

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
ENGENHARIA DE SOFTWARE-ES41

Antônio Carlos Ferreira
Gustavo Pukanski Schatzmann
Luiz Fernando Moreira Domênico

NOTESYNC

CORNÉLIO PROCÓPIO
JUNHO, 2025

Antônio Carlos Ferreira
Gustavo Pukanski Schatzmann
Luiz Fernando Moreira Domênico

NOTESYNC

Projeto elaborado na disciplina de Programação Orientada a Objetos 2 do curso de Engenharia da Computação, do Campus Cornélio Procópio da Universidade Tecnológica Federal do Paraná.

Professora: Gisele Alves Santana

CORNÉLIO PROCÓPIO

JUNHO, 2025

RESUMO

Este sistema tem como objetivo auxiliar no ensino de teoria musical por meio de uma abordagem interativa e visual. A plataforma apresenta acordes, escalas (maiores e menores) e progressões harmônicas, permitindo ao usuário visualizar as notas em um teclado virtual, além de escutá-las em tempo real. A metodologia combina recursos de áudio, visualização gráfica e navegação intuitiva para facilitar o aprendizado prático e teórico. Os resultados esperados incluem o aprimoramento da compreensão musical, o reconhecimento auditivo e a familiarização com estruturas harmônicas. Conclui-se que a integração entre som e imagem torna o processo de aprendizagem mais eficaz, especialmente para iniciantes.

Palavras-chave: teoria musical, acordes, escalas, progressões harmônicas, áudio interativo.

SUMÁRIO

1	Introdução	1
2	FERRAMENTAS E TECNOLOGIAS	1
3	DESENVOLVIMENTO	1
3.1	Levantamento dos Requisitos	1
3.2	Diagrama de Casos de Uso	1
3.3	Especificação dos Casos de Uso	1
3.4	Diagrama de Classes	1
3.5	Boas Práticas	2
3.6	Banco de Dados	2
3.7	Telas do sistema	2
3.8	Link do vídeo da apresentação	2
4	CONCLUSÕES	2
	REFERÊNCIAS	3

1 INTRODUÇÃO

A teoria musical, por sua natureza abstrata, frequentemente apresenta desafios significativos para estudantes, especialmente os iniciantes. Embora exista atualmente uma ampla gama de ferramentas digitais, que vão desde complexas estações de trabalho de áudio até softwares de notação musical, muitas delas carecem de uma abordagem integrada, voltada especificamente para o aprendizado prático.

Diante desse cenário, este projeto tem como objetivo desenvolver uma plataforma interativa destinada ao ensino prático da teoria musical. A proposta consiste em oferecer ao usuário a possibilidade de visualizar, de forma clara e didática, notas, acordes, escalas e progressões harmônicas, ao mesmo tempo em que escuta sua sonoridade em tempo real. Ao integrar recursos visuais e auditivos, espera-se que a ferramenta contribua significativamente para o aprimoramento da compreensão teórica, para o desenvolvimento da percepção musical e para a familiarização com estruturas harmônicas essenciais.

2 FERRAMENTAS E TECNOLOGIAS

IDEs: Eclipse e NetBeans

SGBD: MySQL

Linguagem: Java

Diagramação: Astah

Plataforma de hospedagem: Github

3 DESENVOLVIMENTO

3.1 Levantamento dos Requisitos

Requisitos Funcionais

Identificador	Descrição	Prioridade
RF01	O sistema deve permitir a visualização das diferentes escalas, campos harmônicos e acordes.	Alta
RF02	O sistema deve ter filtros para facilitar a busca.	Alta
RF03	O sistema deve permitir a inclusão de progressões que o usuário criou.	Alta
RF04	O sistema deve permitir a importação de progressões.	Alta
RF05	O sistema deve permitir a exportação de progressões.	Alta
RF06	O usuário pode buscar por diferentes acordes e escalas.	Alta
RF07	O usuário pode favoritar um acorde ou escala.	Média
RF08	O sistema pode exibir informações complementares sobre acordes e escala.	Média
RF09	O usuário pode transpor notas.	Média
RF10	O usuário pode inverter acordes.	Média
RF11	O sistema pode reproduzir sonoramente as escalas, os acordes, as notas e as progressões.	Baixa
RF12	O sistema deve reconhecer os dispositivos MIDI conectados.	Baixa
RF13	O sistema pode se conectar com dispositivos MIDI.	Baixa
RF14	O sistema pode reconhecer as notas tocadas em um teclado/piano conectado por USB.	Baixa
RF15	O usuário pode fazer anotações.	Baixa
RF16	O usuário pode buscar informações sobre músicas já existentes.	Baixa

Requisitos Não Funcionais

Identificador	Descrição	Tipo
RNF01	O sistema deve ter um design responsivo, garantindo uma resposta em no máximo 3 segundos	Desempenho
RNF02	O sistema deve ser compatível com Windows, macOS e Linux	Compatibilidade
RNF03	O sistema deve ter uma interface intuitiva	Produto
RNF04	O sistema deve oferecer suporte para múltiplos idiomas, como Português, Espanhol, Inglês e Mandarim	Produto
RNF05	O sistema deve estar disponível 24 horas por dia	Disponibilidade
RNF06	O sistema deve fazer backup das progressões do usuário a cada 3 dias	Confiabilidade
RNF07	O sistema deve oferecer recursos de acessibilidade, como contraste ajustável	Acessibilidade
RNF08	O sistema deve permitir o uso offline do aplicativo	Desempenho
RNF09	O sistema deve implementar as atualizações automaticamente	Manutenção
RNF10	O sistema deve registrar e reportar erros em um log automaticamente	Manutenção

3.2 Diagrama de Casos de Uso

Diagrama do caso de uso geral do programa:

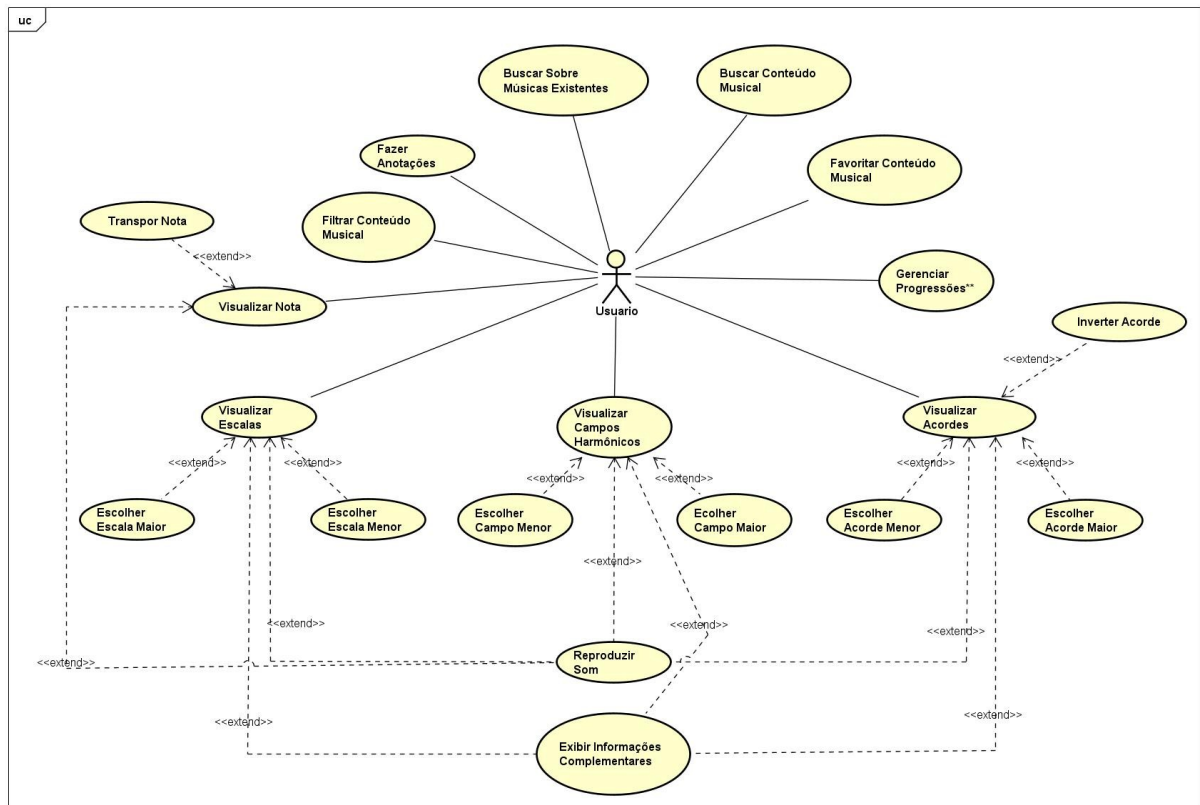


Diagrama do gerenciar progressões:

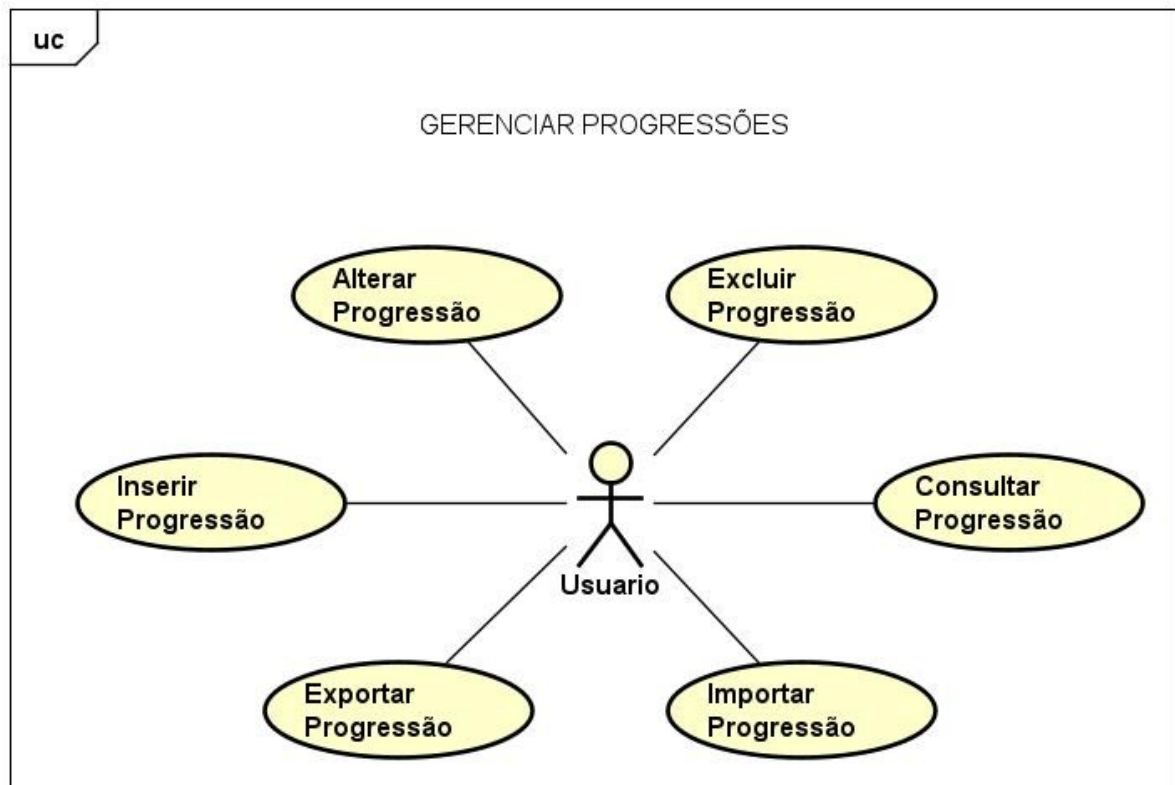


Diagrama de classes seguindo o MVC:

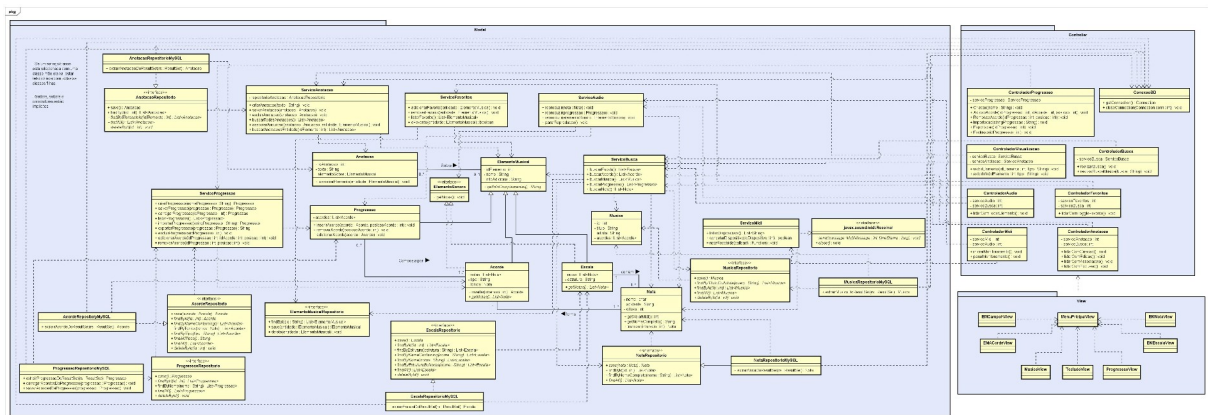


Diagrama do pacote Model simplificado para melhor visualização:

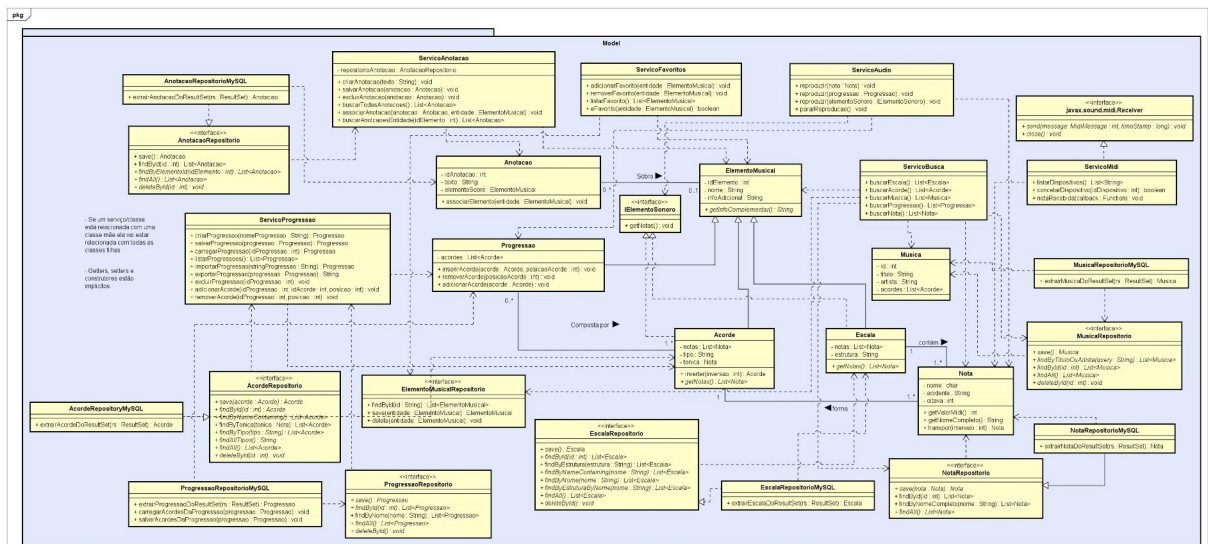
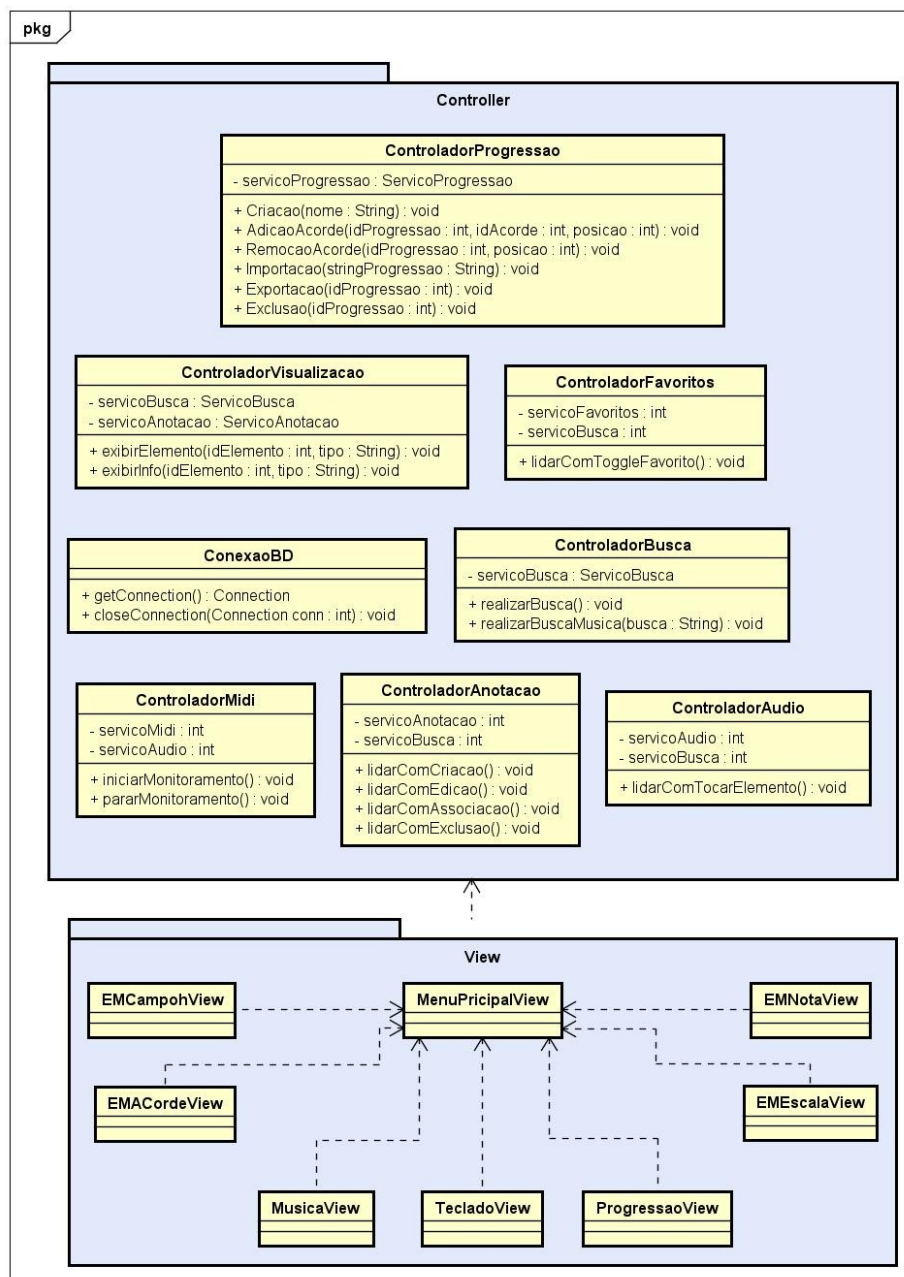


Diagrama dos pacotes Controller e View simplificados:



3.4 Boas Práticas

3.4.1 SRP

De início, foi usada a classe 'ControladorInteracao', da camada controller, mas ela tinha muitas responsabilidades, como está abaixo. Por isso, ela foi fragmentada em outras quatro classes: ControladorAnotacao, ControladorAudio, ControladorFavoritos e ControladorMidi.

```
public class ControladorInteracao {  
    public void lidarComToggleFavorito(int idElemento) {}  
    public void lidarComTocarElemento(int idElemento) {}  
    public void iniciarEscutaMidi() {}  
    public void pararEscutaMidi() {}  
    public void lidarComCriacaoAnotacao(String texto, Integer idElementoAssociado)  
    {}  
    public void lidarComEdicaoAnotacao(int idAnotacao, String novoTexto) {}  
    public void lidarComExclusaoAnotacao(int idAnotacao) {}  
}
```

```
public class ControladorAnotacao {  
    public Anotacao lidarComCriacao(String texto) {}  
    public void lidarComEdicao(int idAnotacao, String novoTexto) {}  
    public void lidarComAssociacao(int idAnotacao, Integer idElemento) {}  
    public void lidarComExclusao(int idAnotacao) {}  
}
```

```
public class ControladorAudio {  
    public void lidarComTocarElemento(int idElemento) {}  
}
```

```
public class ControladorFavoritos {  
    public void lidarComToggleFavorito(int idElemento) {}}
```

```
public class ControladorMidi {  
    public void iniciarMonitoramento() {}
```

```

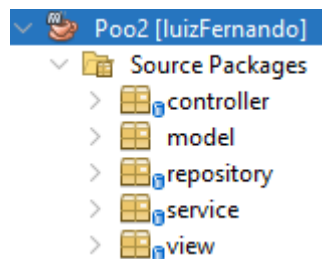
    public void pararMonitoramento() {}
}

```

3.5 MVC

Foram usadas as camadas Model, View e Controller como base e foi expandido para outras camadas mais específicas como Repository, responsável pelo Banco de dados e Service, responsável por aplicar a lógica da camada Model.

Abaixo está uma imagem de como os pacotes ficaram organizados no projeto:



3.6 Banco de Dados

Algumas das principais tabelas:

Query 1

Limit to 1000 rows

1 • SELECT * FROM ACORDE;

<

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	AcordeId	Nome	Tipo	TonicaId	EstruturaNotas	ElementoId
▶	1	C	Maior	1	C - E - G	1
	2	D	Maior	2	D - F# - A	2
	3	E	Maior	3	E - G# - B	3
	4	F	Maior	4	F - A - C	4
	5	G	Maior	5	G - B - D	5
	6	A	Maior	6	A - C# - E	6
	7	B	Maior	7	B - Eb - Gb	7
	8	C	Menor	8	C - Eb - G	8
	9	D	Menor	9	D - F - A	9
	10	E	Menor	10	E - G - B	10
	11	F	Menor	11	F - Ab - C	11
	12	G	Menor	12	G - Bb - D	12
	13	A	Menor	13	A - C - E	13
	14	B	Menor	14	B - D - Gb	14
•	NULL	NULL	NULL	NULL	NULL	NULL

Query 1

Limit to 1000 rows

1

SELECT * FROM MUSICA;

Result Grid

Filter Rows:

Edit

Export/Import:

Wrap Cell Content:

	MusicaId	Titulo	Artista	ANO	Tonica	Estilo
10		Wonderwall	Oasis	1995	G Maior	Rock
11		Highway to Hell	AC/DC	1979	A Maior	Rock
12		Seven Nation Army	The White St...	2003	A Maior	Rock
13		Come As You Are	Nirvana	1991	Dm	Rock
14		Água de Beber	Tom Jobim	1963	Bb Maior	MPB
15		Garota de Ipanema	Tom Jobim	1962	Db Maior	MPB
16		Chega de Saudade	Tom Jobim	1962	Am	MPB
17		Construção	Chico Buarque	1971	Cm	MPB
18		Anunciação	Alceu Valença	1977	E Maior	MPB
19		Cálce	Chico Buarqu...	1973	Dm	MPB
20		Aquarela do Brasil	Ary Barroso	1939	Eb Maior	Samba
21		O Caderno	Toquinho	1983	C Maior	MPB
22		Catedral	Zélia Duncan	1995	G Maior	MPB
23		Wave	Tom Jobim	1967	D Maior	Bossa Nova
24		Corcovado	Tom Jobim	1960	C Maior	Bossa Nova
25		Insensatez	Tom Jobim	1961	Fm	Bossa Nova
26		Desafinado	João Gilberto	1958	F#m	Bossa Nova
27		Samba de Uma Nota Só	Tom Jobim	1961	C Maior	Bossa Nova
28		O Morro Não Tem Vez	Tom Jobim	1960	C Maior	Bossa Nova

MUSICA 7

Apply

Query 1 x

Limit to 1000 rows

1 • SELECT * FROM ESCALA3

Result Grid

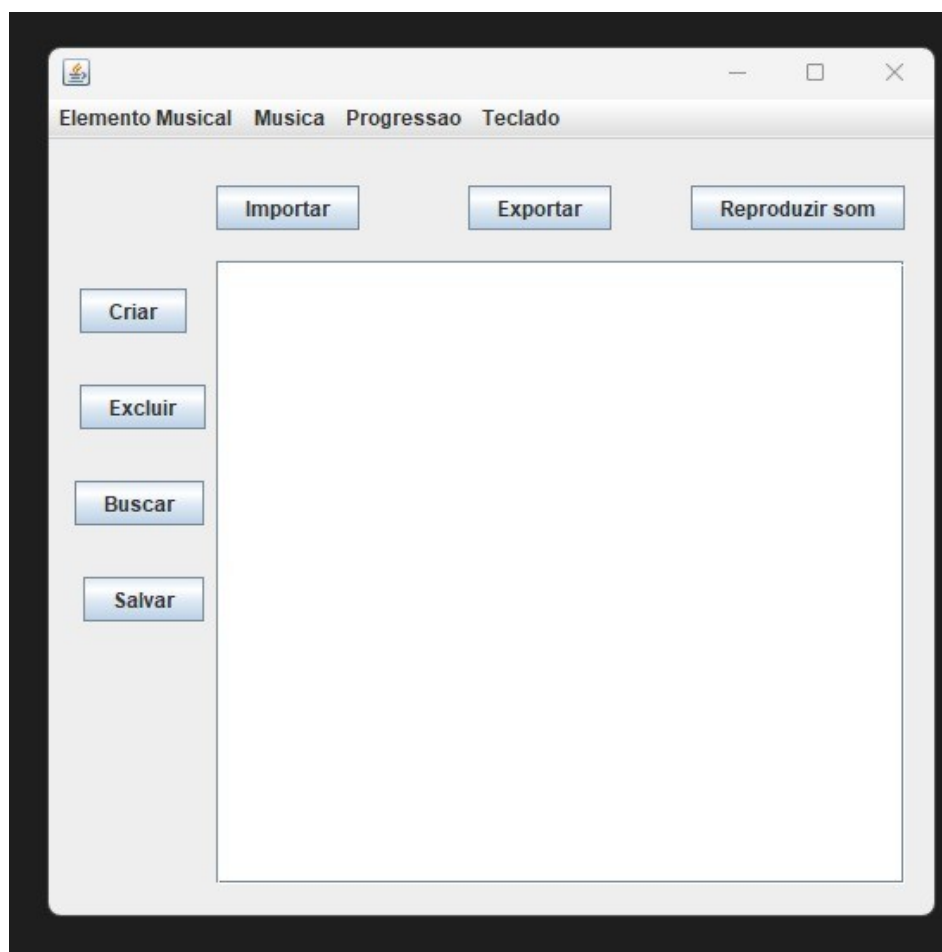
EscalaId	Estrutura	ElementoId	Tipo	Nome
1	C-D-E-F-G-A-B	1	Maior	C
2	D-E-F#-G-A-B-C#	2	Maior	D
3	E-F#-G#-A-B-C-D#	3	Maior	E
4	F-G-A-Bb-C-D-E	4	Maior	F
5	G-A-B-C-D-E-F#	5	Maior	G
6	A-B-C#-D-E-F#-G#	6	Maior	A
7	B-C#-D#-E-F#-G#-A#	7	Maior	B
9	C-D-Eb-F-G-A-Bb	8	Menor	Cm
10	D-E-F-G-A-Bb-C	9	Menor	Dm
11	E-F#-G-A-B-C-D	10	Menor	Em
12	F-G-Ab-Bb-C-Db-Eb	11	Menor	Fm
13	G-A-Bb-C-D-Eb-F	12	Menor	Gm
14	A-B-C-D-E-F-G	13	Menor	Am
15	B-C-D-E-F#-G-A	14	Menor	Bm
*	NULL	NULL	NULL	NULL

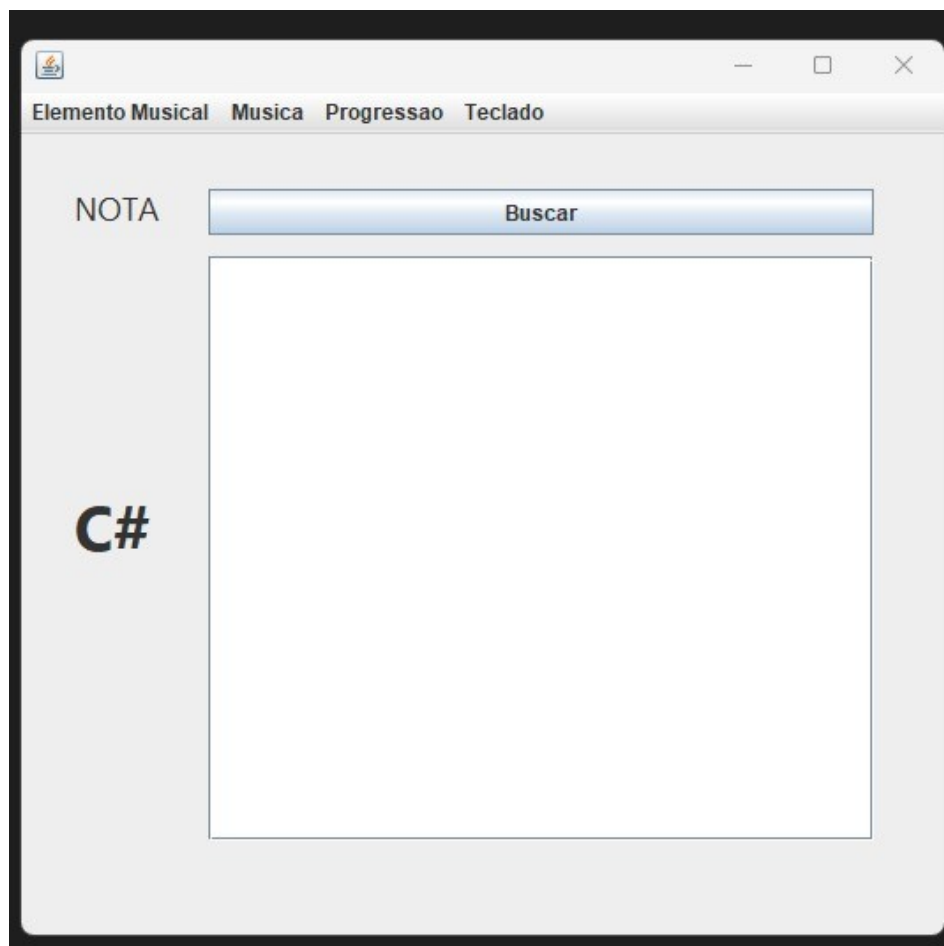
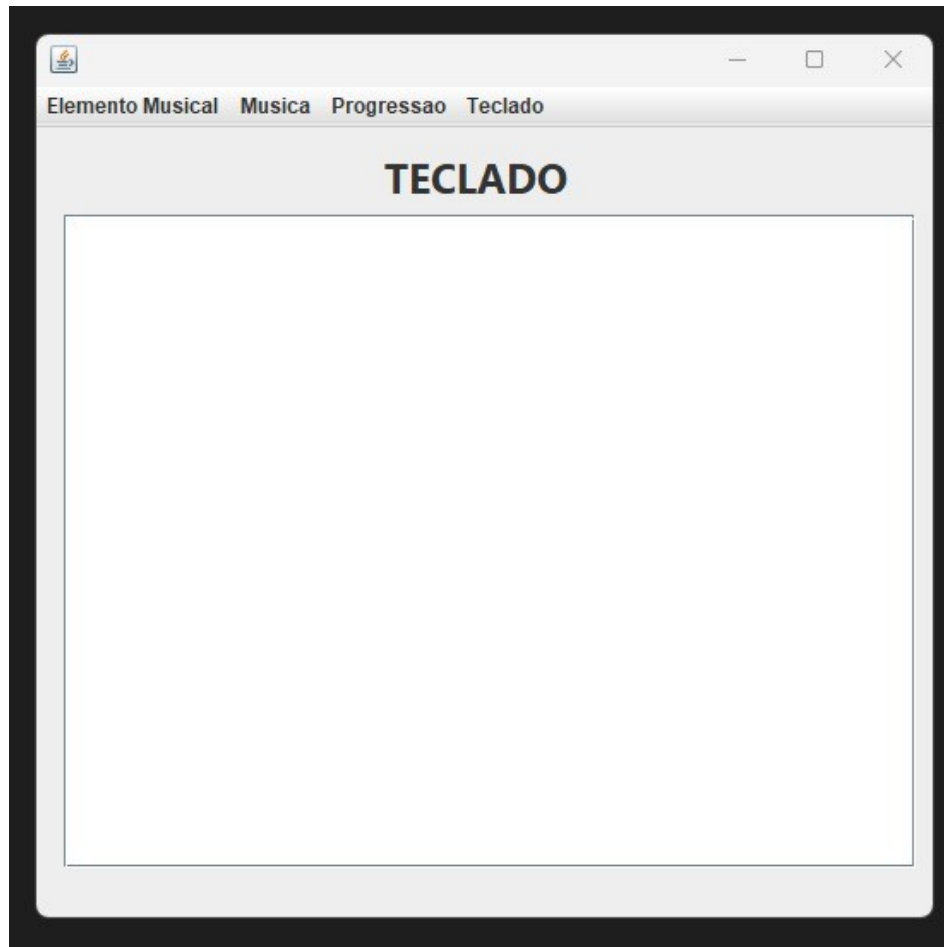
ESCALA 8 x

Apply

3.7 Telas do sistema

As telas selecionadas foram as telas de progressão, do teclado MIDI e da nota.





3.8 Link do repositório

<https://github.com/Lfdomenico/TrabalhoPOO2>

4 CONCLUSÕES

Os maiores aprendizados deste projeto foram saber trabalhar em equipe e também aplicar as boas práticas no código. Em relação às melhorias no sistema, a interface e algumas funções, como a busca de músicas, podem ser aprimoradas e integradas de forma melhor em outras partes do aplicativo. Na divisão de trabalho e organização, poderíamos ter nos organizado melhor e feito um uso mais eficiente do nosso tempo. O maior desafio foi se organizar e dividir o trabalho de forma eficiente, e também na parte das interfaces nós tivemos alguns problemas técnicos.

REFERÊNCIAS

ORACLE. Java Sound Programmer Guide. Disponível em: https://docs.oracle.com/javase/8/docs/technotes/guides/sound/programmer_guide/contents.html. Acesso em: 14 maio 2025.

ORACLE. Interface Receiver (Java SE 8). Disponível em: <https://docs.oracle.com/javase/8/docs/api/javax/sound/midi/Receiver.html>. Acesso em: 14 maio 2025.