

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

BALAIO SONORO: UM SISTEMA DE RECOMENDAÇÃO DE
MÚSICA

PATRÍCIA GONSALVES

BLUMENAU
2015

2015/2-13

PATRÍCIA GONSALVES

**BALAIO SONORO: UM SISTEMA DE RECOMENDAÇÃO DE
MÚSICA**

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

Prof. Aurélio Faustino Hoppe - Orientador

**BLUMENAU
2015**

2015/2-13

BALAIO SONORO: UM SISTEMA DE RECOMENDAÇÃO DE MÚSICA

Por

PATRÍCIA GONSALVES

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, Orientador - FURB

Membro: _____
Prof. Alexander Roberto Valdameri, Mestre – FURB

Membro: _____
Prof. Miguel Alexandre Wisintainer, Mestre – FURB

Blumenau, 10 de dezembro de 2015.

Dedico este trabalho à minha família e meus amigos por sempre estarem ao meu lado me apoiando e me motivando.

AGRADECIMENTOS

A Deus por todas as oportunidades em minha vida.

À minha família pelo amor, valores ensinados e compartilhados que me tornaram a pessoa que sou hoje. Agradeço também por sempre me apoiar e me incentivar nos estudos.

Aos meus amigos pelo suporte nessa difícil jornada e pelo apoio nos momentos de desespero.

Ao meu orientador, Aurélio Faustino Hoppe por toda a paciência, compreensão e acompanhamento durante o desenvolvimento desse trabalho. Sendo sempre prestativo e disposto a ajudar, repassando seu conhecimento e sugestões.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

A lot of times, people don't know what they want until you show it to them.

Steve Jobs

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação para recomendação de música. A aplicação web *responsive* desenvolvida proporciona aos usuários uma nova opção de rede social voltada à música, onde a interação entre os usuários se faz presente. A possibilidade de descobrir, compartilhar e interagir com novos artistas e usuários é o grande objetivo da aplicação. Para efetuar as recomendações, utiliza-se a técnica de filtragem híbrida, tendo como base as ações realizadas pelo usuário dentro da aplicação. A partir de testes de usabilidade foi possível perceber que a aplicação é boa e de fácil entendimento.

Palavras-chave: Sistemas de recomendação. Música. Rede social. MusicBrainz. EchoNest.

ABSTRACT

This work presents the development of an application for music recommendation. The developed responsive web application gives users a new option for social network focused on music, where interaction between users is present. The ability to find out, share and interact with new artists and users is the great object of application. To make recommendations, it uses the hybrid filtering technique, based on the actions performed by the user within the application. From usability testing it was revealed that the application is good and easy to understanding.

Key-words: Recommender systems. MusicBrainz. EchoNest. Music. Networking.

LISTA DE FIGURAS

Figura 1 - Processo de recomendação	16
Figura 2 - Matriz A de avaliação dos usuários. Cada célula corresponde à avaliação do usuário u para o item i . A tarefa é estimar o valor da avaliação $A_{a,i}$ para o usuário a .	18
Figura 3 - Filtragem Híbrida.....	22
Figura 4 - Tela principal do Last.fm.....	23
Figura 5 - Tela principal do ActivUFRJ	24
Figura 6 - Tela de eventos recomendados	25
Figura 7 - Diagrama de Caso de Uso.....	29
Figura 8 - Fluxo geral da aplicação	31
Figura 9 - Tela inicial da aplicação	33
Figura 10 - Criar conta.....	33
Figura 11 - Tela principal da aplicação	34
Figura 12 - Alterar perfil	34
Figura 13 - Buscar artista.....	35
Figura 14 - Convidar amigo	36
Figura 15 - Listar favoritos.....	37
Figura 16 - Convidar amigo	37
Figura 17 - Meus Amigos.....	38
Figura 18 - Notificações	38
Figura 19 - Nova solicitação de amizade.....	38
Figura 20 - Lista de notificações de sugestão.....	39
Figura 21 - Avaliar sugestão.....	39
Figura 22 - Recomendações iniciais.....	42
Figura 23 - Recomendações após alteração da pontuação	43
Figura 24 - Modelo de Entidade de Relacionamento	53

LISTA DE QUADROS

Quadro 1 - Comparativo entre os trabalhos correlatos	26
Quadro 2 - Trecho de código para pontuar pesquisa	35
Quadro 3 - Trecho e código para salvar favorito	35
Quadro 4 - Trecho de código pontuar avaliação.....	36
Quadro 5 - Trecho de código para sugerir artista	36
Quadro 6 - Trecho de código para convidar amigo	37
Quadro 7 - Trecho de código para pontuar avaliação.....	39
Quadro 8 - Trecho de código da pontuação.....	40
Quadro 9 - Agrupa estilo	40
Quadro 10 - Trecho de código para selecionar estilos.....	40
Quadro 11 - Trecho de código estilo musical do usuário	40
Quadro 12 - Trecho de código utilizando API EchoNest	41
Quadro 13 - Artistas e Estilos musicais.....	41
Quadro 14 - Artista e Pontuação.....	41
Quadro 15 - Estilo Musical e sua Pontuação	42
Quadro 16 - Pontuação do Artista	43
Quadro 17 - Estilo e Pontuação	43
Quadro 18 - Perfil dos usuários envolvidos no teste de usabilidade	45
Quadro 19 - Usuários envolvidos no teste de tarefas recomendação e sugestão.....	45
Quadro 20 - Usuários envolvidos no teste de tarefas de convites	46
Quadro 21 - Respostas quanto à usabilidade do protótipo	47
Quadro 22 - Comparativo de resultados	47
Quadro 23 - UC1 Fazer Login.....	51
Quadro 24 - UC02 Manter Perfil do Usuário	51
Quadro 25 - UC03 Listar informações	51
Quadro 26 - UC04 Buscar artistas	51
Quadro 27 - UC05 Recomendar artistas.....	52
Quadro 28 - UC06 Avaliar itens sugeridos	52
Quadro 29 - UC07 Construir base de conhecimento.....	52
Quadro 30 - UC08 Recomendar artistas.....	52
Quadro 31 - UC09 Notificar recomendações	52

Quadro 32 - UC10 Notificar solicitações de amizade	52
Quadro 33 - Questionário de perfil do usuário	55
Quadro 34 - Lista de tarefa.....	56
Quadro 35 - Questionário de usabilidade	58

LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*

CSS – *Cascading Style Sheets*

FI – Filtragem de Informação

IDE - *Integrated Development Environment*

HTML – *HyperText Markup Language*

MER – Modelo de Entidade de Relacionamento

RF – Requisito Funcional

RI – Recuperação de Informações

RNF – Requisito não Funcional

SGBDR - Sistema Gerenciador de Banco de Dados Relacional

UC – Use Case

UML – *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS.....	15
1.2 ESTRUTURA.....	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 SISTEMAS DE RECOMENDAÇÃO.....	16
2.2 ESTRATÉGIAS DE RECOMENDAÇÃO	19
2.2.1 Recomendação baseada no conteúdo	20
2.2.2 Recomendação colaborativa.....	20
2.2.3 Recomendação baseada na utilidade.....	21
2.2.4 Recomendação híbrida	22
2.3 TRABALHOS CORRELATOS.....	23
2.3.1 Last.fm	23
2.3.2 ActivUFRJ	24
2.3.3 O que tá valendo?.....	25
2.3.4 Comparativo entre os trabalhos correlatos	26
3 DESENVOLVIMENTO	28
3.1 LEVANTAMENTO DE INFORMAÇÕES	28
3.2 ESPECIFICAÇÃO	29
3.2.1 Casos de Uso	29
3.2.2 Diagrama de Atividades.....	30
3.3 IMPLEMENTAÇÃO	32
3.3.1 Técnicas e ferramentas utilizadas.....	32
3.3.2 Fluxo geral da aplicação.....	32
3.4 RESULTADOS E DISCUSSÕES.....	41
3.4.1 Validação do sistema de recomendação.....	41
3.4.2 Experimento de usabilidade	44
3.4.3 Comparação com trabalhos correlatos	47
4 CONCLUSÕES	48
4.1 EXTENSÕES	48
REFERÊNCIAS	49
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO	51

APÊNDICE B – DICIONÁRIO DE DADOS.....	53
APÊNDICE C – ROTEIRO E QUESTIONÁRIO DE AVALIAÇÃO DE USABILIDADE.....	55

1 INTRODUÇÃO

Na sociedade atual, todos necessitam de informações para realizar suas atividades diárias, sejam estas para tarefas profissionais ou pessoais. Nesse sentido, a Internet se tornou um meio poderoso para facilitar a busca por informação, por ser um local onde todos podem expressar suas opiniões, interesses e experiências. O grande desafio está em encontrar informações realmente relevantes e confiáveis, frente à sobrecarga de informação ocasionada pela crescente quantidade de dados existente na Internet (ZANETTE, 2008).

Segundo Oliveira (2011), para tentar suprir esta necessidade, inicialmente, foram utilizados sistemas de Recuperação de Informações (RI), como por exemplo, a ferramenta de buscas do Google. Apesar de muito úteis, as ferramentas de busca retornam uma quantidade de informação proporcional à existente na internet, o que muitas vezes obriga o usuário a realizar uma busca manual por resultados relevantes dentre os obtidos.

Diante disso, recentemente surgiram os sistemas de recomendação cujo objetivo é produzir recomendações individualizadas, ou que guie o usuário de forma a apresentar conteúdo de seu interesse dentre uma variedade de opções. Como sugerem Reategui e Cazella (2005), os sistemas de recomendação auxiliam no aumento da capacidade e eficácia deste processo de indicação já bastante conhecido na relação social entre seres humanos. Segundo Cazella, Nunes e Reategui (2013) um dos grandes desafios deste tipo de sistema é realizar a combinação adequada entre as expectativas dos usuários e os produtos, serviços e pessoas a serem recomendados aos mesmos, ou seja, definir e descobrir este relacionamento de interesses é o grande problema.

Com os avanços das pesquisas, diferentes informações dos usuários são utilizadas para gerar as recomendações, entre elas: redes sociais, dados demográficos, conteúdo dos itens, histórico de navegação do usuário, entre outras (CAZELLA; NUNES; REATEGUI, 2013). Segundo Castro (2004) para diminuir as dúvidas e necessidades que tem-se frente à escolha entre alternativas, geralmente confia-se nas recomendações que são passadas por outras pessoas, as quais podem chegar de forma direta ou através de textos de recomendação, opiniões de revisores de filmes e livros, impressos de jornais, dentre outros.

Com a popularização da internet e das redes sociais, diversas tecnologias relacionadas vêm se consolidando como, por exemplo o Twitter, que é um serviço de *microblogging* de caráter altamente social que permite que você poste mensagens com 140 caracteres ou menos; essas mensagens são chamadas de *tweets*. O LinkedIn, que é um popular site de rede social concentrado em relacionamentos profissionais e de negócios e o Facebook que permite

informar aos seus amigos o que estão pensando, trocar mensagens mais longas, participar de *chats* em tempo real e mais uma dúzia de coisas (CAZELLA; NUNES; REATEGUI, 2013).

No entanto, quando se fala em tecnologias voltadas a troca de informações musicais, tem-se o Last.fm, Amazon ou Spotify. Entretanto, segundo Russell (2011, p. 135) nesses ambientes o usuário tem dificuldades para encontrar novos artistas ou bandas, pois as recomendações são limitadas aos artistas/banda que o usuário adiciona em sua biblioteca. As indicações de amigos não são relevantes, pois ao indicar uma banda é enviada uma simples mensagem ao invés de uma notificação.

Segundo Castro (2004) esta dificuldade resultou no surgimento de sistemas de recomendação no âmbito musical cujo objetivo é apresentar, para os usuários, músicas que sejam do seu interesse. Estes sistemas agrupam os indivíduos com interesses, assuntos, comportamento e gostos em comum, onde permite-se a troca de experiência, possibilitando uma troca de informações mais eficiente num espaço menor tempo.

Diante desse cenário, este trabalho propõe o desenvolvimento de uma aplicação que seja capaz de identificar as preferências do usuário e a partir delas recomendar artistas que sejam do seu interesse.

1.1 OBJETIVOS

O objetivo deste trabalho é o desenvolvimento de uma aplicação *web responsive* usando técnicas de recomendação para sugestões de artistas.

Os objetivos específicos deste trabalho são:

- a) coletar e armazenar informações dos usuários;
- b) estabelecer critérios e estatísticas para transformar as informações coletadas em recomendações conforme perfil do usuário;
- c) disponibilizar através de uma interface *web*, um painel (*dashboard*) para visualizar e gerenciar notificações/sugestões de outros usuários.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. O primeiro capítulo apresenta a introdução do trabalho e os objetivos. O segundo capítulo apresenta a fundamentação teórica sobre técnicas de recomendação. No terceiro capítulo é demonstrado o desenvolvimento do trabalho com requisitos, especificação, implementação, resultados e operacionalidade da aplicação. No quarto capítulo são relatadas as conclusões e também possíveis extensões.

2 FUNDAMENTAÇÃO TEÓRICA

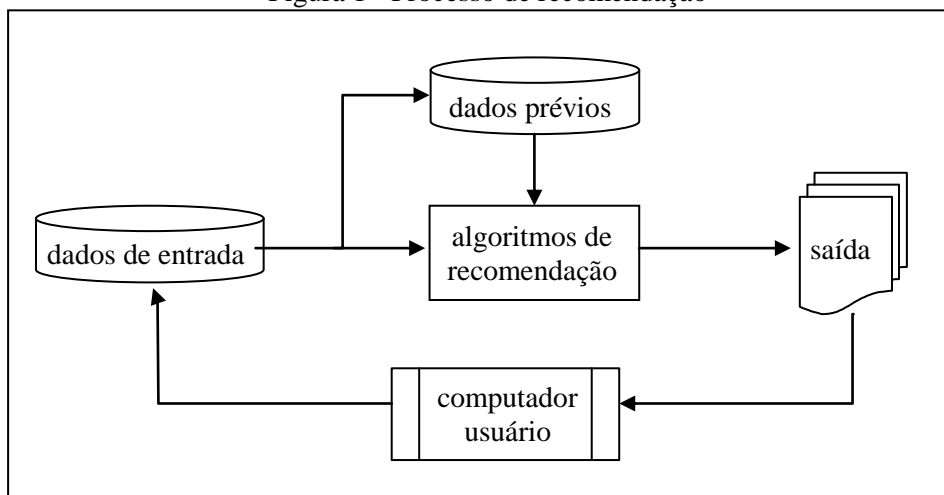
Nas próximas seções são apresentados os principais conceitos relacionados a sistemas de recomendação. Na seção 2.1 é descrito sobre sistemas, estratégias e técnicas de recomendação. E, na seção 2.3 são apresentados três trabalhos correlatos.

2.1 SISTEMAS DE RECOMENDAÇÃO

Os sistemas de recomendação são definidos, segundo Resnick e Varian (1997), como sistemas que auxiliam e aumentam a eficácia do processo natural de recomendações “boca-a-boca” de livros, filmes, restaurantes, etc. Através destes sistemas é possível recomendar, aos usuários, produtos de uma determinada categoria, baseando-se no conhecimento e utilizações prévias (compras, navegação, mensagens, etc.) de outras pessoas presentes no sistema.

Os sistemas de recomendação auxiliam no aumento da capacidade e da eficácia deste processo de indicação já bastante conhecido na relação social entre seres humanos (REATEGUI; CAZELLA, 2005). Segundo Schafer, Konstan e Riedl (1999) os sistemas de recomendação são usados pelos sites de comércio eletrônico para sugerir produtos para seus clientes e fornecer informações que procuram auxiliar os clientes sobre qual produto deve ser adquirido. De acordo com Burke (2002) os sistemas de recomendação são formados por dados prévios armazenados, dados de entrada ou do usuário e do algoritmo de recomendação. Esta estrutura pode ser vista na Figura 1.

Figura 1 - Processo de recomendação



Fonte: adaptado de Burke (2002, p. 5).

Os dados prévios ou armazenados correspondem ao conjunto de itens que será recomendado, ou seja, os produtos, páginas ou conteúdo que estão armazenados no sistema e que serão indicados ao usuário.

Os dados de entrada referem-se às informações que o usuário fornecerá ao sistema para que seja possível a construção do seu perfil. Estas informações podem ser coletadas de forma explícita ou implícita. A primeira forma geralmente é feita através do preenchimento de um formulário ou questionário que mostre possíveis interesses do usuário, mas também pode ser feita através da análise de um conteúdo. A segunda é realizada através da análise do comportamento do usuário, como por exemplo, verificando quais páginas ele navegou, qual produto visualizou ou comprou etc. (OLIVEIRA, 2007). A definição de quais informações serão necessárias para realizar a coleta é um fator essencial para gerar recomendações que sejam realmente relevantes para o usuário.

Outro ponto relevante referente à captura dos dados de entrada é a identificação de usuários. Sem essa identificação o sistema de recomendação passa a “não existir”, já que não se sabe através de que perfil deve ser feita a filtragem ou indicação do conteúdo existente. Vale lembrar que pode existir uma exceção, caso o software utilize dados temporários de um usuário, como por exemplo, sua geolocalização. Nesse caso, o sistema não está realmente identificando um usuário ou fazendo uso de um perfil para realizar a recomendação, mas sim analisando dados que o “visitante” da aplicação possui e que pode ser utilizado para cumprir mesmo que parcialmente sua funcionalidade (OLIVEIRA, 2007).

De acordo com Cazella, Nunes e Reategui (2013), as duas maneiras mais comuns de realizar a identificação de usuários em um sistema web são:

- a) identificação no servidor: disponibiliza ao usuário uma área de cadastro com informações pessoais. Estas informações ficam armazenadas em um banco de dados no servidor. Ao acessar o sistema o usuário fará sua autenticação informando os dados cadastrados como *login* e senha. Este mecanismo permite com que a aplicação identifique qual o usuário conectado;
- b) identificação no cliente: utiliza normalmente *cookies*, mecanismo pelo qual um *website* pode armazenar informações no browser cliente. Este método assume que a máquina conectada é utilizada sempre pela mesma pessoa. Logo, ao identificar a máquina, o *website* está na realidade identificando seu usuário. Trata-se de um mecanismo mais simples do que a identificação através do servidor, porém menos confiável, principalmente se o computador identificado for utilizado por mais de uma pessoa, ou se os dados de *cookies* forem apagados pelo usuário.

Os dados de saída de um sistema de recomendação podem ser uma predição ou uma recomendação. A predição consiste em apresentar um item associado a uma escala que prediz

sua relevância para o usuário. Um exemplo de predição ocorre ao apresentar um filme classificado com uma estrela, significando “não recomendado ao usuário”, enquanto outro que possui cinco estrelas representando “altamente recomendado”.

Segundo Oliveira (2007), a recomendação é representada por uma lista de N itens mais relevantes para o usuário. A lista costuma apresentar itens não comprados, vistos ou avaliados pelo usuário. Para Medeiros (2013), de forma geral, o problema de recomendação é estruturado como mostrado na Figura 2. As preferências são representadas como uma matriz A de n usuário e m itens onde cada célula $A_{u,i}$ corresponde à avaliação, nota, do usuário u ao item i .

Tipicamente essa matriz é esparsa, pois os usuários geralmente avaliam um subconjunto muito menor que o conjunto I . Portanto, o problema central dos sistemas de recomendação é extrapolar a função de utilidade u para todo o conjunto $U \times I$. O usuário em questão no processo de recomendação é referenciado como o usuário alvo.

Figura 2 - Matriz A de avaliação dos usuários. Cada célula corresponde à avaliação do usuário u para o item i . A tarefa é estimar o valor da avaliação $A_{a,i}$ para o usuário a .

		Itens					
		1	2	...	i	...	m
Usuários	1	5	3		5		
	2		2		4	1	3
	·			5			
	·						
	a	4	2		?	1	
	·			3			4
	·						
	n				2		5

Fonte: adaptado de Medeiros (2013, p. 6).

Ao construir um sistema de recomendação é necessário definir como será feito o processo de recomendação. Para a construção deste processo é necessária a adoção de uma técnica de Filtragem de Informação (FI). De acordo com Machado (2006), filtragem de informação consiste em uma série de processos que envolvem a entrega da informação para quem as necessita. Utilizando a FI, o sistema mantém um perfil dos interesses do usuário e busca continuamente informações que podem ser recomendadas.

2.2 ESTRATÉGIAS DE RECOMENDAÇÃO

Segundo Cazella, Nunes e Reategui (2013), os principais objetivos dos Sistemas de Recomendação são a fidelidade e o consequente aumento da satisfação de seus usuários. Diferentes estratégias podem ser personalizadas para um usuário, cada uma exigindo um grau de complexidade distinto no tratamento de informações coletadas. As estratégias mais utilizadas são descritas a seguir:

- a) recomendações por associação: este tipo de recomendação é obtido através de técnicas capazes de encontrar em uma base de dados associações entre itens avaliados por usuários (comprados, lidos e outros). Este tipo de recomendação é a forma mais complexa de recomendação. Ela exige uma análise mais profunda dos hábitos do usuário para a identificação de padrões e recomendação de itens com base nestes padrões;
- b) associação por conteúdo: também é possível fazer recomendações com base no conteúdo de determinado item, por exemplo, um autor, um compositor, um editor, entre outros. Para possibilitar este tipo de recomendação, é necessário que se encontrem associações num escopo mais restrito. Por exemplo, determinado livro contém assuntos que são relacionados a assuntos contidos em outras obras;
- c) análise de sequências de ações: sequências são um tipo importante de dados que podem ocorrer com frequência em diferentes domínios, como por exemplo, na medicina, negócios, segurança dentre outros. Na *web*, sequências podem ser utilizadas para capturar o comportamento de usuários através de históricos de atividade temporal, como em *weblogs* e histórico de compras de clientes. As informações encontradas nestas sequências temporais podem ser empregadas nas identificações de padrões de navegação e consumo, que em seguida podem servir aos Sistemas de Recomendação. Se comparado à recomendação por associação, a análise de sequências permite encontrar padrões que consideram a variável tempo, deste modo levando em consideração a ordem com que determinadas ações ocorrem. Em função do tipo de aplicação, tal característica pode ou não ser importante.

Segundo Lorenzi (2010), existe uma grande variedade de técnicas e formas diferentes de se calcular as estimativas tais como métodos de aprendizagem de máquinas, teoria da aproximação e uma quantidade variada de heurísticas. Os sistemas de recomendação são comumente classificados de acordo com a forma como eles obtêm essas estimativas.

Para Herlocker et al. (2000), os sistemas de recomendação são geralmente classificados em três categorias de acordo com a técnica de filtragem de informação que utilizam:

- a) recomendação baseada em conteúdo ou filtragem baseada em conteúdo;
- b) recomendação colaborativa ou filtragem colaborativa;
- c) recomendação baseada em utilidade;
- d) abordagem híbrida ou filtragem híbrida.

Nas próximas seções serão apresentadas e descritas as técnicas mais utilizadas pelos sistemas de recomendação.

2.2.1 Recomendação baseada no conteúdo

Na filtragem baseada no conteúdo o usuário de forma implícita ou explícita determina suas preferências e restrições, sendo assim, a recomendação baseada no conteúdo visa recomendar objetos classificados ou similares aos objetos do perfil do usuário. Como é definido por Salton (1989), “a descrição de interesses do usuário é obtida através de informações fornecidas por ele próprio ou através de ações, como seleção e aquisição de itens”.

O processo de recomendação baseado em conteúdo seleciona os itens mais similares aos itens identificados no perfil de interesse do usuário. Para isso ele compara o perfil do usuário com o perfil de cada item. O problema então se torna uma tarefa de recuperação de informação, onde o conteúdo associado ao perfil de usuário é tratado como uma consulta (*query*) e os documentos (itens) não avaliados são classificados e ranqueados de acordo com a relevância/similaridade em relação a essa consulta (BALABANOVIĆ; SHOHAM, 1997, p. 66-72).

2.2.2 Recomendação colaborativa

A recomendação colaborativa é realizada com o auxílio de indivíduos, baseando-se somente nas opiniões dos usuários sobre itens específicos. Essa técnica é bastante utilizada para agrupar pessoas através de comunidades, onde a recomendação será de pessoas mais confiáveis pelo fato de possuir gostos semelhantes.

Ao contrário da filtragem por conteúdo, a técnica de filtragem colaborativa, ou sistemas de recomendação colaborativos, tenta predizer a avaliação de um determinado

usuário para um item baseado nas avaliações dos outros usuários (similares a ele) para este item (SARWAR, 2001). Em outras palavras, a ideia é usar a inteligência coletiva de um grupo de pessoas para fazer recomendações para outras (SEGARAN, 2008).

Por exemplo, o usuário X define suas preferências e, a partir deste momento, o sistema buscará usuários com perfil semelhante. Considerando que essa busca tenha como resultado o usuário Y, o sistema recomendará para o usuário X os mesmos itens recomendados para Y (LORENZI, 2010).

A capacidade de produzir recomendações de qualidade depende das avaliações positivas e negativas fornecidas pelo usuário. Para um sistema automatizado é muito difícil perceber a qualidade de um livro, por exemplo, baseando-se na análise do texto do livro, quando o usuário avalia esse livro fica mais simples para o sistema produzir recomendações de qualidade (LORENZI, 2010).

Dentre as vantagens em utilizar a filtragem colaborativa, Lorenzi (2010) destaca “a capacidade de o sistema recomendar produtos inesperados, mas com qualidade, ou seja, uma recomendação que surpreenda positivamente o usuário, essa ação é chamada de *serendipity*”.

2.2.3 Recomendação baseada na utilidade

A filtragem baseada na utilidade parte do princípio de que os usuários tendem a interessar-se por itens similares aos que demonstraram interesse no passado, definindo então, a similaridade entre os itens (HERLOCKER et al., 2000).

Segundo Burke (2002), o grande problema está em como desenvolver uma função de utilidade que supra as necessidades de cada usuário especificamente. As informações do perfil do usuário formam uma função de utilidade e o sistema emprega a técnica de satisfação de restrições na busca da obtenção do melhor resultado.

No cálculo de utilidade, além de considerar as características dos itens cadastrados, o sistema pode acrescentar variáveis que não estão diretamente relacionadas aos mesmos como, por exemplo, prazo de entrega, localização do produto, forma de pagamento, entre outros.

Dentre as vantagens da utilização de filtragem baseada em conteúdo, podem-se citar: não é necessário que um novo item tenha algum tipo de qualificação, bastando apenas que este seja semelhante ao perfil do usuário (BURKE, 2002).

Um dos problemas apresentados pela filtragem baseada na utilidade, segundo Burke (2002) a diversidade de itens recomendados fica restrita à similaridade de itens avaliados pelo

usuário, sem conseguir recomendar itens não relacionados que poderiam ser de interesse do cliente.

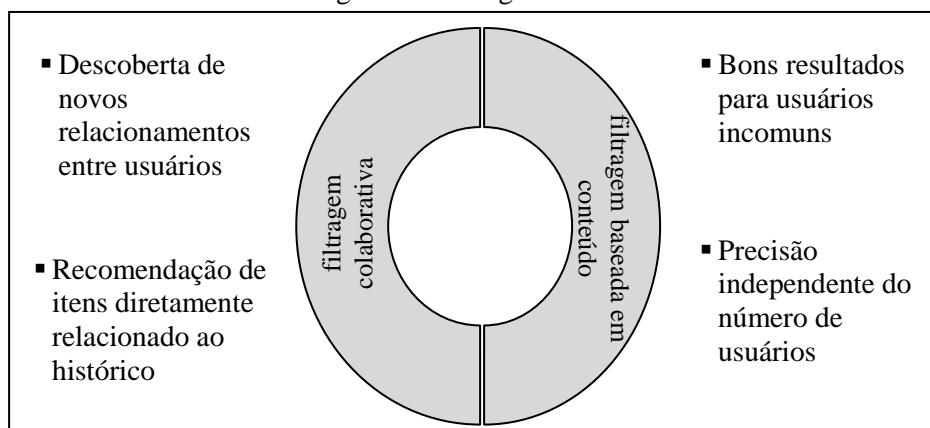
2.2.4 Recomendação híbrida

As filtragens mais utilizadas citadas nas seções anteriores, possuem desvantagens conhecidas. Por exemplo, na filtragem colaborativa há a desvantagem da independência de conteúdo e ao mesmo tempo existe a dependência da avaliação do conteúdo para um item ser recomendado. Enquanto na filtragem baseada no conteúdo, possui a desvantagem de recomendar qualquer produto, mas elimina o problema da dependência de que algum item precisa ser avaliado pelo usuário.

A filtragem híbrida procura combinar filtragem colaborativa e filtragem baseada em conteúdo visando criar um sistema que possa melhor atender às necessidades do usuário (HERLOKER, et al., 2000). Sistemas de recomendação híbridos são a combinação de mais de uma técnica de recomendação. Seu objetivo é explorar os pontos fortes de cada técnica a fim de minimizar as limitações apresentadas no uso de apenas uma técnica (BURKE, 2002).

Dessa forma, Reategui e Cazella (2005) definem sistemas de recomendação híbridos como a combinação da recomendação colaborativa e da recomendação baseada em conteúdo, onde os pontos fortes de cada uma dessas técnicas eliminam as fraquezas que elas apresentam isoladamente, visando criar um sistema que possa melhor atender às necessidades do usuário, como pode ser visto na Figura 3.

Figura 3 - Filtragem Híbrida



Fonte: adaptado de Reategui e Cazella (2005, p. 38).

A única desvantagem que a junção das técnicas de recomendação apresenta, é a de não resolver o problema da *start-up*, que refere-se ao tempo que o sistema leva para conseguir informações relevantes do usuário para gerar boas recomendações. Por isso, até que o sistema consiga informações para fazer recomendações, alguns autores sugerem recomendar uma lista

de itens mais vendidos ou utilizar dados demográficos como idade, sexo, cidade onde mora, para gerar uma recomendação.

2.3 TRABALHOS CORRELATOS

Esta seção tem como objetivo apresentar alguns trabalhos relacionados a recomendação de conteúdos. Dentre os trabalhos selecionados estão: o trabalho de Zanette (2008) que realiza a recomendação de filmes. O trabalho de Busatto (2013) que sugere eventos culturais da cidade de Porto Alegre e o Lastfm (2014) que cria mapas de gostos musicais e recomenda artistas de acordo com o perfil do usuário.

2.3.1 Last.fm

Last.fm (LASTFM, 2014) é um serviço de recomendação musical. Com o objetivo construir uma rede social de música na Internet, o Sistema de Recomendação Last.fm foi criado em 2003 como rádio online. Porém, em agosto de 2005 incorporou tanto o software Audioscrobbler (que registra os hábitos de escuta dos ouvintes) como a interface de acesso as músicas através das *tags* criadas por seus usuários. Na Figura 4 é apresentada a tela principal do Last.fm.

Figura 4 - Tela principal do Last.fm



Fonte: Lastfm (2014).

A Last.fm estimula seus usuários a criarem *tags* para: 1) construir *playlists* de canções baseadas na classificação colaborativa da música pela comunidade; 2) categorizar o perfil e gosto de cada ouvinte; 3) melhorar as recomendações do sistema baseado nas tags 4) que a

comunidade de usuários colabore na classificação do “maior catálogo global de música” (LAST.FM, 2014).

O Sistema de Recomendação da Last.fm ajuda o usuário à administrar o problema do excesso de informação, recomendando itens que correspondam aos seus critérios pessoais de relevância. Para isso, o Last.fm articula informações relevantes na lógica de difusão e consumo da música na Internet através da classificação colaborativa da música. Ele, utiliza recomendações baseada na *folksonomia*¹, mapas de gostos musicais dos usuários e das redes sociais de ouvintes e dessa forma, aponta para as tendências dos usos sociais da música nestes novos contextos (SANTINI; SOUZA; CALVI, 2009).

A Last.fm constrói um perfil público detalhado do gosto musical de cada usuário mostrando suas músicas e artistas favoritos. O perfil de todos os usuários da comunidade é utilizado para listar artistas parecidos, músicas mais tocadas, pessoas com interesses semelhantes e a interação de usuário uns com outros usuários.

2.3.2 ActivUFRJ

Zanette (2008) apresentou um sistema de recomendação baseado na rede de confiança do usuário. Este sistema foi batizado de ActivUFRJ e, consiste em uma alternativa para que o usuário alvo consiga identificar qual a importância de um determinado item recomendado frente a sua confiança nos recomendadores. Na Figura 5 é apresentada a tela principal do ActivUFRJ.

Figura 5 - Tela principal do ActivUFRJ



Fonte: Zanette (2008, p. 79).

¹ Folksonomia é uma técnica que possibilita a classificação de conteúdo por meio do processo de marcação (SANTINI; SOUZA; CALVI, 2009)

Um conceito aplicado ao ActivUFRJ (ZANETTE, 2008) é o de mostrar ao usuário uma alternativa onde ele consiga identificar qual é a importância de um item recomendado por recomendadores confiáveis, possibilitando a integração e fidelização de usuários. A ideia apresentada é que as avaliações e recomendações de um item por pessoas confiáveis tem grau relevância superior para o indivíduo.

O estudo desenvolvido por Zanette (2008) constatou que para a utilização da técnica de filtragem colaborativa é necessário que se tenha um perfil com as preferências do usuário para gerar recomendações. Isso afeta os novos usuários que não podem receber recomendações, pois o sistema de recomendação não consegue identificar suas preferências e ao considerar a rede de confiança do usuário é possível fazer com que novos usuários comecem a receber recomendações mais rapidamente.

Através dos resultados dos experimentos realizados, Zanette (2008) evidenciou a relevância da rede de confiança no processo de recomendação, sendo que como resultado final verificou-se a diminuição dos problemas relativos à filtragem colaborativa, e principalmente, apresentando-se como uma boa solução para o problema do novo usuário.

2.3.3 O que tá valendo?

No trabalho de conclusão de curso de Busatto (2013), ele apresentou um sistema web de divulgação e recomendação de eventos culturais que estão ocorrendo em Porto Alegre. Chamado de "*O que tá valendo?*", seu diferencial é recomendar atividades que sejam do interesse de cada usuário. Na Figura 6 é apresentada a tela de eventos recomendados ao usuário.

Figura 6 - Tela de eventos recomendados



Fonte: Buzatto (2013, p. 53).

Para o desenvolvimento da aplicação foi implementado um processo de coleta de dados, onde, as informações sobre os eventos a serem divulgados são extraídas dinamicamente dos principais portais de eventos e casas noturnas previamente definidos.

A arquitetura do sistema tem no *framework* Yii a sua base de funcionamento, sendo ele responsável por toda a interação entre os diversos componentes presentes no sistema. O componente de “Web crawling” tem como objetivo extrair informações de sites definidos e adicionar novos eventos ao sistema. O componente de “Autenticação com o Facebook ” é responsável por realizar o cadastro e *login* de usuários, utilizando para isso, os dados do perfil da rede social, através do uso da Facebook API. O componente de “Banco de Dados” é responsável por manter a persistência dos dados que serão utilizados pelo componente de “Recomendação”, que por sua vez realiza a busca do histórico de uso do sistema e, através de diversos critérios e cálculos, tenta priorizar os eventos que serão mais interessantes àquele usuário específico.

A aplicação desenvolvida atingiu o seu objetivo de auxiliar na escolha de eventos e atividades de lazer, usufruindo de uma aplicação web que possibilita a divulgação de eventos da cidade de Porto Alegre, recomendando eventos baseado no conceito *treemapping*², método para exibir dados hierárquicos usando retângulos alinhados.

2.3.4 Comparativo entre os trabalhos correlatos

O Quadro 1 apresenta um comparativo entre as principais características dos trabalhos correlatos.

Quadro 1 - Comparativo entre os trabalhos correlatos

	Lastfm (2014)	Zanette (2008)	Buzatto (2013)
Área de atuação	Música	Filme	Evento
Conceito	Folksomia	Rede de confiança	<i>Treemapping</i>
Técnica de recomendação	Recomendação colaborativa	Recomendação colaborativa	Não Informado
Plataforma	Web	Web	Web

A partir do Quadro 1 é possível perceber que os sistemas de recomendação normalmente utilizam técnicas de recomendações associadas a um conceito para gerar uma recomendação mais exata. A técnica de recomendação colaborativa é a mais utilizada por se basear nas avaliações feitas pelos usuários. No trabalho do Zanette (2008), optou-se pelo uso do conceito de rede confiança para tentar encontrar uma correlação entre os usuários e assim

gerar recomendações mais confiáveis e permitir que novos usuários comecem a receber recomendações mais rapidamente. Já o trabalho do Bussato (2013) utilizou a técnica *treemaps* por demonstrar claramente quais as *tags* mais relevantes. E, o Last.fm (2014) opta por utilizar o conceito de *folksonomia* por permitir que os usuários adicionem *tags* facilitando a busca e o agrupamento de informações relacionadas. Outro ponto que pode ser observado é que todos os sistemas foram desenvolvidos para a plataforma Web.

² *Treemaps* é uma técnica que mapeia dados para retângulos 2D, onde informações com maior relevância são exibidas em retângulos maiores e a de menor importância, em retângulos que ocupam espaços menores (BUSSATO, 2013).

3 DESENVOLVIMENTO

Neste capítulo é apresentado o desenvolvimento da aplicação. A seção 3.1 apresenta os requisitos da aplicação desenvolvida. Na seção 3.2 está a especificação contendo os diagramas da parte principal da aplicação. Na seção 3.3 está descrita a implementação das principais partes da aplicação, assim como as ferramentas utilizadas e a operacionalidade da aplicação. Por fim, na seção 3.4 são apresentados os testes realizados e os resultados obtidos.

3.1 LEVANTAMENTO DE INFORMAÇÕES

A aplicação do trabalho de conclusão de curso tem como requisitos:

- a) permitir ao usuário manter e atualizar o perfil do usuário (Requisito Funcional – RF);
- b) permitir que o usuário se autentique utilizando seu *login* e senha cadastrado (RF);
- c) listar as preferências musicais do usuário (RF);
- d) permitir que o usuário busque artistas de seu interesse (RF);
- e) permitir que o usuário possa sugerir músicas (RF);
- f) permitir que o usuário avalie os itens recomendados por outros usuários (RF);
- g) construir uma base de conhecimento a partir das ações realizadas pelo usuário na aplicação (RF);
- h) recomendar artistas a partir do perfil do usuário e da associação de informações contidas na base de conhecimento (RF);
- i) notificar o usuário quando alguém recomendar artistas (RF);
- j) notificar o usuário quando uma solicitação de amizade for solicitada (RF);
- k) ser desenvolvida em Python utilizando o *framework* Django (Requisito não funcional – RNF);
- l) ser *webresponsive* (RNF);
- m) utilizar banco de dados PostgreSQL (RNF);
- n) ter uma interface de fácil utilização pelo usuário (RNF);
- o) ser acessível pelos navegadores Mozilla Firefox e Microsoft Internet Explorer 9.0 ou versão superior (RNF).

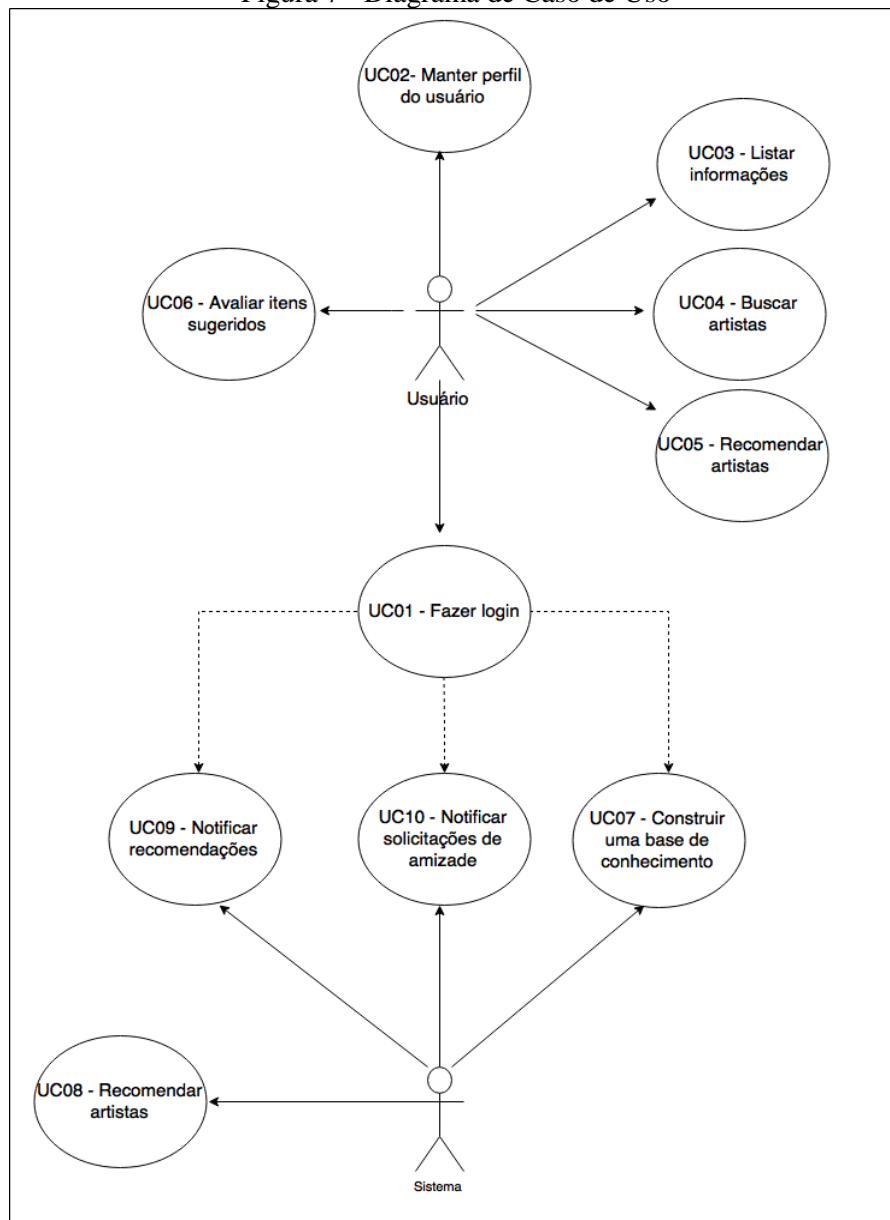
3.2 ESPECIFICAÇÃO

A especificação da aplicação foi desenvolvida de acordo com os diagramas da *Unified Modeling Language* (UML), utilizando a ferramenta ArgoUML 0.34. Neste trabalho, foram elaborados os diagramas casos de uso, atividades e de classes. O diagrama de Entidade Relacionamento está no Apêndice B.

3.2.1 Casos de Uso

Nesta seção é apresentado o diagrama de casos de uso (UC) do sistema. Na Figura 7 são apresentadas as funcionalidades que os atores *Sistema* e *Usuário* podem realizar na aplicação.

Figura 7 - Diagrama de Caso de Uso

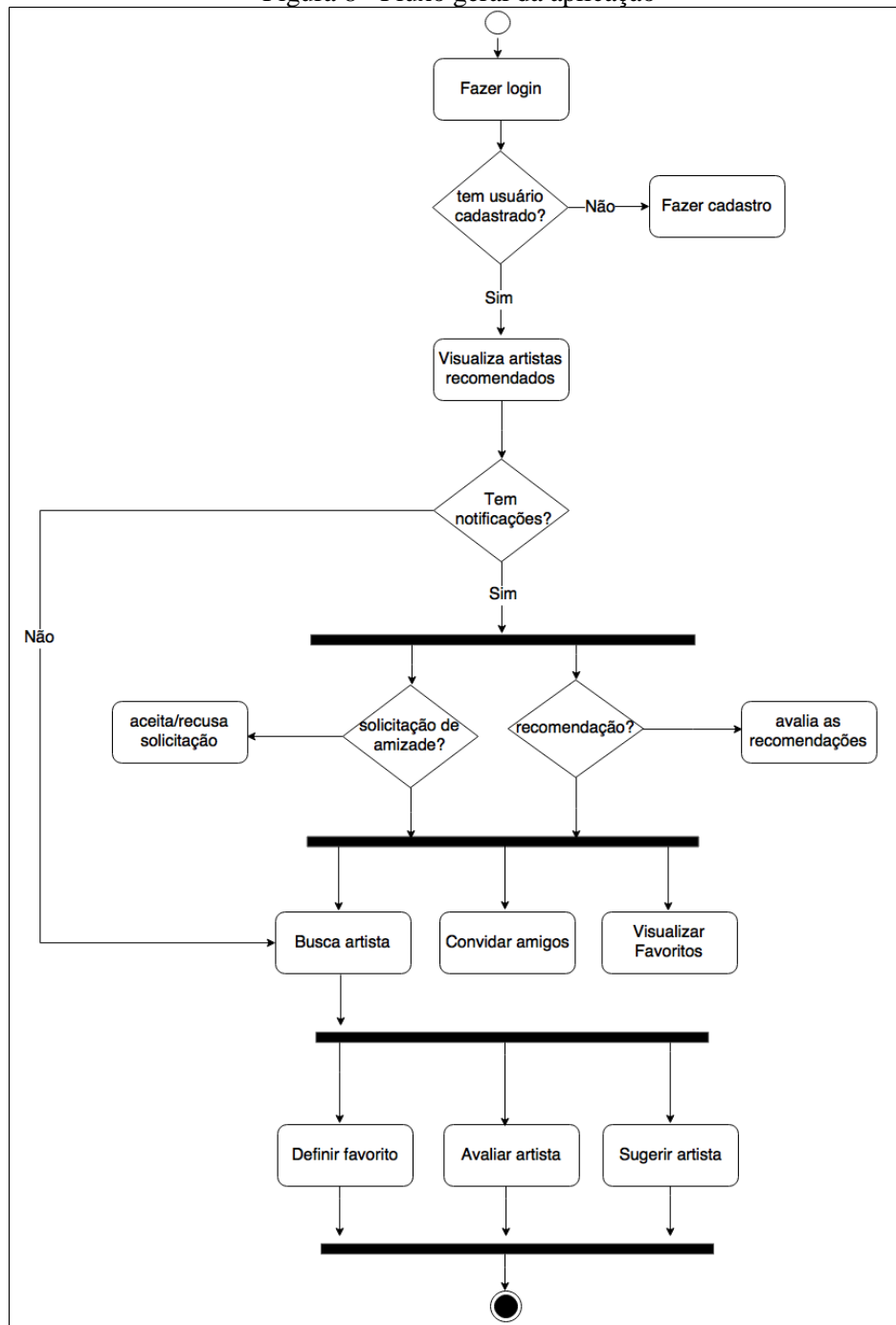


No caso de uso UC01: Fazer login o usuário, através de seu email e senha, pode acessar a aplicação. No caso de uso UC02: Manter perfil usuário o usuário pode alterar suas informações. O caso de uso UC03: Listar informações mostra as preferências musicais do usuário. Através do caso de uso UC04: Buscar artistas o usuário pode pesquisar os artistas de seu interesse. Já no caso de uso UC05: Recomendar artistas o usuário pode recomendar qualquer artista para outro usuário. O caso de uso UC06: Avaliar itens sugeridos permite ao usuário avaliar uma sugestão recebida. No caso de uso UC07: Construir uma base de conhecimento o sistema armazena as informações das ações do usuário para fazer recomendações. No caso de uso UC08: Recomendar artistas a aplicação calcula a pontuação e faz recomendações com base no perfil do usuário que foi obtido a partir da base de conhecimento. O caso de uso UC09: Notificar recomendações é responsável por notificar o usuário sempre existir novas recomendações. No caso de uso UC10: Notificar solicitações de amizade o sistema deve notificar o usuário caso exista novas solicitações de amizade. O detalhamento dos casos de uso pode ser visualizado no Apêndice A.

3.2.2 Diagrama de Atividades

Para um melhor entendimento da estrutura da aplicação foi elaborado o diagrama de atividades da Figura 8 representando o fluxo principal da aplicação.

Figura 8 - Fluxo geral da aplicação



A partir da Figura 8 é possível perceber que para acessar a aplicação o usuário precisa efetuar o cadastro. Após realizar o *login* na aplicação, o usuário poderá visualizar as recomendações de artistas que foram geradas a partir do seu gosto musical. Posteriormente, ele pode checar suas notificações, sendo elas de solicitação de amizade ou de sugestão musical. O usuário também pode avaliar uma sugestão indicando se “Gostei” ou “Não Gostei”. Seguindo o fluxo, o usuário também pode realizar buscas por novos artistas, podendo avaliar, compartilhar e definir um artista como favorito. A aplicação também permite ao

usuário convidar novos amigos para participar da aplicação, assim como, enviar convites de amizades e visualizar seus artistas prediletos.

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas pesquisadas e utilizadas, detalhando algumas das principais rotinas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

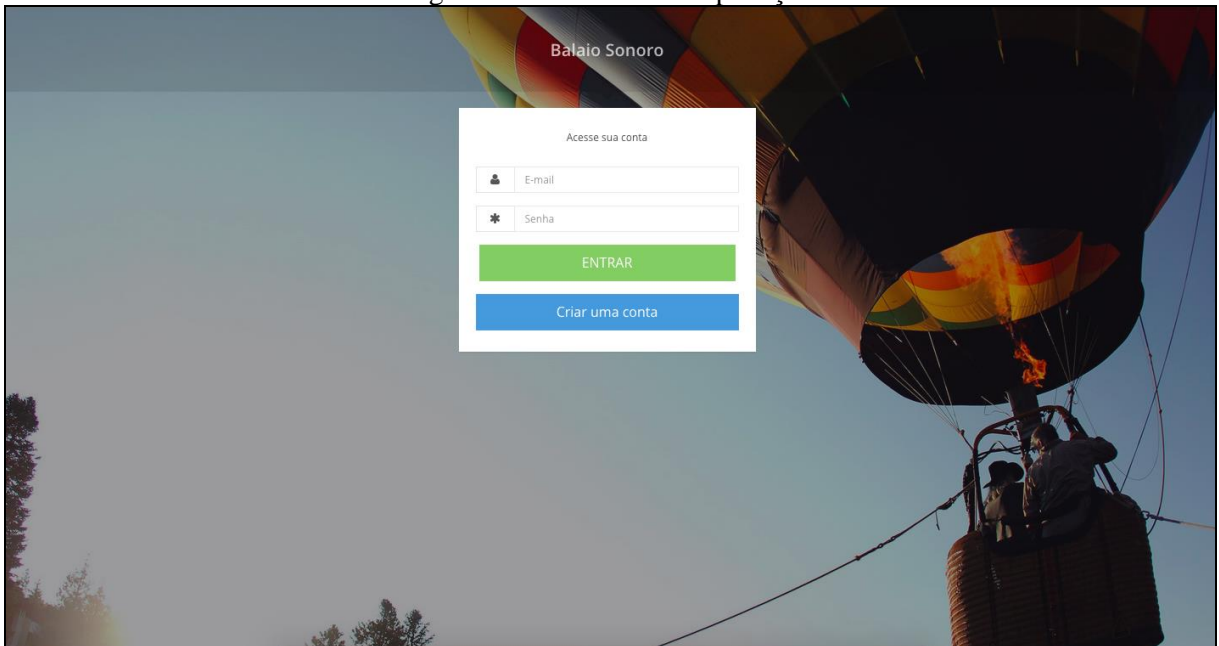
Para o desenvolvimento da aplicação foi utilizada a linguagem de programação JavaScript que é uma linguagem de *script* incorporada a um documento HTML, sendo altamente dependente do navegador que chama a página *web* e ao mesmo tempo não necessita de um compilador. A aplicação foi desenvolvida utilizando o *framework* Django, que é desenvolvido na linguagem de programação Python, seguindo as filosofias de desenvolvimento rápido e limpo. A *Integrated Development Environment* (IDE) escolhida foi o Vim. O banco de dados escolhido foi o PostgreSQL que é um Sistema Gerenciador de Banco de Dados Relacional (SGBDR) que é compatível com principais sistemas operacionais e suporta o armazenamento de grandes objetos binários. Para o desenvolvimento da interface foi utilizado a biblioteca Bootstrap, que é *webresponsive* e que utiliza os artefatos do HTML5 e CSS3.

Para realizar as implementações de busca, lista de álbuns, estilo dos artistas foi extraída uma base de dados da MusicBrainz que é um banco de dados musical com código aberto. Para coletar informações como capa do álbum e artistas similares foi utilizada a API EchoNest que é uma das maiores empresas de inteligência musical que usa uma combinação de aparelho de escuta e varredura intensa para criar seu próprio banco de dados, fornecendo aos desenvolvedores uma compreensão mais profunda da música.

3.3.2 Fluxo geral da aplicação

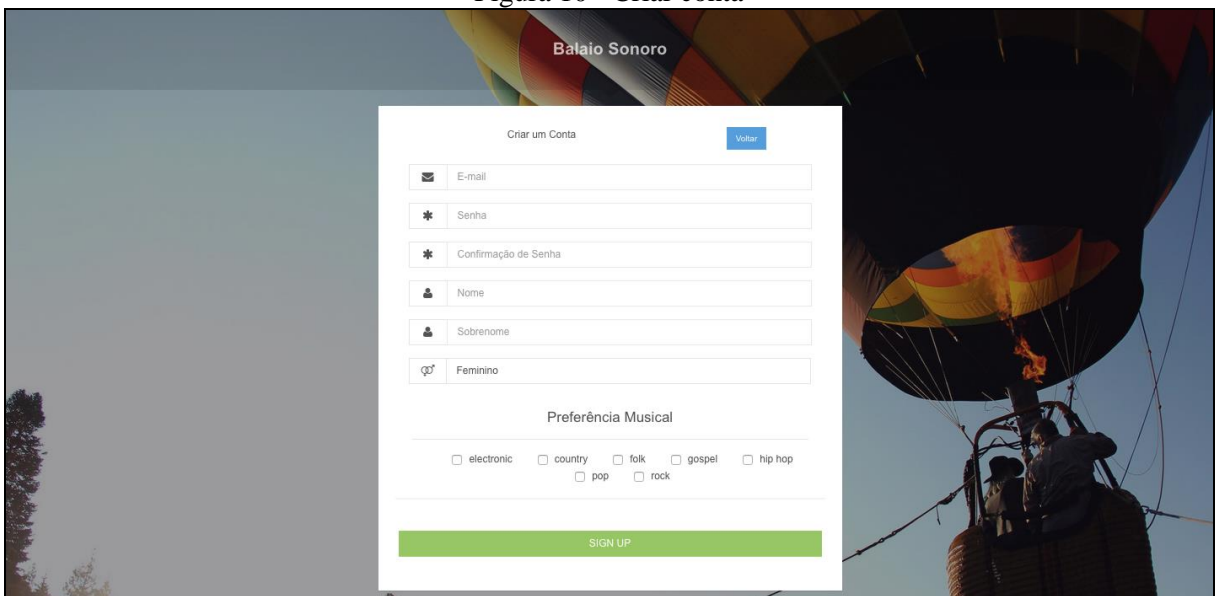
Ao acessar a aplicação tem-se a tela de autenticação do usuário, onde, o usuário precisa informar seu endereço eletrônico e sua senha para prosseguir. A Figura 9 contém a tela inicial da aplicação.

Figura 9 - Tela inicial da aplicação



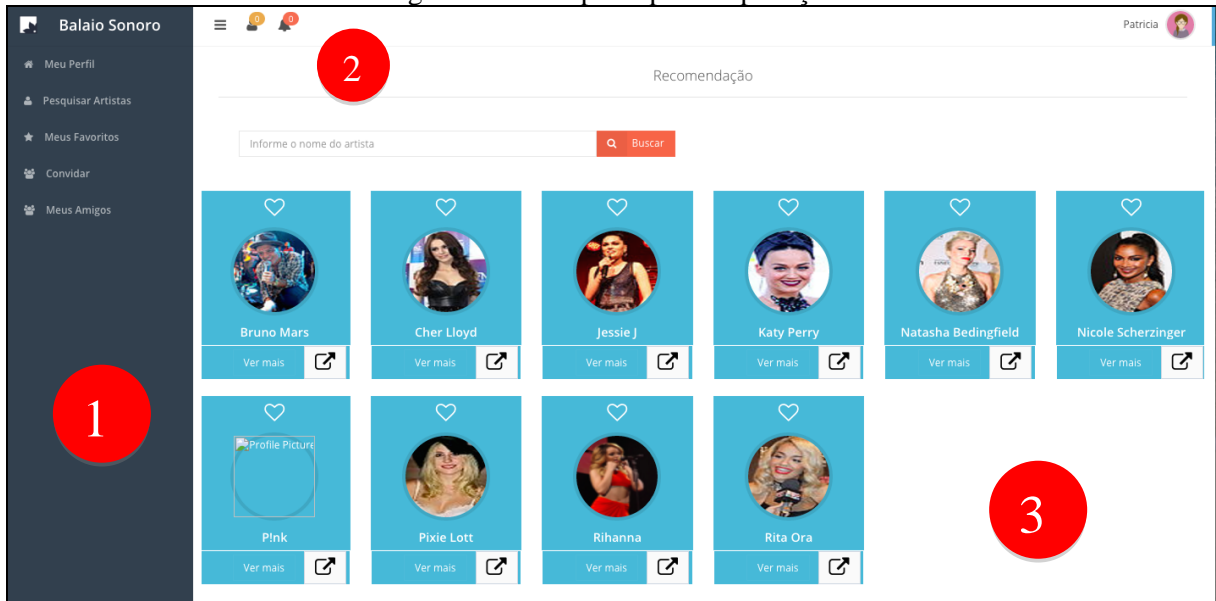
Caso o usuário não esteja cadastrado, ele tem a opção de realizar o cadastro na aplicação. A Figura 10 apresenta a tela para criação de uma nova conta. Nela, o usuário deverá informar o seu endereço eletrônico, sua senha, confirmar sua senha, seu nome e sobrenome, seu gênero e, principalmente, definir sua preferência musical, que futuramente será utilizada para a aplicação fazer recomendações de artistas.

Figura 10 - Criar conta



Após efetuar o cadastro e a autenticação, o usuário será redirecionado para a tela principal da aplicação, conforme mostra a Figura 11.

Figura 11 - Tela principal da aplicação

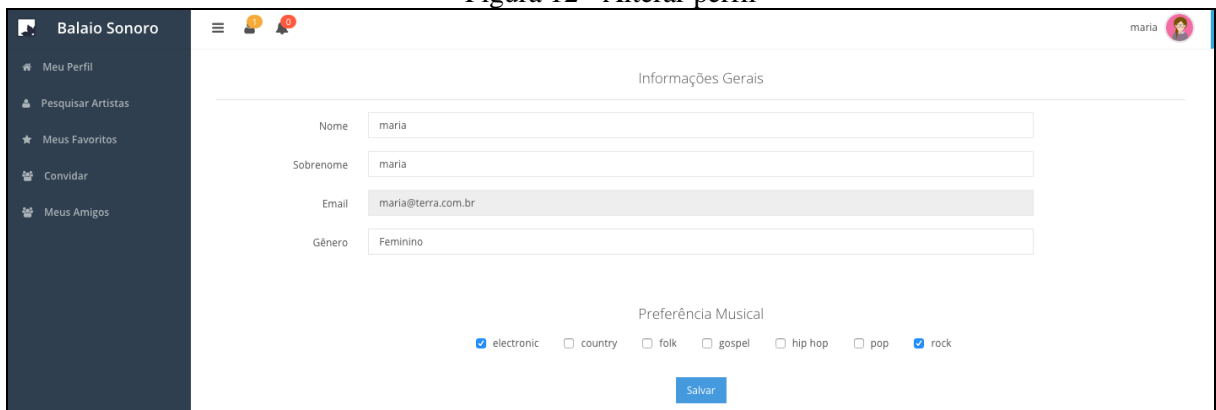


Na Figura 11, pode-se observar que a tela principal da aplicação está subdividida em 3 áreas. Na primeira (item 1), encontra-se o menu de opções. A segunda (item 2), apresenta as notificações de solicitação de amizade e sugestão de artista. E, na terceira (item 3), encontra-se uma lista de artistas recomendados a partir da análise do perfil musical indicado no cadastro de usuário. Nas próximas seções serão detalhadas as funcionalidades de cada área.

3.3.2.1 Menu de opções do usuário

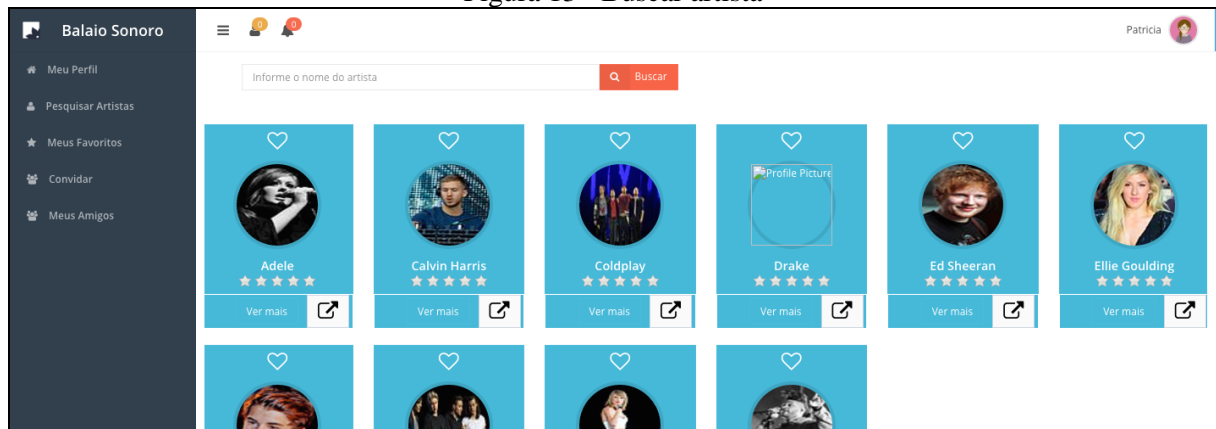
Ao acessar a opção `Meu perfil` na barra de menu, o usuário poderá atualizar as suas informações cadastrais (nome, sobrenome, gênero e preferência musical), exceto o endereço eletrônico, pois ele é o principal mecanismo de comunicação entre os usuários da aplicação. A Figura 12 apresenta a tela que permite ao usuário alterar seu perfil.

Figura 12 - Alterar perfil



A partir da opção `Pesquisar Artistas`, o usuário pode procurar artistas existentes e, que foram importados da base musical MusicBrainz. A Figura 13 apresenta a tela de busca de artistas.

Figura 13 - Buscar artista



É importante ressaltar que cada ação (busca, avaliação, recomendação e favorito) possui uma pontuação específica, ao qual serão utilizadas para definir as recomendações para o usuário.

Ao realizar a ação de busca, é verificado se existe vínculo entre o artista e o usuário. Se existir, é incrementado 0,1 no atributo `vl_pesquisa`. Do contrário, cria-se um novo registro. O Quadro 2 contém o trecho de código que pontua cada pesquisa realizada pelo usuário.

Quadro 2 - Trecho de código para pontuar pesquisa

```
01 if Pontuacao.objects.filter(usuario=user,artista=new_artista).exists():
02     pontuacao=Pontuacao.objects.get(usuario=user, artista=new_artista)
03     pontuacao.vl_pesquisa = pontuacao.vl_pesquisa + 0.1
04     pontuacao.save()
05 else:
06     Pontuacao.objects.create(usuario=user,artista=new_artista,\
07         vl_pesquisa=0.1)
```

Quando o usuário clicar para visualizar as informações de um determinado artista, ele tem a opção de defini-lo como favorito. Para isso, basta clicar sobre o ícone na forma de coração. Mediante a esta ação, cria-se ou verifica-se a existência do vínculo entre o usuário e o artista. Se existir, o atributo `vl_favorito` recebe nota 1.0. O Quadro 3 contém o trecho de código responsável por salvar um artista como favorito.

Quadro 3 - Trecho e código para salvar favorito

```
01 if Pontuacao.objects.filter(usuario=user, artista=new_artista).exists():
02     pontuacao = Pontuacao.objects.get(usuario=user, artista=new_artista)
03     pontuacao.vl_favorito = 1.0 if favorito else 0.0
04     pontuacao.save()
05 else:
06     if favorito:
07         Pontuacao.objects.create(usuario=user,artista=new_artista,
08     vl_favorito=1.0)
```

O usuário também tem a possibilidade de avaliar um artista. Para isso, é necessário selecionar as estrelas, quanto mais estrelas selecionadas, melhor será o conceito do artista em relação ao gosto musical do usuário. Cada estrela, num total de 5 estrelas, vale 0,2 pontos no cálculo da recomendação. Para salvar a pontuação é validado se existe vínculo entre o usuário

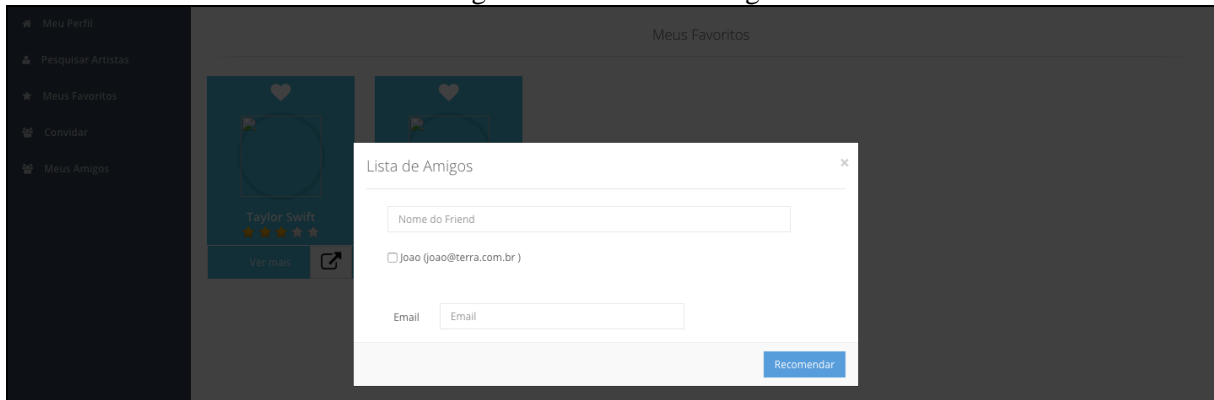
e o artista. Se existir, o atributo `vl_stars` é atualizado, do contrário, um novo registro será criado. O Quadro 4 contém o trecho de código responsável por pontuar uma avaliação.

Quadro 4 - Trecho de código pontuar avaliação

```
01 if Pontuacao.objects.filter(usuario=user, artista=artista).exists():
02     pontuacao = Pontuacao.objects.get(usuario=user, artista=artista)
03     pontuacao.vl_stars = float(request.POST.get("nota")) * 0.2
04     pontuacao.save()
05 else:
06     Pontuacao.objects.create(usuario=user, artista=new_artista,\
07         vl_stars=float(request.POST.get("nota")) * 0.2)
```

O usuário ainda tem a opção de sugerir um artista para outros usuários. Para isso, é necessário clicar no botão de compartilhar e posteriormente selecionar o usuário ou informar um endereço eletrônico. A Figura 14 contém a tela para sugerir o artista.

Figura 14 - Convidar amigo



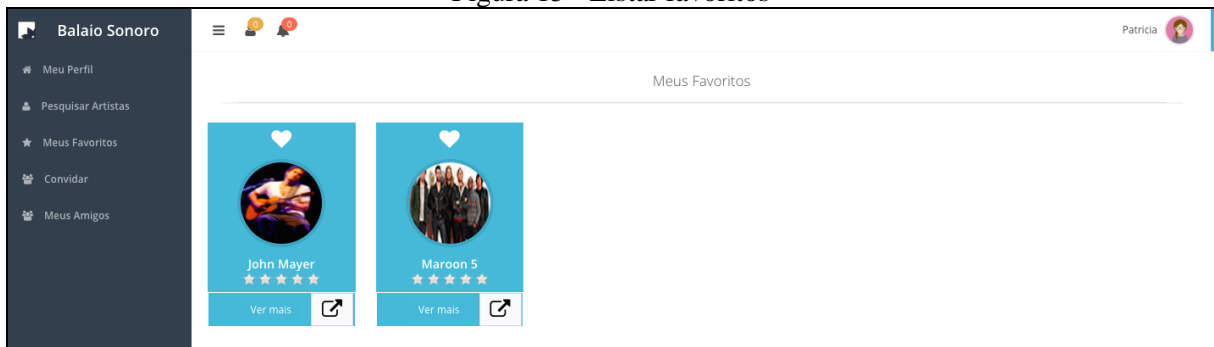
Ao sugerir um artista para um usuário, é gerada uma notificação avisando-o de uma nova recomendação. O Quadro 5 contém o trecho de código responsável por salvar uma nova notificação de sugestão.

Quadro 5 - Trecho de código para sugerir artista

```
01 for friend in friend_list:
02     amigo = FacebookUser.objects.get(pk=friend)
03     Notificacao.objects.create(descricao=json.dumps({"nome_artista":\
04         nome_artista[0]["name"],"id_artista":artista_id,"gid_artista":\
05         request.POST.get('gid')}),ativo=True, visualizada=False,\
06         data=datetime.now(),from_usuario=usuario, to_usuario=\
07         amigo.email,tipo=TipoNotificacao.objects.get(pk=4))
08
09 if request.POST.get("email"):
10     Notificacao.objects.create(descricao=json.dumps({"nome_artista":\
11         nome_artista[0]["name"],"id_artista":artista_id,"gid_artista":\
12         request.POST.get('gid')}),ativo=True, visualizada=False,\
13         data=datetime.now(),from_usuario=usuário, to_usuario=\
14         request.POST.get("email"),tipo=\
15         TipoNotificacao.objects.get(pk=4))
```

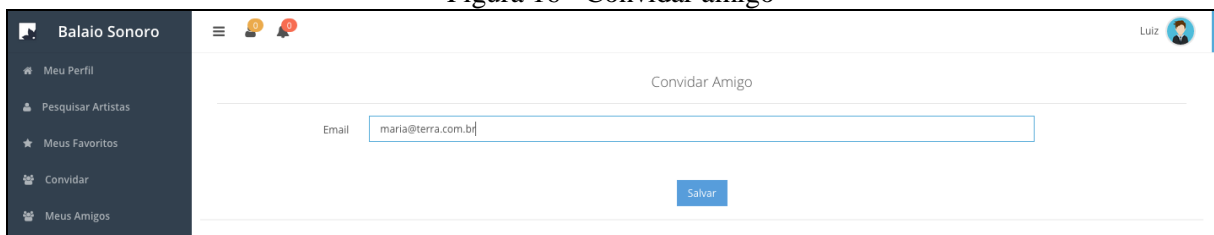
Seguindo o menu de opções, o usuário também tem a possibilidade de consultar os seus artistas favoritos. Para isso, ele precisa selecionar a opção “Meus Favoritos”. A Figura 15 contém a lista dos artistas favoritos.

Figura 15 - Listar favoritos



Para remover o artista de sua lista de favoritos, basta clicar sobre o ícone representado pelo coração, desmarcando-o. O sistema também permite ao usuário convidar alguém para participar da aplicação ou enviar um convite de amizade. Para isso é necessário acessar o menu “Convidar” e informar um endereço eletrônico. A Figura 16 apresenta a tela para enviar convite.

Figura 16 - Convidar amigo



Se o endereço eletrônico já estiver cadastrado na aplicação, é enviada uma notificação de solicitação de amizade. Do contrário, é enviado um email convidando a pessoa desejada para conhecer a aplicação. O Quadro 6 contém o trecho de código responsável por convidar amigos.

Quadro 6 - Trecho de código para convidar amigo

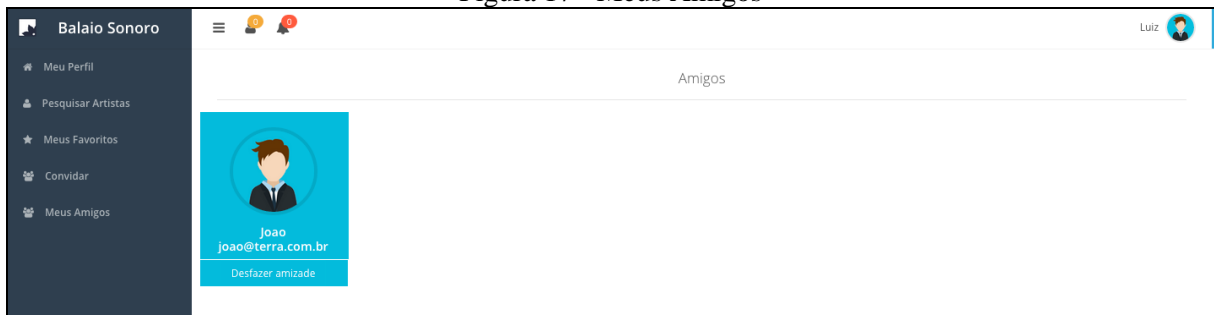
```

01 if FacebookUser.objects.filter(email=email).exists():
02     Notificacao.objects.create(descricao=json.dumps(\
03         {"descricao": "Convite de Amizade"}), ativo=True,\
04         visualizada=False, data=datetime.now(), from_usuario=usuario,\
05         to_usuario=email, tipo=TipoNotificacao.objects.get(pk=2))
06 else:
07     msg = "Um sistema de recomendação de música,\
08         venha conhecer em http://balaiosonoro.com.br"
09     send_mail(u"Venha conhecer Balaio Sonoro", msg,\
10         'Balaio Sonoro <balaiosonoro.app@gmail.com>', [email],\
11         fail_silently=False)

```

Para visualizar seus amigos ou desfazer amizade, o usuário precisa acessar o menu “Meus Amigos”. Para desfazer a amizade, é necessário clicar no botão “Desfazer Amizade”. A Figura 17 apresenta a lista de amigos.

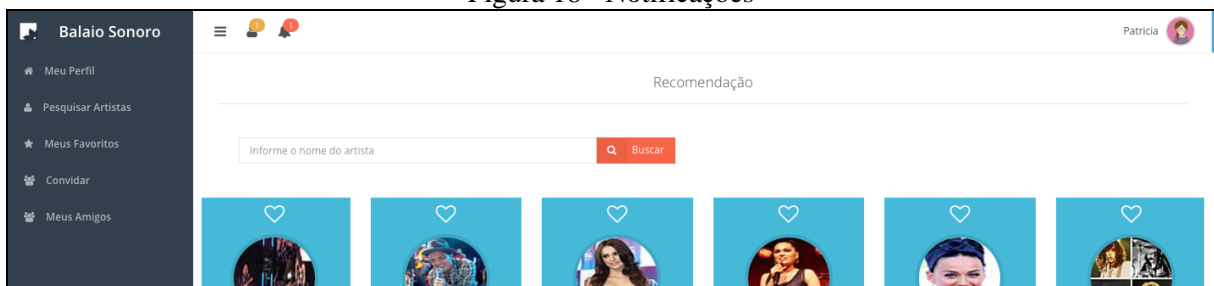
Figura 17 - Meus Amigos



3.3.2.2 Notificações

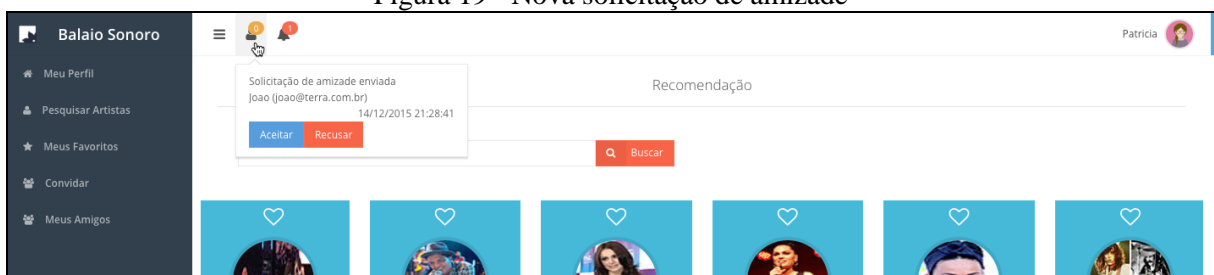
Após acessar a aplicação, o usuário pode checar suas notificações (item 1 da Figura 18), sendo estas de solicitação de amizade ou de sugestão de artista. O número que é apresentado sobre o ícone, representa a quantidade de novas notificações. A Figura 18 contém o aviso de novas notificações.

Figura 18 - Notificações



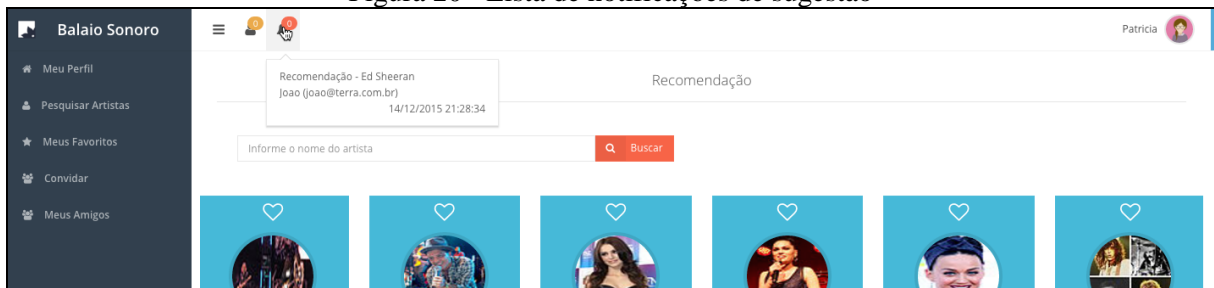
Caso a notificação seja de solicitação de amizade, o usuário tem a opção de aceitar ou recusar a nova solicitação. Se a solicitação for aceita, é criado um vínculo de amizade entre os usuários. A Figura 19 apresenta a tela com novas solicitações de amizade.

Figura 19 - Nova solicitação de amizade



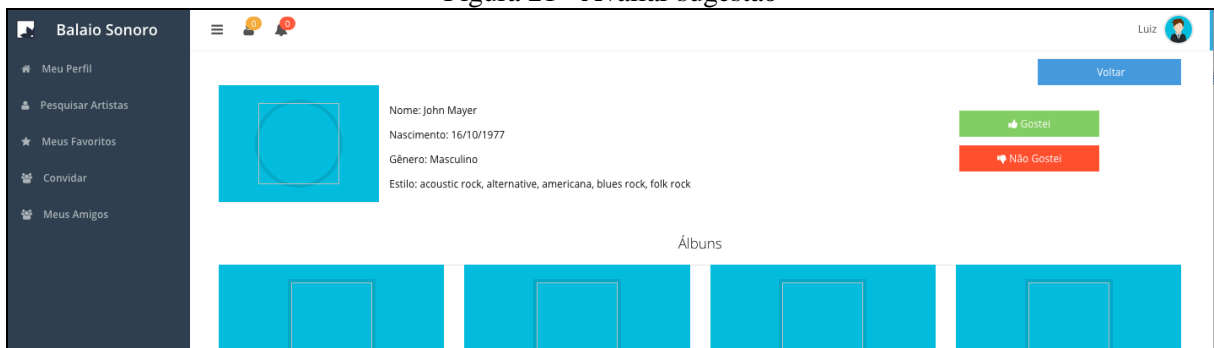
Caso a notificação seja de sugestão de artista, o usuário pode visualizar qual foi o artista recomendado juntamente com o usuário e a data em que foi realizada a recomendação. A Figura 20 apresenta a tela com as notificações de sugestão.

Figura 20 - Lista de notificações de sugestão



Quando o usuário recebe uma recomendação de outro usuário, ele tem a opção de avaliar a sugestão. Para isso, é necessário clicar sobre a sugestão e clicar no botão “Gostei” ou “Não Gostei”. A Figura 21 apresenta a tela de avaliação da sugestão.

Figura 21 - Avaliar sugestão



Se o usuário clicar sobre botão “Gostei”, é gerada uma pontuação. Onde, é validado se existe vínculo entre o usuário e o artista, caso exista, o atributo `vl_sugestao` é atualizado com o valor 0,5. Caso não exista o vínculo, ele é criado. O Quadro 7 contém o trecho de código responsável por salvar a pontuação de uma sugestão.

Quadro 7 - Trecho de código para pontuar avaliação

```

01  if Pontuacao.objects.filter(usuario=user, artista=artista).exists():
02      pontuacao = Pontuacao.objects.get(usuario=user, artista=artista)
03      pontuacao.vl_sugestao = 0.5
04      pontuacao.save()
05  else:
06      Pontuacao.objects.create(usuario=user, artista=artista, \
07                              vl_sugestao=0.5)

```

3.3.2.3 Recomendação

Ao acessar a aplicação, o usuário tem acesso a uma lista de artistas recomendados. Para gerar esta lista de recomendações, a aplicação realiza um cálculo considerando a pontuação por estilo musical dos artistas vinculados ao usuário. Para efetuar o cálculo, leva-se em consideração as avaliações, sugestões, pesquisas e definição de favorito, conforme explicado nas seções anteriores. O Quadro 8 contém um trecho de código responsável por realizar o somatório da pontuação conforme o estilo do artista.

Quadro 8 - Trecho de código da pontuação

```

01 for art in Pontuacao.objects.filter(usuario=usuario.pk).values(\
02     "artista_genero", "vl_stars", "vl_favorito", "vl_sugestao", "vl_pesquisa"):
03     artista_pontuacao.append({"genero":\
04         art["artista_genero"].replace(",","").split(), \
05         "total": ((art.get("vl_stars") or 0.0) + \
06             (art.get("vl_favorito") or 0.0)+(art.get("vl_sugestao") or 0.0) +
07             (art.get("vl_pesquisa") or 0.0))})

```

Após realizar o somatório, a aplicação agrupa todos os valores de acordo com o estilo. O Quadro 9 contém o trecho de código responsável por agrupar os estilos e calcular a sua pontuação geral.

Quadro 9 - Agrupa estilo

```

01 genero_pont = {}
02     for pont in artista_pontuacao:
03         for tag in pont["genero"]:
04             if not tag in genero_pont:
05                 genero_pont[tag] = 0
06                 genero_pont[tag] += pont["total"]

```

Após a definição da pontuação por estilo, a aplicação considera apenas os três primeiros estilos com maior pontuação para mostrar na lista que o usuário visualizará na tela inicial da aplicação. O Quadro 10 contém o trecho de código responsável por definir os estilos mais pontuados.

Quadro 10 - Trecho de código para selecionar estilos

```

01 if genero_pont:
02     max = max(genero_pont.iteritems(), key=operator.itemgetter(1))[0]
03     genero.append(max)
04     genero_pont.pop(max, None)
05 if len(genero_pont) > 2:
06     max = max(genero_pont.iteritems(), key=operator.itemgetter(1))[0]
07     genero.append(max)
08     genero_pont.pop(max, None)
09     max = max(genero_pont.iteritems(), key=operator.itemgetter(1))[0]
10     genero.append(max)
11 else:
12     max = max(genero_pont.iteritems(), key=operator.itemgetter(1))[0]
13     genero.append()

```

Os estilos musicais do usuário e os definidos a partir da pontuação, são adicionados em uma lista única, onde são removidos os estilos repetidos. Os estilos gerados a partir da pontuação são considerados mais relevantes e mostrados para o usuário na tela inicial da aplicação. O Quadro 11 apresenta o trecho de código responsável por retornar a lista de estilo.

Quadro 11 - Trecho de código estilo musical do usuário

```

01 genero = genero + genero_perfil
02 genero = list(OrderedDict.fromkeys(genero))

```

Para retornar os artistas recomendados, é utilizada a API EchoNest, onde é realizada a chamada do método `format`, que tem como parâmetro o estilo musical e retorna os artistas conforme o estilo musical. O Quadro 12 apresenta a chamada da API EchoNest enviando como parâmetro o estilo musical.

Quadro 12 - Trecho de código utilizando API EchoNest

```

01 r = requests.get("http://developer.echonest.com/api/v4/
02     genre/artists?api_key={0}&format=json&results=10&name={1}".\
03     format(settings.ECHONEST_API, genero[0]))
04 result = r.json()
05 rs = result.get('response').get('artists')

```

3.4 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os experimentos feitos com o protótipo. Na seção 3.4.1, detalha-se os testes realizados para validar o sistema de recomendação e, na seção 3.4.2, apresenta o teste de usabilidade do protótipo, onde são apresentados o questionário de perfil dos usuários, a aplicação dos testes, o questionário de usabilidade, a análise e interpretação dos dados coletados.

3.4.1 Validação do sistema de recomendação.

Para validar se a aplicação efetua as recomendações conforme critérios de pontuação e estilo musical pré-definido no cadastro do usuário, realizou-se um teste unitário, onde inicialmente selecionou-se quatro artistas, Taylor Swift, John Mayer, Aloe Blacc e Of Monsters and Men, que possuem estilos musicais diferentes. O Quadro 13 apresenta os artistas com seus respectivos estilos musicais.

Quadro 13 - Artistas e Estilos musicais

Artista	Estilo Musical
Taylor Swift	Pop, Country
John Mayer	Pop, Rock
Aloe Blacc	Hip-hop
Of Monsters and Men	Indie Folk, Folk

Depois de definidos os artistas, realizou-se algumas ações em relação aos mesmos dentro da aplicação. No caso da artista Taylor Swift foram realizadas 3 pesquisas (0,1 por pesquisa), ela foi definida como favorita (1,0 valor único por definição), avaliada com 3 estrelas (0,2 por estrela), e ninguém à sugeriu (0,5 por sugestão aceita), totalizando individualmente 1,9 pontos para os estilos Pop e Country. Os resultados da simulação e das pontuações associadas/alcançadas pelos artistas podem ser visualizadas no Quadro 14.

Quadro 14 - Artista e Pontuação

	Estilo Musical	Pesquisa	Favorito	Avaliação	Gostou da sugestão	Total de pontos
Taylor Swift	Pop, Country	0,3	1,0	0,6	0	1,9
John Mayer	Pop, Rock	0,1	0	0,6	0	0,7
Aloe Blacc	Hip-hop	0,1	0	0,2	0	0,3
Of Monsters and Men	Indie Folk, Folk	0,2	0	0,4	0	0,6

A partir da pontuação de cada artista, o sistema de recomendação agrupa a pontuação, por estilo musical, para definir quais artistas/gêneros serão apresentados para o usuário na tela inicial da aplicação. O Quadro 15 mostra o estilo musical e sua respectiva pontuação.

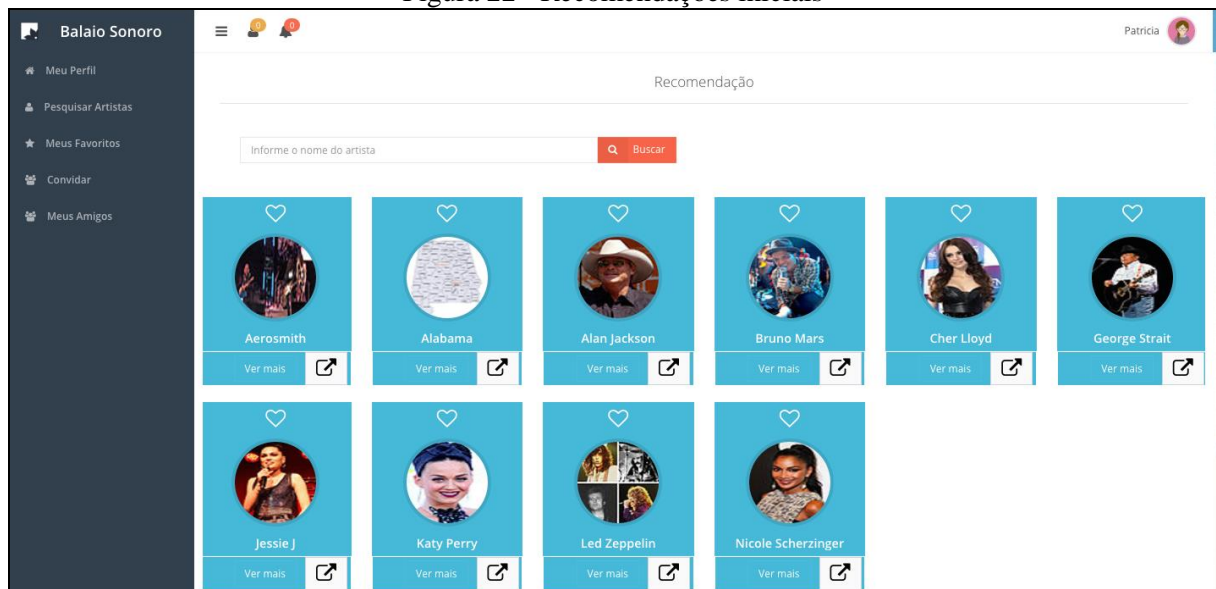
Quadro 15 - Estilo Musical e sua Pontuação

Pop	2,6
Country	1,9
Rock	0,7
Indie Folk	0,6
Folk	0,6
Hip-hop	0,3

Depois de definido os estilos musicais e suas respectivas pontuações, a aplicação seleciona os três estilos com maior pontuação. Observando a tabela acima, constata-se que os estilos mais relevantes são os: Pop, Country e Rock.

A partir disso, a aplicação realiza consultas na API do EchoNest com os estilos Pop, Country e Rock que foram definidos como os estilos mais relevantes, apresentando-os para o usuário na listagem inicial da aplicação. A Figura 22 apresenta os artistas sugeridos após a consulta a API EchoNest por estilo.

Figura 22 - Recomendações iniciais



Dando continuidade ao teste, foi aumentada a pontuação do artista Of Monsters and Men, fazendo com que seu estilo musical seja o mais bem pontuado. Para isso, foram feitas mais 4 pesquisas (+0,4), ele foi marcado como favorito (+1,0) e 2 usuários sugeriram o artista em questão, alterando a pontuação total de 0,6 para 3,0. O Quadro 16 apresenta a pontuação do artista que sofreu as alterações.

Quadro 16 - Pontuação do Artista

	Estilo Musical	Pesquisa	Favorito	Avaliação	Gostou da sugestão	Total
Taylor Swift	Pop, Country	0,3	1,0	0,6	0	1,9
John Mayer	Pop, Rock	0,1	0	0,6	0	0,7
Aloe Blacc	Hip-hop	0,1	0	0,2	0	0,3
Of Monsters and Men	Indie Folk, Folk	0,6	1,0	0,4	1,0	3,0

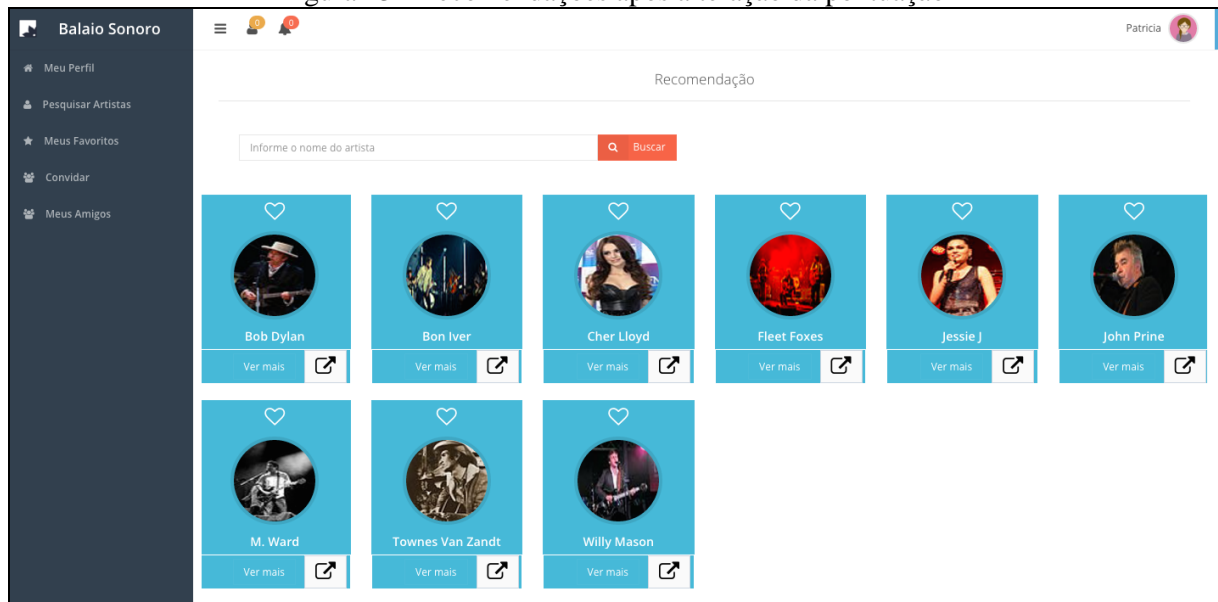
Novamente, o estilo musical é agrupado a partir da sua pontuação total. O Quadro 17 apresenta os estilos e suas respectivas pontuações após aumentada a pontuação do artista Of Monsters and Men.

Quadro 17 - Estilo e Pontuação

Indie Folk	3,0
Folk	3,0
Pop	2,6
Country	1,9
Rock	0,7
Hip-hop	0,3

Com isso, percebe-se que houve uma alteração nos estilos musicais mais relevantes, onde, os estilos musicais mais bem pontuados agora são: Indie Folk, Folk e Pop. Após a definição dos novos estilos musicais, novas sugestões são apresentadas para o usuário. A Figura 23 apresenta as novas recomendações.

Figura 23 - Recomendações após alteração da pontuação



A partir deste teste, pode afirmar que a técnica de associação e agrupamento de ações realizadas pelo usuário e utilizada no sistema de recomendação desenvolvido, funciona corretamente, pois, foram testados todos os elementos que estabelecem a pontuação dos artistas e gêneros musicais que serão sugeridos para o usuário ao entrar/navegar pela aplicação.

3.4.2 Experimento de usabilidade

O experimento de usabilidade foi realizado com quadro usuários para avaliar a eficiência e a facilidade de utilização das funcionalidades implementadas no protótipo.

3.4.2.1 Metodologia

O experimento aconteceu durante o mês de novembro por meio de teste individual com os usuários em computadores com configurações diversas. Foi fornecido a cada usuário um questionário de perfil, uma lista de tarefas e um questionário de usabilidade, que estão disponíveis no Apêndice C.

3.4.2.2 Aplicação do teste

Para iniciar a avaliação, os participantes foram orientados a preencher um questionário de perfil de usuário. Após preenchido o questionário, cada voluntário foi orientado sobre o objetivo dos testes de usabilidade e sobre o objetivo do protótipo. A lista de tarefas foi composta por trinta e duas tarefas e buscou contemplar todas as funcionalidades implementadas e disponíveis no protótipo. Ao finalizar cada tarefa, foi solicitado que o usuário informasse se a tarefa foi executada, se foi identificado algum problema ou se havia alguma observação a fazer.

O questionário de usabilidade foi composto por cinco perguntas, quatro delas fechadas e uma aberta. Para as quatro questões fechadas foram disponibilizadas a opção de colocar uma observação sobre a resposta escolhida. As perguntas procuraram obter as impressões do usuário sobre o protótipo, sobre a facilidade de utilização e a importância das funcionalidades disponíveis. Ao responder o questionário os voluntários também puderam deixar suas reclamações e/ou sugestões sobre o protótipo. Os resultados deste experimento são apresentados na próxima seção.

3.4.2.3 Análise e interpretação dos dados coletados

A primeira análise realizada foi a dos dados coletados através do questionário de perfil de usuário. No Quadro 18 são exibidos os perfis dos usuários envolvidos no teste de usabilidade.

Quadro 18 - Perfil dos usuários envolvidos no teste de usabilidade

Sexo	100% masculino
Idade	75% menos de 21 anos 25% entre 21 e 30 anos
Nível de escolaridade	100% ensino superior incompleto
Relação com a música	100% gostam e escutam frequentemente
Rede social de música	25% utilizam 75% não utilizam
Compartilha músicas com os amigos	75% não 25% sim

A partir do Quadro 18 percebe-se que os avaliadores se interessam por música e, que de alguma forma compartilham e consomem serviços musicais. Com as informações de perfil dos usuários voluntários para os testes, prosseguiu-se para a avaliação dos resultados obtidos com a lista de tarefas.

3.4.2.3.1 Análise dos resultados da lista de tarefas

No Quadro 19 são apresentados os resultados quanto às questões envolvendo o cenário de recomendação e sugestão da aplicação.

Quadro 19 - Usuários envolvidos no teste de tarefas recomendação e sugestão

	Perguntas/Respostas	Sim	Não
Recomendação	1. Analisar se a maioria dos artistas apresentados na página inicial não faz parte do seu gosto musical	75%	25%
	2. Analisar se a maioria dos artistas apresentados na página inicial faz parte do seu gosto musical.	75%	25%
Sugestão	1. Fazer uma sugestão de artista para o endereço eletrônico balaiosonoro.app@gmail.com .	100%	
	2. Acessar a lista de sugestões.	100%	
	3. Selecionar uma sugestão.	100%	
	4. Avaliar a sugestão.	100%	

Ao questionar o usuário sobre as sugestões musicais exibidas na tela inicial da aplicação, tinha-se o intuito de verificar se o sistema de recomendação estava levando em consideração apenas o perfil musical cadastrado pelo usuário. A partir das respostas dos usuários, constata-se que este critério de recomendação está funcionando corretamente.

Ao solicitar para o usuário realizar a pesquisa de três artistas que não fazem parte do seu gosto musical, tinha-se o intuito de verificar se a aplicação levaria em consideração o critério de pontuação pesquisar, pois ele influencia na recomendação de artistas. Nesta avaliação, dos quatro avaliadores, três apontaram que o sistema apresentou, em sua listagem inicial, artistas que não eram do seu gosto musical. Isso, comprova que o sistema de recomendação também está levando em consideração as pesquisas realizadas pelo usuário ao fazer o cálculo da pontuação.

Ao pedir para o usuário marcar um artista de seu gosto musical como favorito, pretendia-se que a pontuação associada ao perfil musical do usuário fosse aumentada. Neste item, todos os usuários indicaram que, na listagem inicial, começaram a aparecer artistas do seu gosto musical. Fato, que evidencia a influência do critério favorito na pontuação da recomendação.

Ao solicitar para o usuário efetuar uma sugestão musical, tinha-se o objetivo de testar se a aplicação estava notificando a existência de sugestões de outros usuários. Ao receber uma sugestão, os usuários conseguiram selecionar e posteriormente avaliar a sugestão. Neste item, todos os usuários perceberam a existência da notificação e aqueles que aceitaram a sugestão, tiveram sua pontuação de recomendações modificada.

No Quadro 20 são apresentados resultados das questões envolvendo o cenário de convite da aplicação.

Quadro 20 - Usuários envolvidos no teste de tarefas de convites

	Perguntas/Respostas	Sim	Não
Convite	1. Enviar um convite de amizade para o endereço eletrônico joao@terra.com.br.	100%	
	2. Acessar a sua lista de amigos.		
	3. Verificar se o usuário joão@terra.com.br aparece em sua lista de amigos.		
	4. Acessar a lista de solicitações de amizades.		
	5. Aceitar a solicitação de amizade do usuário balaiosonoro.app@gmail.com		
	6. Verificar se os usuários balaiosonoro.app@gmail.com e o joao@terra.com.br aparecem em sua lista de amigos.		

Ao solicitar o envio de convite para outro usuário, tinha-se o intuito de verificar se após o aceite da solicitação de amizade, o mesmo iria aparecer na lista de amigos. Neste item, todos os avaliadores apontaram que a aplicação mostrava a vinculação de amizade.

Ao pedir que o usuário acessasse a lista de solicitações de amizades pendentes e, posteriormente aceitasse uma nova solicitação, tinha-se o objetivo de validar se a lista de amigos do usuário sofreria modificações. Neste item, todos os avaliadores relataram que após aceitar o convite, o usuário começou a fazer parte de sua rede de contatos.

3.4.2.3.2 Análise quanto à usabilidade

Após analisados os resultados da lista de tarefas e perfil dos usuários, foram analisados os resultados obtidos do questionário de usabilidade aplicado. Estes resultados constam no Quadro 21.

Quadro 21 - Respostas quanto à usabilidade do protótipo

Perguntas	Respostas
7. Você gostou da interface?	100% Todas
8. Você achou fácil usar a aplicação?	75% sim
9. A partir das suas impressões da aplicação, você utilizaria o Balaio Sonoro como rede social?	75% não usariam
10. Qual a sua avaliação sobre a aplicação?	100% boa

A partir do Quadro 21 observa-se que a maioria dos usuários considera a aplicação fácil e de boa utilização. Já em relação à utilização da aplicação como rede social, a maioria dos usuários apontou que não utilizaria a aplicação. Isso, se deve ao fato de que a maioria dos usuários não usam ou participam de redes sociais voltadas para música, conforme constam nas respostas do questionário de perfil do usuário.

3.4.3 Comparação com trabalhos correlatos

Após discussões dos trabalhos correlatos, das considerações sobre sistemas de recomendação, da especificação, desenvolvimento e operacionalidade do sistema e dos resultados obtidos, apresenta-se a análise comparativa entre as características da aplicação desenvolvida com as características existentes nos trabalhos correlatos. O Quadro 22 apresenta um comparativo entre a aplicação desenvolvida e os trabalhos correlatos.

Quadro 22 - Comparativo de resultados

	Lastfm (2014)	Zanette (2008)	Buzatto (2013)	Sistema Proposto
Área de atuação	Música	Filme	Eventos	Música
Conceito	Folksomia	Rede de confiança	Treemaping	
Técnica de recomendação	Recomendação colaborativa	Recomendação colaborativa	Não Informado	Recomendação híbrida
Plataforma	Web	Web	Web	Web

A partir do Quadro 22, pode-se observar que o sistema proposto contempla a maioria das características existentes nos trabalhos correlatos, sendo que, a maior diferença do sistema desenvolvido é a técnica de recomendação escolhida, onde no sistema proposto foi adotada como técnica de recomendação a filtragem híbrida, por fortalecer as vantagens e minimizar as principais desvantagens de seus dois componentes, a filtragem baseada em conteúdo e a filtragem colaborativa.

Todos os trabalhos descritos utilizam a plataforma web para desenvolvimento de seus trabalhos, isso pelo fato de que sistemas de recomendação analisam e consideram as ações realizadas pelo usuário na web.

Contudo, pode-se verificar que o sistema desenvolvido atende a maioria das características enumeradas.

4 CONCLUSÕES

Esse trabalho apresentou o desenvolvimento de uma aplicação de recomendação de música utilizando a técnica de recomendação híbrida, tendo como principal objetivo a recomendação de artistas a partir das ações realizadas pelo usuário dentro da aplicação.

A aplicação foi implementada em uma arquitetura cliente-servidor. A camada cliente foi desenvolvida utilizando a linguagem de programação Javascript com o auxílio do HTML5, CSS3 e Bootstrap. Foi utilizada a base de dados da MusicBrainz para obter a lista de artista e suas respectivas informações e a API do EchoNest para consultar os artistas mais comentados na semana e para fazer recomendações. A camada servidor foi implementada com a linguagem de programação Python utilizando o framework Django.

Os resultados obtidos a partir de testes com usuários mostraram-se satisfatórios com um total de 75% de usuários que acham que a aplicação atende o objetivo de recomendar artistas. Também é possível citar que 100% dos usuários que testaram a aplicação a consideraram boa.

A principal limitação da aplicação é apresentar falhas ao avaliar um usuário. Outra limitação apresentada é a de não permitir buscas por artistas brasileiros, apenas estrangeiros.

A partir da aplicação desenvolvida foram apresentadas técnicas e ferramentas que podem servir para o desenvolvimento de novas aplicações. Também foi desenvolvido um algoritmo que é capaz de realizar recomendações com base em uma pontuação. Finalmente, a aplicação desenvolvida pode ser utilizada por usuários que gostam de música e de conhecer novos artistas.

4.1 EXTENSÕES

Algumas extensões possíveis para esse trabalho são:

- a) permitir que o usuário se cadastre na aplicação através de uma API de rede social (Facebook; Twitter; Google);
- b) permitir a busca de artistas brasileiros;
- c) permitir escutar as músicas dos artistas;
- d) permitir que o usuário crie sua *playlist* de música. Com isso, atribuir uma pontuação ao artista da música;
- e) apresentar mais informações do artista (agenda de shows; últimos *tweets* do Twitter; últimas fotos do Instagram);
- f) criar aplicativo (IOs e/ou Android).

REFERÊNCIAS

BALABANOVIĆ, Marko; SHOHAM, Yoav. Fab: content-based, collaborative recommendation. [S.l.], 1997. **Communications of the ACM**. Disponível em: <http://www.ischool.utexas.edu/~i385q/readings/Balabanovic_Shoham-1997-Fab.pdf>. Acesso em: 31 mar. 2014.

BALABANOVIĆ, Marko; SHOHAM, Yoav. Fab: content-based, collaborative recommendation. [S.l.], 1997. **Communications of the ACM**, University of Texas, Austin, 1997. Disponível em: <http://www.ischool.utexas.edu/~i385q/readings/Balabanovic_Shoham-1997-Fab.pdf>. Acesso em: 31 mar. 2014.

BURKE, Robin. **Hybrid recommender systems: Survey and experiments**. User Modeling and User Adapted Interaction, Department of Information Systems and Decision Sciences, California State University, v.12, n. 6, p. 331–370, 2002. Disponível em: <<http://josquin.cti.depaul.edu/~rburke/pubs/burke-umuai02.pdf>>. Acesso em: 1 abr. 2014.

BUSATTO, Claudio J. S. **O que tá valendo? Um sistema web de recomendação de eventos**, Universidade Federal de Ciências da Saúde de Porto Alegre, Porto Alegre, 2013. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/86266/000910130.pdf?sequence=1>> Acesso em: 25 mar. 2014.

CASTRO, Eliane. B. **Marketing de Varejo**. Curitiba: Ibpex, 2007. 182p.

CAZELLA, Sílvio César; NUNES, Maria Augusta S. N.; REATEGUI, Eliseo Berni. A Ciência da Opinião: Estado da arte em Sistemas de Recomendação. XXX Congresso da Sociedade de Computação: Computação Verde – desafios científicos e tecnológicos, Puc-Minas, **Procedimentos ...** Belo- Horizonte – MG, 2010. Disponível em: <http://www.do.ufgd.edu.br/WillianAmorim/TAIC022013_arquivos/Artigo2.pdf>. Acesso em: 13 abr. 2014.

ECHONEST. disponível em <<http://the.echonest.com/>>. Acesso em: 23 nov. 2015.

HERLOCKER, Jonathan et al. Explaining collaborative filtering recommendations. In: ACM CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2000, Philadelphia. **Proceedings ...** Philadelphia: 2000. p.241-250.

LASTFM. disponível em <<http://www.lastfm.com>>. Acesso em: 10 abr. 2014.

LORENZI, Fabiana. **Uma abordagem multiagente de recomendação baseada em suposições e confiança para cenários dinâmicos**, Universidade Federal de Ciências da Saúde de Porto Alegre, Porto Alegre, 2010. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/26969/000762251.pdf?sequence=1>> Acesso em: 3 abr. 2014.

MACHADO, Marcos O. **RecImóveis**: Um Sistema de Recomendação para auxiliar consumidores na decisão de compra de imóveis, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2006. Disponível em: <http://www.unisinos.org.br/inf/files/melhores_tccs/2006_2/tcc_marcosdeoliveiramachado.pdf>. Acesso em: 20 mar. 2014.

MEDEIROS, Igor R. G. Estudo sobre Sistemas de Recomendação Colaborativos, Universidade Federal e Pernambuco, Recife, 2013. Disponível em: <<http://www.cin.ufpe.br/~tg/2012-2/irgm.pdf>>. Acesso em: 25 out. 2015.

MUSICBRAINZ. disponível em < <http://musicbrainz.org/>>. Acesso em: 23 nov. 2015.

REATEGUI, Eliseo B.; CAZELLA, Sílvio C. **Sistemas de Recomendação**. XXV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2005, São Leopoldo. Anais do Encontro Nacional de Inteligência Artificial. São Leopoldo, 2005, p. 306- 348.

RESNICK, P.; VARIAN, H. R. **Recommender Systems**. Communications of the ACM, Berkeley, v.40, n. 3, p. 56-58, 1997.

SALTON, Gerard. **Automatic text processing**: The Transformation, Analysis, and Retrieval of Information by Computer. Boston: Addison-Wesley, 1989.

SARWAR, Badrul. et al. Item-based collaborative filtering recommendation algorithms. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 10., 2001, Hong Kong. **Proceedings** ... New York: ACM, 2001.

SCHAFER, Ben J. ; KONSTAN, Joseph; RIEDL, John. Recommender systems e-commerce. In: ACM CONFERENCE ON ELECTRONIC COMERCE, 1., 1999. Denver. **Proceedings** ... New York, 2000.

SEGARAN, Toby. **Programando a inteligência coletiva**. Rio de Janeiro: Alta Books, 2008. 282p.

OLIVEIRA, Leonardo G. **Sistema de recomendação de meios de hospedagem baseado em filtragem colaborativa e informações contextuais**, Universidade Federal de Santa Catarina, Florianópolis, 2007. Disponível em: <<http://btd.egc.ufsc.br/wp-content/uploads/2010/06/Leonardo-Gomes-de-Oliveira.pdf>>. Acesso em: 30 mar. 2014.

ZANETTE, Leonardo R. **Sistema de recomendação de itens baseado na rede de confiança do usuário**, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008. Disponível em: <http://www.nce.ufrj.br/ginape/publicacoes/dissertacoes/d_2008/d_2008_leonardo_rosa_zanette.pdf> Acesso em: 25 mar. 2014.

APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos principais casos de uso.

Quadro 23 - UC1 Fazer Login

UC01	Fazer <i>login</i>
Descrição	Permitir que o usuário faça o <i>login</i> na aplicação utilizando seu email e senha.
Ator	Usuário
Pré-condição	Usuário possuir um email.
Fluxo principal	<ol style="list-style-type: none"> Usuário preenche seu login e senha; A aplicação valida os dados; Sistema direciona o usuário para a página inicial.
Fluxo alternativo (a)	<ul style="list-style-type: none"> Nome de usuário e/ou senha inválido(s)
Pós-condição	Usuário tem acesso as informações.

Quadro 24 - UC02 Manter Perfil do Usuário

UC02	Manter perfil do usuário.
Descrição	Permitir que o usuário possa alterar as informações do seu perfil.
Ator	Usuário
Pré-condição	Usuário deve estar logado no sistema.
Fluxo principal	Usuário acessa o menu “Perfil do Usuário“
Fluxo alternativo (a)	<ul style="list-style-type: none"> Campo(s) obrigatório(s) não preenchidos(s). Alerta com mensagem “Favor preencher todos os campos obrigatórios“ é mostrada.
Cenário – Edição	<ol style="list-style-type: none"> Sistema mostra o perfil do usuário; Sistema mostra as informações para edição; Usuário realiza as alterações necessárias; Usuário seleciona a opção alterar; Sistema altera o registro e apresenta a mensagem “Alteração concluída com sucesso”.
Pós-condição	Cadastro atualizado.

Quadro 25 - UC03 Listar informações

UC03	Listar informações de seu interesse.
Descrição	Permitir que o usuário visualize seus artistas favoritos.
Ator	Usuário
Pré-condição	Usuário deve estar logado no sistema.
Fluxo principal	Usuário acessa o menu “Meus Favoritos“
Cenário - Edição	<ol style="list-style-type: none"> Aplicação mostra todos os artistas favoritos; Usuário clica no ícone coração e o artista é removido da lista de favoritos; Aplicação atualiza lista de favoritos.

Quadro 26 - UC04 Buscar artistas

UC04	Buscar artistas de seu interesse.
Descrição	O usuário pode buscar informações de seu interesse.
Ator	Usuário
Fluxo Principal	Usuário deve indicar o nome do artista de seu interesse.

Quadro 27 - UC05 Recomendar artistas

UC05	Recomendar artistas para outros usuários.
Descrição	O usuário pode recomendar artistas para outros usuários.
Ator	Usuário.
Fluxo Principal	A partir da navegação e interação do usuário com o sistema, ele pode recomendar artistas que atendam ao gosto musical de outros usuários.

Quadro 28 - UC06 Avaliar itens sugeridos

UC06	Avaliar itens sugeridos por outros usuários.
Descrição	O usuário pode avaliar itens sugeridos por outros usuários.
Ator	Usuário
Fluxo Principal	O usuário avalia se o item sugerido por outro usuário está de acordo com suas preferências musicais.

Quadro 29 - UC07 Construir base de conhecimento

UC07	Construir uma base de conhecimento a partir das ações realizadas.
Descrição	O sistema deve armazenar informações (perfil, artistas favoritos, itens sugeridos) para montar a base de conhecimento para realizar recomendações.
Ator	Sistema
Fluxo Principal	Ao acessar na aplicação o sistema deve armazenar e atualizar informações necessárias e atualizar o usuário com novas sugestões.

Quadro 30 - UC08 Recomendar artistas

UC08	Recomendar artistas a partir do perfil do usuário e da associação de informações contidas na base de conhecimento.
Descrição	O sistema deve armazenar informações (perfil, artistas favoritos, itens sugeridos) para montar a base de conhecimento para realizar recomendações.
Ator	Sistema
Fluxo Principal	Ao acessar na aplicação o sistema deve armazenar e atualizar informações necessárias e atualizar o usuário com novas sugestões.

Quadro 31 - UC09 Notificar recomendações

UC09	Notificar a existência de novas recomendações.
Descrição	O sistema deve notificar o usuário quando existirem novas recomendações.
Ator	Sistema.
Fluxo Principal	Ao acessar a aplicação o sistema deve notificar o usuário sobre a existência de novas recomendações.

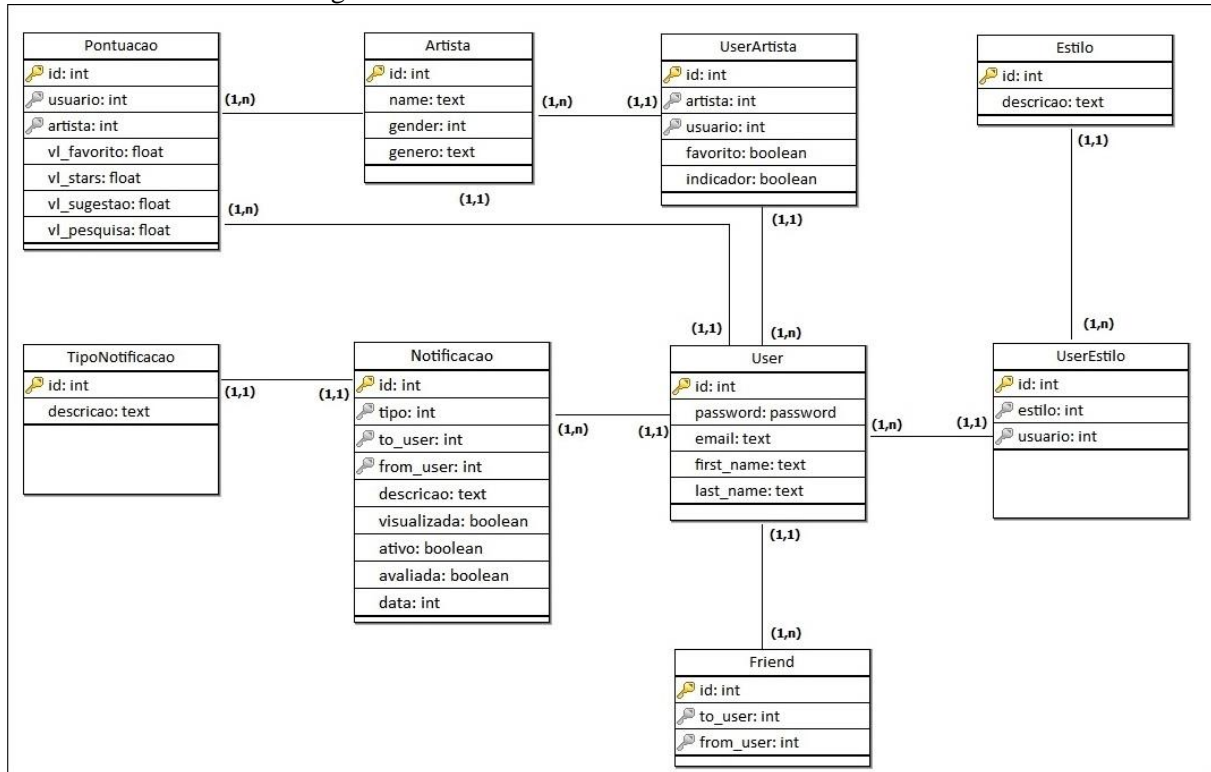
Quadro 32 - UC10 Notificar solicitações de amizade

UC10	Notificar a existência de novas solicitações de amizade.
Descrição	O sistema deve notificar o usuário quando existirem novas solicitações de amizade.
Ator	Sistema.
Fluxo Principal	Ao acessar a aplicação o sistema deve notificar o usuário sobre a existência de novas solicitações de amizade.

APÊNDICE B – Dicionário de Dados

O Modelo de Entidade-Relacionamento (MER) é um diagrama que descreve o modelo de dados de um sistema com alto nível de abstração. A Figura 24 apresenta o MER com as entidades que serão persistidas no banco de dados do sistema.

Figura 24 - Modelo de Entidade de Relacionamento



A seguir é apresentada uma breve descrição das entidades utilizadas para o desenvolvimento da aplicação:

- user**: entidade responsável por armazenar informações referente aos usuários cadastrados na aplicação;
- friend**: entidade responsável por armazenar informações referente aos amigos do usuário na aplicação;
- estilo**: entidade responsável por armazenar os estilos cadastrados na aplicação;
- userEstilo**: entidade responsável por armazenar informações dos usuários e seus respectivos estilos;
- artista**: entidade responsável por armazenar informações referente aos artistas cadastrados na aplicação;
- userArtista**: entidade responsável por armazenar o vínculo do usuário com o artista;
- tipoNotificacao**: entidade responsável por armazenar os tipos de notificações

cadastradas na aplicação;

- h) `notificacao`: entidade responsável por armazenar todos os tipo de notificações geradas na aplicação;
- i) `pontuacao`: entidade responsável por armazenar todas as pontuações obtidas pelo usuário para o artista.

APÊNDICE C – Roteiro e Questionário de Avaliação de Usabilidade

Neste apêndice constam os questionários de perfil do usuário e usabilidade, assim como o roteiro de tarefas que deveriam ser seguidas. O

Quadro 33 demonstra o questionário do perfil do usuário. O Quadro 34 demonstra a lista de tarefas que os usuários deveriam reproduzir. No Quadro 35 consta o questionário de usabilidade da aplicação.

Quadro 33 - Questionário de perfil do usuário

QUESTIONÁRIO DE PERFIL DO USUÁRIO	
Observação: Todo o questionário é anônimo e confidencial.	
Sexo:	
<input type="checkbox"/> Feminino <input type="checkbox"/> Masculino	
Idade:	
<input type="checkbox"/> 15 – 20	<input type="checkbox"/> 41 – 50
<input type="checkbox"/> 21 – 30	<input type="checkbox"/> Mais de 50
<input type="checkbox"/> 31 – 40	
Escolaridade:	
<input type="checkbox"/> Ensino Fundamental Completo	
<input type="checkbox"/> Ensino Fundamental Incompleto	
<input type="checkbox"/> Ensino Médio Completo	
<input type="checkbox"/> Ensino Médio Incompleto	
<input type="checkbox"/> Superior Incompleto	
<input type="checkbox"/> Superior Completo	
Gosta de música?	
<input type="checkbox"/> Sim <input type="checkbox"/> Não	
Você escuta frequentemente as mesmas músicas?	
<input type="checkbox"/> Sim	
<input type="checkbox"/> Não	
O que você utiliza para escutar música?	
<input type="checkbox"/> Rádio	
<input type="checkbox"/> Computador	
<input type="checkbox"/> Celular	
<input type="checkbox"/> Internet	
Você participa de alguma rede social de música?	
<input type="checkbox"/> Sim. Qual? <input type="checkbox"/> Last.fm <input type="checkbox"/>	
<input type="checkbox"/> Não.	
Você teria interesse de compartilhar gostos/informações musicais com seus amigos?	
<input type="checkbox"/> Sim	
<input type="checkbox"/> Não	

Quadro 34 - Lista de tarefa

INSTRUÇÕES

Com esse questionário buscamos avaliar a utilização da aplicação de recomendação de música. Com base nas instruções de uso da aplicação, solicitamos que prossiga nos testes conforme as orientações abaixo.

Um dos objetivos do protótipo é avaliar a recomendação de música com base no gosto musical do usuário permitindo assim que novos artistas sejam apresentados para o usuário. Outro objetivo é disponibilizar aos usuários uma rede social musical permitindo que os usuários interajam entre si em torno da música.

Você pode utilizar o protótipo livremente por um período de 5 a 10 minutos para se ambientar. Ao finalizar, solicitamos que prossiga nos testes conforme as orientações abaixo.

Lista de tarefas a serem executadas:**Tarefa nº1:** Cadastro da aplicação.

A sua primeira tarefa é realizar o seu cadastro na aplicação, informando nome, email e perfil musical.

A tarefa foi executada? () Sim () Não

Tarefa nº2: Acessar a aplicação.

Sua segunda tarefa é acessar a aplicação, para isso é necessário fazer o login.

A tarefa foi executada? () Sim () Não

Tarefa nº3: Verificar recomendação.

Sua terceira tarefa é analisar se os artistas apresentados na página inicial se aproximam do seu gosto musical.

A tarefa foi executada? () Sim () Não

Tarefa nº4: Realizar pesquisa na aplicação.

Sua quarta tarefa é pesquisar por artistas que você NÃO gosta.

A tarefa foi executada? () Sim () Não

Tarefa nº5: Sair da aplicação.

Sua quinta tarefa é sair da aplicação, para isso é necessário fazer logout.

A tarefa foi executada? () Sim () Não

Tarefa nº6: Acessar a aplicação.

Sua sexta tarefa é acessar a aplicação novamente, para isso é necessário fazer o login.

A tarefa foi executada? () Sim () Não

Tarefa n°7: Verificar recomendação.

Sua sétima tarefa é analisar se a maioria dos artistas apresentados na página inicial não faz parte do seu gosto musical.

A tarefa foi executada? () Sim () Não

Tarefa n°8: Pesquisar na aplicação.

Sua oitava tarefa é pesquisar por dois artistas que fazem parte do seu gosto musical.

A tarefa foi executada? () Sim () Não

Tarefa n°9: Sair da aplicação.

Sua nona tarefa é sair da aplicação, para isso é necessário fazer logout.

A tarefa foi executada? () Sim () Não

Tarefa n°10: Acessar a aplicação.

Sua décima tarefa é acessar a aplicação novamente, para isso é necessário fazer o login.

A tarefa foi executada? () Sim () Não

Tarefa n°11: Verificar recomendação.

Sua décima primeira tarefa é analisar se a maioria dos artistas apresentados na página inicial faz parte do seu gosto musical.

A tarefa foi executada? () Sim () Não

Tarefa n°12: Favorito.

Sua décima segunda tarefa é marcar um artista como favorito.

A tarefa foi executada? () Sim () Não

Tarefa n°13: Sair da aplicação.

Sua décima terceira tarefa é sair da aplicação, para isso é necessário fazer logout.

A tarefa foi executada? () Sim () Não

Tarefa n°14: Acessar a aplicação.

Sua décima quarta tarefa é acessar a aplicação novamente, para isso é necessário fazer o login.

A tarefa foi executada? () Sim () Não

Tarefa n°15: Verificar recomendação.

Sua décima quinta tarefa é analisar se a maioria dos artistas apresentados na página inicial faz parte do seu gosto musical.

A tarefa foi executada? () Sim () Não

Quadro 35 - Questionário de usabilidade

QUESTIONÁRIO DE USABILIDADE

Você gostou da interface (cor, botões, etc...)?

- Sim
 Não
-

Você achou fácil usar a aplicação?

- Sim
 Não
-

A partir das suas impressões da aplicação, você utilizaria o Balaio como rede social?

- Sim
 Não
-

Qual a sua avaliação sobre a aplicação?

- Ótima
 Boa
 Regular
 Ruim
 Péssima
-

Considerações sobre a o uso da aplicação?
