma ciência multidisciplinar que estuda as formas de trabalho em grupo auxiliadas por tecnologia e comunicação. O surgimento do termo CSCW, apesar de ter sido oficializado apenas em 1986, ocorreu no ano de 1984, num *workshop* realizado nos Estados Unidos por Iren Greif e Paul Cashman, onde neste encontrava-se um grupo de pessoas com o intuito comum de utilizar a tecnologia para auxiliar no ambiente de trabalho.

Como o nome sugere, o CSCW visa o estudo e desenvolvimento de aplicações que suportem grupos de pessoas envolvidos em uma mesma tarefa e/ou objetivos em comum. Fluckiger (1995) define CSCW como o campo relativo ao projeto de sistemas baseados em computador para o suporte e melhoria do trabalho em grupo de usuários envolvidos em tarefas ou objetivos comuns, e ao entendimento dos efeitos da utilização de tais sistemas. As aplicações desenvolvidas neste contexto são denominadas habitualmente de *groupware*, sendo as aplicações de correio eletrônico uma das mais antigas.

Existem algumas confusões em torno destes termos, onde *groupware* é usado para especificar única e exclusivamente, os sistemas computacionais utilizados por pessoas para realizarem trabalhos em conjunto. Já o CSCW refere-se a um campo genérico focalizando a área profissional.

2.2.1 Groupware

Apesar de muitas vezes o termo *groupware* ser utilizado como sinônimo para CSCW, estes são dois termos distintos, porém ambos buscam auxiliar o trabalho de grupos de usuários. Segundo Souza (2006), o CSCW é um conceito mais amplo onde o *groupware* está inserido.

Castilho (2008) cita que Robbins e Finley definem *groupware* como um software distribuído e interativo utilizado em redes de computadores voltado para o trabalho em grupo. Como exemplo, podem-se citar aplicações como correios eletrônicos, videoconferência, *chats*, editores colaborativos, gerenciadores de fluxo de trabalho (*workflow*), sistemas de apoio à tomada de decisões, sistemas de comunicação síncrona e assíncrona, sistemas para gerenciamento eletrônico de documentos (GED), entre outros.

2.2.2 Requisitos gerais do CSCW

Botelho e Vidal (2005) propõe a ideia de que os requisitos das ações e atividades em um CSCW referem-se às funcionalidades do sistema, podendo estar divididas em gerais ou específicas e consideradas básicas não apenas num *groupware*, mas em quaisquer interações humano-computador. A seguir, são listados os requisitos gerais de acordo com Schneiderman (1992), e citados também por Botelho e Vidal (2005):

- a) identidade individual dos membros: saber quem está autenticado;
- b) posição do usuário: nomes e/ou ícones visíveis na tela;
- c) conhecimento: outros integrantes saibam da presença de um usuário no grupo;
- d) visão pessoal ou padrão: visão especificada por cada usuário para um problema;
- e) visão convergente: sendo mais recomendado o princípio *What You See Is What I See* (WYSIWIS), cuja tradução remete a algo equivalente a " O que você vê é o que vejo";
- f) visões particular e pública: partes podem ser vistas por grupos de usuários específicos;
- g) controle da palavra: algumas situações necessitam de somente um usuário manipulando um aplicativo ou emitindo opiniões;
- h) controle de acesso: direitos de acesso para cada usuário;
- i) atualização e sincronismo: para não haver divergências na edição e/ou entrada e saída de dados;
- j) coordenação da informação: coordenação de atividade ou grupo de atividades, aumentando a eficiência.

Quando se trata do CSCW aplicado às tecnologias de dispositivos móveis, há desafios extras a serem considerados, devido às limitações encontradas nos dispositivos no que diz respeito à capacidade de memória, tamanho da tela e de energia disponível. A curta vida útil das baterias, por exemplo, pode vir a causar interrupções inesperadas que levam a consequências ruins ao trabalho colaborativo. Além disso, as redes sem fio onde estes dispositivos se comunicam, tem baixa largura de banda e apresentam muitas vezes atrasos significativos devido a desligamentos temporários (PAPADOPOULOS, 2005).

2.3 IHC

A área destinada a estudar não só o *design* de interface, como também o interesse e demandas dos usuários do sistema foram descritas como Interface Humano-Computador

(IHC), na década de 80, iniciada pelo psicólogo cognitivista Donald Norman, quando o mesmo passou a estudar a associação do homem interagindo com o computador (SANTOS et al., 2012). Rocha et al. (2003) definem IHC como a área preocupada com o *design*, avaliação e implementação de sistemas computacionais interativos para o uso humano e com o estudo dos principais fenômenos subjacentes a eles.

Baseando-se nestes conceitos, o IHC tem por objetivo estudar e definir métodos para projetos de sistemas ou dispositivos, de forma que sejam de fácil utilização, eficazes, eficientes e que ofereçam conforto ao usuário (AGNER, 2006). Segundo Silva (2014), é possível citar três ondas durante a história da área de IHC:

- a) primeira onda: é voltada aos fatores humanos, tendo foco no indivíduo e tomando o usuário como um conjunto de mecanismos de processamento da informação;
- b) segunda onda: é voltada para atores humanos, tendo o foco em grupos. Nessa onda é apresentada uma abordagem qualitativa e não mais quantitativa, prototipação e *design* contextual, levando em consideração a integração da pessoa em determinado ambiente;
- c) terceira onda: tem foco em aspectos culturais e estéticos, se expandindo do cognitivo ao emocional, levando a tecnologia a extrapolar os limites de trabalho e tornar-se parte da cultura, vida e casa das pessoas.

A expansão da telefonia móvel gera oportunidades tanto para fornecedores de serviços e aplicações, bem como para pesquisadores e profissionais que atuam no desenvolvimento das interfaces com o usuário dos equipamentos e de seus aplicativos (CYBIS; BETIOL; FAUST, 2010). De acordo com Yang, Zheng e Ni (2007), estes são os fatores de maior importância a serem considerados durante o desenvolvimento de aplicações:

- a) eficiente armazenamento e rede adaptativa: buscando o melhor uso do armazenamento local, a fim de obter melhor desempenho;
- b) Interface Gráfica do Usuário (GUI, do inglês *Graphical User Interface*) simples e *user-friendly*: visando facilitar a entrada de dados, visto que a digitação em um dispositivo móvel, por vezes, é mais difícil que um teclado de computador;
- c) computação leve e gerenciamento de energia: buscando direcionar a maior parte do processamento intensivo a um servidor, que normalmente é mais poderoso que um processador móvel, permitindo uma maior velocidade no lado do cliente e realizar um gerenciamento de energia eficaz aumentando o tempo de vida da bateria;

- d) alta segurança: visando a segurança de dados pessoais do usuário, que são armazenados e transmitidos através da rede.

Segundo Cybis, Betiol e Faust (2010), os aplicativos para dispositivos móveis demandam de forte atenção aos detalhes e à qualidade, da mesma forma dos aplicativos tradicionais, e mesmo apresentando diferenças, nenhuma delas é considerável ao ponto de que práticas de planejamento, arquitetura, desenvolvimento e teste de aplicativos não possam ser empregadas.

2.4 APLICATIVO PARA CRIAÇÃO DE COMPOSIÇÕES MUSICAIS

Conforme Weitgenant (2014, apud ALVARENGA, 2013), o aplicativo desenvolvido é apresentado como uma ferramenta que visa fornecer recursos necessários para criação de composições musicais por usuários leigos, através de uma representação musical estruturada em uma *grid*, onde cada peça representa um pequeno trecho da composição. A Figura 1 apresenta a interface do aplicativo, juntamente com um exemplo de música carregada pelo mesmo.

Figura 1 - Interface para criação de composições musicais



Fonte: Weitgenant (2014, p.49).

O aplicativo permite ações de abertura de composições criadas no aplicativo, edição, gravação e reprodução das peças da composição. Segundo Weitgenant (2014), ele foi desenvolvido com o uso da ferramenta *Android Developer Tools* (ADT), que se trata de um pacote que contém o ambiente de desenvolvimento Eclipse, bem como os *plug-ins* necessários para o desenvolvimento de aplicativos Android.

Weitgenant (2014) relata que “[...] foi possível tornar o aplicativo em um cenário colaborativo, com as implementações de controle de versão, envio/recebimento de arquivos do servidor, bem como comunicação entre usuários por meio de *chat*”. Foi apontado por

alguns usuários a necessidade de um menu mais interativo e a possibilidade de alterar a representação que é de forma textual para uma representação mais gráfica, com a utilização de imagens e ícones. Como limitações, destaca-se a ausência de informação quanto a usuários online em relação aos usuários que não estão acessando o *chat* e a falta de salas de bate-papo (WEITGENANT, 2014).

2.5 TRABALHOS CORRELATOS

Dentre as diversas ferramentas relacionadas ao processo de composição e/ou aprendizado musical, quatro são comentadas a seguir, selecionadas por possuírem ligações com as funcionalidades propostas neste estudo:

- a) musicLine: desenvolvido pela CreateApps e disponibilizada através da loja *online* da Google, a Google Play (GOOGLE PLAY, 2015);
- b) Sound Club: desenvolvido pela Bluemoon Interactive (BLUEMOON INTERACTIVE, 2014);
- c) Best Composer: desenvolvido pela Netigen (NETIGEN, 2014);
- d) Sibelius: desenvolvido pela Sibelius Software, que agora pertence à Avid Technology (AVID TECHNOLOGY, 2015).

2.5.1 musicLine

É um aplicativo desenvolvido pela empresa CreateApps. Está disponível para plataforma *mobile*, mais especificamente, para o sistema operacional Android. O aplicativo tem por objetivo oferecer recursos para a composição musical de forma rápida e descomplicada, oferecendo cerca de 100 tipos de sons de instrumentos (GOOGLE PLAY, 2015).

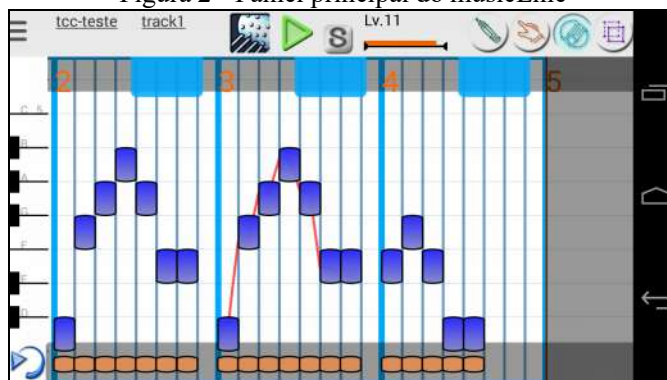
As composições criadas no aplicativo podem ser compartilhadas para amigos, diretamente do menu. São oferecidos diversos *timelines* onde cada um representa uma nota musical, onde o usuário seleciona o tempo de execução de cada nota.

O aplicativo oferece funcionalidades diversas como controle individual de volume por instrumento, mudança de tempo (batidas por minuto – BPM), repetição de partes de solos, entre outras. As funcionalidades oferecidas são bem abrangentes, mas a maior parte delas é liberada conforme o uso, sendo que quanto maior o *level*, maior o número de recursos liberados para utilização.

A Figura 2 apresenta a interface principal do musicLine. No topo da tela exibe-se o menu para acesso às opções de salvar, novo, abrir, fechar, exportar, configurações, entre

outras. Também são apresentadas as opções de acesso rápido, como nome do arquivo, parâmetros da trilha, menu de instrumentos, comandos de execução e de edição. Na lateral esquerda apresenta-se o painel de instrumentos, onde são listadas as notas musicais a serem selecionadas para a composição. O restante da tela é ocupado pela linha de execução onde são marcadas as notas a serem tocadas em cada compasso da música.

Figura 2 - Painel principal do musicLine



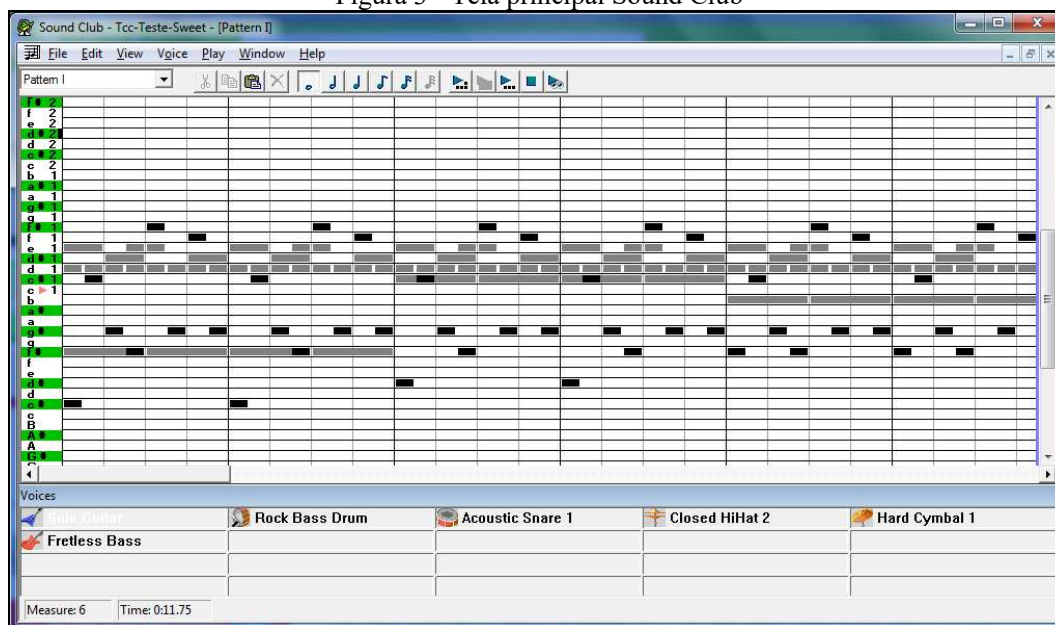
Fonte: Google Play (2014).

A interface é bem intuitiva, sendo possível entender o funcionamento de forma bem eficaz. O aplicativo também conta com um tutorial que auxilia nos primeiros-passos para utilização. O musicLine está disponível no Google Play de forma gratuita.

2.5.2 Sound Club

Desenvolvido para a plataforma Windows, o Sound Club, segundo Baixaki (2008), é um aplicativo para composição musical que oferece uma série de recursos interessantes para a criação de faixas de áudio, com diversos sons de instrumentos musicais distintos. A Figura 3 apresenta a tela principal do Sound Club. A interação com o programa pode ser realizada através de mouse ou teclado, sendo que cada tecla representa uma nota musical, que é representada no programa através de linhas horizontais.

Figura 3 - Tela principal Sound Club



Fonte: Bluemoon Interactive (2014).

Como pode ser observado na Figura 3, o programa apresenta um menu na parte superior da tela, seguindo o padrão do sistema operacional. Na parte inferior são apresentados os instrumentos musicais (aqui sendo chamados de *voices*) que farão parte da trilha que está sendo composta. Cada instrumento musical adicionado à trilha é representado por marcações na *timeline* de execução conforme a nota musical que o mesmo deve produzir. Estas marcações são realizadas conforme a escala cromática que é disponibilizada na lateral esquerda da tela. Quando determinado *voice* é selecionado, suas marcações são destacadas em relação às demais.

A aplicação também oferece recursos quanto à parametrização de tempo de duração das notas, opções de execução de todos os instrumentos em conjunto, ou apenas um instrumento e/ou um trecho da composição. Ao término da composição é possível exportar a mesma para um arquivo em formato WAV (*Waveform audio format*). Apesar de ser uma aplicação antiga, mostrou-se muito usual e intuitiva para uso e entendimento.

2.5.3 Best Composer

O Best Composer é um aplicativo para o sistema operacional Android, desenvolvido pela empresa Netigen e oferece recursos para composição de amostras musicais através dos sons disponibilizados pelo aplicativo. A Figura 4 apresenta a tela principal do Best Composer.

Figura 4 - Tela principal do Best Composer



Fonte: Netigen (2014).

A interface contém as funções básicas para a criação das amostras musicais. Conforme observado na Figura 4, disposto na parte superior do aplicativo, estão os botões para acesso às funções de abertura de arquivo, salvamento e exportação, o seletor de batidas por minuto, os demais comandos de controle de execução e um comando para limpeza dos parâmetros inseridos.

Na lateral esquerda são disponibilizadas seis opções de sons a serem escolhidos para fazer parte da trilha que está sendo composta. Na parte inferior da tela é apresentada uma barra que marca o ponto de execução da faixa. O restante da tela é ocupado pela grade de marcadores onde serão parametrizados os sons a serem tocados.

O Best Composer apresenta as opções de exportação do áudio criado para os formatos mp3, ogg e wav, mas não possui funcionalidade para compartilhamento direto do arquivo. Devido ao fato de não apresentar uma escala de notas, torna-se bastante limitado para composições, mesmo com vários sons disponíveis para seleção.

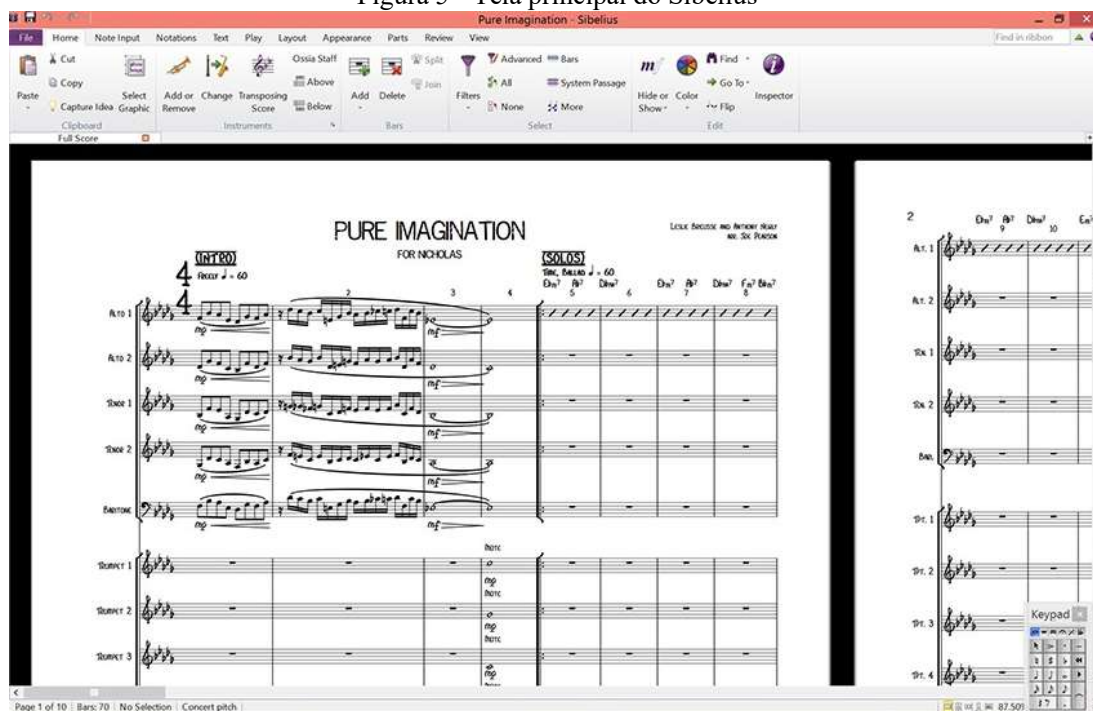
2.5.4 Sibelius

O Sibelius é um software desenvolvido pela empresa Sibelius Software (agora pertencente à Avid Technology) e se destina à edição de partituras, oferecendo ferramentas de fácil utilização a compositores, arranjadores, editores, educadores e estudantes (AVID TECHNOLOGY, 2015).

Através do Sibelius é possível escrever composições musicais de forma a poder consultá-las mais tarde ou até mesmo, ouvi-las em tempo real. É necessário, contudo estar familiarizado com a terminologia e os símbolos da escrita musical para poder utilizar o

aplicativo com mais eficiência (BAIXAKI, 2013). A Figura 5 apresenta a tela principal do Sibelius, onde é visualizada toda a estrutura da partitura para edição.

Figura 5 - Tela principal do Sibelius



Fonte: Avid Technology (2015).

Segundo Marques (2014), “as partituras podem ser enviadas por e-mail, publicadas como *music sheet* no ScoreExchange.com ou compartilhadas como arquivo de vídeo ou áudio no Youtube, Facebook ou SoundCloud”. Marques (2014) também cita que “o Sibelius automaticamente otimiza a partitura para a tela do iPad, com especial atenção à orientação da página, quantidade de pautas, tamanho das margens e muito mais”.

2.5.5 Comparação entre os trabalhos correlatos

No Quadro 1 é apresentado um comparativo entre características dos trabalhos apresentados:

Quadro 1 – Características dos trabalhos correlatos

características / trabalhos correlatos	Google Play (2015)	Bluemoon Interactive (2014)	Netigen (2014)	Avid Technology (2015)
usuário-alvo	Leigos	Leigos/Avançados	Leigos	Avançados
plataforma	<i>mobile</i>	<i>desktop</i>	<i>mobile</i>	<i>mobile/desktop</i>
sistema operacional	Android	Windows	Android	Windows/MAC OS/RISC OS
permite edição nota-à-nota	Sim	Sim	Não	Sim
oferece escala cromática	Não	Sim	Não	Sim
permite uso de múltiplos instrumentos	Sim	Sim	Sim	Sim
permite salvar as músicas criadas	Sim	Sim	Sim	Apenas na versão paga
permite exportar as composições	Sim	Sim	Sim	Sim
oferece funcionalidade de compartilhamento	Sim	Não	Não	Sim
criação colaborativa	Não	Não	Não	Não
forma de execução	loop	Opcional: Loop ou início e fim	loop	início e fim

Comparando-se os trabalhos, é possível observar que o programa Sound Club (BLUEMOON INTERACTIVE, 2014), apesar de mais antigo, oferece recursos interessantes em comparação aos outros. Também há pontos que deixam a desejar em questão de funcionalidades oferecidas pelos aplicativos.

O aplicativo musicLine (GOOGLE PLAY, 2015) não apresenta uma escala cromática, o que limita a composição apenas às sete notas musicais básicas. No caso do Best Composer (NETIGEN, 2014), a situação se apresenta de forma ainda mais problemática, visto que cada trilha da *timeline* marca apenas uma nota.

O aplicativo Sibelius (AVID TECHNOLOGY, 2015) oferece uma gama maior de funcionalidades, mas o programa foge ao conceito dos demais, pois trabalha com escrita musical através de partituras voltado à produção profissional. Esse tipo de escrita musical requer do usuário um conhecimento maior quanto à linguagem musical, destarte se tornando um aplicativo complexo para o uso por pessoas leigas na composição musical.

Entre os quatro programas apresentados o Sound Club é o que oferece mais recursos quando levado em consideração o uso por usuários leigos na linguagem musical. Em comparação com os demais o Sound Club possui toda a gama de notas musicais necessárias para composições básicas até composições um pouco mais elaboradas. A interface também se apresenta de forma bem organizada, o que é comum entre as aplicações. Devido às particularidades das plataformas, os programas musicLine e Best Composer necessitam apresentar uma interface reduzida e o mais funcional possível.

Todos os aplicativos atendem de alguma forma à função de composição musical, apresentando melhores recursos ou deixando a desejar em alguns pontos. As similaridades estão nas características mais básicas, que são quanto à questão de instrumentos e sons disponíveis à criação, a forma como a criação é feita através de marcações específicas em um *timeline*, às funções de arquivamento dos áudios criados e a exportação dos arquivos.

No que diz respeito aos programas apresentados para a plataforma *mobile*, que será a também a plataforma base para o desenvolvimento deste trabalho, os aplicativos musicLine, Sound Club e Best Composer, atenderam muito pouco as necessidades quanto a funcionalidades para a composição, onde apesar de possuírem uma boa usabilidade, ainda são bastante limitados quanto às questões musicais. Outras características também elencadas no Quadro 1, são quanto às questões de compartilhamento de áudios e criação colaborativa. Quanto à primeira opção, apenas o programa musicLine e Sibelius oferecem o recurso de compartilhamento. No entanto, nenhum deles possui a possibilidade de criação cooperativa, com controles de versões ou registros de alteração.

3 DESENVOLVIMENTO DO APLICATIVO

Neste capítulo são abordadas as etapas contempladas na prossecução do desenvolvimento dos novos requisitos do aplicativo, sendo abrangidas em quatro seções distintas, que abordam respectivamente: os principais requisitos desta etapa do desenvolvimento evolutivo da aplicação, a apresentação técnica do aplicativo, os detalhes e particularidades da implementação e finalmente, aborda os resultados alcançados.

3.1 REQUISITOS

Os requisitos que envolvem as novas funcionalidades, bem como as alterações de funcionalidades já presentes no aplicativo de composição musical, são os relacionados abaixo:

- a) criar uma área para publicação, sendo esta uma lista, onde as músicas desenvolvidas possam ser disponibilizadas para a audição de outros usuários (RF);
- b) permitir que usuários anônimos possam visualizar as composições no editor, mas não permitir que os mesmos consigam salvar alterações (RF);
- c) criar a área para listar as composições do usuário, bem como as composições que o mesmo tem acesso para edição (RF);
- d) criar a funcionalidade de convites para grupo de trabalho, permitindo que sejam selecionados os usuários integrantes do grupo. Estes receberão um e-mail no momento do convite (RF);
- e) permitir que o aplicativo crie versões da composição conforme realizadas as alterações, bem como criar o versionamento na alteração por usuários distintos (RF);
- f) possibilitar que cada versão possa ser restaurada, de forma que possa ser alterada e salva em uma nova versão (RF);
- g) permitir a criação da lista de discussão (fórum) para cada composição, servindo para descrever alterações ou até discutir opiniões quanto às composições (RF);
- h) empregar a linguagem de desenvolvimento Java na implementação dos novos requisitos da aplicação e de suas alterações (RNF – Requisito Não Funcional);
- i) utilizar a plataforma de desenvolvimento Eclipse como ambiente de desenvolvimento (RNF);
- j) valer-se do *Software Development Kit* (SDK) para Android, como conjunto de ferramentas no desenvolvimento das funcionalidades propostas (RNF);
- k) utilizar do conjunto de ferramentas oferecido pelo *Android Developer Tools* (ADT) para desenvolvimento das funcionalidades da proposta (RNF);

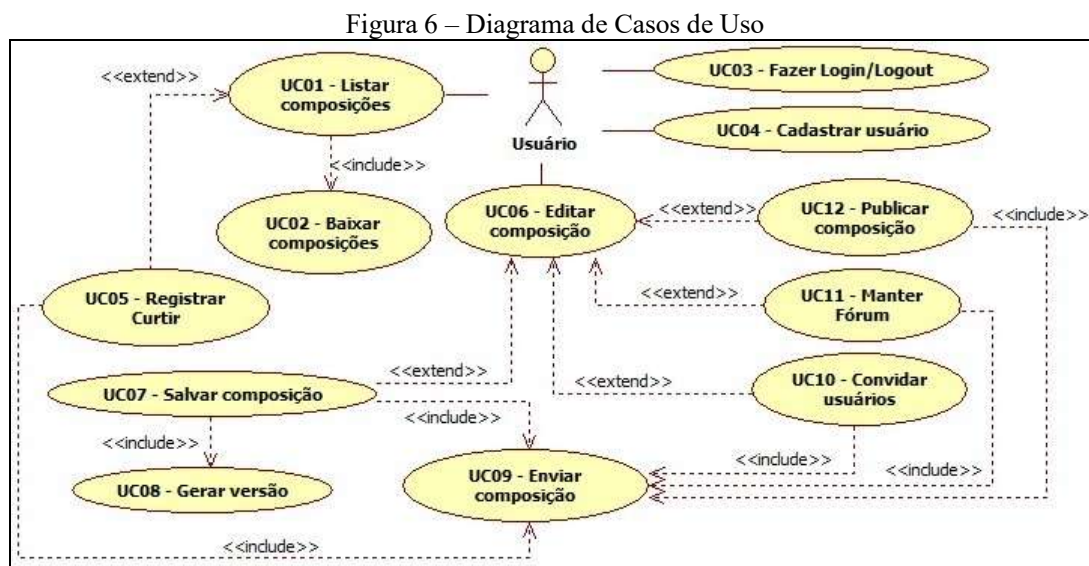
- 1) utilizar as funcionalidades oferecidas pela API QuickBlox, que provê meios para implementação de funcionalidades como comunicação entre usuários, compartilhamento de artefatos, controle de *login*, repositório de arquivos, entre outros (RNF).

3.2 ESPECIFICAÇÃO

Nesta seção serão descritas as especificações das extensões propostas como continuação do projeto apresentado por Weitgenant (2014), bem como os respectivos diagramas. A elaboração dos diagramas foi realizada através do emprego das definições da *Unified Modeling Language* (UML), utilizando para criação dos mesmos, as ferramentas StarUML para o Diagrama de Casos de Uso, a ferramenta Enterprise Architect para os Diagramas de Classe e o Bizagi Modeler para o Diagrama de Atividades, que são apresentados nas próximas sessões.

3.2.1 Diagrama de casos de uso

Esta seção destina-se à contextualização dos casos de uso apresentados no diagrama da Figura 6, onde estão colocados os casos de uso da aplicação, tratando em especial dos casos de uso contemplados na extensão do aplicativo de Weitgenant (2014). Casos de uso contemplados na versão anterior e que não sofreram alterações nesta versão foram abstraídos, como é o caso de todo o ambiente relacionado ao *chat*. As interações com o aplicativo são realizadas por um único ator em especial, este definido como *Usuário*.



Os novos requisitos foram agrupados em doze casos de uso (Apêndice A), conforme o diagrama e são abordados de forma individual.

O caso de uso UC01 - Listar composições trata da lista de composições apresentadas na inicialização do aplicativo, podendo exibir as publicações dos usuários da aplicação, ou as composições do usuário logado. A lista de composições também dá acesso à opção curtir que é tratada pelo caso de uso UC05 - Registrar Curtir, responsável por realizar o registro das curtidas atribuídas às composições publicadas.

O caso de uso UC02 - Baixar composições é utilizado durante a busca dos dados a serem relacionados na lista de composições, sendo que este caso de uso sempre será executado quando a atualização da lista de composições for requisitada.

O caso de uso de UC03 - Fazer Login/Logout trata do registro das sessões de usuários já cadastrados no aplicativo, realizando o controle de criar e manter a sessão do usuário após o login e encerrar a mesma quando requisitado o Logout da aplicação.

O caso de uso 04 - Cadastrar usuário é utilizado por novos usuários da aplicação para que possam realizar o seu cadastro e desfrutar de todas as funcionalidades disponibilizadas pelo aplicativo.

O caso de uso do UC06 - Editar composição da aplicação trata de toda a funcionalidade de visualização, criação e alteração das composições, possibilitando o acesso às várias funcionalidades da aplicação relacionadas à edição das composições.

Quando o usuário seleciona a opção para salvar a composição, dá-se início ao caso de uso UC-07 - Salvar composição, que realiza a gravação das edições realizadas na composição e envia a mesma para o servidor. No momento em que a aplicação salva a composição, o caso de uso UC08 - Gerar versão é acionado para tratar a necessidade de geração de novas versões para a composição. Após a composição estar devidamente salva e sua versão gerada, o caso de uso UC09 - Enviar composição realiza o envio da mesma para o servidor da aplicação.

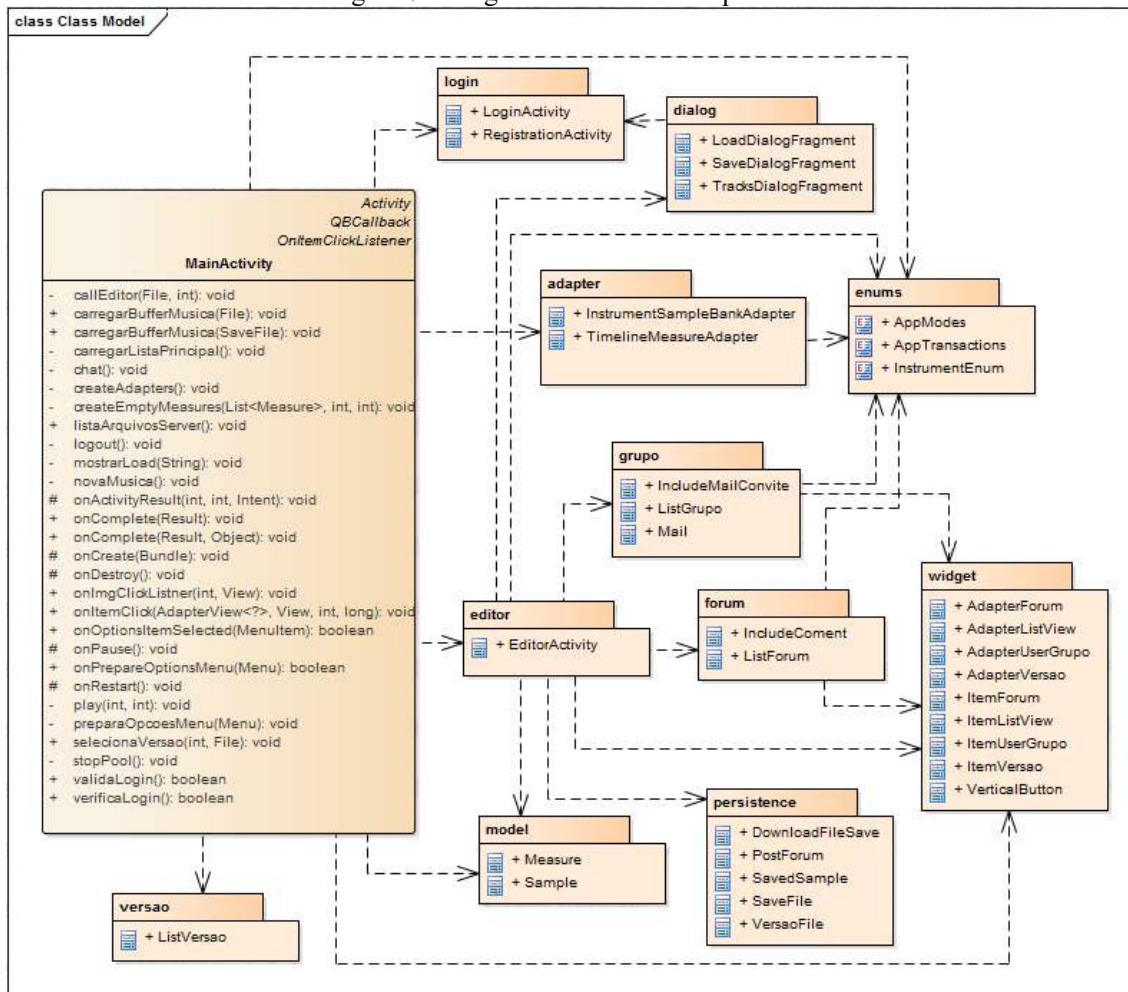
O caso de uso UC10 - Convidar usuários define a funcionalidade que permite convidar outros usuários a editar e trabalhar de forma colaborativa em uma mesma composição. Os usuários podem por meio do caso de uso UC11 - Manter Fórum, contar com as funcionalidades presentes na lista de discussão da composição, para registrar alterações de versões, informações quanto à composição ou sobre o grupo de compositores.

Durante o processo de edição de uma composição, o usuário poderá através do caso de uso UC12 - Publicar composição realizar a publicação da composição para os demais usuários do aplicativo, permitindo que através da lista pública, outros usuários possam contemplá-la.

3.2.2 Diagrama de Classes

Nesta seção são descritos os relacionamentos e estrutura das classes desenvolvidas na implementação das extensões propostas ao aplicativo. Na Figura 7 são demonstrados os agrupamentos lógicos das classes implementadas.

Figura 7 - Diagrama de classes do aplicativo



A classe **MainActivity** é a responsável pela inicialização do aplicativo, armazenando as informações necessárias para que o aplicativo possa se registrar no servidor e realizando as demais inicializações. O método `listaArquivosServer` realiza a leitura dos arquivos do servidor e retorna a lista de arquivos para a aplicação. Esta lista é utilizada pelo método `carregarListaPrincipal` para realizar a listagem dos arquivos para o usuário. Quando o usuário clica em alguma composição o método `play` realiza a chamada do método `carregarBufferMusica`, que realiza a carga dos *samples* da composição e inicia a execução de todo o áudio da composição ou até que o usuário clique novamente sobre a composição, pois nesse caso a aplicação, através do método `stopPool`, encerra a execução do áudio.

Quando o usuário, através da lista de composições públicas, solicita a visualização de uma determinada composição através do clique longo, ocorre a chamada do método `callEditor`, com um parâmetro indicando que a abertura do editor deverá se dar em modo de leitura. Quando o usuário, estando na sua lista privada, realiza a ação de abertura de uma composição, a aplicação verifica se a composição possui mais que uma versão e em caso afirmativo, realiza a chamada da lista de versões, carregada através da classe `ListVersao`, apresentada na Figura 8.

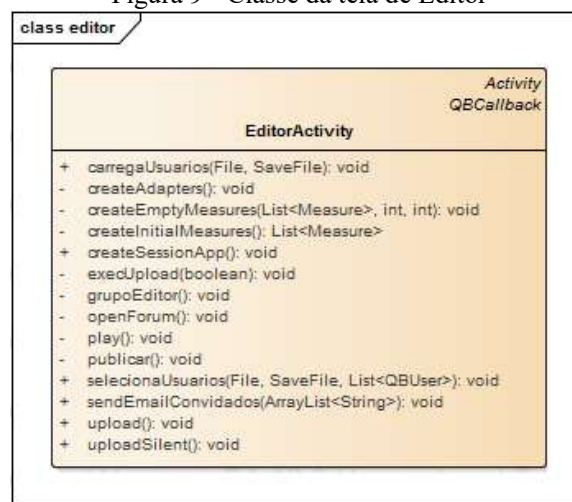
Figura 8 – Classe da tela de Versões



Através do método `carregarListaversao`, todas as versões anteriores da composição são exibidas em uma lista. Quando o usuário indicar a versão desejada, a aplicação finaliza a lista de versões e retorna ao escopo da classe `MainActivity`, prosseguindo com a chamada do editor, com a versão desejada.

A classe `EditorActivity` contém todas as funcionalidades relacionadas à visualização e edição de composições, conforme Figura 9.

Figura 9 - Classe da tela de Editor

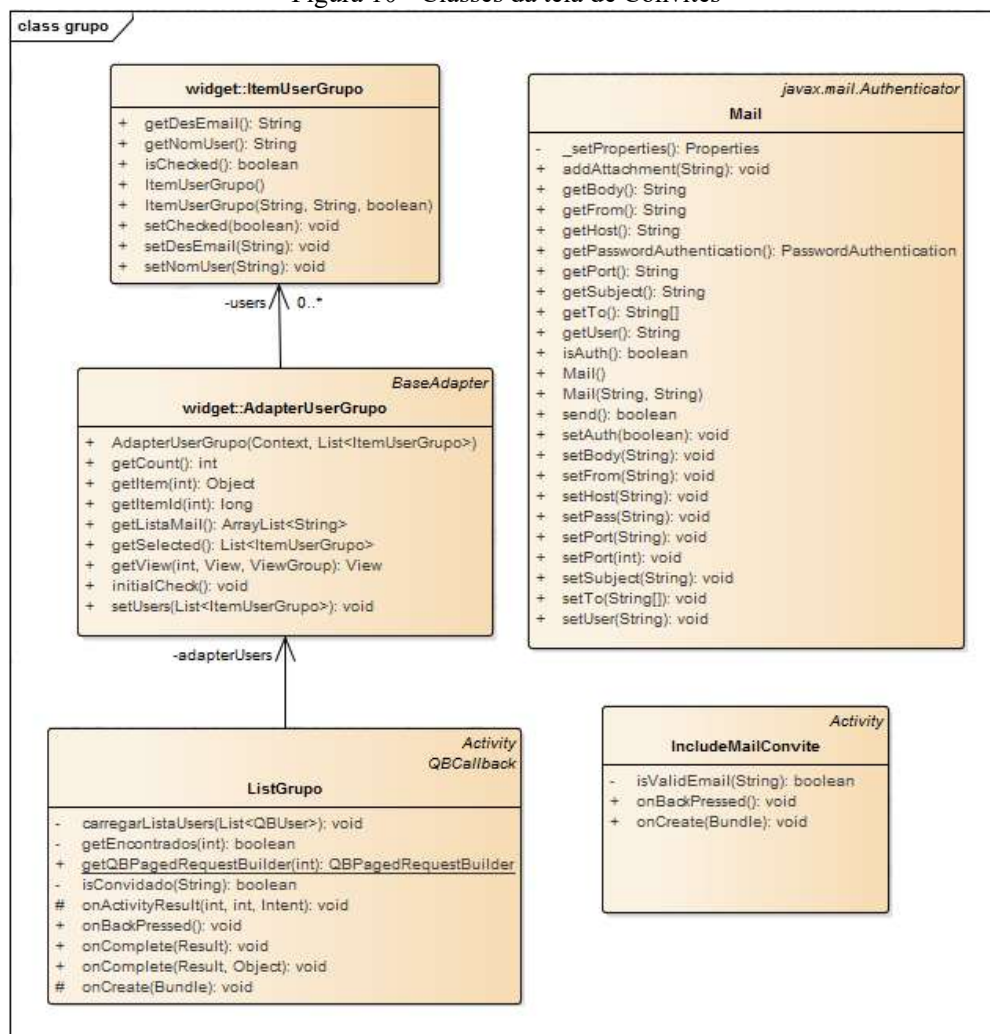


No momento em que a classe `EditorActivity` é inicializada, ela constrói a exibição da composição e a apresenta ao usuário, possibilitando o acesso a diversas funcionalidades como lista de convite, lista de discussão, salvar e publicar composições.

Os métodos de gravação e envio da composição para o servidor são executados através dos métodos `upload` ou `uploadSilent`. Ambos os métodos possuem o mesmo processamento, mas no caso do método `uploadSilent`, nenhuma mensagem é apresentada para o usuário, sendo sua principal utilização ser o envio de atualizações ao servidor de forma transparente ao usuário.

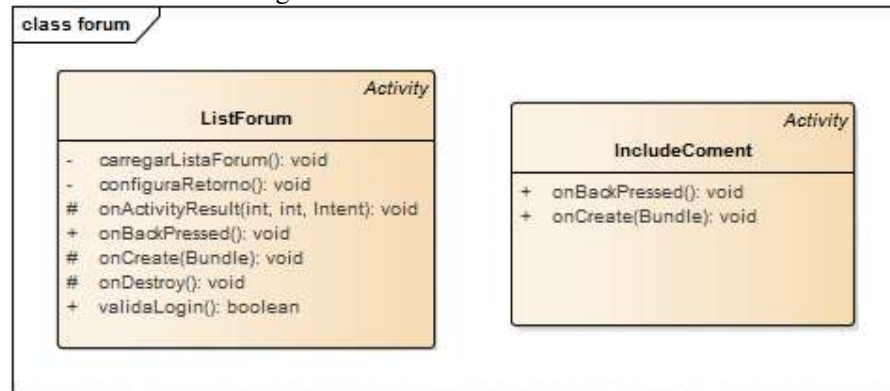
A lista de convite de compositores é implementada através da classe `ListGrupo`, apresentada na Figura 10. Esta classe é responsável por listar os usuários da aplicação que podem ser convidados a colaborar na composição, sendo esta listagem realizada pelo método `carregarListaUsers`, que retorna todos os usuários da aplicação, excluindo da lista o usuário logado. Quando o usuário deseja adicionar um colaborador que não possui cadastro, o aplicativo permite através da classe `IncludeMailConvite` que o usuário informe o e-mail da pessoa que deseja convidar e a aplicação realiza o envio do convite por e-mail utilizando a classe `Mail`.

Figura 10 - Classes da tela de Convites



A Figura 11 apresenta o diagrama da classe `ListForum`, que dá suporte a lista de discussão, juntamente com a classe `IncludeComent`, que permite a inclusão de comentários na lista.

Figura 11 - Classes da tela do Fórum

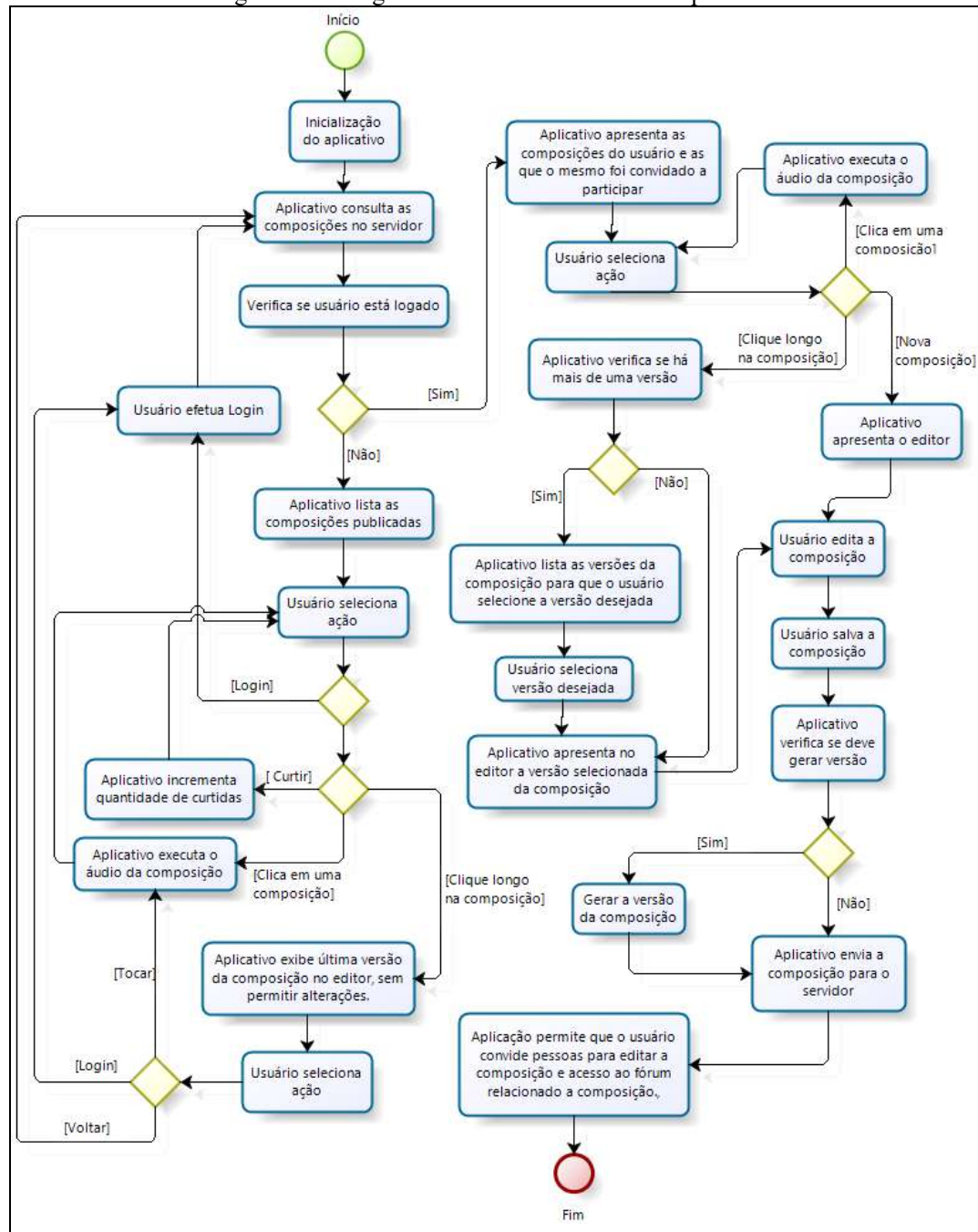


A lista de discussão é disponibilizada de forma única em cada composição e tem por objetivo oferecer aos usuários uma funcionalidade onde seja possível relatar opiniões, registrar detalhes das alterações e tratar demais questões relacionadas à composição. A lista de discussões é uma ferramenta diretamente ligada à lista de convidados, pois ela proporciona uma comunicação mais aberta e direta ao grupo envolvido no processo criativo de determinada composição.

3.2.3 Diagrama de Atividades

O diagrama de atividades apresentado na Figura 12 busca relacionar de uma forma macro as tarefas realizadas pela aplicação durante sua execução, desde sua inicialização até à gravação e publicação de uma composição.

Figura 12 – Diagrama de atividades macro do aplicativo



Analisando o fluxo apresentado na Figura 12, dado a inicialização do aplicativo, é realizada a consulta dos arquivos no servidor. As composições retornadas nesta consulta são exibidas conforme a situação de *login* do usuário, portanto, apresentando duas situações:

- usuário não está logado: neste caso são listadas todas as composições que foram publicadas pelos usuários da aplicação;
- usuário está logado: são exibidas na lista as composições da autoria do próprio usuário e as composições da qual o mesmo foi convidado a participar.

Na Figura 12, também se pode observar que quando o usuário não está logado, ele apenas possui acesso para ouvir a composição, curtir e visualizar a estrutura, mas não possui permissão para fazer alterações e apenas pode contemplar a última versão da composição. Depois de efetuado o *login*, a lista de composições é recarregada e a partir de então o usuário não tem mais permissão para curtir as publicações, mas possui permissão para alterar as composições, convidar novos colaboradores, visualizar e participar dos grupos de discussão através do fórum, criar novas composições e publicar composições que ainda não se encontram nessa situação, contando ainda com o acesso as versões anteriores de cada composição.

3.3 IMPLEMENTAÇÃO

Nesta seção serão abordadas as técnicas e ferramentas utilizadas na implementação das funcionalidades propostas e a operacionalidade das mesmas. Na seção 3.3.1 destacam-se as técnicas e ferramentas utilizadas no desenvolvimento dos requisitos. Na seção 3.3.2 são abordados etapas das implementações realizadas. Na seção 3.3.3 apresenta-se a operacionalidade do aplicativo.

3.3.1 Técnicas e ferramentas utilizadas

O desenvolvimento dos novos requisitos propostos para o aplicativo foram realizadas com o uso da linguagem Java, conforme as versões anteriores. A interface de desenvolvimento utilizada para dar o apoio necessário no desenvolvimento foi o Eclipse em sua versão 4.4.2, denominado de Luna, juntamente com as funcionalidades e ferramentas disponibilizadas pelo *plugin* customizado chamado *Android Developer Tools* (ADT), que fornece um ambiente poderoso e integrado para a construção de aplicações Android. O uso da SDK QuickBlox, que já foi utilizada na versão desenvolvida por Weitgenant (2014), para gerir os controles de *login*, troca de mensagens no *chat* e como repositório para arquivos foi mantida, pois a mesma ofereceu os recursos necessário para os novos requisitos.

Os testes de execução e verificações de funcionamento do aplicativo foram realizados diretamente em um dispositivo móvel através do gerenciamento oferecido pelo *Android Virtual Device (AVD) Manager*. Optou-se por utilizar esse formato para testes, devido ao desempenho dos dispositivos virtuais se apresentarem muito lentos e apresentando também problemas de travamento constante. O dispositivo utilizado para os testes foi um *smartphone* Motorola Moto X, modelo XT1058 e versão do Android 5.1.

3.3.2 Etapas da implementação

Nesta seção são apresentados detalhes das implementações realizadas para atender aos novos requisitos definidos para a aplicação, abordando no decorrer desta, detalhes sobre as listas de composições, edição e visualização das composições, envio dos arquivos para o servidor, convite para composição colaborativa, fórum de discussão e publicação de composições.

3.3.2.1 Listas de composições

As listas de composição da aplicação são responsáveis por exibir ao usuário as composições publicadas para que possam ser ouvidas e visualizadas, sem que sejam permitidas alterações nas mesmas ou exibir as composições do próprio usuário logado e também as composições nas quais ele foi convidado a participar, por este motivo elas são definidas respectivamente como lista pública e lista privada.

No momento em que a aplicação é acessada pelo usuário ou no momento em que o mesmo acessa a funcionalidade da lista, a aplicação verifica se o usuário realizou o *login* na aplicação e através dessa verificação ele define quais os objetos devem ser listados.

Inicialmente, como apresentado no Quadro 2, a aplicação realiza a leitura dos arquivos do servidor, retornando informações de todos os objetos salvos.

Quadro 2 - Leitura dos arquivos do servidor

```

01 public void listaArquivosServer() {
02     // Carregando
03     mostrarLoad("Carregando lista de arquivos...");
04     // Lista que conterà todos os itens
05     listFiles = new ArrayList<SaveFile>();
06     files = null;
07     QBCustomObject.getObjects("composit", new QBCallbackImpl() {
08         @Override
09         public void onComplete(Result result) {
10             if (result.isSuccess()) {
11                 QBCustomObjectLimitedResult coresult =
12                     (QBCustomObjectLimitedResult) result;
13                 ArrayList<QBCustomObject> co =
14                     coresult.getCustomObjects();
15                 HashMap<String, Object> fields =
16                     new HashMap<String, Object>();
17                 // Criar a lista de objetos para baixar
18                 ArrayList<DownloadFileSave> baixando =
19                     new ArrayList<DownloadFileSave>();
20                 for (QBCustomObject qbCustomObject : co) {
21                     // Criar a tarefa
22                     baixando.add(new DownloadFileSave(qbCustomObject));
23                 }
24             }
25         }
26     });

```

Observa-se no Quadro 2, que a consulta dos arquivos no servidor é realizado através do uso do método `QBCustomObject.getObjects`. Os arquivos de composições são gravados

no servidor como objetos customizados, pois este tipo de objeto permite que um usuário tenha acesso aos arquivos gravados pelos demais usuários, visto que não há controle de acesso em nível de usuários para este tipo de objeto.

Cada objeto customizado retornado é inserido em um `ArrayList` de objetos da classe `DownloadFileSave`, esta que por sua vez é uma classe que implementa a classe `Runnable`, oferecendo assim, a possibilidade para realizar *downloads* de cada arquivo em uma `thread` distinta e de forma paralela, conforme visto no Quadro 3.

Quadro 3 – Realizando o início dos *downloads*

```

01     if (baixando.size() > 0) {
02         for (DownloadFileSave downloadFileSave : baixando) {
03             Thread ativar = new Thread(downloadFileSave);
04             process.add(ativar);
05             process.get(process.size() - 1).start();
06         }
07     }
08     try {
09         for (Thread thread : process) {
10             thread.join(); // Aguardar
11         }
12     } catch (InterruptedException e) {
13         e.printStackTrace();
14     }

```

Após ativar cada uma das `Threads` definidas conforme retorno dos objetos customizados, a aplicação aguarda a conclusão das mesmas. Cada `Thread` ao ser executada realiza a chamada do método de *download*, definido no método `run()` da classe `DownloadFileSave`, apresentada no Quadro 4.

Quadro 4 – *Download* das composições

```

01     @Override
02     public void run() {
03
04         Looper.prepare();
05
06         QBCustomObjectsFiles.downloadFile(this.download, "composit"
07             , new QBCallbackImpl() {
08
09             @Override
10             public void onComplete(Result result) {
11                 QBFileDownloadResult downloadResult =
12                     (QBFileDownloadResult) result;
13
14                 if (result.isSuccess()) {
15                     byte[] content = downloadResult.getContent();
16                     InputStream is = downloadResult.getContentStream();

```

Quando o *download* é concluído o método `onComplete` é executado, sendo este responsável por extrair o conteúdo do arquivo vindo no retorno do método `QBCustomObjectsFiles.downloadFile`, para que assim possa ser criado o objeto `SaveFile`, conforme apresentado no Quadro 5.

Quadro 5 – Convertendo o texto do arquivo

```

01 // Prepara o Gson
02 Gson gson = new Gson();
03
04 // Gera o JSon e cria o arquivo
05 SaveFile arquivo = gson.fromJson(texto,
06 SaveFile.class);

```

Após terminar todos os *downloads*, o aplicativo irá realizar a carga da lista a ser apresentada para o usuário, realizando o filtro de quais composições devem ser apresentadas.

3.3.2.1.1 Lista pública

A lista de composições definida como pública é exibida sempre que o usuário acessa a lista de composições sem que o *login* tenha sido realizado. Nela, são listadas todas as composições dos usuários que estão no servidor e que estão com a situação de publicadas. O Quadro 6 destaca como é realizada a verificação para determinar se a composição irá ou não ser exibida ao usuário.

Quadro 6 – Verifica acesso à composição

```

01 boolean userLogin = verificaLogin();
02
03 for (DownloadFileSave downloadFileSave : baixando) {
04     SaveFile save = downloadFileSave.getArquivo();
05     // Se usuário não está logado
06     if (!userLogin) {
07         // Verifica se o arquivo é publico
08         if (save.isPublico()) {
09             listFiles.add(save);
10         }
11     } else { // Se usuário está logado
12         // Guardar e-mail e login do usuário logado
13         String email = ((App) getApplication()).getQbUser().getEmail();
14         String login = ((App) getApplication()).getQbUser().getLogin();
15
16         // Verifica se o usuário logado é convidado ou autor
17         if (save.isAutor(login) || save.isConvidado(email)) {
18             listFiles.add(save);
19         }
20     }
21 }

```

Conforme apresentado no código acima, a aplicação verifica se o *login* foi realizado e para cada um dos arquivos retornados verifica se a composição foi publicada através da chamada do método *isPublico*.

3.3.2.1.2 Lista privada

A lista de composições privada, diferentemente da lista pública apresentada anteriormente, exibe todas as composições do usuário e as composições que o mesmo faz parte do grupo colaborativo, isto independente destas estarem publicadas ou não. Esta lista é

apresentada para o usuário a partir do momento em que o *login* é realizado e enquanto a sessão do mesmo perdurar.

Conforme apresentado no Quadro 6, a aplicação percorre todos os arquivos que foram baixados do servidor e realiza a verificação para determinar se a composição irá ou não ser exibida ao usuário, levando em consideração o resultado dos métodos `isAutor` e `isConvidado` que indicam, respectivamente, se o usuário é autor ou convidado da composição.

3.3.2.2 Edição e visualização das composições

O editor de composições pode ser utilizado tanto para a edição da composição como para a simples visualização da estrutura das composições. Quando um usuário está acessando a lista pública, ele apenas tem permissão de ouvir as músicas ou visualizar a estrutura da mesma, dessa forma, quando uma composição é visualizada, o editor tem as funções de menu relacionadas à edição e à lista de *samples* ocultada.

Devido às novas funcionalidades de listas agregadas à aplicação, tornou-se necessário que o editor receba informações através de parâmetros, principalmente do arquivo que o mesmo deve abrir. No Quadro 7 destaca-se o trecho de código onde os parâmetros são recebidos pelo editor.

Quadro 7 – Recepção de parâmetros pelo editor

```

01 // Receber os parametros através do Intent
02 Intent it = this.getIntent();
03 // Se recebeu o Intent
04 if (it != null) {
05     Bundle params = it.getExtras();
06     // Se encontrou os parametros
07     if (params != null) {
08         // Buscar o modo da tela
09         this.appMode = params.getInt("AppModes");
10         this.versionLoad = params.getInt("versao");
11         // Arquivo
12         File fileOpen = (File) params.getSerializable("file");
13         // Verificar se foi passado arquivo para ser aberto
14         if (fileOpen != null) {
15             // Criar o fragmet
16             this.onDialogLoadClick(new LoadDialogFragment(), fileOpen);
17             fileMusic = fileOpen;
18         }
19     }
20 }
21

```

Conforme pode ser observado, o editor recebe como parâmetros três informações para serem utilizadas na sua configuração inicial:

- a) `appMode`: define se o editor deve ser exibido em modo de edição ou visualização;

- b) `versionLoad`: indica a versão que está sendo enviada para exibição;
- c) `fileOpen`: recebe o objeto `File` que contem a estrutura da composição.

Quando o editor é aberto em modo de visualização, o valor do parâmetro `versionLoad`, sempre é “-1” (um negativo). Este valor é utilizado para indicar à aplicação que a versão a ser utilizada é a composição mais atual. Nesse caso, como o usuário público não tem acesso às versões anteriores da aplicação, o parâmetro possui este valor pré-fixado.

3.3.2.3 Envio dos arquivos para o servidor

O envio de arquivos para o servidor é feito em vários momentos pela aplicação. A cada alteração na composição, o arquivo é reenviado ao servidor para que a alteração possa ser exibida aos demais usuários. O *upload* do arquivo para o servidor é realizado através do método apresentado no Quadro 8.

Quadro 8 – Método de *upload* dos arquivos

```

01 public void upload() {
02     final File file = this.getSaved();
03     final QBCustomObject qbCustomObject;
04
05     if (this.customObjID == null) {
06         qbCustomObject = new QBCustomObject("composit");
07         QBCustomObjects.createObject(qbCustomObject
08                                     , new QBCallbackImpl() {
09             @Override
10             public void onComplete(Result result) {
11                 if (result.isSuccess()) {
12                     QBCustomObjectsFiles.uploadFile(file, qbCustomObject
13                                                     , "composit", new QBCallbackImpl() {
14                 @Override
15                 public void onComplete(Result result) {
16                     if (result.isSuccess()) {
17                         System.out.println("Arquivo salvo com sucesso!");
18                     } else {
19                         System.out.println(result.getErrors()
20                                           .toString());
21                     }
22                     file.delete();
23                 }
24             });
25         } else {
26             Log.e("Errors", result.getErrors().toString());
27         }
28     }
29     qbCustomObject = new QBCustomObject("composit",
30                                       this.customObjID);
31     QBCustomObjectsFiles.uploadFile(file, qbCustomObject,
32                                     "composit", new QBCallbackImpl() {
33     });
34 }

```

O método `upload()`, ao realizar o envio da composição, inicialmente verifica se há valor para a variável `customObjID`. Esta variável indica se o arquivo já possui um identificador que é retornado pelo servidor no momento da leitura das composições, o que indica neste caso, que a composição já existe no servidor de destino e não necessita ser criada, apenas atualizada. Caso a variável não apresente este valor, como no caso de novas composições, a aplicação necessita requisitar inicialmente a criação do objeto, utilizando-se para isto da chamada do método `QBCustomObjects.createObject`. O envio do arquivo físico contendo os dados da composição é realizado apenas no momento em que o método `QBCustomObjectsFiles.uploadFile` é chamado. Logo após a conclusão do envio o arquivo físico criado pela chamada do método `getSaved()`, é excluído através da execução do comando `file.delete()`.

3.3.2.4 Convite para composição colaborativa

A funcionalidade de convite de usuários para colaborarem em uma composição é acessada através da opção “Convidar”, na tela do editor. Quando a opção é acessada, a aplicação exibe a lista de todos os usuários da aplicação e os e-mails dos usuários convidados que ainda não realizaram o cadastro. Conforme pode-se ver no Quadro 9, a aplicação utiliza o método `QBUsers.getUsers` para realizar a consulta dos usuários que serão exibidos na lista para seleção e envio do convite.

Quadro 9 – Selecionar os usuários para convite

01	<code>QBUsers.getUsers(getQBPageRequestBuilder(currentPage),</code>
02	<code>new QBCallback() {</code>
03	<code> @Override</code>
04	<code> public void onComplete(Result result) {</code>
05	<code> if (result.isSuccess()) {</code>
06	<code> QBUserPagedResult usersResult = (QBUserPagedResult) result;</code>
07	<code> List<QBUser> users = usersResult.getUsers();</code>
08	<code> }</code>
09	<code> }</code>

Após a seleção do grupo de usuários a serem convidados, a aplicação envia um e-mail aos usuários selecionados, utilizando para isto a classe `Mail`, conforme mostra o Quadro 10.

Quadro 10 – Método para envio de e-mails

```

01 // Método para realizar o envio de e-mails
02 public void sendEmailConvidados (ArrayList<String> mails) {
03
04     if (mails.size() == 0) {
05         return;
06     }
07
08     String subject = "MyMusic - Convite para Grupo de Composição";
09     String message = "Gostaria de convidá-lo a participar"
        + " do grupo de composição da música \""
        + saveFile.getNome()
        + "\". \n\n"
        + "Caso ainda não possua o aplicativo MyMusic, "
        + "acesse a Play Store, faça o download "
        + "e cadastre-se.\n\n"
        + "Venha criar com a gente!";
10
11     Mail m = new Mail(loginMail,passMail);
12
13     // Dados conexão BOL
14     m.setHost("smtps.bol.com.br");
15     m.setPort(587);
16     m.setFrom("furb2015.mymusic@bol.com.br");
17     m.setSubject(subject);
18     m.setBody(message);
19
20     // Percorrer todos os e-mails
21     for (String mail : mails) {
22         String[] toArr = m.setTo(toArr);
23         try {
24             if (m.send()) {
25                 Log.i("MailApp", "E-mail para \"" + mail
26                     + "\" enviado com sucesso");
27             } else {
28                 Log.i("MailApp", "E-mail para \"" + mail
29                     + "\" não pode ser enviado.");
30             }
31         } catch (Exception e) {
32             Log.e("MailApp", "Could not send email", e);
33         }
34     }
35 }

```

O método `sendEmailConvidados()`, apresentado no Quadro 10, recebe como parâmetro a lista de e-mails que devem receber o convite e realiza a configuração da mensagem a ser enviada aos convidados, além dos parâmetros necessários para o acesso serviço de e-mail.

3.3.2.5 Fórum de discussão

O requisito que trata da lista de discussão para as composições foi implementado no formato de fórum, onde os usuários podem incluir seus comentários. Cada composição possui um fórum exclusivo onde todas as mensagens estão atreladas unicamente à composição e às suas versões. O acesso ao fórum é realizado através da função “Fórum”, presente no editor

quando o usuário acessa estando devidamente logado e visualizando uma composição salva. No momento do acesso ao fórum, a aplicação realiza a consulta de todas as postagens realizada para a composição, conforme pode ser visto no Quadro 11.

Quadro 11 – Carregar a lista de postagens do fórum

```

01 private void carregarListaForum () {
02     // Lê o list view
03     listView = (ListView) findViewById(R.id.forumList);
04
05     // Buscar a lista de posts
06     ArrayList<PostForum> forumList = saveFile.getForum();
07     posts = new ArrayList<ItemForum>();
08
09     // Se encontrou registros do fórum
10     if (forumList != null) {
11         // Percorrer os itens retornados
12         if (forumList.size() > 0) {
13             // Copiar o array
14             ArrayList<PostForum> forumReverse = forumList;
15
16             Collections.reverse(forumReverse);
17
18             for (PostForum postForum : forumReverse) {
19                 // CRIAR O ITEM DE POST
20                 ItemForum post = new ItemForum(postForum.getUser(),
21                                                 postForum.getData(), postForum.getTexto());
22                 // Adicionar ao arraylist
23                 posts.add(post);
24             }
25     }

```

A carga das postagens é feita a partir da recepção das informações do fórum através do método `getForum()`. Este método é responsável por retornar o `ArrayList` contendo a lista de objetos do fórum, objetos da classe `PostForum`, que contém as informações do autor da postagem, o texto e a data e hora da mesma. Devido a cada postagem ser adicionada ao final do `ArrayList`, conforme pode-se observar no quadro Quadro 12, utiliza-se o método `Collections.reverse()`, para que a ordem do `ArrayList` seja alterado e assim as postagens sejam escritas em uma ordem cronológica decrescente.

Quadro 12 – Adicionar postagens ao Fórum

```

01 Bundle params = data.getExtras();
02 String comentario = params.getString("coment");
03
04 // Adicionar o comentário ao fórum
05 saveFile.addPost(((App) getApplication())
06                 .getQbUser().getLogin(), comentario);
07
08 saveFile.upload();
09
10 this.carregarListaForum();

```

O método `addPost()`, responsável por registrar a postagem, é chamado logo após a leitura do parâmetro que guarda o comentário do usuário. Além do comentário o método

também recebe por parâmetro o usuário, que é lido da aplicação através da chamada do método `((App) getApplication()).getQbUser().getLogin()`. Após a inclusão do comentário do usuário na lista de postagens da composição, observa-se que o arquivo é enviado para o servidor, através da chamada `saveFile.upload()`. Assim os demais usuários que acessarem o fórum desta composição poderão visualizar a postagem. Por fim, é realizado a chamada do método `carregarListaForum()`, que carrega novamente a lista do fórum, exibindo também a postagem realizada pelo usuário.

3.3.2.6 Publicação de composições

A indicação de uma composição estar na situação de publicada ou não, é realizada através de um atributo da composição. Quando a composição é criada, o atributo `idPublico` é definido inicialmente como *false*. Dessa forma é possível realizar filtros para que a composição não seja exibida na lista pública de composições.

Uma composição se torna visível ao público a partir do momento em que o usuário, através do menu auxiliar do editor, seleciona a opção “Publicar”. Conforme mostra o Quadro 13, quando a opção é selecionada a composição tem o atributo `idPublico` alterado para *true* por intermédio da chamada do método `setIdPublico(true)`.

Quadro 13 – Publicar a composição

```

01 private void publicar() {
02
03     if (saveFile.isPublico()) {
04         Toast.makeText(this, "Composição já esta publica.",
05                         Toast.LENGTH_SHORT);
06     } else {
07         // Setar o arquivo como publico
08         saveFile.setIdPublico(true);
09
10         // Envia o arquivo para o servidor
11         saveFile.upload();
12
13         Toast.makeText(this, "Composição publicada com sucesso.",
14                         Toast.LENGTH_SHORT);
15     }
16 }

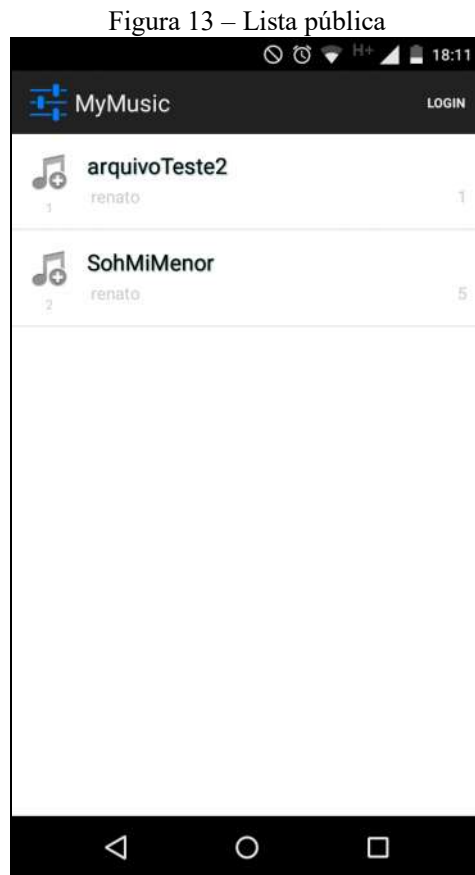
```

3.3.3 Operacionalidade da implementação

Esta seção apresenta a operacionalidade dos requisitos implementados, abordando principalmente a representação visual da aplicação, sendo esta a visão apresentada ao usuário. A seção foi dividida com base no diagrama de atividades apresentado na Figura 12, apresentando as interfaces que foram implementadas para atender aos requisitos propostos.

3.3.3.1 Lista Pública

No momento da inicialização do aplicativo *MyMusic*, é realizada a carga e exibição da lista de composições públicas, conforme pode ser observado na Figura 13.



A lista pública de composições é exibida a todos os usuários da aplicação, mesmo os que ainda não tenham realizado o cadastro na mesma. A lista apresenta todas as composições nas quais os seus respectivos compositores indicaram como publicadas, dessa forma indicando que a composição está liberada para ser contemplada pelos usuários da aplicação. A lista pública permite ao usuário as ações básicas de reprodução do áudio da composição, a opção de curtir a composição, realizar a chamada da tela para visualização da estrutura da composição ou ainda efetuar o *login* na aplicação.

Para executar o áudio da composição, basta o usuário clicar sobre composição desejada na lista que a aplicação iniciará a reprodução do áudio e fará a indicação da composição que está sendo executada, conforme a Figura 14.

Figura 14 – Reproduzindo áudio da composição

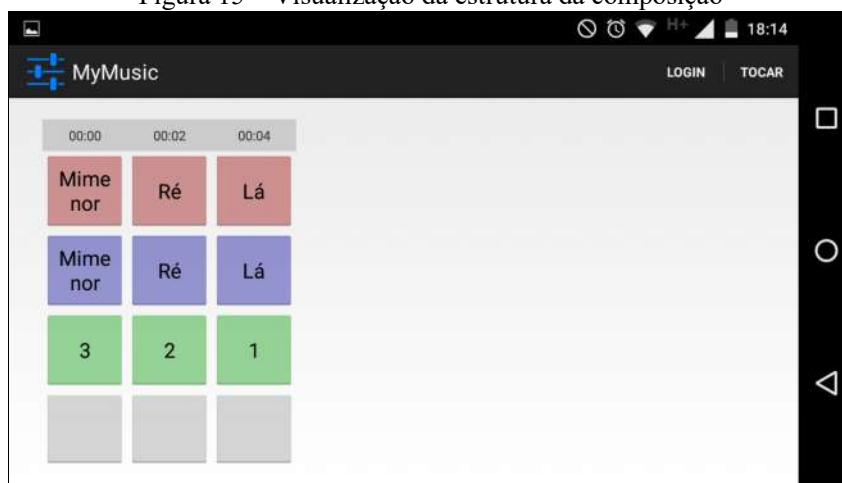


Na Figura 14 se observa a presença de uma imagem que é utilizada para atribuir o “curtir” à composição quando a mesma é clicada. Abaixo da imagem é apresentado o contador com a quantidade de curtidas que a composição recebeu. O número apresentado à direita da lista indica a versão atual da composição, sendo esta a versão executada na reprodução do áudio da composição.

A lista pública de composições ainda permite ao usuário a chamada da tela de visualização da estrutura da composição. Para acessar a visualização, o usuário deve aplicar um clique longo (pressionar e segurar) sobre a composição desejada.

Quando o usuário realiza a solicitação de visualização da estrutura da composição através da lista pública, a tela apresentada na Figura 15 é carregada.

Figura 15 – Visualização da estrutura da composição

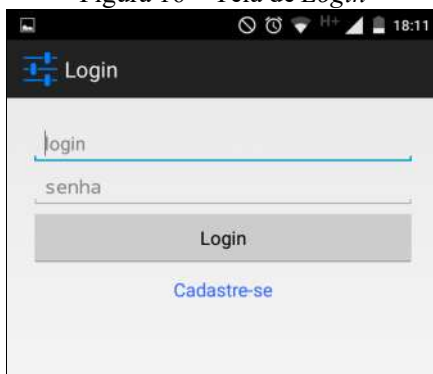


Na tela de visualização é exibida a distribuição dos *samples* na *timeline* e as únicas funcionalidades oferecidas ao usuário são o acesso ao *login* e a opção de executar o áudio da composição (Tocar). Devido ao fato do usuário, quando não logado, não possuir acesso às versões anteriores da composição, a versão apresentada para execução e visualização será exclusivamente a última versão criada para a composição.

3.3.3.2 Login e Cadastramento de usuários

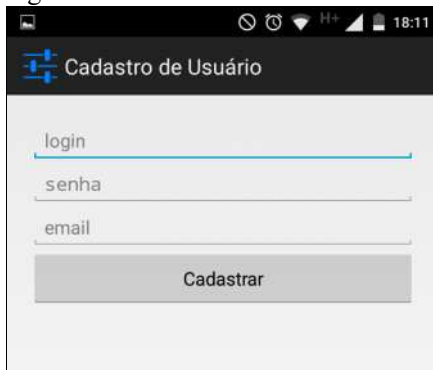
O acesso às funcionalidades de composição do aplicativo é permitido somente após o *login* ser realizado pelo usuário. Para isto, o mesmo necessita atualizar a tela de *login* da aplicação, conforme apresentado na Figura 16, e informar o seu respectivo *login* e *senha*.

Figura 16 – Tela de *Login*



Caso o usuário não possua cadastro na aplicação, a tela de *login* oferece um acesso à tela de cadastramento, através do *link* “Cadastre-se”. Quando o usuário clica no *link*, é direcionado para a tela de cadastro apresentada na Figura 17.

Figura 17 – Tela do cadastro de usuário

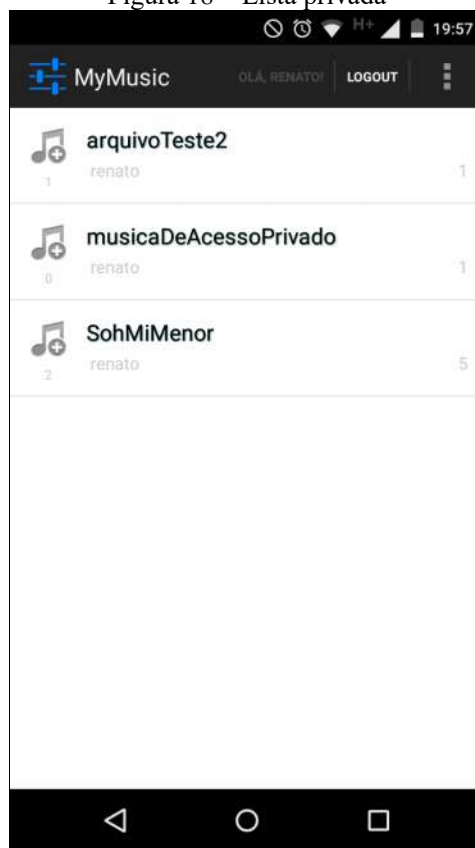


É na tela de cadastro que o usuário informa o *login* que deseja utilizar, a *senha* e o seu e-mail. Após informar os dados e selecionar “Cadastrar” a aplicação verifica se o *login* e o e-mail são únicos e em caso afirmativo, realiza a criação do usuário no servidor, efetua o *login* na aplicação e retorna a lista de composições.

3.3.3.3 Lista Privada

Após a realização do *login* por parte do usuário, a aplicação realiza a carga e exibição da lista de composições do usuário e as que ele recebeu convite para participar, conforme apresentado na Figura 18.

Figura 18 – Lista privada



A lista privada permite ao usuário o acesso a várias funcionalidades de apoio para edição e compartilhamento das composições. A funcionalidade de execução do áudio da composição está disponível na lista privada da mesma maneira como é disponibilizada na lista pública, porém as funcionalidades de curtir e de abrir a composição no visualizador não estão disponíveis. A opção curtir no caso da lista privada, fica bloqueada, de forma que o usuário não tenha acesso a curtir suas próprias composições. Já a ação para a execução do visualizador é substituída pela funcionalidade de abertura da composição no editor, que dará acesso à área de edição da composição, além das demais funções.

3.3.3.4 Lista de versões da composição

A funcionalidade referente às versões das composições é acessada quando o usuário solicita, através do clique longo, a abertura de uma determinada composição, sendo exibida a lista de versões caso a composição selecionada possua mais que uma versão, conforme mostra a Figura 19.

Figura 19 – Tela de versões da composição

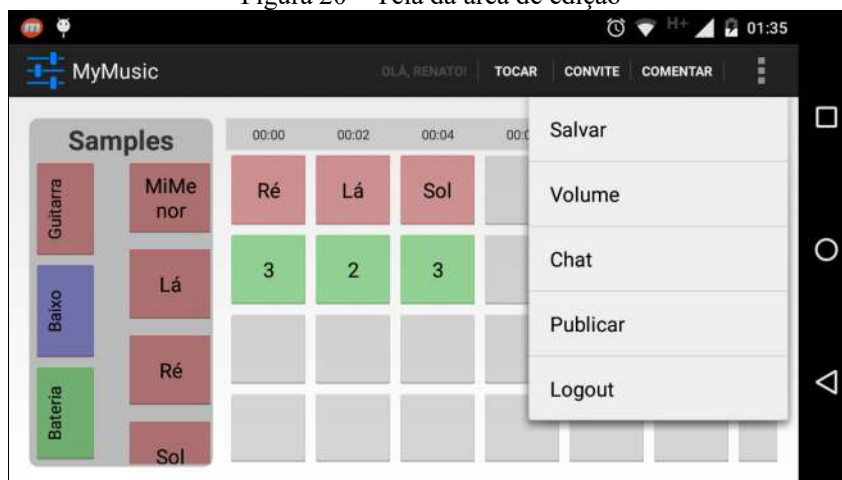


Conforme apresentado na Figura 19, a tela de versões lista todas as versões da composição, destacando em cor diferente a última versão e disponibilizando também a opção “Versão Atual”, que possibilita selecionar a versão atual da composição, sem ter que selecionar a mesma na lista.

3.3.3.5 Editor

O editor proporciona acesso às principais funções da aplicação. Na Figura 20 pode-se observar que é nesta tela que o usuário, além de editar a sua composição através da distribuição dos *samples* na *timeline*, realiza o acesso às telas de convite, ao fórum, salva as suas composições e pode publicar as mesmas.

Figura 20 – Tela da área de edição



O menu principal da tela, conforme pode ser observado na Figura 20, apresenta as opções de “Tocar”, “Convite” e “Comentar”, porém as opções Convite e Comentar só estão disponíveis para composições salvas, ou seja, para que em novas composições o usuário possa

selecionar convidados ou visualizar o fórum, primeiramente ele necessita salvar a composição.

No menu auxiliar, destacam-se as opções “Salvar”, responsável por realizar a gravação do arquivo no servidor, e a opção “Publicar”, que altera a composição de forma que ela seja inclusa na lista pública, para apreciação de outros usuários. Uma composição após sua publicação, não pode mais ser oculta, dessa forma a indicação de publicação da composição é realizada através do bloqueio da opção Publicar do menu e alterado o texto da opção para “Publicado”, conforme destaque na Figura 21.

Figura 21 – Indicação de composição publicada



3.3.3.6 Convidar usuários para composição

A lista de convidados para colaborar no desenvolvimento da composição é selecionada através da lista acessada pelo usuário através da função “Convite” do menu do editor. Conforme mostra a Figura 22, o aplicativo apresenta a lista de usuários cadastrados para que o usuário conectado possa selecionar a quem o convite de colaboração deva ser estendido.

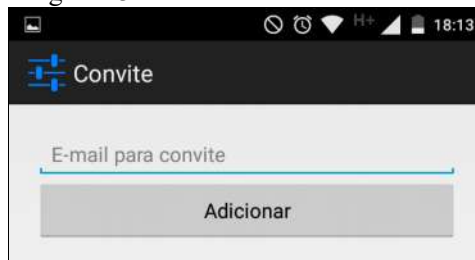
Figura 22 – Lista de usuários para convite



Um usuário convidado para participar da composição não pode mais ser removido do grupo. Dessa forma, após o envio do convite a tela bloqueia todo *checkbox* que já fora

marcado, impossibilitando desta forma o desmarque. Usuários que não possuem cadastro na aplicação também podem ser convidados através da opção “Adicionar”, que ao ser selecionada pelo usuário, apresenta a tela apresentada na Figura 23.

Figura 23 – Adicionar e-mail de convite



A tela requisita a informação do e-mail ao qual o convite deve ser enviado. Este e-mail será incluído na lista de convidados, onde será exibida da forma apresentada na Figura 24, enquanto não for realizado o cadastro por parte do convidado, sendo que após a realização do cadastro, a aplicação passa a exibi-lo também como um usuário cadastrado.

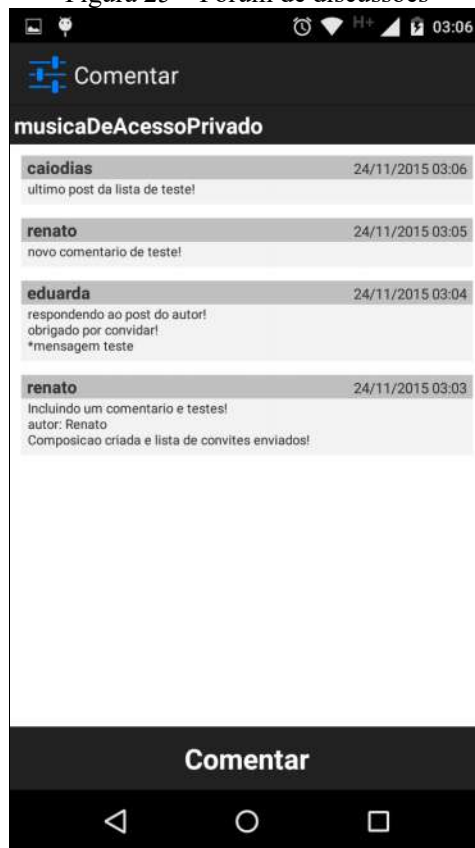
Figura 24 – Convidado inserido na lista



3.3.3.7 Fórum de discussão

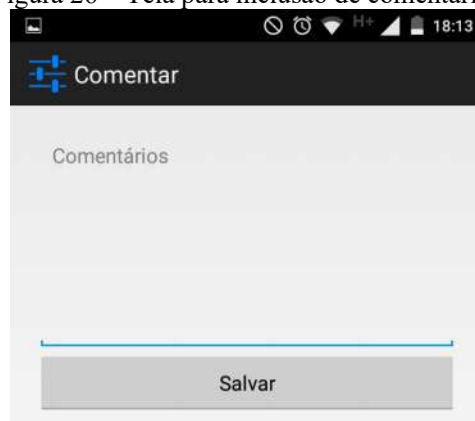
O fórum de discussão, que é acessado através da opção “Comentar” disponível no menu principal do editor, apresenta os comentários atrelados à composição. Estes comentários podem ser lidos e incluídos pelo autor da composição e/ou qualquer um dos convidados a colaborar com o desenvolvimento da composição. Conforme mostra a Figura 25, a aplicação apresenta a lista de comentários de forma ordenada, onde a mensagem mais recente é a primeira da lista.

Figura 25 – Fórum de discussões



Através da opção “Comentar”, o usuário pode tecer os comentários desejados na tela apresentada na Figura 26 e incluí-los ao fórum para que os demais usuários possam contemplar suas colocações.

Figura 26 – Tela para inclusão de comentários



A tela para inclusão dos comentários possui um campo para a digitação da mensagem e a opção “Salvar”, utilizada para adicionar os comentários à lista.

3.4 RESULTADOS E DISCUSSÕES

Nesta seção é detalhada a metodologia de avaliação adotada, perfil dos usuários, aplicação do teste, análise e interpretação dos dados coletados e comparação entre as funcionalidades adicionadas ao aplicativo de composição musical e os trabalhos correlatos.

A seção 3.4.1 apresenta o experimento de usabilidade utilizado para avaliar a usabilidade do aplicativo pelo usuário final. Nela são descritos a metodologia empregada na elaboração das avaliações da aplicação, descreve a etapa de aplicação do teste relatando os resultados obtidos com base nas instruções apresentadas para os usuários e por fim, apresentando a análise e a interpretação dos resultados obtidos nos testes. Por fim, a seção 3.4.2 realiza o comparativo entre este trabalho e os trabalhos correlatos.

3.4.1 Experimento de usabilidade

O experimento de usabilidade contou com a participação de cinco usuários que aceitaram participar da avaliação da aplicação e responder ao questionário elaborado com o roteiro para avaliação das funcionalidades da aplicação.

Os usuários convidados apresentam perfis compatíveis com o público-alvo do aplicativo, sendo três deles totalmente leigos em música e os demais possuindo certo nível conhecimento. O cenário foi configurado neste formato, a fim de possibilitar também uma avaliação de uso entre usuários de níveis diferentes.

3.4.1.1 Metodologia

A avaliação foi realizada no mês de novembro, de forma individual e assistida, quando possível, para que além da avaliação quanto às respostas apresentadas no questionário, também houvesse uma avaliação quanto ao nível de dificuldade apresentado pelos avaliadores durante o uso da aplicação.

A aplicação foi testada a partir do *smartphone* utilizado durante o processo de implementação, sendo o mesmo cedido a cada usuário no momento dos testes, juntamente com o questionário de usabilidade (Apêndice A).

Segundo Nielsen (1993), para se obter resultados satisfatórios em testes de usabilidade, são necessários apenas cinco usuários, alegando que, a partir do quinto usuário, se desperdiça tempo em observar praticamente os mesmos problemas repetidamente, sem encontrar novos erros.

3.4.1.2 Aplicação do teste

Inicialmente cada participante recebeu uma explicação sobre o projeto, a aplicação implementada e alguns detalhes quanto ao objetivo da pesquisa de usabilidade. Também foi realizada uma explicação sobre como o questionário deveria ser utilizado durante os testes.

Os usuários receberam o questionário e inicialmente preencheram os dados quanto ao perfil pessoal. O questionário foi elaborado de forma que cada questão fosse executada dando continuidade à questão apresentada anteriormente, buscando assim, expor um roteiro onde o usuário pudesse apreciar todas as funcionalidades da ferramenta. A cada passo efetuado, o usuário indicava no questionário se conseguiu efetuar a ação solicitada, caso contrário, o mesmo deveria informar o motivo pelo qual o mesmo não conseguiu executar a ação proposta. Ao final do questionário disponibilizou-se também um espaço para sugestões e críticas.

Foi permitido aos usuários que os mesmos utilizassem a aplicação por alguns minutos antes de iniciar o questionário, mas os usuários optaram por iniciar o questionário juntamente com os testes da aplicação. O tempo de avaliação durou para cada usuário de dez a vinte minutos aproximadamente, onde usuários mais diretos executaram o roteiro e apresentaram suas respostas de forma mais rápida, enquanto outros buscaram utilizar a aplicação mais detalhadamente realizando um teste mais aprofundado.

3.4.1.3 Análise e interpretação dos dados coletados

Os dados dos testes foram analisados inicialmente com base nos dados relacionados ao perfil do usuário, conforme apresentado no Quadro 14 abaixo.

Quadro 14 – Resultados do questionário de perfil dos usuários

Sexo	60% masculino 40% feminino
Idade	60% entre 18 e 25 anos 40% entre 25 e 35 anos
Nível de escolaridade	20% ensino médio completo 60% ensino superior incompleto 20% ensino superior completo
Grau de familiaridade com a música	20% leigo em música 20% escutam música casualmente 40% músico amador 20% músico experiente

Após análise do perfil do usuário, realizou-se a interpretação dos resultados obtidos a partir do questionário de usabilidade. Estes resultados estão apresentados no Quadro 15 e foram obtidos a partir da análise do retorno quanto à execução do roteiro proposto aos usuários.

Quadro 15 – Resultados do questionário de usabilidade

Tarefa proposta		A tarefa foi executada?	
		Sim	Não
1	Acesse a aplicação e ouça algumas das composições publicadas.	100%	
2	Curta alguma(s) composições.	100%	
3	Abra uma das composições para que possa visualizar sua estrutura.	80%	20%
4	Execute a música que está sendo visualizada.	100%	
5	Realize o <i>login</i> na aplicação (caso ainda não tenha realizado o cadastro, realize-o neste momento).	100%	
6	Crie uma nova composição.	100%	
7	Salve a composição.	100%	
8	Convide outros usuários a colaborar em sua composição.	100%	
9	Realize a postagem de algum comentário no Fórum.	100%	
10	Publique a sua composição.	100%	
11	Solicite o convite para colaborar em uma composição, depois retorne a sua lista privada e verifique se a composição foi exibida.	100%	
12	Abra a composição que você foi convidado, selecionando uma das versões da mesma.	100%	
13	Altere a versão conforme desejado, salve e verifique se foi gerada uma nova versão.	100%	
14	Efetue o <i>logout</i> da aplicação.	100%	
15	Verifique e curta a sua composição na lista pública.	100%	

Analisando os resultados das ações propostas, observa-se que os usuários conseguiram efetuar todas as ações propostas, não considerando o grau de dificuldade que tiveram para realizá-las. Apenas um usuário necessitou de ajuda para realizar a abertura da composição para visualização (etapa “3”, apresentada no Quadro 15). Analisando cada um dos usuários realizando a execução do roteiro de forma assistida, foi possível notar que alguns usuários foram mais cautelosos quanto ao uso da aplicação, evitando testar toques em áreas da tela para buscar as funcionalidades, o que leva a conclusão que alguns usuários necessitam de indicações mais presentes das funcionalidades da tela. Exemplos que podem ser citados são que alguns usuários levaram certo tempo para encontrar a função para curtir a aplicação, ou levaram certo tempo para encontrar a função de visualização da consulta através do clique longo na lista de composições e ainda houve situações que os usuários levaram certo tempo para identificar se a visualização da lista de composições apresentada tratava-se da sua lista pública ou da lista privada.

Apesar de terem encontrado algumas dificuldades em determinados pontos no uso da aplicação, apenas um usuário registrou uma crítica no questionário, onde mencionou que os menus possuem pouco destaque, pois se mostram pequenos demais, dificultando a localização de algumas funções. Outro detalhe colocado pelo usuário foi quanto aos botões das telas, pois

alguns que se apresentam com cor similar ao restante da tela ficaram sem destaque. Apenas um usuário realizou o registro de sugestão, onde mencionou que seria interessante se a aplicação disponibilizasse um conjunto mais variado de notas musicais.

Com os resultados obtidos nos testes realizados, foi possível concluir que apesar de alguns usuários terem tido problemas para encontrar algumas funções da aplicação, todos conseguiram realizar a maior parte das ações propostas no roteiro de avaliação.

3.4.2 Comparação com trabalhos correlatos

Através da análise dos resultados obtidos, reformulou-se o quadro de características dos trabalhos correlatos (Quadro 1 da seção 2.5.5), incluindo o aplicativo de composição musical para leigos *MyMusic*, para fins de comparação. O Quadro 16 apresenta a análise comparativa realizada entre as características do aplicativo desenvolvido com as características dos trabalhos correlatos apresentados.

Quadro 16 - Comparação com os trabalhos correlatos

características / trabalhos correlatos	Google Play (2015)	Bluemoon Interactive (2014)	Netigen (2014)	Avid Technology (2015)	<i>MyMusic</i>
usuário-alvo	Leigos	Leigos/Avançados	Leigos	Avançados	Leigos
plataforma	<i>mobile</i>	<i>desktop</i>	<i>mobile</i>	<i>mobile/desktop</i>	<i>Mobile</i>
sistema operacional	Android	Windows	Android	Windows/MAC OS/RISC OS	Android
permite edição nota-à-nota	Sim	Sim	Não	Sim	Não
oferece escala cromática	Não	Sim	Não	Sim	Não
permite uso de múltiplos instrumentos	Sim	Sim	Sim	Sim	Sim
permite salvar as músicas criadas	Sim	Sim	Sim	Apenas na versão paga	Sim
permite exportar as composições	Sim	Sim	Sim	Sim	Não
oferece funcionalidade de compartilhamento	Sim	Não	Não	Sim	Não
criação colaborativa	Não	Não	Não	Não	Sim
forma de execução	Loop	Opcional: Loop ou início e fim	loop	início e fim	início e fim

Conforme o Quadro 16 pode-se observar que o aplicativo proposto contempla grande parte das características oferecidas nos demais trabalhos correlatos (vide seção 2.5.5), visando como diferencial, apresentar um ambiente simples, voltado a usuários leigos em música através do uso da plataforma Android.

3.4.3 Alterações realizadas na aplicação

O aplicativo *MyMusic* sofreu diversas alterações nesta versão, quando comparado à versão anterior, principalmente no que se refere à inclusão de novas telas. Algumas das principais evoluções realizadas da versão anterior para a versão desenvolvida no trabalho podem ser observadas no Quadro 17.

Quadro 17 – Comparando a evolução do trabalho

funções/evolução do trabalho	Weitgenant (2014)	Trabalho Proposto
montar composição	X	X
tocar composição	X	X
salvar composição no dispositivo	X	
enviar composição para o servidor	X	X
baixar composição do servidor	X	X
controlar versão das evoluções das composições	X	X
possibilitar ao usuário o resgate das versões		X
apresentar a lista pública ao usuário, contendo composições publicadas pelo usuário		X
apresentar a lista privada com as composições do usuário logado		X
permitir que o usuário selecione convidados para colaborar na sua composição		X
disponibilizar aos usuários uma lista de discussão		X

Nesta versão, o Editor deixa de ser a tela inicial do aplicativo, que passa a apresentar a lista de composições como tela inicial, que por sua vez, oferece os acessos às telas de *login* e visualização das composições quando o usuário não está logado e após o *login*, oferece acesso às telas de versões, editor e *chat*.

O editor recebeu novas funcionalidades, podendo agora exibir determinada composição em modo de visualização para o usuário, sem que sejam permitidas alterações, além da inclusão de acessos, quando em modo de edição, para as telas de convite e lista de discussão, além de funcionalidades como a de publicação.

As funcionalidades relacionadas ao *chat* não sofreram alterações, apenas foi disponibilizado um acesso ao mesmo através da lista inicial, além do acesso que já era disponibilizado no menu do editor, mantido também nesta versão.

As funcionalidades de sincronismo disponíveis na versão anterior e as demais funcionalidades de *download* e *upload*, oferecidas de forma explícita, deixaram de existir. A aplicação agora mantém as composições em um servidor de forma sincronizada, onde toda ação de salvar é realizada através de *upload*, enviando o arquivo ao servidor, e toda função de leitura é realizada através de *download*, onde a leitura é feita diretamente do servidor, tornando assim os arquivos disponíveis a todos os usuários sem que haja a necessidade de sincronia de forma manual.

3.4.4 Aplicando requisitos do CSCW

Os requisitos do CSCW se mostram presentes na aplicação dissolvidos nos requisitos implementados. Questões quanto à identidade e controle de acesso são abordadas através do cadastro e *login* do usuário na aplicação, onde o usuário passa a estar identificado e assim podendo integrar grupos de compositores de uma ou mais composições.

Os requisitos implementados relacionados ao contexto público e privado, apresentados aos usuários pela aplicação, definem não somente as questões relacionadas às visões distintas para cada caso, como também são a primeira abordagem a remeter à utilização de conceitos de visões convergentes, onde apesar de os usuários serem diferentes é possível contar com uma área de visão comum, onde podem compartilhar informações específicas ao entorno da composição através de uma lista de discussão.

Por fim, o contexto colaborativo adotado na implementação evolutiva do trabalho, remeteu à necessidade do emprego de técnicas responsáveis por tratar da situação de sincronismo, onde a atualização das informações passou a ser realizada de forma *online*, devido ao fato de que cada ação efetuada por um usuário do grupo deve ser refletida e disponibilizada aos demais usuários.

4 CONCLUSÕES

Este trabalho abordou a criação de funcionalidades em um ambiente colaborativo, onde leigos consigam criar e compartilhar composições, dando continuidade ao trabalho desenvolvido por Weitgenant (2014) e adicionando funcionalidades como ambientes distintos de listas de composição privada e pública, lista de versões e versionamento das composições, inclusão de usuários como editores colaborativos na criação de determinada composição e fórum de discussão para registro de mensagens entre os usuários.

O resultado obtido através dos testes com os usuários mostrou que a aplicação necessita de alguns ajustes para auxiliar ainda mais o usuário no seu processo de criação. Todavia, a aplicação pôde ser utilizada de forma satisfatória para atender aos requisitos propostos, de forma a permitir que os usuários conseguissem através do ambiente proporcionado, criar e compartilhar suas composições, resgatar versões e realizar o registro de alterações através do fórum.

Em relação aos objetivos específicos apresentados, foi possível melhorar o cenário colaborativo da aplicação, apresentando para os usuários ambientes diferenciados onde no ambiente público é possível contemplar composições publicadas pelos usuários e um ambiente privado, onde o usuário vai poder trabalhar de forma privada e colaborativa para o desenvolvimento de suas composições. Alguns pontos apresentados pelos usuários tratam principalmente de detalhes relacionados à apresentação das funcionalidades da tela, onde em alguns casos não ocorre de forma muito clara, principalmente menus e botões.

Dentre as limitações da aplicação, destacam-se pontos como o tempo de resposta para as transações entre a aplicação e o servidor, devido ao fato da persistência dos dados ser realizada por arquivos, o que demanda *downloads* e *uploads* constantes por parte da aplicação. Esta característica também traz algumas dificuldades para a implementação de controles mais refinados a algumas opções da tela. Entre os desafios encontrados na implementação das novas funcionalidades, pode-se apontar inicialmente a falta de conhecimento relacionado ao desenvolvimento Android, o que demandou a busca de conteúdo teórico para melhor compreensão da tecnologia. Outro desafio encontrado foi quanto ao uso da API *QuickBlox*, que é oriunda da versão anterior, e devido a extrema deficiência na própria documentação, alinhado ao pouco conhecimento da mesma, contribuiu para o surgimento de erros que demandaram muito tempo para serem solucionados, mesmo sendo problemas causados por detalhes simples onde uma documentação mais completa e/ou detalhada poderia sanar o problema com maior brevidade.

Dentre as vantagens que a aplicação oferece, uma delas é ter adotado como ambiente a plataforma Android, o que torna a aplicação acessível a uma quantidade muito grande de pessoas, visto a popularização da plataforma e a facilidade de acesso a mesma. Outro detalhe a ser apontado, é que o aplicativo mostrou-se divertido e em conjunto com as funcionalidades de colaboração, propõe uma experiência agradável ao usuário, instigando no mesmo o interesse pela composição musical.

4.1 EXTENSÕES

Como possibilidades de extensões para este trabalho, sugerem-se:

- a) realizar alterações do *layout* buscando destacar as funções oferecidas pela aplicação por meio de menu e botões;
- b) alterar a representação musical para que esta apresente um ambiente menos textual, de forma a utilizar símbolos e imagens para a representação musical;
- c) disponibilizar uma gama maior de *samples* para o usuário, em especial quanto a quantidade de notas musicais de cada instrumento;
- d) alterar a criação de composições por *samples* pré-fixados para um conceito de aplicação de notas musicais de uma escala musical cromática sobre uma *timeline* que trabalhe seu compasso por batidas por minuto (BPM) e não mais pela duração dos *samples*;
- e) aprimorar a função curtir, de forma que ela identifique usuário que curtiu a composição, permitindo também ao usuário divulgar a composição através das redes sociais;
- f) oferecer um meio de exportar o áudio criado, para algum formato de áudio conhecido;
- g) alterar a forma que a aplicação realiza a persistência de dados, retirando as informações de arquivos e utilizando banco de dados para tal tarefa, buscando assim além de uma melhora na performance da aplicação, reduzir o tráfego de dados desnecessários na rede.

REFERÊNCIAS

- AGNER, Luiz. **Ergodesign e arquitetura de informação: trabalhando com o usuário**. Rio de Janeiro: Quartet, 2006.
- ALVARENGA, Gustavo G. **Ferramenta para criação de composições musicais para Android**. 2013. 58 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <http://www.bc.furb.br/docs/MO/2013/353880_1_1.pdf>. Acesso em: 29 mar. 2015.
- AVID TECHNOLOGY. **Sibelius**. [S.l.], [2015]. Disponível em: <<http://www.avid.com/US/products/sibelius#overview>>. Acesso em: 21 mai. 2015.
- AZEVEDO, Fernando. **Como Compor Músicas Facilmente**. Rio de Janeiro: Bruno Quaino Editores, 1982.
- BAIXAKI. **Sibelius download – Baixaki**. [S.l.], [2013]. Disponível em: <<http://www.baixaki.com.br/download/sibelius.htm>>. Acesso em: 21 mai. 2015.
- BAIXAKI. **Sound Club download – Baixaki**. [S.l.], [2008]. Disponível em: <<http://www.baixaki.com.br/download/sound-club.htm>>. Acesso em: 30 mar. 2015.
- BLUEMOON INTERACTIVE. **Sound Club**. [S.l.], 2014. Disponível em: <<http://www.bluemoon.ee/history/>>. Acesso em: 29 mar. 2015.
- BOTELHO, Eulina X., VIDAL, Jorgiano M. B. **CSCW: Trabalho cooperativo suportado por computador**. [S.l.], 2005. Disponível em: <http://www.academia.edu/6886676/Resumo_-_CSCW>. Acesso em: 30 mar. 2015.
- CAIADO, Elen Campos. **A importância da música no processo de ensino-aprendizagem - Educador Brasil Escola**. [S.l.], [2015]. Disponível em: <<http://educador.brasilecola.com/sugestoes-pais-professores/a-importancia-musica-no-processo-ensinoaprendizagem.htm>>. Acesso em: 30 mar. 2015.
- CASTILHO, Marcelo. **O que é CSCW?** [S.l.], 2008. Disponível em: <<http://www.dimensaotech.com/2008/11/o-que-e-cscw/>>. Acesso em: 30 mar. 2015.
- CYBIS, Walter A.; BETIOL, Adriana H.; FAUST, Richard. **Ergonomia e Usabilidade – Conhecimentos, Métodos e Aplicações**. 2. ed. São Paulo: Editora Novatec, 2010.
- FARIA, Márcia Nunes. **A música, fator importante na aprendizagem**. 2001. 40f. Monografia (Especialização em Psicopedagogia) – Centro Técnico-Educacional Superior do Oeste Paranaense, Paraná.
- FLUCKIGER, François. **Understanding networked multimedia**. Indiana, U.S.A.: Prentice Hall, 1995.
- GAIGHER, Jobert M. **A música consumida nos dias atuais, está cada vez mais fora dos processos convencionais de produção musical**. [S.l.], 2012. Disponível em: <<http://gaigher.wordpress.com/2012/04/25/a-musica-consumida-nos-dias-atuais-esta-cada-vez-mais-fora-dos-processos-convencionais-de-producao-musical/>>. Acesso em: 30 mar. 2015.
- GOOGLE PLAY. **3-Minute Composition musicLine - Apps para Android no Google Play**. [S.l.], 2015. Disponível em: <<https://play.google.com/store/apps/details?id=jp.gr.java.conf.createapps.musicline>>. Acesso em: 29 mar. 2015.

- MARÇAL, Wanessa. **Música**. [S.l.], 2011. Disponível em: <<http://nuvendigital.com/blog/2011/06/27/musica-2/>>. Acesso em: 29 mar. 2015.
- MARQUES, Keila. **Sibelius 7.5 facilita composição e notação musical**. [S.l.], jan. 2014. Disponível em: <<http://www.panoramaaudiovisual.com.br/2014-01-sibelius-75-facilita-composicao-e-notacao-musical-9903>>. Acesso em: 21 mai. 2015.
- MARTIN, George. **Fazendo Música: o guia para compor, tocar e gravar**. Brasília: Editora Universidade de Brasília, São Paulo: Imprensa Oficial do Estado, 2002.
- MED, Bohumil. **Teoria da música**. 4.ed. Brasília: Musimed, 1996.
- NETIGEN. **Best Composer Pro – NETIGEN**, 2014. Disponível em: <<http://netigen.pl/best-composer-pro>>. Acesso em: 30 mar. 2015.
- NIELSEN, Jakob. **Usability Engineering**, San Francisco, USA: Morgan Kaufmann Publishing, 1993.
- PAPADOPOULOS, Constantinos. **Improving Awareness in Mobile CSCW**. Manchester, [2005]. Disponível em: <<http://scgroup.hpclab.ceid.upatras.gr/scgroup/Variou/IEEE-TMC-Manuscript.pdf>>. Acesso em: 19 mai. 2015.
- ROCHA, H. V. da et al. **Design e avaliação de interfaces humano-computador**. Campinas: Unicamp, 2003.
- SANTOS, Paloma M. et al. **A importância da utilização de abordagens centrada no usuário para o domínio da TV Digital**. [S.l.], ago. 2012. Disponível em: <http://datagramazero.org.br/ago12/Art_05.htm>. Acesso em: 18 mai. 2015.
- SCHNEIDERMAN, Ben. **Designing the User Interface: Strategies for Effective Human-Computer Interaction**. Addison-Wesley, 1992.
- SEVERO, Felipe. **Uma notícia que é música para os ouvidos**. Bagé, dez. 2013. Disponível em: <<http://www.jornalminuano.com.br/VisualizarNoticia/4909/uma-noticia-que-e-musica-para-os-ouvidos-.aspx>>. Acesso em: 14 mai. 2015.
- SILVA, Marcus V. O. **A metacomunicação em interfaces digitais de aplicativos educacionais: repensando a experiência do aprendizado**. 2014. 28 f. Projeto de pesquisa (Mestrado em Sistemas de Informação e Gestão do Conteúdo) - Universidade FUMEC, Belo Horizonte. Disponível em: <<http://www.fumec.br/revistas/sigc/article/download/2534/1514>>. Acesso em: 25 mai. 2015.
- SOUZA, Cleidson R. B. **GROUPWARE & CSCW: Conceitos básicos**. Pará, abr. 2006. Disponível em: <<http://pt.scribd.com/doc/218610021/Groupware-e-Cscw#scribd>>. Acesso em: 30 mar. 2015.
- SWANWICK, Keith; FRANÇA, Cecília C. Composição, apreciação e performance na educação musical: teoria, pesquisa e prática. **EM PAUTA**, Porto Alegre: v.13, n.21, p. 5-41, dez. 2002. Disponível em: <<http://seer.ufrgs.br/index.php/EmPauta/article/download/8526/4948>>. Acesso em: 30 mar. 2015.
- WEITGENANT, Fabrício. **My Music: Aplicativo de composição musical para leigos**. 2014. 75 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- YANG, Baijian; ZHENG, Pei; NI, Lionel M. **Professional Microsoft Smartphone Programming**. Indianapolis: Editora John Wiley & Sons, 2007.

APÊNDICE A – Detalhamento dos casos de uso

Este apêndice detalha os casos de uso especificados para a aplicação de desenvolvimento de composições para leigos, destacando os roteiros de cada caso de uso, com seus respectivos cenários.

Os casos de uso são apresentados nos quadros abaixo, onde o Quadro 18 apresenta o caso de uso UC01 - Listar composições, o Quadro 19 apresenta o caso de uso UC02 - Baixar composições, o Quadro 20 apresenta o caso de uso UC03 - Fazer Login/Logout, o Quadro 21 apresenta o caso de uso UC04 - Cadastrar usuário, o Quadro 22 apresenta o caso de uso UC05 - Registrar Curtir, o Quadro 23 apresenta o caso de uso UC06 - Editar composição, o Quadro 24 apresenta o caso de uso UC07 - Salvar composições, o Quadro 25 apresenta o caso de uso UC08 - Gerar versão, o Quadro 26 apresenta o caso de uso UC09 - Enviar composição, o Quadro 27 apresenta o caso de uso UC10 - Convidar usuários, o Quadro 28 apresenta o caso de uso UC11 - Manter Fórum e por fim o Quadro 29 apresentando o caso de uso UC12 - Publicar composição.

Quadro 18 - UC01 - Listar composições

Número	01
Caso de Uso	Listar composições
Descrição	Define a listagem das composições publicadas ou do próprio usuário.
Ator	Usuário
Pré-condições	Aplicativo já previamente instalado no dispositivo e conexão de internet ativa.
Pós-condições	Listar as composições disponíveis para contemplação dos demais usuários quando o Usuário não estiver logado ou listar as composições relacionadas ao próprio Usuário quando o mesmo houver efetuado o <i>login</i> .
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário abre o aplicativo no dispositivo móvel; 2. O aplicativo exibe a interface principal; 3. O aplicativo inicia a leitura das composições; 4. As composições publicadas são listadas na tela do dispositivo; 5. O Usuário realiza o <i>login</i>; 6. Passo 3 é repetido pelo aplicativo; 7. As composições do Usuário são listadas na tela do dispositivo; 8. O Usuário realiza o <i>logoff</i>; 9. Passos 3 e 4 são repetidos; 10. Usuário sai o aplicativo.
Cenário alternativo	No passo 3, caso não haja uma conexão ativa com a internet, as composições não poderão ser lidas do servidor.

Quadro 19 - UC02 - Baixar composições

Número	02
Caso de Uso	Baixar composições
Descrição	Define o caso de uso da funcionalidade de <i>download</i> das composições.
Ator	Usuário
Pré-condições	Aplicativo apresenta a interface e inicia a listagem das composições
Pós-condições	Retornar todos os arquivos gravados no servidor
Cenário principal	<ol style="list-style-type: none"> 1. O aplicativo inicia a busca das composições a serem exibidas na lista; 2. O aplicativo conecta no servidor e realiza o download das composições; 3. As composições baixadas são apresentadas para o Usuário conforme o caso de uso UC01.
Cenário alternativo	No passo 2, caso não haja uma conexão ativa com a internet, as composições não poderão ser baixadas do servidor.

Quadro 20 - UC03 - Fazer Login/Logout

Número	03
Caso de Uso	Fazer Login/Logout
Descrição	Define as ações de <i>login</i> e <i>logout</i> realizadas pelo usuário.
Ator	Usuário
Pré-condições	Inicializar o aplicativo e dispor de conexão de internet ativa.
Pós-condições	Registrar ou encerrar a sessão do usuário que conecta a aplicação ao servidor.
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário seleciona a opção Login no menu do aplicativo; 2. A tela para entrada dos dados de <i>login</i> é exibida ao Usuário; 3. O usuário informa seu <i>login</i> e sua senha e seleciona a opção Login; 4. A aplicação envia as informações ao servidor para validação; 5. A aplicação cria a sessão para o Usuário; 6. O Usuário seleciona a opção Logout no menu do aplicativo; 7. A aplicação encerra a sessão do Usuário.
Cenário alternativo	<p>Caso não haja a validação dos dados no passo 4 o Usuário deverá revisar e informar novamente os dados do <i>login</i>.</p> <p>Ao chegar ao passo 3, caso o Usuário ainda não possua cadastro, o mesmo deverá selecionar a opção “Cadastre-se”, iniciando assim o caso de uso UC04.</p>

Quadro 21 - UC04 - Cadastrar usuário

Número	04
Caso de Uso	Cadastrar usuário
Descrição	Viabiliza o cadastro de novos usuários.
Ator	Usuário
Pré-condições	Inicializar o aplicativo, dispor de conexão de internet ativa e possuir um endereço de e-mail não cadastrado anteriormente.
Pós-condições	Usuário cadastrado e registrado.
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário seleciona a opção <i>Cadastre-se</i> na tela de Login; 2. A aplicação exibe a tela com os campos do cadastro; 3. O Usuário informa o seu <i>login</i>, senha e email; 4. O Usuário seleciona a opção <i>Cadastrar</i>; 5. O aplicativo valida os campos do cadastro; 6. O aplicativo cadastra e registra o Usuário no servidor.
Cenário alternativo	<p>Caso não conexão com a internet no momento no passo 4, o aplicativo exibe um alerta e retorna ao passo 2.</p> <p>Caso no passo 5 alguma informação não for aceita retorna ao passo 2.</p>

Quadro 22 - UC05 - Registrar Curtir

Número	05
Caso de Uso	Registrar Curtir
Descrição	Possibilita que usuários indique que gostou de uma composição.
Ator	Usuário
Pré-condições	O Usuário esteja visualizando a lista de composições publicadas
Pós-condições	Contador de curtidas é incrementado.
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário seleciona a opção <i>Like</i> através do clique da imagem da lista de composições; 2. A aplicação incrementa o contador de curtidas; 3. A aplicação envia a composição para o servidor conforme o caso de uso UC09.

Quadro 23 - UC06 - Editar composição

Número	06
Caso de Uso	Editar composição
Descrição	Permite a criação e alterações das composições.
Ator	Usuário
Pré-condições	Possuir o aplicativo de composição musical instalado no dispositivo móvel e conexão de internet ativa.
Pós-condições	Composição editada.
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário acessa o editor realizando a abertura de uma composição ou selecionando a opção de nova composição no menu do aplicativo ; 2. A aplicação exibe a estrutura para edição e visualização de composições; 3. Se o Usuário estiver logado a lista de <i>samples</i> é apresentada; 4. O Usuário realiza a disposição de <i>samples</i> na <i>timeline</i> de edição através da funcionalidade de clicar e arrastar; 5. O Usuário salva a composição através do menu conforme caso de uso UC07; 6. O Usuário tem a opção de convidar outros usuários para colaborarem com a composição através do caso de uso UC10; 7. O Usuário tem a opção de realizar a inclusão de comentários no fórum através do caso de uso UC11; 8. O Usuário tem a opção de publicar a composição através do caso de uso UC12.
Cenário alternativo	<p>Caso no passo 3 o usuário não tenha realizado o <i>login</i> na aplicação, a lista de <i>samples</i> não é apresentada e o usuário pode apenas visualizar a estrutura da composição.</p> <p>Caso o Usuário finalize a tela do editor antes de efetuar o passo 5, a composição não será salva.</p>

Quadro 24 - UC07 - Salvar composições

Número	07
Caso de Uso	Salvar composições
Descrição	Realiza a gravação da composição.
Ator	Usuário
Pré-condições	Realizar a criação de uma composição que contenha ao menos um <i>sample</i> .
Pós-condições	Composição salva no servidor.
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário seleciona a opção salvar; 2. A aplicação solicita o nome da mesma ao Usuário; 3. A aplicação grava os dados da <i>timeline</i>; 4. O arquivo é enviado ao servidor conforme o caso de uso UC09.
Cenário alternativo	<p>Caso no passo 1 não seja encontrado nenhum <i>sample</i> na composição é exibida uma mensagem ao Usuário informando que não há composição para salvar.</p> <p>Caso seja uma aplicação já salva, o passo 2 é ignorado.</p>

Quadro 25 - UC08 - Gerar versão

Número	08
Caso de Uso	Gerar versão
Descrição	Realiza o controle de geração de versões
Ator	Usuário
Pré-condições	Há uma composição para ser salva.
Pós-condições	Composição é salva no servidor com a versão correta
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário seleciona a opção salvar; 2. A aplicação verifica se deve gerar nova versão ou atualizar a atual; 3. A versão do arquivo é salva.

Quadro 26 - UC09 - Enviar composição

Número	09
Caso de Uso	Enviar composição
Descrição	Realiza o envio do arquivo da composição para o servidor
Ator	Usuário
Pré-condições	A composição está salva.
Pós-condições	Arquivo é atualizado no servidor.
Cenário principal	<ol style="list-style-type: none"> 1. A aplicação realiza o <i>upload</i> do arquivo; 2. A versão do arquivo é gravada no servidor.

Quadro 27 - UC10 - Convidar usuários

Número	10
Caso de Uso	Convidar usuários
Descrição	Permitir a seleção de usuários para coparticipar da criação da composição
Ator	Usuário
Pré-condições	A composição está salva.
Pós-condições	Arquivo é atualizado no servidor e ocorre o envio de e-mail aos convidados.
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário acessa a opção Convite no editor; 2. A aplicação apresenta uma tela onde são listados os usuários disponíveis para envio de convite; 3. O Usuário seleciona os participantes e seleciona Enviar convite; 4. A aplicação atribui os usuários à composição e solicita o <i>upload</i> da mesma; 5. A aplicação envia os e-mails aos convidados.
Cenário alternativo	No passo 3, caso o Usuário queira convidar pessoas que não sejam usuárias da aplicação, o mesmo pode selecionar a opção Adicionar, onde a aplicação apresenta uma tela para que seja informado o e-mail da pessoa a ser convidada. Após adicionar o e-mail da pessoa, a aplicação inclui o e-mail da pessoa a lista de usuários disponíveis para convite e retorna ao passo 2.

Quadro 28 - UC11 - Manter Fórum

Número	11
Caso de Uso	Manter Fórum
Descrição	Possibilita o acesso a uma lista de discussão da composição
Ator	Usuário
Pré-condições	A composição está salva.
Pós-condições	Cada novo comentário inserido no fórum é gravado na composição
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário acessa a opção Comentar no editor; 2. A aplicação apresenta uma tela onde são listados os comentários; 3. O Usuário seleciona a opção comentar; 4. A aplicação apresenta a tela para que seja informado o comentário; 5. O Usuário salva seu comentário; 6. A aplicação grava o comentário na composição; 7. O arquivo é enviado ao servidor conforme o caso de uso UC09; 8. Aplicação retorna ao passo 2.

Quadro 29 - UC12 - Publicar composição

Número	12
Caso de Uso	Publicar composição
Descrição	Possibilita que a composição seja publicada para os demais usuários
Ator	Usuário
Pré-condições	A composição está salva.
Pós-condições	A publicação poderá ser vista por todos os usuários na lista principal.
Cenário principal	<ol style="list-style-type: none"> 1. O Usuário acessa a opção Publicar através do menu secundário do Editor; 2. A aplicação define a composição como publicada e realiza a atualização no servidor através do caso de uso UC09; 3. A composição passa a ser listada para todos os usuários na lista principal.
Cenário alternativo	Caso a opção publicar esteja inativa no passo 1, significa que a composição já foi publicada.

APÊNDICE B – Questionário de usabilidade

Este apêndice apresenta o questionário destinado à avaliação por parte dos usuários avaliadores do aplicativo. O questionário possui algumas instruções quanto ao uso do aplicativo e apresenta um roteiro para que os usuários possam realizar a avaliação das funcionalidades da aplicação.

O questionário apresenta inicialmente alguns questionamentos destinados a traçar o perfil do usuário, conforme pode ser visto no Quadro 30. No Quadro 31 contemplam-se algumas instruções para a realização dos testes e, por fim, no Quadro 32 estão descritos os passos a serem executados pelo usuário, de forma que o mesmo possa indicar se alcançou o objetivo, ou em caso negativo, descreva o motivo. Também há uma área destinada a observações caso o usuário queira sugerir algo.

Quadro 30 – Questionário de perfil do usuário

<h1>QUESTIONÁRIO</h1> <h2>Perfil do Usuário</h2> <p style="text-align: center;"><i>* Observação: todos os dados informadores serão mantidos de forma confidencial.</i></p> <p>Sexo:</p> <p><input type="checkbox"/> Masculino</p> <p><input type="checkbox"/> Feminino</p> <p>Idade:</p> <p><input type="checkbox"/> Tenho menos de 18 anos</p> <p><input type="checkbox"/> Tenho entre 18 e 25 anos</p> <p><input type="checkbox"/> Tenho entre 25 e 35 anos</p> <p><input type="checkbox"/> Tenho mais de 35 anos</p> <p>Nível de escolaridade:</p> <p><input type="checkbox"/> Ensino fundamental incompleto</p> <p><input type="checkbox"/> Ensino fundamental completo - 1º grau</p> <p><input type="checkbox"/> Ensino médio incompleto</p> <p><input type="checkbox"/> Ensino médio completo - 2º grau</p> <p><input type="checkbox"/> Ensino superior incompleto</p> <p><input type="checkbox"/> Ensino superior completo</p> <p>Indique seu grau de familiaridade com música:</p> <p><input type="checkbox"/> Sou leigo em música</p> <p><input type="checkbox"/> Escuto música casualmente</p> <p><input type="checkbox"/> Sou apreciador de música, consigo identificar ritmos e instrumentos</p> <p><input type="checkbox"/> Sou músico amador, sei tocar um instrumento</p> <p><input type="checkbox"/> Sou músico experiente, sei tocar um ou mais instrumentos e compor</p>
--

Quadro 31 – Instruções iniciais

INSTRUÇÕES:

O objetivo desta avaliação é validar os novos requisitos agregados ao aplicativo de composição musical buscando avaliar a usabilidade oferecida pelo formato adotado e analisando a facilidade de entendimento e operação do mesmo. Ao final da avaliação, busca-se obter o retorno das impressões do usuário, para que assim o aplicativo possa ser melhorado e adequado às necessidades dos usuários.

Utilize o aplicativo livremente por algum tempo e quando se sentir confortável, realize as ações listadas a seguir.

Quadro 32 – Roteiro para avaliação

<p>1. Acesse a aplicação e ouça algumas das composições publicadas. A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>2. Curta alguma(s) composições. A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>3. Abra uma das composições para que possa visualizar sua estrutura. A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>4. Execute a musica que está sendo visualizada. A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>5. Realize o login na aplicação (caso ainda não tenha realizado o cadastro, realize-o neste momento). A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>6. Crie uma nova composição. A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>7. Salve a composição. A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>8. Convide outros usuários a colaborar em sua composição. A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>9. Realize a postagem de algum comentário no Fórum. A tarefa foi executada? () Sim () Não Motivo: _____</p>
<p>10. Publique a sua composição. A tarefa foi executada? () Sim () Não Motivo: _____</p>

11. Solicite a mim, ou outro usuário, para que o convide para colaborar em uma composição e depois retorne a sua lista privada e verifique se a composição foi exibida na lista.

A tarefa foi executada?

() Sim () Não Motivo: _____

12. Abra a composição que você foi convidado, selecionando uma das versões da mesma.

A tarefa foi executada?

() Sim () Não Motivo: _____

13. Altere a versão conforme desejado, salve e verifique se foi gerado uma nova versão da mesma.

A tarefa foi executada?

() Sim () Não Motivo: _____

14. Efetue o logout da aplicação.

A tarefa foi executada?

() Sim () Não Motivo: _____

15. Verifique e curta sua composição na lista pública.

A tarefa foi executada?

() Sim () Não Motivo: _____

Caso você deseje registrar alguma observação quanto ao aplicativo (dificuldades, sugestões, etc.), utilize as linhas abaixo:

Obrigado por sua participação!