

Reprodutor Musical em Java: Explorando as Possibilidades do Swing

Bárbara Pereira Schneider¹, Franco Vivan Michelin¹, Pedro Bampi Gollo¹

¹Área de Conhecimento de Ciências Exatas e Engenharia –
Universidade de Caxias do Sul (UCS) – Caxias do Sul – RS – Brasil

Abstract. *This article will address the development process of the Pulse application, which aimed to meet the demand for an MP3 music player that prioritizes simplicity and usability.*

Resumo. *No presente artigo será abordado o processo de desenvolvimento da aplicação Pulse, que teve como objetivo atender à demanda por um reprodutor de músicas MP3 que priorize a simplicidade e a usabilidade.*

1. Introdução

A música é parte essencial da vida humana, e muitas pessoas têm coleções de álbuns em formato MP3 em seus computadores. No entanto, encontrar um reprodutor de música que seja fácil de usar, rápido e não sobrecarregado com recursos desnecessários pode ser um desafio.

Neste artigo será descrita a atividade realizada durante a disciplina Projeto Temático I do curso de Análise e Desenvolvimento de Sistemas (ADS), que teve como proposta desenvolver uma aplicação relacionada ao tema exposto anteriormente. Dessa forma, a aplicação denominada Pulse consiste em um aplicativo de software desktop desenvolvido em Java (Java™ SE Development Kit 20.0.2 - JDK 20.0.2) projetado para tornar a reprodução de músicas uma experiência simples e eficiente.

Sendo assim, na primeira parte do artigo será exposto o processo de compreensão do problema. Na segunda parte serão apresentados os requisitos funcionais e não funcionais, assim como o diagrama dos casos de uso. Em seguida, será feita a discussão sobre a modelagem do software. A quarta parte consiste no detalhamento do funcionamento da aplicação e apresentação do protótipo visual. E, por fim, será detalhada a solução proposta, os recursos utilizados e os resultados obtidos.

2. Compreensão do Problema

Para a condução deste projeto, adotou-se metodologias de engenharia de software que desempenharam papel crucial na compreensão do problema e elaboração da documentação essencial para orientar a fase de desenvolvimento do projeto.

A proposta do trabalho perpassa pela aplicação de processos de desenvolvimento de software aplicados à consolidação deste em um contexto desktop. Teve como escopo estruturante, etapas de análise de negócio, análise de requisitos, análise de sistema, projeto do software, bem como a definição da sua arquitetura.

Como a tarefa solicitava um aplicativo que girasse em torno da temática musical definiu-se que o produto final seria um reprodutor de músicas. Os requisitos de um reprodutor musical podem ser vários, mas a parte elementar dele é reproduzir músicas porque

sem essa funcionalidade ele não cumpre o seu papel. Essa é a premissa de classificação de requisitos do tipo funcional.

Consta nos pré-requisitos do trabalho que o aplicativo deve ser voltado para desktop, não havendo portanto necessidade de comunicação online. A definição por si só do que o produto não deveria ser, orientou o desenvolvimento na mesma proporcionalidade da definição do que ele deveria ser.

O modelo de Prototipagem facilita a criação de uma compreensão das necessidades do produto de software ainda na fase de análise e especificação de requisitos, e foi escolhido entre outros por este critério. Tal modelo, associado à entregas incrementais, contribui para um desenvolvimento mais assertivo ao passo em que gera um software inicialmente descartável e a seguir, o transforma em produto. Opera em um fluxo rápido, o que atende às necessidades presentes no contexto em que um semestre é o tempo limite do trabalho de desenvolvimento.

Sendo o protótipo um modelo preliminar, referências de ferramentas já desenvolvidas, adequadas em termos de escala, escopo e aplicações, aceleraram a elicitação de requisitos. Esta, por sua vez, auxilia na visualização da 'definition of done' - que é a definição do que se configura como o produto a ser alcançado. Partiu-se do pressuposto de não reinventar o que já foi consolidado e portanto, está estabelecido quanto à usabilidade, sendo esta naturalmente reconhecida pelos usuários.

Qualquer princípio ordenador deve nos interessar muito mais por sua substância do que por sua aparência, já que esta é caracterizada pela impermanência e é resultado das circunstâncias de cada projeto [Mahfuz 2007].

No início do processo foi alinhado que seria dado um peso maior ao background da aplicação. Ou seja, que esta funcionasse acima de todos os outros requisitos. E que a entrega do que foi solicitado, mensurado e documentado pudesse ser efetiva, com um escopo claro a ser seguido. Desenvolver a aplicação de um reprodutor musical em linguagem Java trouxe a complexidade como um fator de escopo limitante de projeto. A partir desse conceito, delineou-se outro pré-requisito: deve ser um software de visualização simples e minimalista o suficiente para que os esforços fossem concentrados no indispensável.

Ao definir-se que a interface conteria o essencial de um reprodutor musical pesquisou-se exemplos desses aplicativos e, dos diversos produtos no mercado que possibilitam a reprodução musical, procurou-se extrair dados de usabilidade desejados.

Uma referência foi escolhida para fazer o que a própria palavra comunica: referenciar. O ato de referenciar gera, para o analista, sentido de espaço e orientação de um objeto em relação ao outro - possibilidade de retorno, de refazer aferências, medir, aferir e gerar métricas. Nos momentos de dúvidas, sejam elas de escopo, relações ou definições de regras de negócio que ocorreram durante o desenvolvimento do trabalho, recorreu-se ao aplicativo referência o qual, por sua vez, contribuiu para criar direcionamentos consistentes.

Foi escolhido como modelo o aplicativo Groove, que é o Reprodutor Multimídia padrão do Windows. Outras referências também foram analisadas, como o Apple Music, o Amazon Music e o Wizard.io. A partir dessas referências, foram feitos testes e análises

da interface gráfica de todos, para então entender quais ofereciam uma interação mais facilitada, sem redundância de elementos e com clareza visual para que tornasse mais simples a implementação em Java. Nesta etapa, compreendeu-se quais elementos estão presentes em um reprodutor musical e então os casos de uso foram escritos. O Groove se mostrou uma referência pertinente por ser o mais próximo, entre os outros, de uma aplicação desenvolvida que é voltada para desktop e sem conectividade externa. Sua interface possui poucos elementos visuais em contrapartida aos outros aplicativos analisados e foi a partir dela que o Reprodutor Musical Pulse foi desenhado. Para fins de aprendizado, foi incluída a funcionalidade de login, que não está presente na referência.

3. Levantamento dos Requisitos e Casos de Uso

Com o objetivo de gerenciar requisitos e os transformar em produto, o processo de desenvolvimento do software foi apoiado em casos de uso, ou seja, orientado às necessidades do usuário. Foram gerados diversos artefatos ao longo do processo, que foram sendo revisitados à medida em que a compreensão dos problemas e de suas soluções possíveis avançava, em um processo incremental e contínuo de desenvolvimento.

O levantamento de requisitos (também chamado elicitación de requisitos) combina elementos de solução de problemas, elaboração, negociação e especificação. Para estimular uma abordagem colaborativa e orientada a equipes em relação ao levantamento de requisitos, os envolvidos trabalham juntos para identificar o problema, propor elementos da solução, negociar diferentes abordagens e especificar um conjunto preliminar de requisitos da solução [Pressman and Maxim 2016].

A partir da análise do referencial, definiram-se cenários para o protótipo do software, que seriam traduzidos para requisitos funcionais e então entregues de forma iterativa e incremental. O usuário deve poder se cadastrar e então, realizar seu login no sistema (RF01). O usuário deve poder importar arquivos de música que estão no seu computador para a biblioteca de músicas do aplicativo (RF02). A partir disso, o usuário deve poder editar ou remover músicas da biblioteca do aplicativo (RF03). O usuário deve poder criar, editar ou remover playlists (RF04). O usuário deve poder adicionar músicas da biblioteca à essas playlists, assim como removê-las (RF05). O usuário deve poder reproduzir músicas, bem como controlar a execução das músicas (RF06). O usuário deve poder buscar músicas na biblioteca (RF07). O usuário deve poder adicionar e remover músicas da fila de reprodução, tanto a partir da biblioteca quanto de suas playlists (RF08). O sistema deve permitir a reprodução contínua da fila de reprodução, passando automaticamente para a próxima música quando a música atual terminar (RF09). Até este ponto, os requisitos funcionais foram definidos. À medida que os requisitos são reunidos, uma visão geral das funções e características começa a se materializar [Pressman and Maxim 2016].

Quanto aos requisitos não funcionais, foram definidos a partir de questionamentos quanto aos desejos da equipe quanto ao software e, mediante discussões e deliberações, definiram-se pontos chave. A interface do usuário deve ser intuitiva (RNF01). O aplicativo deve ser compatível com o sistema operacional Windows (RNF02). A interface do aplicativo deve ser responsiva (RNF03). O aplicativo deve ser capaz de reproduzir o formato de áudio MP3 (RNF04). O aplicativo não deve depender de conexão com a internet (RNF05). O aplicativo será desenvolvido na linguagem de programação Java (RNF06).

Um caso de uso é um documento narrativo que descreve a sequência de eventos

de um ator que usa um sistema para completar um processo [Jacobson et al. 1999]. Os casos de uso e os requisitos foram delineados em um processo contíguo. Nas primeiras fases do desenvolvimento alterações foram feitas em ambos, visto que sua existência é interdependente e a explicitidade desta relação trouxe clareza para os envolvidos quanto aos desafios que o negócio deveria contemplar, antes de chegar à granularidade do código em si. Será apresentado o diagrama de casos de uso para ilustrar as principais funcionalidades do sistema. Os detalhes específicos de cada caso de uso podem ser encontrados no apêndice.

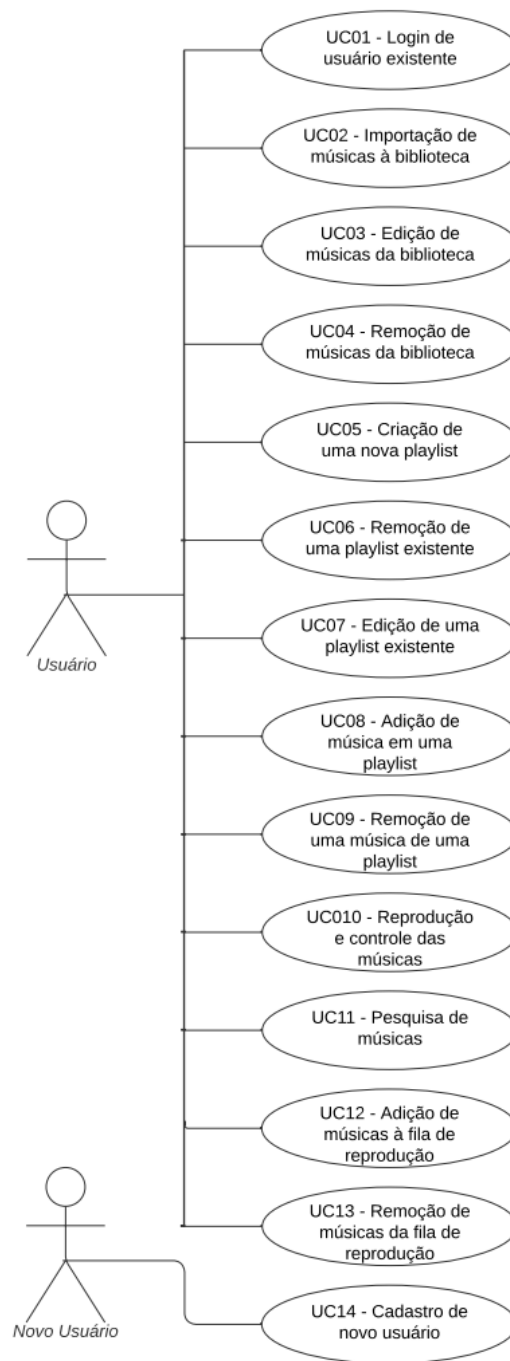


Figura 1. Diagrama dos Casos de Uso

4. Modelagem do Software

A arquitetura de software é uma abordagem para projetar e estruturar sistemas de software de forma a permitir que eles sejam flexíveis, testáveis, escaláveis e facilmente mantidos ao longo do tempo [Martin 2017]. Essa disciplina essencial no desenvolvimento de software busca criar uma base sólida para o sistema, proporcionando uma estrutura que não apenas atenda aos requisitos funcionais, mas também otimize a qualidade do código, a eficiência do desenvolvimento e a adaptabilidade a mudanças futuras.

O padrão de arquitetura de software escolhido para a modelagem do sistema foi o MVC (Model-View-Controller). Nesta arquitetura, a lógica do programa é dividida em três componentes principais: o Modelo (Model), a Visão (View) e o Controlador (Controller). Cada componente tem uma responsabilidade específica e trabalha em conjunto para criar uma aplicação modular e de fácil manutenção. O objetivo do MVC é fornecer uma estrutura clara para separar a lógica de negócios, a apresentação dos dados e a interação do usuário [Gamma et al. 1994].

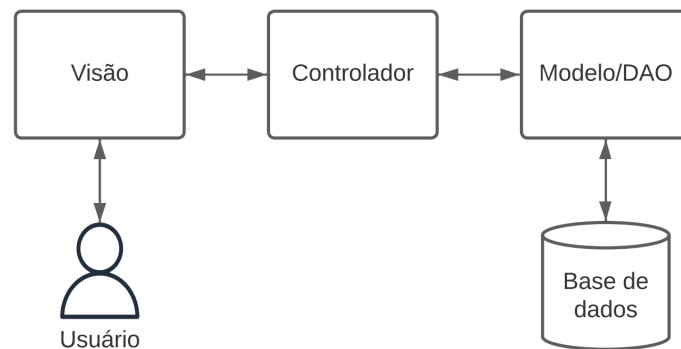


Figura 2. Diagrama da Arquitetura MVC

O Modelo é responsável por estruturar e organizar as informações de maneira coesa, fornecendo uma representação abstrata que facilita a interação com o restante da aplicação. Ele define como os dados são armazenados, recuperados e manipulados, garantindo consistência e integridade ao longo do ciclo de vida da aplicação.

Dentro do modelo do aplicativo, encontramos classes fundamentais como Usuário, Música e Playlist. Estas classes são estruturadas para conter atributos específicos que refletem as características essenciais relacionadas aos usuários, músicas e playlists no contexto do sistema.

Para operacionalizar uma interação eficiente com o banco de dados, foi implementado o padrão DAO (Data Access Object). O DAO atua como uma ponte entre a lógica de negócios do Modelo e a camada de persistência no banco de dados. Com isso, é possível isolar as operações relacionadas ao banco de dados, proporcionando uma abstração que simplifica o acesso e a manipulação dos dados.

A Visão exerce um papel crucial ao apresentar os dados atualizados do Modelo e receber as interações do usuário. Funcionando como a interface através da qual o usuário interage com o sistema, a Visão notifica o Controlador sobre as ações do usuário e, simultaneamente, é informada das mudanças no Modelo.

No aplicativo, a visão é representada pelas classes de interface gráfica desenvolvidas em Swing, como a LoginGUI, PrincipalGUI, e assim por diante. Essas classes são responsáveis por apresentar os dados ao usuário de uma maneira visualmente atraente e interativa, permitindo que o mesmo possa interagir com a aplicação.

Já o Controlador atua como intermediário entre a Visão e o Modelo. O Controlador recebe as entradas do usuário da Visão e as traduz em ações a serem executadas no Modelo [Martin 2003]. Ele também recebe notificações do Modelo e atualiza a visão de acordo.

A implementação do Controlador é evidenciada por meio das classes controladoras, como UsuarioController, MusicaController e PlaylistController. Estes controladores desempenham um papel crucial na organização e execução das operações relacionadas às entidades específicas do sistema.

5. Funcionamento do Software e Protótipo Visual

Por meio de uma interface de usuário limpa e de fácil navegação, o Pulse proporciona uma experiência descomplicada na reprodução de músicas. Controles tradicionais, como play, pause, avançar e retroceder, garantem controle total sobre a experiência auditiva, enquanto informações essenciais como título da música e duração enriquecem cada faixa reproduzida. Adicionalmente, o Pulse simplifica a criação e o gerenciamento de playlists personalizadas, contribuindo para uma organização eficaz das preferências musicais dos usuários. Não menos importante, a inclusão da fila de reprodução oferece uma dimensão adicional, permitindo que os usuários planejem dinamicamente suas próximas faixas, ajustando a sequência de reprodução de acordo com seus gostos momentâneos.

O processo de funcionamento do aplicativo inicia na interface gráfica de login, representada pela classe LoginGui. Nesta interface, os usuários inserem suas credenciais, como nome de usuário e senha.

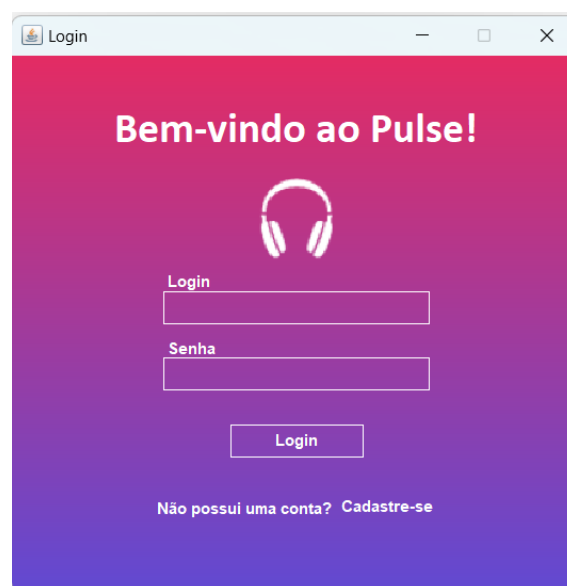


Figura 3. Interface de Login

Para os novos usuários, a opção "Cadastrar-se" direciona-os para a interface refe-

rente à classe CadastroGui, onde devem preencher informações essenciais, como nome, login e senha, para criar uma nova conta. As informações de cadastro fornecidas são então armazenadas no banco de dados MySQL (MySQL 8.0.34), estabelecendo um registro na tabela correspondente.

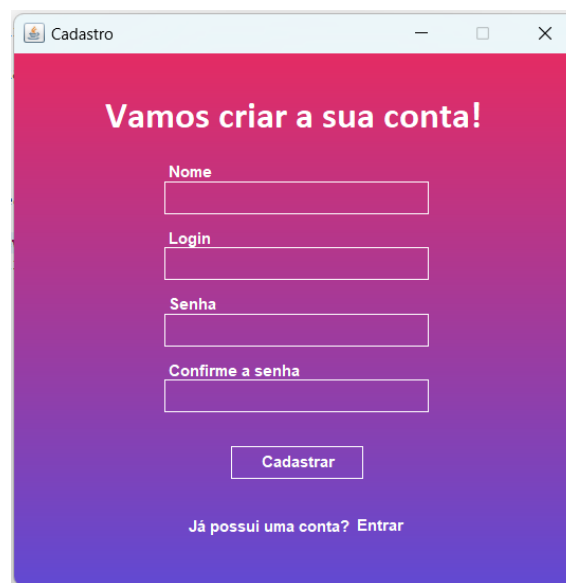


Figura 4. Interface de Cadastro

Quando um usuário insere suas credenciais e clica em "Login", o sistema verifica essas informações no banco de dados. Se as credenciais estão corretas, o usuário é autenticado com sucesso e redirecionado para a interface PrincipalGui.

A classe PrincipalGui representa a janela principal do aplicativo após um login bem-sucedido. Essa interface representa o núcleo da experiência do usuário, dividindo-se em diversas sessões principais que abordam as diferentes facetas da interação com o aplicativo.

Na seção "Início", os usuários são recebidos com uma visão imediata das músicas e playlists recentemente tocadas, proporcionando uma experiência dinâmica e personalizada. Essa funcionalidade permite que o usuário recorde facilmente suas experiências musicais recentes, adicionando uma camada de conveniência à navegação.

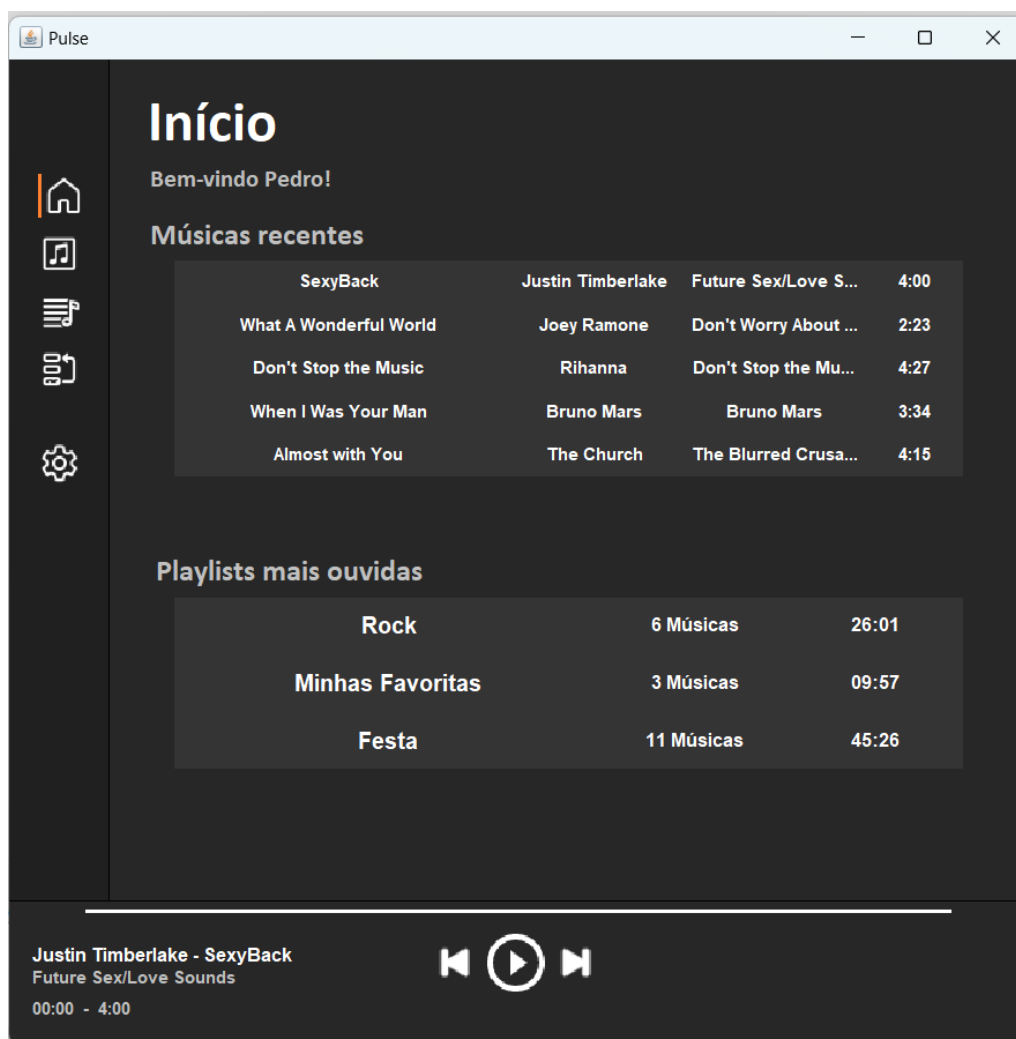


Figura 5. Seção de início da interface principal

A seção "Biblioteca" é dedicada à organização das músicas importadas pelo usuário. Aqui, os usuários podem explorar e gerenciar sua coleção pessoal, proporcionando fácil acesso às faixas e uma maneira eficiente de lidar com o conteúdo musical pessoal.

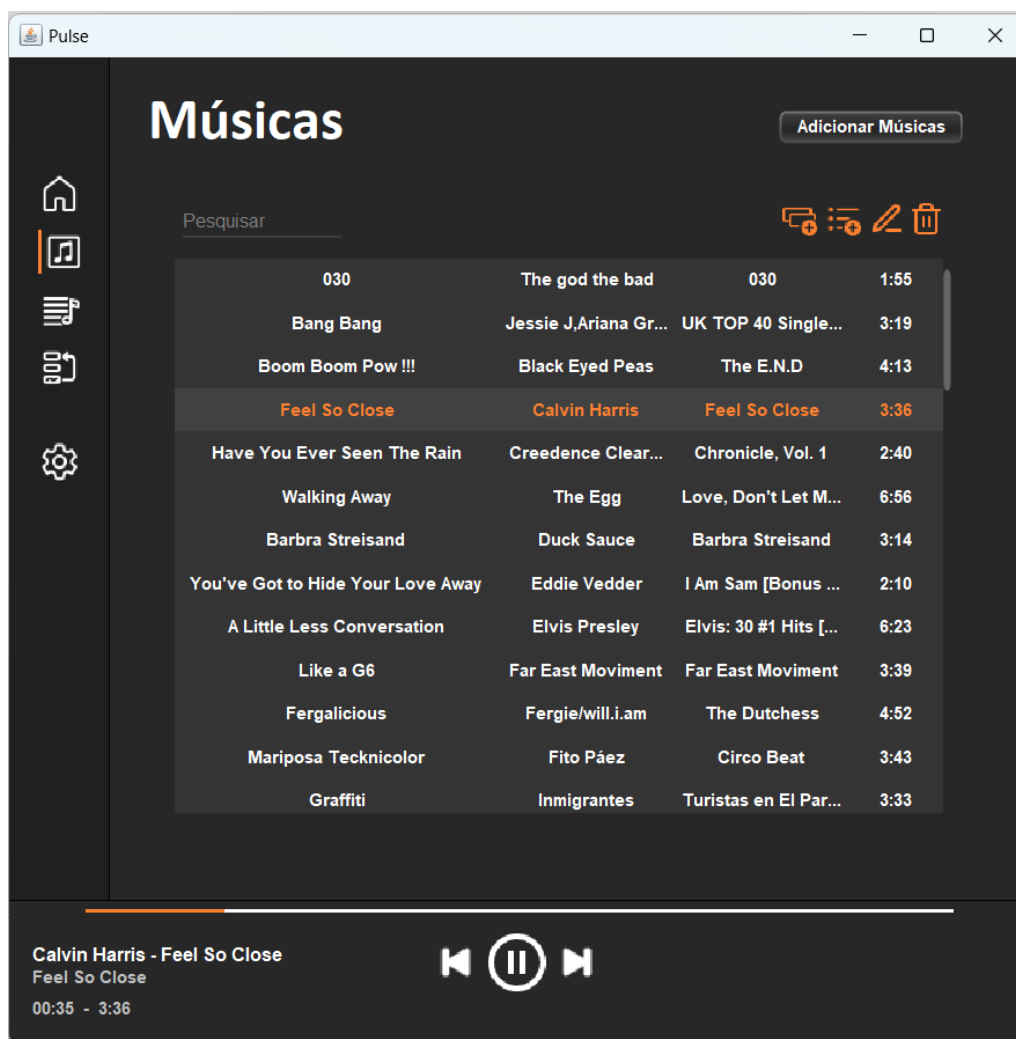


Figura 6. Seção de biblioteca da interface principal

Na biblioteca do aplicativo, os usuários têm a facilidade de importar suas próprias músicas do computador utilizando o botão "Adicionar Músicas". Ao clicar nesse botão, é acionado o JFileChooser da biblioteca Swing, possibilitando aos usuários navegar e selecionar as músicas desejadas em seus diretórios locais.

Além disso, a biblioteca do aplicativo oferece quatro funcionalidades acessíveis por meio de botões discretos. O botão "Remover Música" permite a exclusão da música selecionada, exibindo previamente uma tela de confirmação. Já o botão "Editar Música" abre uma nova janela, representada pela classe "EditMusicaGUI", dedicada à edição de informações da música. Na interface de edição de músicas, os usuários têm a capacidade de modificar informações específicas da música, como título, artista e álbum.

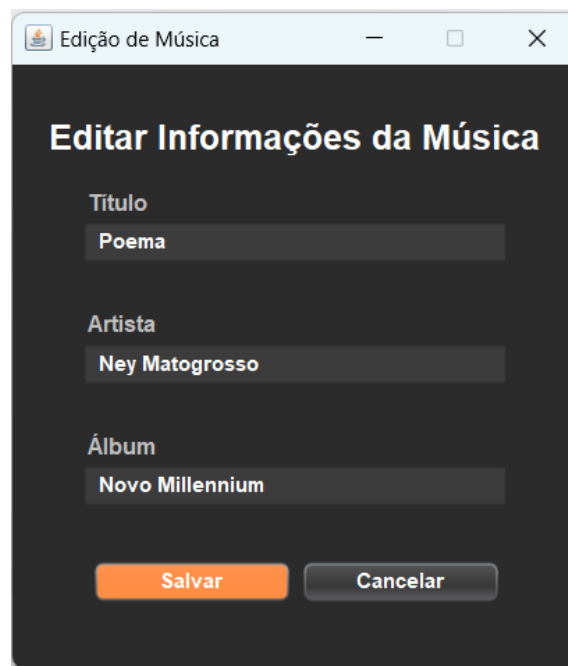


Figura 7. Interface de edição de música

A funcionalidade "Adicionar Música à Playlist" dispara a classe "AddMusicaPlaylistGUI", que por sua vez abre uma nova janela que possibilita ao usuário escolher em qual playlist específica a música selecionada será incluída. Nesta interface, os usuários são apresentados a uma lista das playlists personalizadas que eles criaram.

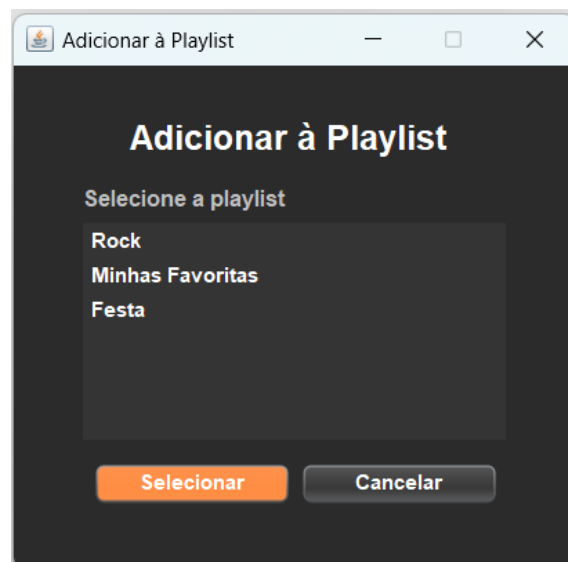


Figura 8. Interface de adição de música em playlist

Por fim, o botão "Adicionar Música à Fila" permite a adição da música selecionada à fila de reprodução do aplicativo.

Na seção "Playlists" são centralizadas as funcionalidades de criação, gerencia-

mento e acesso a playlists personalizadas. Essa abordagem permite que os usuários organizem suas músicas de acordo com preferências específicas.

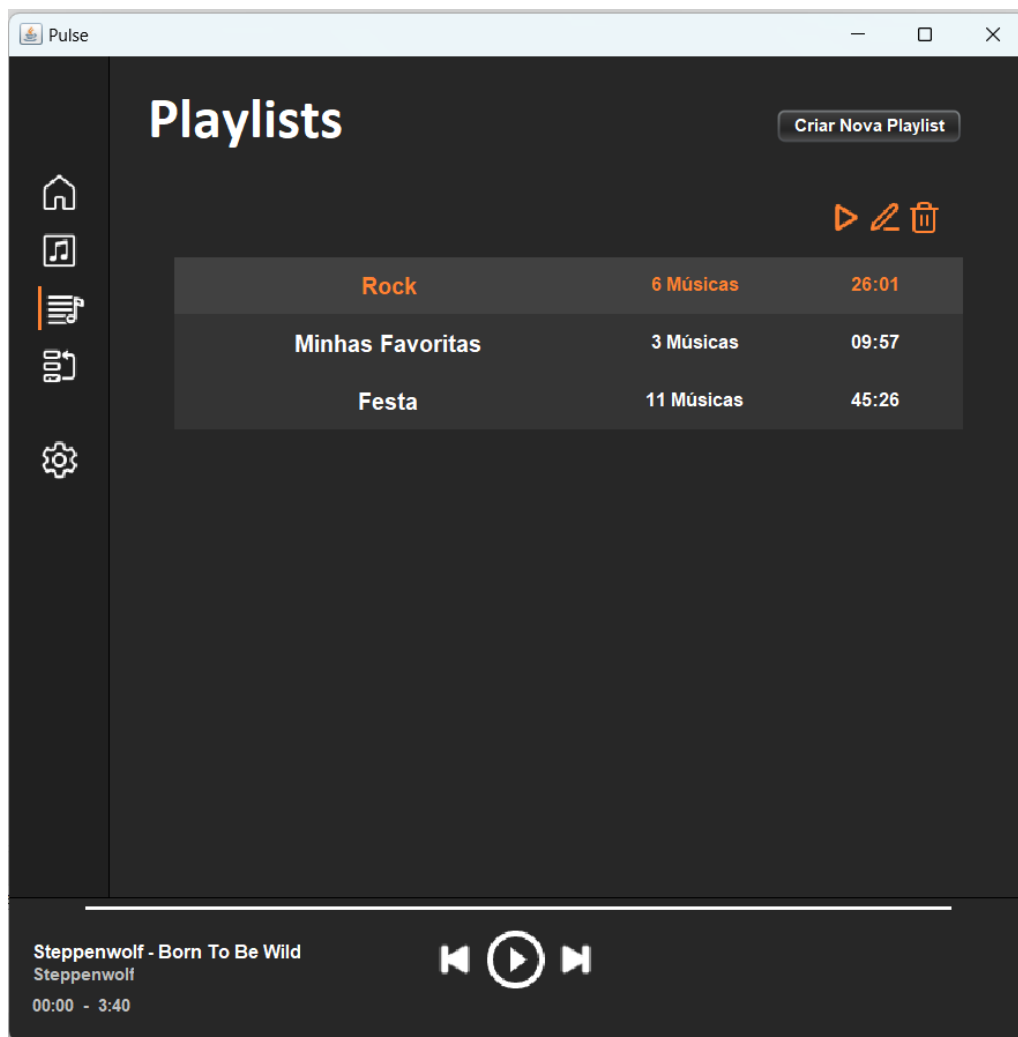


Figura 9. Seção de playlists da interface principal

Ao clicar no botão "Criar Nova Playlist", uma nova janela representada pela classe "AddPlaylistGUI" é imediatamente aberta, permitindo que o usuário forneça inicialmente o nome desejado para a nova playlist. Importante ressaltar que, conforme mencionado anteriormente, as músicas a serem adicionadas a essa nova playlist serão gerenciadas a partir da biblioteca do aplicativo.

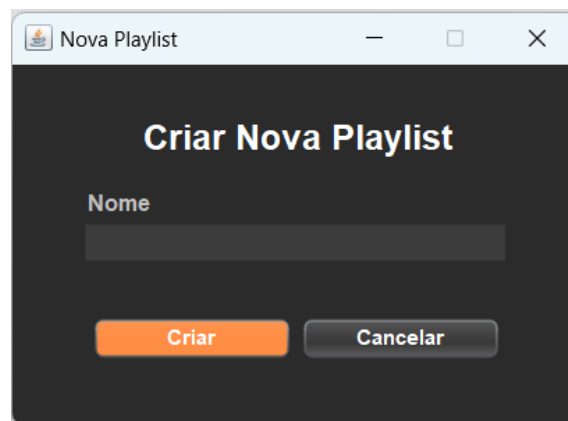


Figura 10. Interface de criação de playlist

Além disso, na seção de playlists, estão disponíveis três funcionalidades adicionais. O botão "Remover Playlist" deleta a playlist e suas músicas, requerendo confirmação do usuário. O botão "Editar Playlist" abre uma nova janela da classe "EditPlaylistGUI" que permite a edição das informações da playlist. Nessa interface, os usuários têm a capacidade de modificar não apenas o nome da playlist, mas também a ordem em que as músicas são dispostas e reproduzidas. Além disso, é possível remover músicas da playlist diretamente por meio dessa interface de edição.

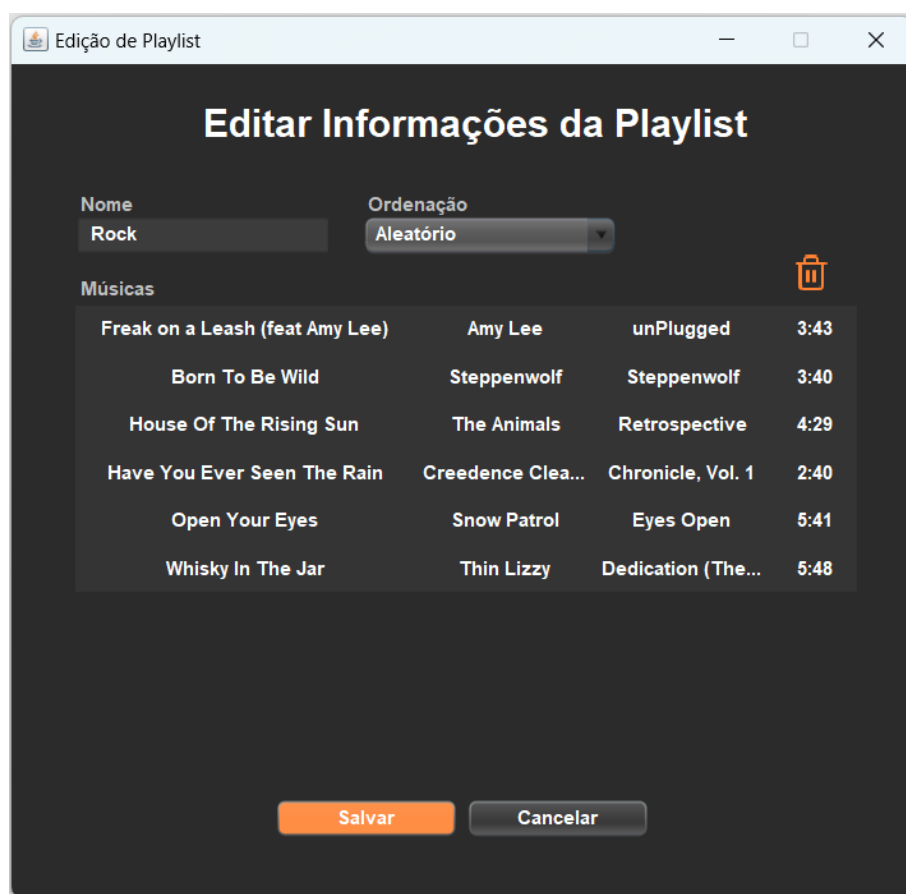


Figura 11. Interface de edição de playlist

Já a funcionalidade "Reproduzir Playlist" abre uma nova janela da classe "See-PlaylistGUI", onde é possível visualizar todas as músicas da playlist selecionada. Os usuários têm a opção de escolher entre reproduzir todas as músicas imediatamente ou adicionar as músicas da playlist à fila de reprodução do aplicativo.

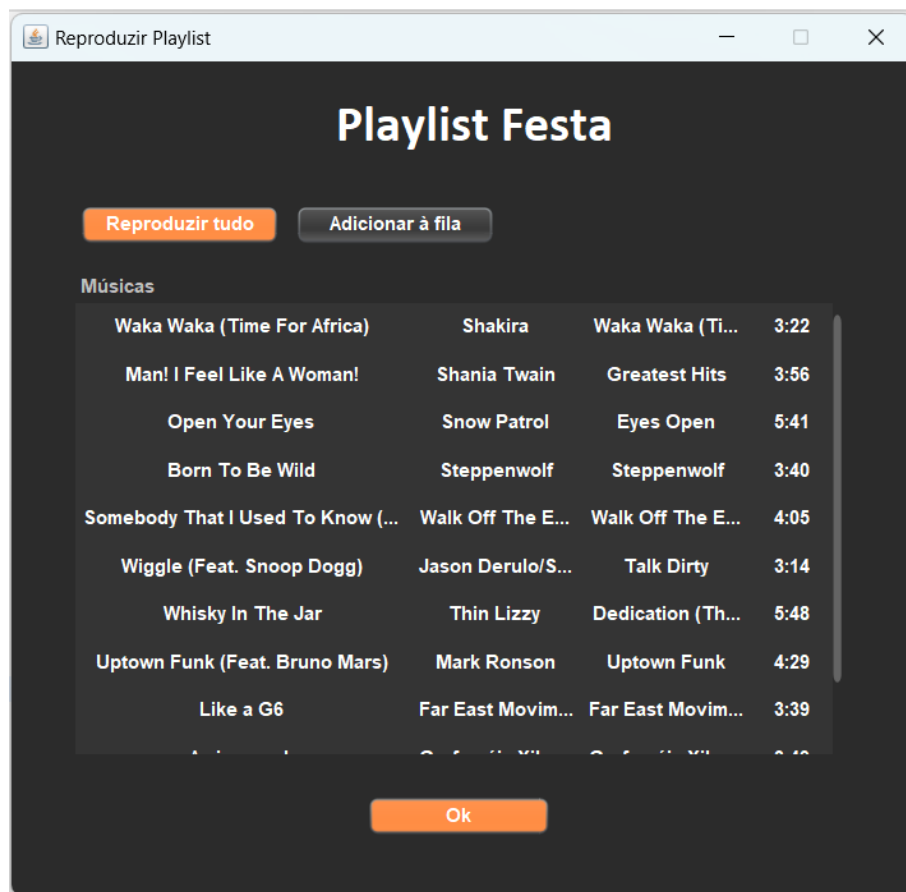


Figura 12. Interface de reprodução de playlist

A "Fila de Reprodução" é uma seção dinâmica que exibe as músicas programadas para reprodução subsequente. Isso concede aos usuários a capacidade de ajustar dinamicamente a lista de músicas, oferecendo uma experiência de audição contínua e adaptada aos seus desejos momentâneos.

As funcionalidades adicionais na fila de reprodução são representadas por dois botões discretos. O botão "Remover Música" permite a exclusão da música selecionada da fila, enquanto o botão "Limpar Fila" remove todas as músicas da fila de reprodução, efetivamente limpando-a.

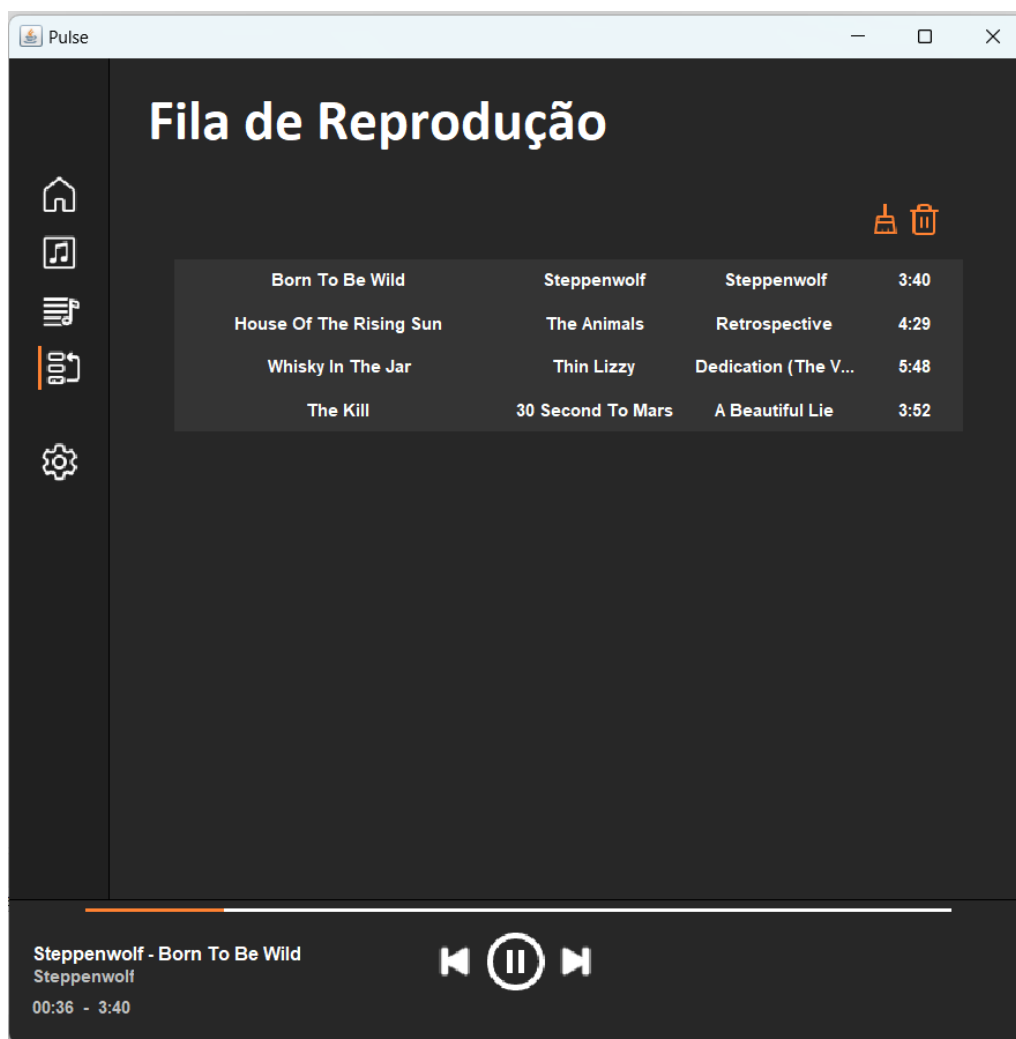


Figura 13. Seção da fila de reprodução da interface principal

A seção "Configurações" adiciona uma camada de controle ao proporcionar opções adicionais. Os usuários podem realizar o logoff para encerrar a sessão atual de forma segura. Além disso, essa área inclui uma opção para acessar a documentação do projeto, fornecendo informações detalhadas sobre o funcionamento do aplicativo.



Figura 14. Seção da configurações da interface principal

Cada sessão na PrincipalGui foi meticulosamente projetada para oferecer ao usuário uma experiência completa e intuitiva, desde a organização de músicas e play-lists até a personalização da fila de reprodução. O objetivo é tornar a navegação pelo aplicativo fluida e adaptada às preferências individuais, promovendo uma experiência de audição envolvente e eficiente.

A integração estratégica de bibliotecas externas desempenha um papel fundamental no funcionamento geral do aplicativo. Essas bibliotecas oferecem soluções especializadas que simplificam e aprimoram aspectos cruciais do software, permitindo uma experiência mais robusta e enriquecedora para o usuário.

Para viabilizar a reprodução das músicas no ambiente Java, foi adotada a biblioteca Jlayer (JLayer 1.0.1 JavaZoom). A integração desta biblioteca possibilitou a implementação de todas as funcionalidades essenciais relacionadas à reprodução de áudio. Com o suporte do jlayer, o aplicativo é capaz de executar operações como play, pause, avançar e retroceder de maneira eficiente e consistente.

Em relação à obtenção dos dados das músicas durante o processo de importação

para o sistema, foi adotada a utilização da biblioteca externa JaudioTagger (JaudioTagger 3.0.1). Essa biblioteca desempenha um papel crucial ao facilitar a extração de metadados relevantes das faixas musicais, como título, artista, álbum e duração. A integração do JaudioTagger no processo de importação garante uma coleta precisa e organizada das informações, enriquecendo assim a biblioteca do usuário com dados completos e precisos sobre suas músicas.

A integração efetiva com o banco de dados MySQL foi alcançada através do uso do JDBC (Java Database Connectivity 8.1.0) específico para o MySQL. Essa escolha estratégica do JDBC do MySQL proporcionou uma conexão robusta e eficiente entre o aplicativo e o banco de dados, permitindo operações como inserção, consulta, atualização e exclusão de dados de maneira confiável.

Dessa forma, a combinação de uma interface intuitiva, funcionalidades bem projetadas e a integração eficiente de bibliotecas externas e banco de dados culmina em um aplicativo coeso e envolvente.

6. Organização de software

Para aprimorar a compreensão da organização interna do software, será apresentado o diagrama de classes, uma representação visual que destaca a estrutura estática do sistema. Esse diagrama é essencial para fornecer uma visão gráfica das classes presentes no projeto, suas inter-relações, atributos e métodos.

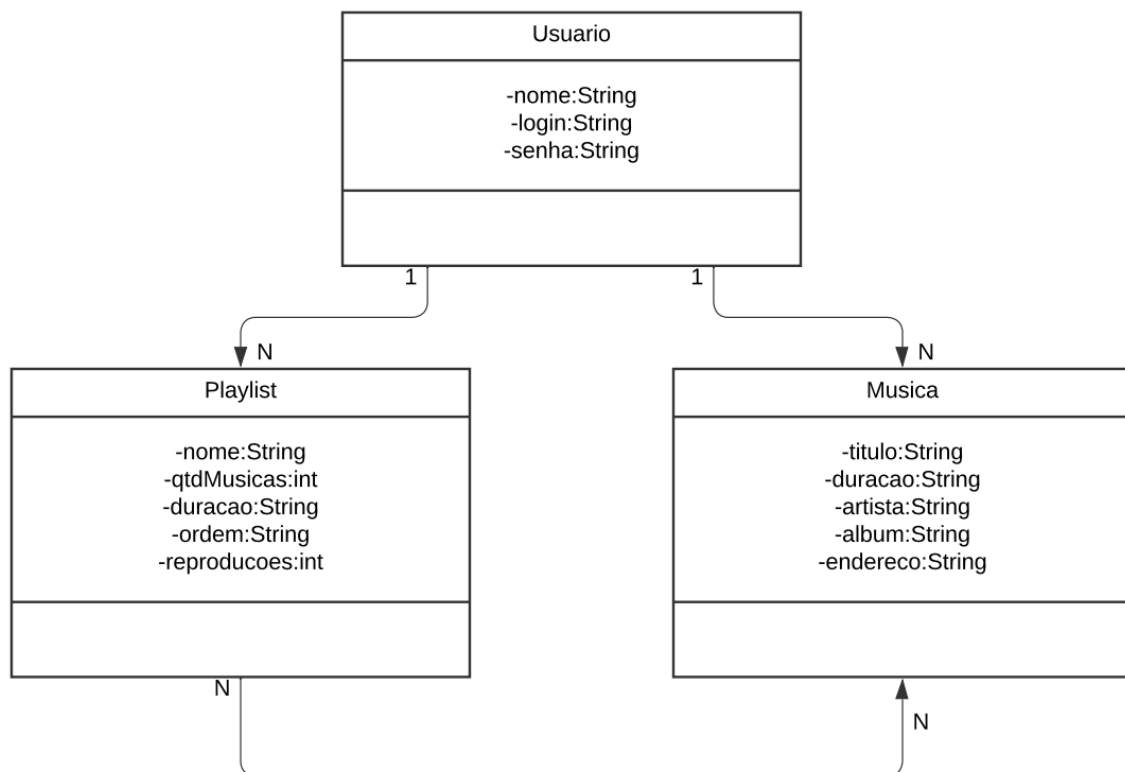


Figura 15. Diagrama de classes

Essas três classes, "Usuario", "Musica" e "Playlist", desempenham papéis cruciais

como as classes de negócio fundamentais da solução, representando as entidades principais no contexto do software. A classe "Usuario" encapsula informações específicas do usuário, como nome de usuário e preferências. Por sua vez, a classe "Musica" abrange os detalhes associados às músicas, como título, artista e álbum. A classe "Playlist" atua como uma entidade organizadora, permitindo a composição e manipulação de listas personalizadas de músicas. Essas classes essenciais não apenas encapsulam dados pertinentes, mas também incorporam a lógica de negócio essencial para o funcionamento do sistema.

O sistema foi organizado em pacotes que visam estruturar e modularizar as diferentes funcionalidades. Essa abordagem permite uma organização mais eficiente do código, facilitando a manutenção e o desenvolvimento contínuo do software. A seguir, uma tabela apresenta os pacotes principais e suas funcionalidades específicas, oferecendo uma visão sistêmica das diferentes partes do sistema e suas respectivas responsabilidades.

Tabela 1. Tabela da estrutura de pacotes

Pacote	Descrição
conexao	Estabelece a conexão com o banco de dados
config	Configura o banco de dados e recupera a sessão de login
control	Atua como intermediário entre o modelo e a visão
dao	Realiza operações no banco de dados
model	Representa as entidades do projeto
player	Responsável pelo funcionamento da reprodução das músicas
principal	Contém o método <i>main</i> da aplicação
view	Responsável pela interface do usuário
view.tablemodel	Abrange a lógica de modelagem de tabelas na interface do usuário

Com o objetivo de avaliar o nível de aceitação e usabilidade efetiva do sistema, implementou-se um questionário contendo cinco perguntas. Dez usuários com diferentes níveis de experiência foram selecionados para participar dessa avaliação, proporcionando uma visão diversificada sobre a eficácia do software em diferentes contextos de uso. Cada resposta é expressa por meio de uma escala de 0 a 5 estrelas, permitindo uma quantificação clara e comparativa das percepções individuais.

Vale ressaltar que esses usuários previamente experimentaram e utilizaram o software, contribuindo com insights baseados em suas experiências práticas. A análise dos resultados será apresentada por meio de uma tabela, proporcionando uma visão consolidada das avaliações e destacando áreas de sucesso e potenciais melhorias, conforme percebidas pelos usuários.

Tabela 2. Tabela de feedback dos usuários

Pergunta	Resultado
Como você avalia a facilidade de navegação e localização das funcionalidades no software?	4,6
Qual é a sua percepção sobre o desempenho geral do sistema em termos de velocidade e responsividade?	4,8
O sistema permite que você execute suas tarefas de maneira eficiente e sem complicações?	4,7
O software fornece feedback visual adequado, como mensagens de sucesso ou erro, para suas ações?	4,9
Considerando sua experiência global com o software, qual seria sua avaliação geral?	4,8

Com base nos dados coletados, a análise das respostas dos usuários revelou uma tendência positiva em relação à usabilidade e desempenho geral do software. Esses dados fornecem insights valiosos para reforçar os pontos fortes do sistema e direcionar futuras melhorias, se necessário, visando aprimorar ainda mais a experiência do usuário.

7. Considerações Finais

Reprodutor Musical em Java: Explorando as Possibilidades do Swing - o título do presente artigo cita a ferramenta Swing, que é uma biblioteca usada para a programação da interface gráfica, aplicada à linguagem Java. A primeira definição de requisitos da interface do Pulse nasceu da análise sobre qual forma pertinente a parcela visual que compõe o desenvolvimento do software assumiria, tornando-se parte consistente na relação entre forma, função e estrutura do sistema. Associadas a isso, as expectativas do próprio grupo em relação às possibilidades formais do que um software desenvolvido em Java poderia entregar foram quebradas sucessivas vezes no decorrer do desenvolvimento. Uma estrutura formal é um princípio ordenador segundo o qual uma série de elementos, governados por relações precisas, adquirem uma determinada estrutura, citando Hélio Pinon.

O processo de desenvolvimento do Pulse foi repleto de desafios e aprendizados que, ao longo de cada etapa, cresciam conjuntamente. Destacou-se no seu curso o momento ao qual os primeiros resultados práticos foram obtidos. A codificação ocorreu apenas depois de estarem delineadas as etapas de definição de escopo e requisitos e foi premissa da equipe que se respeitasse a metodologia proposta. As entregas realizadas de forma incremental e sistemática partiam da subdivisão dos requisitos como metas e foram fator motivador.

Ao final, o aplicativo atendeu a todos os requisitos que foram planejados. Não obstante, foi além, superando as expectativas da equipe. Uma das decisões iniciais acerca do desenvolvimento era concentrar-se esforços em criar um backend funcional e com uma interface minimamente apresentável. Porque esse mesmo foco da equipe sempre orbitou em torno dos requisitos funcionais, os tendo como premissa. O fato de que foram desenvolvidos mockups como estratégia para definições de projeto associado a pessoas e circunstâncias extraordinárias trouxeram gratas surpresas ao processo como um todo. A descoberta de que é possível criar uma interface gráfica agradável e sobretudo funcional também, partindo tão somente de bons princípios de design, usando Java e com bibliotecas

default da linguagem, fez com que houvesse espaço para discussões mais profundas sobre a usabilidade do aplicativo, simulando trocas reais do mercado de trabalho.

A determinação vista por um desenvolvedor ao se delinear o que é possível fazer em um sistema traz um crescimento e aprendizado ímpar que nenhuma abordagem limitante constrói. O aplicativo em questão pode e deve ser melhorado se assim for necessário, mas trouxe crescimentos que estão além do código em si, da elicitação de requisitos, do escopo claro, dos desafios que envolvem tornar funcional uma aplicação que reproduz, gerencia e incorpora músicas. Os ganhos se dão a nível humano e ainda estratégico, nos quais o processo de crescimento profissional acontece depois do pessoal.

A música e a arte são indissociáveis do fluxo da vida humana e a sabedoria mora no reconhecimento das grandezas que surgem em meio à pequenez do dia-a-dia, no pulsar do peito, como é quando se conclui um projeto. “E tudo o que você cria, e tudo o que você destrói, e tudo o que você faz, e tudo o que você diz, e tudo o que é agora, e tudo o que já passou, e tudo que virá, e tudo sob o Sol está em perfeita sintonia, mas o sol está coberto pela lua. . .” - nota traduzida de um trecho da música Eclipse, Pink Floyd (Álbum Pulse, 1995).

Referências

- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.
- Jacobson, I., Christensen, M., Jensen, P., and Linder, M. (1999). *Object-Oriented Software Engineering: A Use-Case Driven Approach*. Addison-Wesley, 2 edition.
- Mahfuz, E. C. (2007). Observações sobre o formalismo de helio piñón - parte 1. *Arquitextos*, 08(089.01):196.
- Martin, R. C. (2003). *Agile Software Development: Principles, Patterns, and Practices*. Pearson Education.
- Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
- Pressman, R. and Maxim, B. (2016). *Engenharia de Software*. Grupo A, 8th edition.

A. Apêndice - Realização dos Casos de Uso

Nesta seção de apêndice, são apresentadas as tabelas com a realização dos casos de uso elaborados na etapa de levantamento de requisitos e casos de uso. Nela serão descritos com detalhes cada um dos casos de uso representados na figura 1, da seção 3 deste artigo.

Tabela 3. Realização de caso de uso 01

Caso de uso	UC 01 - Cadastro de novo usuário
Requisitos funcionais atendidos	RF01
Atores	Novo usuário do sistema
Pré-condições	Estar conectado no sistema
Pós-condições	Nova conta de usuário criada
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário escolhe a opção para cadastrar-se; 2. O sistema exibe a tela para cadastro de usuário; 3. O sistema solicita o nome do usuário, login e senha; 4. O usuário informa os dados solicitados; 5. O sistema grava as informações de login e informa o sucesso da operação;
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 4. a. O usuário não preenche todas informações necessárias (usuário e senha): <ol style="list-style-type: none"> 1. O sistema solicita que sejam preenchidas as informações que estão faltando. 4. b. O usuário informa um nome de usuário que já existe no sistema: <ol style="list-style-type: none"> 1. O sistema exibe uma mensagem de alerta referente ao problema.
Requisitos especiais	Não possui
Frequência	Ocasionalmente

Tabela 4. Realização de caso de uso 02

Caso de uso	UC 02 - Login de usuário existente
Requisitos funcionais atendidos	RF01
Atores	Usuário do sistema
Pré-condições	Possuir um cadastro no sistema
Pós-condições	Entrar no sistema com a sua conta
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O sistema exibe a tela para login de usuário; 2. O sistema solicita o login e a senha do usuário; 3. O usuário informa os dados solicitados; 4. O sistema valida as informações de login; 5. O sistema direciona o usuário para a tela inicial.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 3. a. O usuário não preenche todas informações necessárias (usuário e senha): <ol style="list-style-type: none"> 1. O sistema solicita que sejam preenchidas as informações que estão faltando. 3. b. O usuário informa credenciais inválidas: <ol style="list-style-type: none"> 1. O sistema exibe uma mensagem de falha de autenticação.
Requisitos especiais	Não possui
Frequência	Rotineiramente

Tabela 5. Realização de caso de uso 03

Caso de uso	UC 03 - Importação de músicas à biblioteca
Requisitos funcionais atendidos	RF02
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir músicas no formato MP3 em seu computador
Pós-condições	Músicas importadas para a biblioteca do sistema
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até a seção da biblioteca e clica no botão que importa músicas; 2. O sistema abre uma janela de seleção de arquivos do computador; 3. O usuário seleciona a(s) música(s) ou pasta que deseja importar; 4. O sistema verifica a validade dos arquivos selecionados; 5. O sistema processa, armazena e adiciona esses arquivos à biblioteca de músicas.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 4. a. Os arquivos selecionados não estão no formato MP3: 1. O sistema exibe uma mensagem de erro e instrui o usuário a selecionar arquivos válidos.
Requisitos especiais	Não possui
Frequência	Rotineiramente

Tabela 6. Realização de caso de uso 04

Caso de uso	UC 04 - Edição de músicas da biblioteca
Requisitos funcionais atendidos	RF03
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir músicas na biblioteca
Pós-condições	Música editada na biblioteca do sistema
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até a biblioteca de músicas; 2. O usuário seleciona a música desejada e escolhe a opção de edição; 3. O sistema exibe a tela de edição de músicas; 4. O sistema permite que sejam alterados os dados referentes à música; 5. O usuário informa os dados desejados e clica no botão de salvar; 6. O sistema valida as informações; 7. O sistema edita a música da biblioteca do usuário conforme solicitado.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 5. a. O usuário cancela a operação: 1. O sistema ignora as alterações.
Requisitos especiais	Não possui
Frequência	Frequentemente

Tabela 7. Realização de caso de uso 05

Caso de uso	UC 05 - Remoção de músicas da biblioteca
Requisitos funcionais atendidos	RF03
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir músicas na biblioteca
Pós-condições	Música removida da biblioteca do sistema
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até a biblioteca de músicas; 2. O usuário seleciona a música desejada e escolhe a opção de remoção; 3. O sistema exibe uma tela de confirmação; 4. O usuário confirma a ação; 5. O sistema remove a música da biblioteca do usuário;
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 4. a. O usuário cancela a operação: 1. O sistema ignora a ação e a música é mantida.
Requisitos especiais	Não possui
Frequência	Frequentemente

Tabela 8. Realização de caso de uso 06

Caso de uso	UC 06 - Criação de uma nova playlist
Requisitos funcionais atendidos	RF04
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema
Pós-condições	Nova playlist criada
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até a seção das playlists e seleciona a opção de criar uma nova playlist; 2. O sistema exibe a tela de criação de playlists; 3. O sistema solicita os dados referentes à playlist (nome); 4. O usuário informa os dados necessários para a criação; 5. O sistema valida as informações fornecidas pelo usuário; 6. O sistema cria a nova playlist.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 4. a. O usuário não preenche todas as informações necessárias: 1. O sistema solicita que sejam preenchidas todas as informações. 4. b. O usuário informa um nome de playlist existente: 1. O sistema solicita que já existe uma playlist com aquele nome.
Requisitos especiais	Não possui
Frequência	Rotineiramente

Tabela 9. Realização de caso de uso 07

Caso de uso	UC 07 - Remoção de uma playlist existente
Requisitos funcionais atendidos	RF04
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir playlists
Pós-condições	Playlist removida
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuario navega até a seção das playlists e seleciona a playlist desejada; 2. O usuario escolhe a opção de remover a playlist; 3. O sistema exibe uma tela de confirmação; 4. O usuário confirma a ação; 5. O sistema remove a playlist do usuário e as músicas vinculadas a ela.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 4. a. O usuário cancela a operação: 1. O sistema ignora a ação e a playlist do usuário é mantida.
Requisitos especiais	Não possui
Frequência	Frequentemente

Tabela 10. Realização de caso de uso 08

Caso de uso	UC 08- Edição de uma playlist existente
Requisitos funcionais atendidos	RF04
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir playlists
Pós-condições	Playlist editada
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuario navega até a seção das playlists e seleciona a playlist desejada; 2. O usuario seleciona a opção de editar a playlist desejada; 3. O sistema exibe a tela de edição da playlists; 4. O sistema permite que sejam alterados os dados referentes à playlist; 5. O usuario informa os dados desejados e clica no botão de salvar; 6. O sistema valida as informações; 7. O sistema edita a playlist conforme solicitado.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 5.a. O usuário informa dados inválidos: 1. O sistema solicita que sejam informados dados válidos.
Requisitos especiais	Não possui
Frequência	Frequentemente

Tabela 11. Realização de caso de uso 09

Caso de uso	UC 09 - Adição de música em uma playlist
Requisitos funcionais atendidos	RF05
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema, possuir músicas e playlists
Pós-condições	Música adicionada na playlist
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até a biblioteca de músicas; 2. O usuário seleciona a música desejada e escolhe a opção de adicionar à playlist; 3. O sistema exibe uma tela para selecionar a playlist desejada; 4. O usuário seleciona a playlist que deseja incluir a música e confirma a ação; 5. O sistema inclui a música na playlist do usuário;
Extensão (Fluxos Alternativos)	<p>3.a. O usuário não possui playlists cadastradas:</p> <ol style="list-style-type: none"> 1. O sistema direciona o usuário para a tela de criação de playlists. <p>4.a. O usuário escolhe uma playlist que já está com a música selecionada:</p> <ol style="list-style-type: none"> 1. O sistema exibe uma mensagem de aviso para o usuário e a música não é adicionada à playlist.
Requisitos especiais	Não possui
Frequência	Rotineiramente

Tabela 12. Realização de caso de uso 10

Caso de uso	UC 10 - Remoção de uma música de uma playlist
Requisitos funcionais atendidos	RF05
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir uma playlist com música
Pós-condições	Música removida da playlist
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até suas playlists; 2. O usuário seleciona a playlist desejada e escolhe a opção de edição; 3. O sistema exibe as músicas que estão na playlist; 4. O usuário seleciona a música na qual deseja remover e escolhe a opção de removê-la; 5. O sistema exibe uma tela de confirmação; 6. O usuário confirma a ação; 7. O sistema remove a música da playlist do usuário;
Extensão (Fluxos Alternativos)	<p>6. a. O usuário cancela a ação:</p> <ol style="list-style-type: none"> 1. O sistema ignora a ação e a música é mantida na playlist.
Requisitos especiais	Não possui
Frequência	Rotineiramente

Tabela 13. Realização de caso de uso 11

Caso de uso	UC 11 - Reprodução e controle das músicas
Requisitos funcionais atendidos	RF06, RF09
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir uma música na biblioteca ou em uma playlist
Pós-condições	Música reproduzida
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até a biblioteca ou suas playlists; 2. O usuário seleciona a música desejada e escolhe a opção de reproduzir; 3. O sistema reproduz a música solicitada;
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 4. a. O usuário seleciona um comando de controle na barra de reprodução; 1. O sistema responde à ação do usuário de acordo com o comando escolhido; 4. b. Se existirem músicas na fila de reprodução: <ol style="list-style-type: none"> 1. Após o término da reprodução de uma música, o sistema passa automaticamente para a próxima música da fila e inicia sua reprodução. 2. O processo continua até que todas as músicas da fila de reprodução tenham sido reproduzidas.
Requisitos especiais	Não possui
Frequência	Rotineiramente

Tabela 14. Realização de caso de uso 12

Caso de uso	UC 12 - Pesquisa de músicas
Requisitos funcionais atendidos	RF07
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir músicas na biblioteca
Pós-condições	Lista de músicas filtradas de acordo com a pesquisa
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até a biblioteca de músicas; 2. O usuário seleciona o campo de pesquisar músicas; 3. O sistema exibe uma tela de pesquisa; 4. O sistema solicita os dados que servirão como filtro; 5. O usuário informa os dados desejados; 6. O sistema faz a busca das músicas aplicando os filtros; 7. O sistema mostra os resultados filtrados para o usuário.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 7. a. Não existem músicas para os filtros solicitados; 1. O sistema exibe uma tela de aviso.
Requisitos especiais	Não possui
Frequência	Frequentemente

Tabela 15. Realização de caso de uso 13

Caso de uso	UC 13 - Adição de músicas à fila de reprodução
Requisitos funcionais atendidos	RF08
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir músicas na biblioteca
Pós-condições	Música adicionada na fila de reprodução
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até sua biblioteca de músicas ou playlists. 2. O usuário seleciona uma música da biblioteca ou da playlist. 3. O sistema exibe a opção para adicionar a música à fila de reprodução. 4. O usuário escolhe adicionar a música à fila de reprodução. 5. O sistema adiciona a música à fila de reprodução.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 6. Em seguida, ao terminar uma música, o sistema irá reproduzir a próxima música da fila de reprodução.
Requisitos especiais	Não possui
Frequência	Frequentemente

Tabela 16. Realização de caso de uso 14

Caso de uso	UC 14 - Remoção de músicas da fila de reprodução
Requisitos funcionais atendidos	RF08
Atores	Usuário do sistema
Pré-condições	Estar logado no sistema e possuir músicas uma música na fila de reprodução
Pós-condições	Música removida da fila de reprodução
Cenário de sucesso (Fluxo Principal)	<ol style="list-style-type: none"> 1. O usuário navega até a fila de reprodução; 2. O usuário seleciona a opção para gerenciar a fila de reprodução; 3. O sistema exibe a lista de músicas na fila de reprodução; 4. O usuário seleciona a música que deseja remover; 5. O sistema exibe a opção para remover a música selecionada; 6. O usuário confirma a remoção da música; 7. O sistema remove a música selecionada da fila de reprodução; 8. O sistema atualiza a lista de músicas na fila.
Extensão (Fluxos Alternativos)	<ol style="list-style-type: none"> 6. a. O usuário cancela a ação: 1. O sistema cancela a operação e a música é mantida na fila.
Requisitos especiais	Não possui
Frequência	Frequentemente