

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GABRIEL ALLEGRETTI

**Escala: Um editor de partituras para a *web***

Monografia apresentada como requisito parcial  
para a obtenção do grau de Bacharel em Ciência  
da Computação

Orientador: Prof. Dr. Marcelo de Oliveira Johann

Porto Alegre  
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof<sup>ª</sup>. Patricia Helena Lucas Pranke

Pró-Reitoria de Ensino: Prof<sup>ª</sup>. Cintia Inês Boll

Diretora do Instituto de Informática: Prof<sup>ª</sup>. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter

Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

## RESUMO

Neste trabalho é desenvolvido uma aplicação para a leitura, reprodução e edição de partituras, voltado para estudantes e músicos amadores. Como muitas pessoas não possuem recursos para adquirir um *software* pago, ou simplesmente não têm o sistema operacional compatível, propõe-se uma aplicação gratuita e de código aberto para a *web*. A fim de contextualizar o problema, é revisado brevemente a história dos *softwares* de edição de partituras, assim como as aplicações modernas desse mesmo domínio. Então, as ferramentas e bibliotecas de *software* utilizadas na aplicação são revisadas e explicadas, assim como a arquitetura interna utilizada, preocupações com a experiência do usuário e testes automatizados. Por fim, suas funcionalidades são expostas e comparadas com outras aplicações, demonstrando um nível de completude adequado.

**Palavras-chave:** Plataforma *web*. editor de partitura.

## **Escala: A score editor for the web**

### **ABSTRACT**

This work develops an application for the reading, playing and editing of music scores, aimed at students and amateur musicians. Since many people do not have the resources to acquire paid software, or simply do not have a compatible operating system, a free and open-source application for the web is proposed. In order to contextualize the problem, the history of score editing software is briefly reviewed, as well as modern applications in the same domain. Then, the software tools and libraries used in the application are reviewed and explained, as well as the internal architecture used, concerns with the user experience, and automated tests. Finally, its features are exposed and compared with other applications, demonstrating an adequate level of completeness.

**Keywords:** Web Application. Score Writer.

## LISTA DE FIGURAS

Figura 2.1	Editor SCORE .....	12
Figura 2.2	Editor Concertware .....	13
Figura 2.3	Finale .....	14
Figura 2.4	Sibelius .....	15
Figura 2.5	Tux Guitar .....	15
Figura 2.6	Flat .....	16
Figura 4.1	Notas em uma partitura .....	21
Figura 4.2	Batidas em uma partitura .....	22
Figura 4.3	Vozes e barras em uma partitura .....	23
Figura 4.4	Pauta em uma partitura .....	23
Figura 4.5	Diagrama da arquitetura do editor Escala .....	28
Figura 5.1	Pirâmide de automação de testes .....	33
Figura 5.2	Fluxograma dos testes .....	36
Figura 6.1	Visão geral da aplicação .....	37
Figura 6.2	Cursor selecionando uma nota .....	38
Figura 6.3	Cursor selecionando uma posição válida para uma nota .....	38
Figura 6.4	Cursor sem seleção .....	39
Figura 6.5	Barra das opções de documentos .....	40
Figura 6.6	Edição das informações da partitura .....	41
Figura 6.7	Barra das opções de efeitos .....	43
Figura 6.8	Detalhe das opções de harmônica .....	45
Figura 6.9	Detalhe das opções de bends .....	46
Figura 6.10	Barra das opções de batida .....	51
Figura 6.11	Edição do texto de uma batida .....	51
Figura 6.12	Edição do acorde de Lá menor .....	53
Figura 6.13	Acorde de Lá menor e de Mi na barra de ações .....	53
Figura 6.14	Barra das opções de duração .....	55
Figura 6.15	Barra das opções de dinâmica .....	57
Figura 6.16	Listagem de faixas .....	58
Figura 6.17	Edição de uma faixa .....	59
Figura 6.18	Outras opções .....	60
Figura 6.19	Janela de atalhos .....	61
Figura 6.20	Menu de ajuda .....	62
Figura 6.21	Controles do reproduutor .....	63

## LISTA DE TABELAS

Tabela 4.1	Comandos da interface .....	24
Tabela 4.2	Comandos de modificação.....	25
Tabela 4.3	Implementação dos ícones.....	30
Tabela 6.1	Comparativo dos efeitos sobre uma nota.....	49
Tabela 6.2	Comparativo dos efeitos de slide.....	50
Tabela 6.3	Comparativo dos efeitos de harmônica.....	50
Tabela 6.4	Comparativo dos efeitos sobre uma batida.....	54
Tabela 6.5	Comparativo dos efeitos de duração.....	55
Tabela 6.6	Comparativo dos efeitos de dinâmica.....	57
Tabela 6.7	Comparativo do suporte do modo escuro .....	61

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BPM	Batidas por minuto
CI/CD	Continuous Integration/Continuous Delivery
CSS	Cascading Style Sheet
DOM	Document Object Model
HTML	Hyper Text Markup Language
MIDI	Musical Instrument Digital Interface
OCR	Optical Character Recognition
SWT	Standard Widget Toolkit
UI	User Interface
WYSIWYG	What You See Is What You Get
WYSIWYM	What You See Is What You Mean

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>11</b>
<b>2 CONTEXTUALIZAÇÃO DO PROBLEMA</b>	<b>12</b>
<b>2.1 Notação musical com computadores</b>	<b>12</b>
<b>2.2 Softwares de edição de partituras modernos</b>	<b>14</b>
2.2.1 Finale	14
2.2.2 Sibelius	14
2.2.3 Tux Guitar	15
2.2.4 Flat	16
<b>2.3 Requisitos do Escala</b>	<b>17</b>
2.3.1 Requisitos funcionais	17
2.3.2 Requisitos não funcionais	17
<b>3 TECNOLOGIAS UTILIZADAS</b>	<b>18</b>
3.0.1 HTML e CSS	18
3.0.2 Typescript	18
3.0.3 React	18
3.0.4 MUI	18
3.0.5 Vite	19
3.0.6 SMuFL	19
3.0.7 Bravura	19
3.0.8 Jest	19
3.0.9 Cypress	20
3.0.10 AlphaTab	20
<b>4 ESCALA</b>	<b>21</b>
<b>4.1 Modelagem de dados</b>	<b>21</b>
4.1.1 Nota	21
4.1.2 Batida	21
4.1.3 Voz	22
4.1.4 Barra	22
4.1.5 Pauta	22
4.1.6 Faixa	22
4.1.7 Partitura	23
<b>4.2 Arquitetura</b>	<b>24</b>
4.2.1 Comandos da interface	24
4.2.2 Comandos de modificação	24
4.2.3 Comando do efeito da modificação	25
4.2.4 Agrupamento de comandos	26
4.2.5 Exemplo dos passos necessários para a remoção de uma nota	27
4.2.6 Integração com a biblioteca AlphaTab	27
<b>4.3 Interface do usuário</b>	<b>28</b>
4.3.1 UI/UX	28
4.3.2 Ícones	29
4.3.3 Modo escuro	31
<b>4.4 Implementação</b>	<b>31</b>
4.4.1 Localização do idioma	31
4.4.2 Atalhos do teclado	32
4.4.3 Desfazer e refazer ações	32
<b>5 AUTOMAÇÃO DE TESTES</b>	<b>33</b>
<b>5.1 Objetivos</b>	<b>33</b>

<b>5.2</b>	<b>Categorias de testes.....</b>	<b>33</b>
5.2.1	Testes unitários.....	33
5.2.2	Testes de Serviço.....	34
5.2.3	Testes de UI.....	34
<b>5.3</b>	<b>Integração com o controle de versão .....</b>	<b>35</b>
<b>6</b>	<b>FUNCIONALIDADES .....</b>	<b>37</b>
<b>6.1</b>	<b>Cursor .....</b>	<b>38</b>
<b>6.2</b>	<b>Operações no documento .....</b>	<b>40</b>
6.2.1	Novo arquivo.....	40
6.2.2	Abrir Arquivo.....	40
6.2.3	Informações da partitura .....	41
6.2.4	Desfazer e Refazer .....	42
6.2.5	Imprimir .....	42
6.2.6	Download .....	42
6.2.7	Tempo .....	42
<b>6.3</b>	<b>Operações de efeito sobre uma nota.....</b>	<b>43</b>
6.3.1	Nota Abafada .....	43
6.3.2	Deixe Soar.....	43
6.3.3	Nota Fantasma .....	44
6.3.4	Nota Morta.....	44
6.3.5	<i>Tapping</i> .....	44
6.3.6	<i>Vibrato</i> .....	44
6.3.7	Harmônica.....	44
6.3.8	<i>Hammer On/Pull off</i> .....	45
6.3.9	Nota acentuada & Nota muito acentuada.....	46
6.3.10	<i>Bend</i> .....	46
6.3.11	<i>Slide</i> .....	47
6.3.12	<i>Staccato</i> .....	48
<b>6.4</b>	<b>Operações sobre uma batida.....</b>	<b>51</b>
6.4.1	Texto.....	51
6.4.2	<i>Fermata</i> .....	52
6.4.3	<i>Upstroke e Downstroke</i> .....	52
6.4.4	Acorde.....	52
6.4.5	Repetição.....	53
<b>6.5</b>	<b>Operações na duração .....</b>	<b>55</b>
6.5.1	Duração .....	55
6.5.2	Nota ligada .....	56
<b>6.6</b>	<b>Operações na dinâmica.....</b>	<b>57</b>
<b>6.7</b>	<b>Faixas de música .....</b>	<b>58</b>
6.7.1	Nome da faixa .....	58
6.7.2	Afinação .....	58
6.7.3	Capotraste .....	59
<b>6.8</b>	<b>Outros.....</b>	<b>60</b>
6.8.1	Atalhos no teclado.....	60
6.8.2	Modo escuro.....	60
6.8.3	Seleção de Idioma .....	61
6.8.4	Menu de Ajuda.....	61
<b>6.9</b>	<b>Controle do reprodutor .....</b>	<b>63</b>
6.9.1	Tocar/Parar .....	63
6.9.2	<i>Count In</i> .....	63
6.9.3	Metrônomo.....	63

6.9.4 Repetir.....	64
6.9.5 Volume .....	64
6.9.6 Velocidade de reprodução .....	64
<b>7 CONCLUSÃO .....</b>	<b>65</b>
<b>REFERÊNCIAS.....</b>	<b>66</b>

## 1 INTRODUÇÃO

O ecossistema atual de editores de tablaturas pode se mostrar um tanto difícil de navegar para principiantes. Existem dezenas de editores de tablaturas disponíveis para se escolher (WIKIPEDIA, 2022b). A grande maioria são de aplicações *desktop*, e muitas limitadas a um ou dois sistemas operacionais diferentes, e muitos dos *softwares* não são gratuitos. (WIKIPEDIA, 2022a).

Também é importante notar que nem todos possuem um computador próprio ou possuem um usuário com permissão para instalar aplicações *desktop* (laboratórios de informática, bibliotecas). Estudantes com Chromebooks também têm menos opções, pois aplicações *desktop* convencionais não são suportadas e nem todas as aplicações Android funcionam no aparelho. Além disso, em algumas situações, o estudante pode não ter acesso à loja Google Play.

As opções de editores para a *web* não requerem instalação, mas são mais limitadas e pouco satisfatórias comparadas com as versões *desktop*, seja por possuírem menos funcionalidades ou por serem pagas. Observando essas limitações e a falta de opções, surgiu a ideia do trabalho, que tenta preencher a falta de um editor para a *web*, gratuito e de *software* livre, focado em estudantes e músicos amadores.

Apesar de uma solução com diversas funcionalidades levar muitos anos de desenvolvimento, neste trabalho demonstramos que com a tecnologia da aplicação *web* é possível implementar uma grande quantidade de funções com boa qualidade para músicos amadores, com uma arquitetura de *software* sólida.

Este trabalho está dividido em 7 capítulos. O capítulo 2 aborda um pouco da história da notação musical, editores de tablaturas modernos e define os objetivos do editor Escala. O capítulo 3 introduz as tecnologias usadas no desenvolvimento da aplicação. O Capítulo 4 apresenta uma visão geral da arquitetura e padrões de projeto utilizados na implementação, e o 5 a automação de testes. O capítulo 6 detalha todas as funcionalidades implementadas e compara com outros editores, e o capítulo 7 apresenta as conclusões e possíveis melhorias futuras.

## 2 CONTEXTUALIZAÇÃO DO PROBLEMA

### 2.1 Notação musical com computadores

Desde os primeiros computadores *mainframes*, existiram com projetos com o objetivo de imprimir músicas automaticamente, mas eles eram apenas simples instruções para uma impressora com os comandos de impressão, sem realizar o cálculo do posicionamento dos símbolos. Um caso foi o ILLIAC I da Universidade de Illinois por volta de 1952 (BYRAM-WIGFIELD, 2021).

Um marco notável foi o primeiro *software* comercial para escrita e impressão de partituras amplamente conhecido, o SCORE (Figura 2.1). Seu desenvolvimento foi iniciado em 1967 por Leland C Smith da Universidade de Stanford. Leland publicou o livro *Handbook of Harmonic Analysis*, em 1979. Foi o primeiro livro com partituras impressas completamente via *software* (ELSEA, ; SMITH, ).

É um editor WYSIWYM (*What You See Is What You Mean*), onde a partitura era inserida via texto, usando uma linguagem de definição própria. Apesar de poderoso, foi criticado por ser muito difícil de se usar e ter um tempo de aprendizado longo (PCMA-GAZINE, 1988).

Figura 2.1: Editor SCORE

The screenshot shows the SCORE music editor interface. At the top, the title bar displays 'GTR1.PGE', 'Items=505', 'Zoom= 1.00', and page numbers '5112/14200 13883/31000 (SAVED)'. Below the title bar, the main window contains the following text and musical notation:

\*\*\* Measurements = INCHES  
>\_

to Alan Goldspiel  
**RHAPSODY FOR GUITAR**  
William Holab

Fingering by Alan Goldspiel  
Slowly, doloroso  
cantabile

Allegro, d = ca. 120  
sub.

amiami

At the bottom of the window, a command line shows function key shortcuts: F1-HELP, F2-INPUT, F3-EDIT, F4-TEXT, F5-PLAY, F6-PRINT, F7-DISK, F8-EXIT.

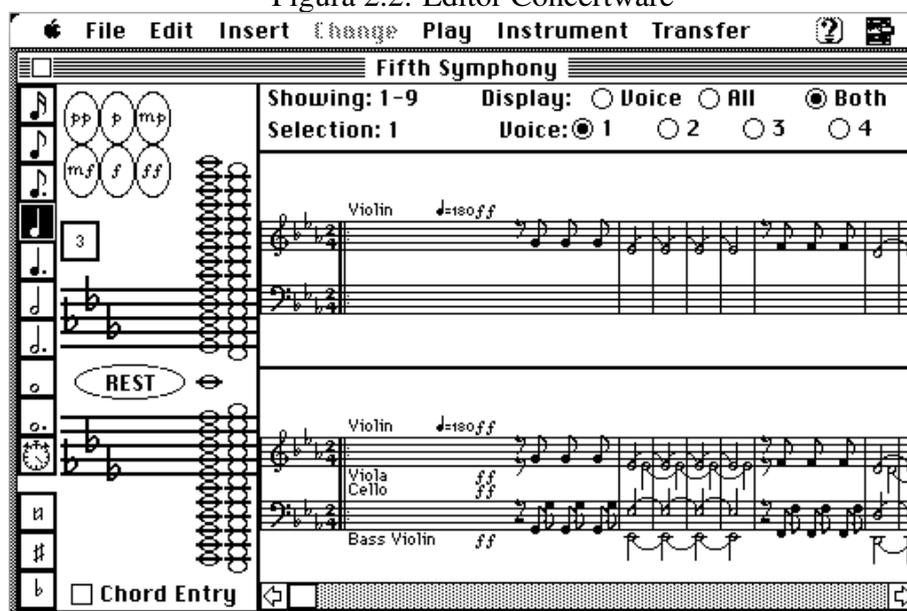
Fonte: Scoring Notes

Já na década de 80, dois pesquisadores da Xerox PARC, Severo M. Ornstein e John

T. Maxwell criaram o Mockingbird. Considerado o primeiro editor WYSIWYG (*What You See Is What You Get*), permitia que a partitura fosse editada diretamente na forma que seria impressa, com o auxílio do mouse. Na mesma década surgiu o MIDI (*Musical Instrument Digital Interface*), um padrão desenvolvido para sintetizadores e equipamentos musicais, que também foi incorporado no Mockingbird.

Com o advento de computadores pessoais mais poderosos, como o primeiro Mac, começaram a surgir mais *softwares* comerciais. Um exemplo foi o *ConcertWare* (Figura 2.2), desenvolvido por Dr. Chad Mitchell e lançado em 1984 (ELSEA, ), que também adotava o WYSIWYG.

Figura 2.2: Editor Concertware



Fonte: Scoring Notes

Um problema que cresceu ao longo dos anos foi a fragmentação do formato dos dados que causavam a pouca interoperabilidade entre esses *softwares*. Por volta dos anos 2000, ocorreu o início do desenvolvimento do MusicXML, com o objetivo de ser um formato comum entre diversos *softwares* de visualização, reprodução e síntese de músicas. Teve sua primeira versão publicada em 2004 e até hoje é um formato muito popular, com mais de 270 aplicações que utilizam o formato (MUSICXML..., 2022).

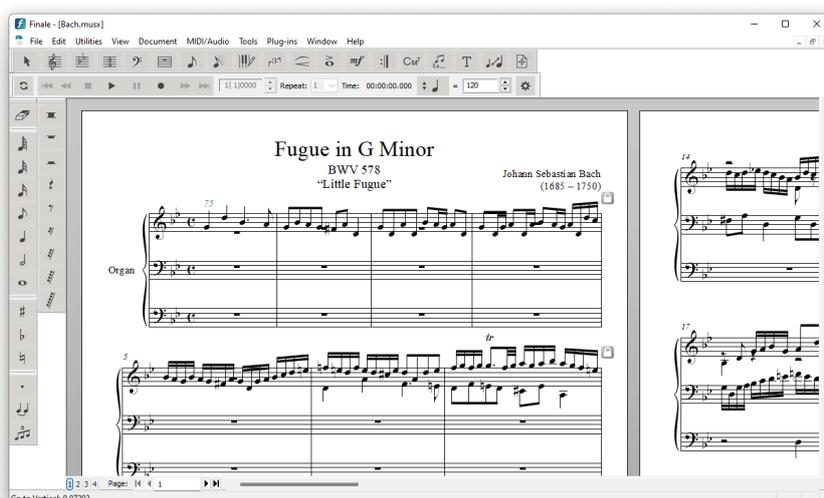
Editores do tipo WYSIWYG com sintetizadores e suporte ao formato MusicXML eventualmente dominaram o mercado e hoje são encontrados em todas as ferramentas profissionais como o Finale e o Sibelius, assim como editores gratuitos como o TuxGuitar e o Flat.

## 2.2 Softwares de edição de partituras modernos

### 2.2.1 Finale

Finale é um editor e reproduzidor para MacOS e Windows. Escrito em C++ e publicado pela primeira vez em 1988, é um dos principais *softwares* de composição do mercado. Atualmente custa entre \$100 e \$300 dólares. Por ser pago, ter foco em profissionais e não ter suporte a *web*, não será usado nas comparações deste trabalho (FINALE..., 2022).

Figura 2.3: Finale



Fonte: Finale

### 2.2.2 Sibelius

Sibelius é um editor e reproduzidor para MacOS e Windows. Escrito em C++ e publicado pela primeira vez em 1993, também é um dos principais *softwares* de composição do mercado. Subscrições anuais custam a partir de \$84 dólares. Por ser pago, ter foco em profissionais e não ter suporte a *web*, não será usado nas comparações deste trabalho (SIBELUS..., 2022).

Figura 2.4: Sibelius

Fonte: Sibelius

## 2.2.3 Tux Guitar

TuxGuitar é um editor e reproduzidor gratuito e de código aberto para Linux, Windows, macOS e FreeBSD. É escrito em Java, utilizando a biblioteca de componentes SWT (*Standard Widget Toolkit*). Não é muito claro quando seu desenvolvimento começou, mas teve sua primeira versão lançada em meados de 2005 (TUXGUITAR, 2022).

Figura 2.5: Tux Guitar

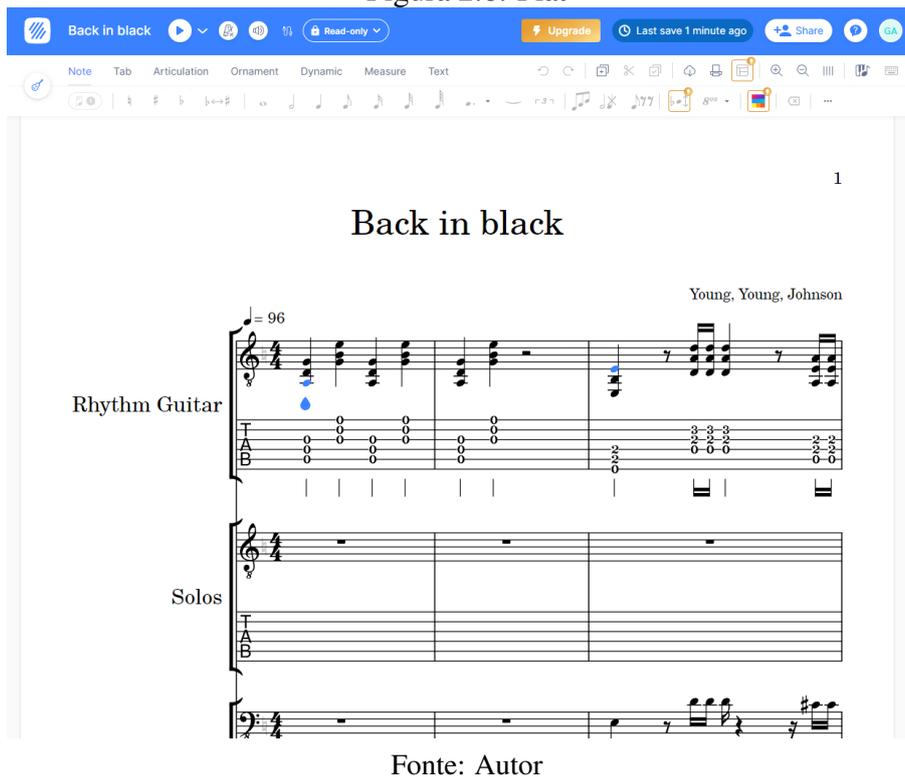
N°	S-M	Nome	Instrumento	Back in black
1	-	Rhythm Guitar	Overdrive Guitar 1	■
2	-	Solos	Distortion Guitar 1	■
3	-	Bass	Fingered Bass 1	■
4	-	Bass Drum	Piano 1	■

Fonte: Autor

## 2.2.4 Flat

Flat é um editor e reproduzidor proprietário para a *web*. Teve seu desenvolvimento iniciado por volta de 2012 e ainda é mantido e desenvolvido até os dias atuais. Possui uma versão gratuita e um plano pago, que disponibiliza funcionalidades extras (FLAT..., 2022).

Figura 2.6: Flat



The image shows a screenshot of the Flat web application interface. The top navigation bar includes the title "Back in black", a play button, a "Read-only" dropdown, an "Upgrade" button, and a "Last save 1 minute ago" indicator. Below the navigation bar is a toolbar with various editing tools. The main content area displays a guitar score for the song "Back in black" by Young, Young, Johnson. The score is for a Rhythm Guitar and includes a Solos section. The tempo is marked as = 96. The score is displayed in 4/4 time and includes a bass line. The source is cited as "Fonte: Autor".

## 2.3 Requisitos do Escala

Para o desenvolvimento do projeto, foram elencados alguns requisitos funcionais, que definem funcionalidades que o sistema deve ter, e não funcionais, que incluem restrições e outras necessidades do sistema. Na definição desses requisitos, utilizou-se do estudo dos editores existentes, assim como os problemas endereçados na introdução.

### 2.3.1 Requisitos funcionais

- O editor deve ser do tipo WYSIWYG (*What You See Is What You Get*).
- O editor deve permitir abrir e exportar partituras.
- O editor deve permitir criar, editar e excluir notas e elementos musicais.
- O editor deve permitir a reprodução de partituras.

### 2.3.2 Requisitos não funcionais

- O editor deve ser de fácil uso.
- O editor deve permitir a edição de partituras de maneira fluída e rápida.
- Devem existir testes automatizados.
- O código deve ser licenciado de maneira *Open source*.
- O editor deve ser acessível através de navegadores *web*.
- A aplicação deve ser disponibilizada gratuitamente em um endereço *web*.

## 3 TECNOLOGIAS UTILIZADAS

### 3.0.1 HTML e CSS

HTML (*Hyper Text Markup Language*) é a linguagem fundamental da *web*, usada para definir o significado e a estrutura do conteúdo da página. CSS (*Cascading Style Sheet*) é o que permite a customização visual da página.

### 3.0.2 Typescript

Desenvolvida pela Microsoft, foi lançada publicamente em 2012. É uma linguagem que compila para Javascript, podendo assim rodar em navegadores *web*. Suas principais vantagens em relação a usar Javascript diretamente são tipagem estática, inferência de tipos e mais. (MICROSOFT, 2022)

### 3.0.3 React

Também conhecido como React.js ou ReactJS, é uma biblioteca para criação de aplicações *web* desenvolvida pelo Facebook e lançada inicialmente em 2013. Seu sucesso foi tanto que segundo a pesquisa do Stack Overflow de 2022 (STACKOVERFLOW, 2022), é a tecnologia de desenvolvimento frontend mais popular do momento, ficando na frente de jQuery e Angular.

Um dos motivos é a sua arquitetura declarativa e funcional: em vez de programar como a tela deve ser atualizada, define-se apenas como a tela deveria estar baseada no seu estado atual.

Outro conceito é o do Componente, um pedaço da UI (*User Interface*) que encapsula lógica de exibição e comportamento, implementando o padrão estrutural de objetos Composite (GAMMA et al., 2000), tornando fácil a reutilização de elementos visuais.

### 3.0.4 MUI

Abreviação de 'Material UI' é um conjunto de componentes implementados em React que utiliza o Material Design (GOOGLE, 2022). Botão, Entrada de texto, Controle

deslizante, Seletor e Divisor são alguns dos componentes utilizados no projeto.

### 3.0.5 Vite

É um conjunto de ferramentas utilizadas para o desenvolvimento de aplicações *web*. Suas principais funcionalidades são:

- *Hot Module Replacement*: Troca, adição ou remoção dos módulos da aplicação durante sua execução, diminuindo o tempo entre uma alteração do código fonte e o *feedback* para o desenvolvedor.
- Otimização: O código final é minificado e empacotado em um número menor de arquivos, resultando em um tempo de carregamento menor da aplicação final.
- Extensibilidade: É possível utilizar de maneira conjunta com o React e Typescript.

### 3.0.6 SMuFL

Abreviação de ‘Standard Music Font Layout’, é uma especificação que mapeia milhares de símbolos musicais para uma área privada do Unicode (TECHNOLOGIES, 2022). Assim como existem diversas fontes para texto, é interessante que se exista um padrão e fontes diferentes para símbolos musicais.

### 3.0.7 Bravura

É a fonte de referência que implementa o SMuFL, desenvolvida por Daniel Spreadbury. Foi utilizada em diversos ícones da aplicação.

### 3.0.8 Jest

Framework de testes que funciona com o Typescript. Utilizado para a implementação dos testes unitários e de UI.

### 3.0.9 Cypress

Ferramenta para criação e automação de testes de integração no navegador. Utilizado de maneira conjunta com o Jest, é utilizado para a implementação dos testes de UI.

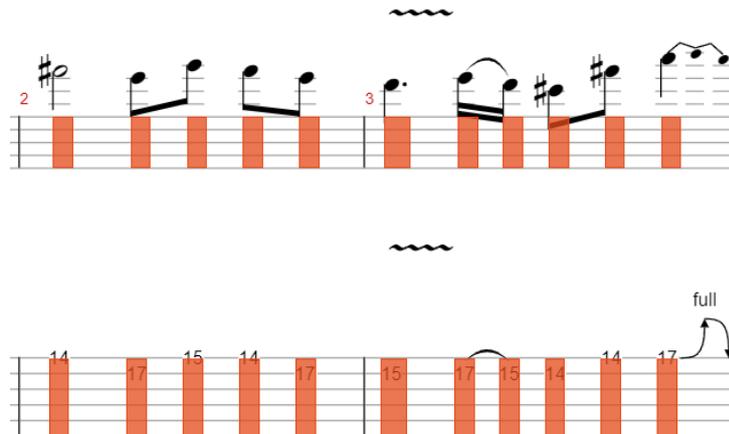
### 3.0.10 AlphaTab

É uma biblioteca que permite a renderização e reprodução de partituras em navegadores da *web*. Escrita em Typescript, possui diversas funcionalidades que influenciaram na escolha para o projeto: (ALPHATAB, 2022).

- Renderização responsiva: Dependendo da resolução da tela, a renderização se adapta para preencher o espaço disponível.
- Reprodução: Contém um sintetizador MIDI que permite a reprodução da tablatura.
- Importação de tablaturas: Importa arquivos no formato do Guitar Pro e MusicXML.
- API (*Application Programming Interface*): Expõe seus dados internos, permitindo sua customização, extensão e alteração da partitura após ela ser carregada.



Figura 4.2: Batidas em uma partitura



Fonte: Alphatab.net

### 4.1.3 Voz

Uma voz é uma sequência de batidas que devem ser reproduzidas consecutivamente. Múltiplas vozes permitem que uma única barra tenha notas independentes com ritmos distintos.

### 4.1.4 Barra

Uma barra, também conhecida como medida, pode possuir uma ou mais vozes que devem ser reproduzidas ao mesmo tempo. Um exemplo é demonstrado na Figura 4.3, com quatro barras, cada uma com uma voz.

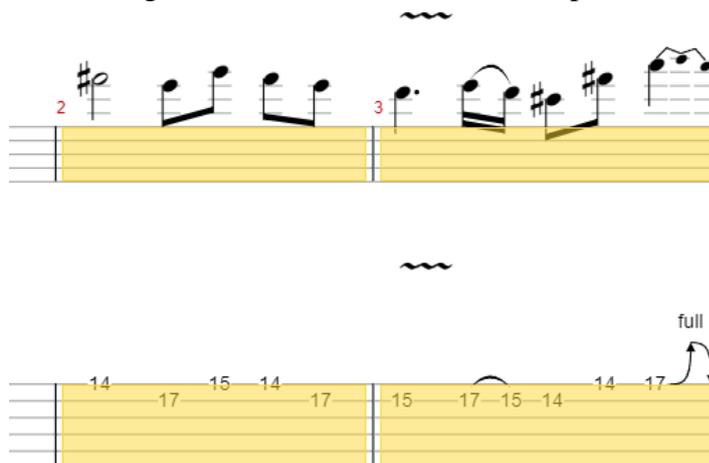
### 4.1.5 Pauta

Uma pauta contém uma ou mais barras que devem ser reproduzidas consecutivamente. Um exemplo de uma pauta simples pode ser visualizado na Figura 4.4.

### 4.1.6 Faixa

Representa um instrumento que deve ser tocado na partitura. Cada faixa contém uma ou mais pautas, além de definir o instrumento e afinação.

Figura 4.3: Vozes e barras em uma partitura



Fonte: Alphatab.net

Figura 4.4: Pauta em uma partitura

O exemplo mostra uma partitura com uma pauta musical e uma tablatura de guitarra. A pauta musical está em 4/4 com uma velocidade de 120 batidas por minuto. A tablatura mostra as cordas T, A e B com fretas 1, 2, 3 e 4.

Fonte: Autor

#### 4.1.7 Partitura

Uma partitura, por fim, contém uma ou mais faixas, que devem ser reproduzidas ao mesmo tempo. É a raiz do modelo da partitura, que também contém informações como o nome dela, autor e artistas.

## 4.2 Arquitetura

O sistema utiliza extensamente o padrão de projeto Command. É um padrão comportamental que “encapsula uma solicitação como um objeto, desta forma permitindo parametrizar clientes com diferentes solicitações, enfileirar ou fazer o registro de solicitações e suportar operações que podem ser desfeitas” (GAMMA et al., 2000).

### 4.2.1 Comandos da interface

São comandos gerados pela interação do usuário com a aplicação, via mouse ou teclado. Tais comandos podem gerar zero ou mais *Comandos de modificação*. Os comandos podem ser vistos na tabela 4.1.

Tabela 4.1: Comandos da interface

<i>Nome</i>	<i>Comandos de modificação</i>
Selecionar nota	Não gera comandos
Deletar nota	Gera um comando
Desselecionar nota	Não gera comandos
Selecionar corda	Não gera comandos
Mover cursor para baixo	Não gera comandos
Mover cursor para esquerda	Não gera comandos
Mover cursor para direita	Gera zero ou mais comandos
Mover cursor para cima	Não gera comandos
Mover cursor para a próxima Barra	Não gera comandos
Mover cursor para a Barra anterior	Não gera comandos
Desfazer ação	Gera um comando
Refazer ação	Gera um comando
Ação da barra superior de ferramentas	Gera zero ou mais comandos
Ação da barra lateral de faixas	Gera zero ou mais comandos
Ação da barra inferior de controle do reprodutor	Não gera comandos

Fonte: Autor

### 4.2.2 Comandos de modificação

É um comando que encapsula uma modificação a ser aplicada na partitura. São sempre gerados a partir de um *Comando da interface*, mas também dependem do estado atual da aplicação. A tabela 4.2 lista os comandos implementados.

Tabela 4.2: Comandos de modificação

<i>Nome</i>	<i>Pode ser desfeito</i>	<i>Renderiza</i>	<i>MIDI</i>	<i>Testes Unitários</i>
Cria barra	Sim	Sim	Sim	Não
Cria batida	Sim	Sim	Sim	Sim
Cria nota	Sim	Sim	Sim	Sim
Edita faixa	Não	Sim	Sim	Não
Remove nota	Sim	Sim	Sim	Sim
Remove faixa	Não	Sim	Sim	Sim
Muda acentuação	Sim	Sim	Sim	Não
Muda <i>bend</i>	Sim	Sim	Sim	Sim
Muda acorde	Sim	Sim	Sim	Sim
Finaliza repetição	Sim	Sim	Sim	Não
Muda nota morta	Sim	Sim	Sim	Não
Muda duração	Sim	Sim	Sim	Sim
Muda dinâmica	Sim	Sim	Sim	Sim
Muda fermata	Sim	Sim	Sim	Sim
Muda casa	Sim	Sim	Sim	Não
Muda nota fantasma	Sim	Sim	Sim	Não
Muda <i>hammer</i>	Sim	Sim	Sim	Sim
Muda harmônica	Sim	Sim	Sim	Não
Muda deixe soar	Sim	Sim	Sim	Não
Inicia repetição	Sim	Sim	Sim	Não
Muda nota abafada	Sim	Sim	Sim	Sim
Muda palhetada	Sim	Sim	Sim	Não
Muda informações	Não	Sim	Não	Sim
Muda slide	Sim	Sim	Sim	Não
Muda slide out	Sim	Sim	Sim	Não
Muda slide in	Sim	Sim	Sim	Não
Muda <i>staccato</i>	Sim	Sim	Sim	Não
Muda <i>tap</i>	Sim	Sim	Sim	Não
Muda tempo	Não	Sim	Sim	Não
Muda texto	Sim	Sim	Não	Sim
Muda nota ligada	Sim	Sim	Sim	Sim
Muda vibrato	Sim	Sim	Sim	Não

Fonte: Autor

### 4.2.3 Comando do efeito da modificação

É um comando que encapsula o resultado da modificação da partitura, gerado após a execução de um *Comando de modificação*. Como a atualização do modelo da partitura é desacoplada de sua renderização e da atualização da faixa MIDI, esse comando contém as seguintes informações referentes a quais dados precisam ser recomputados:

1. É necessário renderizar a partitura.
2. É necessário atualizar a geração do MIDI.

Por exemplo, o comando *Muda texto* que atualiza o texto de uma batida vai requerer a renderização da partitura, mas não será necessária a atualização do MIDI, enquanto o comando *Muda slide*, vai requerer a renderização e a atualização do MIDI. A relação completa de quais atualizações cada comando precisa é disponibilizada na tabela 4.2.

#### 4.2.4 Agrupamento de comandos

Em alguns casos, um único *Comando da Interface* pode gerar mais de um *Comando de modificação*, que após serem executados, vão gerar no total mais de um *Comando do efeito da modificação*. Por exemplo, o comando da interface de *Mover o cursor para a direita* pode necessitar a criação de uma nova batida na pauta (comando de *Cria batida*), caso a nota selecionada esteja na última batida. Porém, se a batida for a última da barra, é necessário criar uma barra antes de criar a batida (comando de *Cria barra*).

Nesse caso, a partitura seria atualizada duas vezes, primeiramente pela criação da barra nova, e em seguida pela criação da batida nova. A fim de evitar uma latência maior entre a ação do usuário e a visualização da mudança, se realiza uma otimização de agrupamento sobre os *Comandos do efeito da modificação*, uma técnica normalmente conhecida como *batching*.

A lógica de agrupar essas mensagens é extremamente fácil de se implementar, e pode ser descrita da seguinte forma:

1. É necessário renderizar a partitura se pelo menos uma das mensagens requer a sua atualização
2. É necessário atualizar a geração do MIDI se pelo menos uma das mensagens requer sua atualização

Seguindo o mesmo exemplo anterior, com o uso dessa técnica é necessário apenas uma renderização da partitura e atualização da faixa MIDI para os dois *Comandos de modificação*.

#### 4.2.5 Exemplo dos passos necessários para a remoção de uma nota

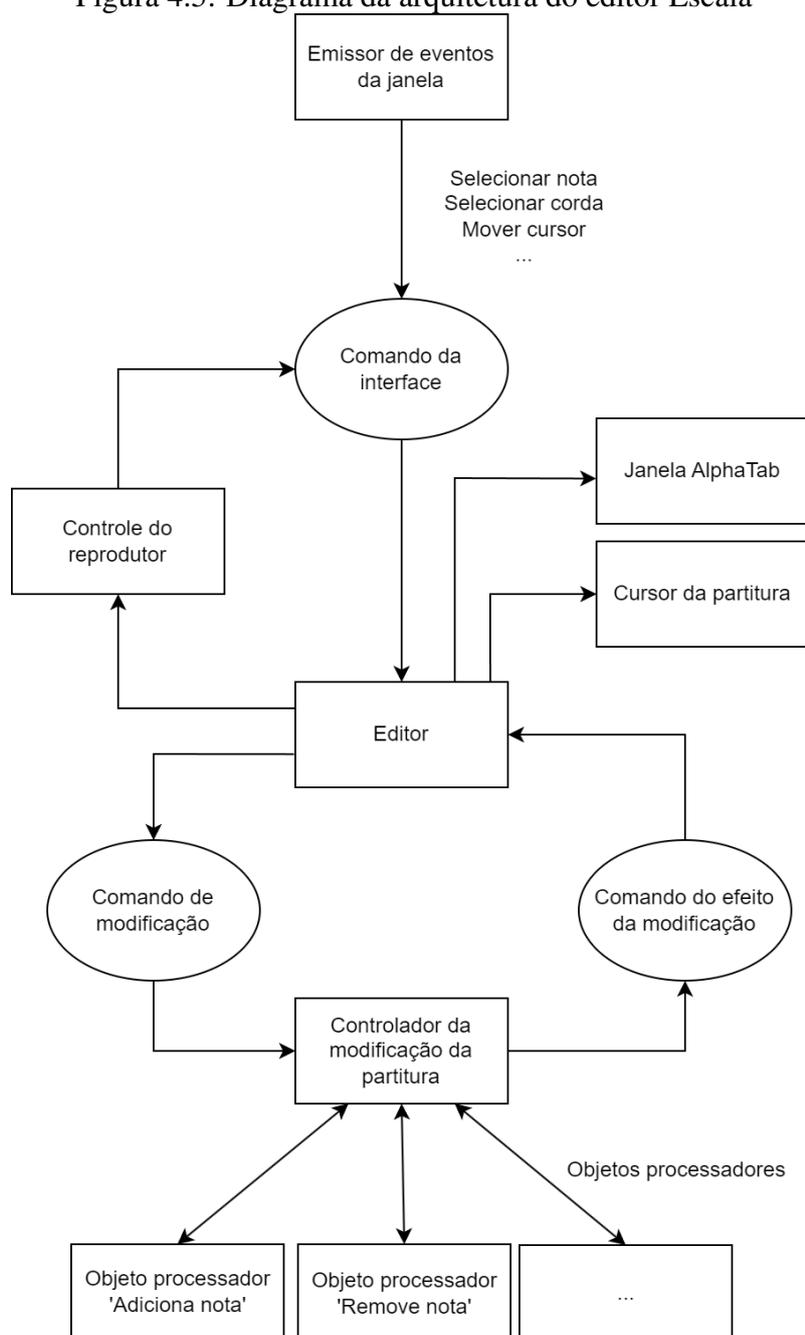
Para exemplificar os diferentes tipos de mensagens na aplicação, demonstra-se um caso de remoção de uma nota da partitura, que pode ser acompanhado com o diagrama da arquitetura apresentada na Figura 4.5.

1. O usuário pressiona a tecla '*Delete*' do seu teclado.
2. O Emissor de eventos da janela detecta a tecla pressionada e gera um *Comando da interface* do tipo 'Deletar nota' e envia para o 'Editor'.
3. Baseado no estado atual da aplicação (assumimos que é de uma nota selecionada como demonstrado na Figura 7.2), o 'Editor' gera um *Comando de modificação* do tipo 'Remove note' com uma referência a essa nota.
4. O Objeto processador de remoção da nota é executado, e gera um *Comando do efeito da modificação*, com a informação de que é necessário uma renderização nova da partitura e atualização da sua faixa MIDI.
5. Baseado nesta mensagem, o Editor realiza as atualizações necessárias na Janela AlphaTab.

#### 4.2.6 Integração com a biblioteca AlphaTab

Um dos desafios no desenvolvimento da aplicação Escala foi a integração com a biblioteca AlphaTab, responsável pela renderização e reprodução da partitura. Apesar de ser possível, seu código não foi feito com a intenção de permitir alterações no modelo da partitura após o seu carregamento inicial. Problemas como um *lookuptable* das notas de uma batida que não era atualizado quando uma nota era removida, ou uma detecção errada da nota clicada foram encontrados.

Figura 4.5: Diagrama da arquitetura do editor Escala



Fonte: Autor

### 4.3 Interface do usuário

#### 4.3.1 UI/UX

Ao planejar a interface, optou-se em agrupar os controles em funcionalidades em quatro grandes áreas:

- Controles do editor

- Faixas da partitura
- Renderização da partitura
- Controles do reprodutor

A área dos controles do editor precisou ser dividida novamente devido ao grande número de ações disponíveis (48 no total). Dividiu-se em ‘Documento’, ‘Efeitos’, ‘Batida’, ‘Duração’ e ‘Dinâmica’:

- Documento: Ações que operam na partitura como um todo.
- Efeitos: Ações que operam sobre uma nota.
- Batida: Ações que operam sobre uma única batida.
- Duração: Ações que operam a duração de uma única nota.
- Dinâmica: Ações que operam sobre a dinâmica de uma única nota.

O resultado pode ser observado na Figura 6.1.

#### 4.3.2 Ícones

Inicialmente pensou-se em utilizar ícones prontos para o projeto. Porém, não foi encontrado um único pacote de ícones gratuito e que abrangesse todas as funcionalidades implementadas. Então, foi utilizado uma combinação de ícones prontos, do *Material Design*, com ícones próprios.

O *Material Design* é um padrão de *Design System Open Source* criado pela Google (GOOGLE, 2022). Ele define diretrizes sobre cores, espaçamentos, acessibilidade e a disposição dos elementos. Além disso, é disponibilizado uma biblioteca de ícones que seguem essas diretrizes, utilizados na aplicação.

Os ícones próprios tem como objetivo serem visualmente semelhantes aos do *Material Design*, para manter a consistência da interface. Foi utilizada a fonte Bravura, que implementa alguns dos símbolos musicais necessários, além de elementos como linhas e textos para montar o ícone final. A Tabela 4.3 detalha quais ícones foram criados e quais foram usados do *Material Design*.

Tabela 4.3: Implementação dos ícones

<i>Ícone</i>	<i>Implementação</i>
Novo Arquivo	Material Design
Abrir Documento	Material Design
Informações da Partitura	Material Design
Desfazer	Material Design
Refazer	Material Design
Imprimir	Material Design
Exportar	Material Design
Tempo	Material Design
Nota Abafada	Própria
Deixe Soar	Própria
Nota Fantasma	Própria
<i>Tapping</i>	Própria
Vibrato	Própria
Harmônica	Própria
Nota Acentuada	Própria
Nota muito acentuada	Própria
<i>Pre-Bend</i>	Própria
<i>Bend</i>	Própria
<i>Release</i>	Própria
<i>Slide in</i>	Própria
<i>Slide</i>	Própria
<i>Slide out</i>	Própria
<i>Staccato</i>	Própria
Texto	Própria
Fermata	Própria
Toca para cima	Própria
Toca para baixo	Própria
Toca para baixo	Própria
Acorde	Própria
Iniciar Repetição	Própria
Terminar Repetição	Própria
Duração	Própria
Dinâmica	Própria
Atalhos do teclado	Material Design
Modo escuro	Material Design
Idioma	Material Design
Ajuda	Material Design
Tocar	Material Design
Contar	Material Design
Metrônomo	Própria
Repetir	Material Design
Volume	Material Design

Fonte: Autor

### 4.3.3 Modo escuro

O modo escuro tem ganhado popularidade ultimamente, sendo cada vez mais adotado por aplicações e sistemas operacionais. Ele proporciona ao usuário a opção de renderização dos elementos em tons claros sobre tons escuros, em vez do normal onde elementos escuros são renderizados sobre tons claros. Vantagens como redução da fadiga ocular e facilidade em utilizar aplicações em ambientes mal iluminados ou escuros também são descritos em (EISFELD; KRISTALLOVICH, 2020).

Em alguns tipos de painéis, como OLED (*Organic Light-Emitting Diode*), em que cada subpixel gera sua própria luz, ainda existe o benefício de economia de energia. Conforme demonstrado em (DASH; HU, 2021), tal economia pode passar de 60% em alguns casos.

O modo escuro foi implementado utilizando o conceito de temas. Um tema contém definições básicas de como um componente deve ser renderizado, como espaçamentos e cores. O tema pode ser modificado durante a execução da aplicação, por exemplo, quando o modo escuro é alternado entre ligado e desligado conforme explicado no capítulo 6.8.2. Quando isso acontece, todos os componentes são notificados e automaticamente atualizados para refletir o estado atual do tema.

## 4.4 Implementação

### 4.4.1 Localização do idioma

Para facilitar o acesso da aplicação a um público maior, foi realizada a localização da aplicação em inglês e português. O idioma selecionado por padrão é detectado automaticamente baseado nas configurações do navegador do usuário, mas também é possível alterá-lo manualmente, conforme descrito no capítulo 6.8.3. Caso o navegador suporte armazenamento local, a linguagem selecionada será salva para a próxima vez que a aplicação for aberta.

#### 4.4.2 Atalhos do teclado

Em um estudo conduzido na Universidade de Nova Gales do Sul, (PETERSON; SCHUBERT, 2007), participantes gastaram em média 58.1% do tempo acessando menus e ferramentas do *software* de edição de tablatura. Apesar de ser possível que os participantes estivessem concorrentemente decidindo como realizar a composição, é um tempo considerável a ser otimizado.

Foram implementados apenas alguns atalhos, que podem ser vistos na Figura 6.19. O reconhecimento de quando um atalho é executado é feito pelo *Emissor de eventos da janela*, que gera um *Comando da interface*, conforme pode ser visualizado na Figura 4.5.

#### 4.4.3 Desfazer e refazer ações

Existem duas estratégias para se implementar um sistema que permita desfazer e refazer ações. A primeira consiste salvar o estado inteiro da aplicação em memória a cada operação, e apenas trocar entre esses estados quando é realizado uma operação de desfazer ou refazer. A segunda consiste em armazenar quais ações foram realizadas no estado, em vez de armazenar o estado inteiro. Essa foi a estratégia adotada, por necessitar de menos memória para funcionar corretamente.

A implementação desse sistema constitui de uma fila de *Comando de modificação* e um índice da fila. Para o funcionamento do sistema, as seguintes regras se aplicam:

- A fila é inicializada vazia, e o índice indefinido.
- Quando uma nova ação é feita, todos os comandos após o índice são descartados, um novo comando é adicionado ao final da fila e o índice recebe a posição desse último comando.
- Quando uma ação é desfeita, o comando do índice é desfeito, e o cursor volta em uma posição.
- Quando uma ação é refeita, o comando do índice é executado, e o cursor avança em uma posição.
- Caso o índice esteja na última posição da fila, ou seja indefinido, não é possível refazer uma ação.
- Caso o índice esteja na primeira posição da fila, ou seja indefinido, não é possível desfazer uma ação.

## 5 AUTOMAÇÃO DE TESTES

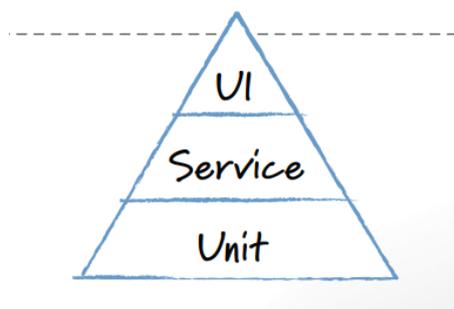
### 5.1 Objetivos

Como disse Beizer, “Os *bugs* se escondem pelos cantos e congregam-se nas fronteiras” (BEIZER, 1990). E a fronteira mais frágil é dividida com o código da biblioteca AlphaTab. Portanto, ao desenvolver os testes automatizados, o objetivo foi priorizar essa área do código.

### 5.2 Categorias de testes

Conforme elaborado em (COHN, 2009), existem três níveis de testes possíveis que podem ser implementados: Unitários, de Serviço e de UI. Segundo o autor, a proporção de testes de cada categoria segue o formato de uma pirâmide, conforme exemplificado na Figura 5.1.

Figura 5.1: Pirâmide de automação de testes



Fonte: (COHN, 2009)

#### 5.2.1 Testes unitários

É um teste realizado sobre uma função, classe ou um método de uma classe, de forma isolada, sem a necessidade da execução da aplicação ou de uso de outros serviços ou componentes. São testes mais rápidos de serem feitos e executados.

Foram criados testes unitários para alguns dos *Objetos processadores* (ver tabela 4.2), que processam um *Comando de modificação*. O pseudocódigo utilizado para implementar cada um dos testes foi o seguinte:

1. Inicializa um *Comando de modificação*.
2. Inicializa um estado inicial para a partitura.
3. Aplica a operação de ‘fazer’ do *Comando de modificação* na partitura.
4. Verifica se o estado final da partitura foi modificado conforme o esperado.
5. Aplica a operação de ‘desfazer’ do *Comando de modificação* na partitura.
6. Verifica se o estado final da partitura é igual ao inicial.

### 5.2.2 Testes de Serviço

Não necessariamente apenas para arquiteturas orientadas a serviços, são testes que utilizam mais de um serviço lógico da aplicação. No contexto da aplicação Escala, existem duas implementações desses testes:

- Múltiplos *Comandos de modificação* executados em sequência.
- Múltiplos *Comando da interface* executados em sequência

### 5.2.3 Testes de UI

Um teste de UI contempla a execução da aplicação completa do ponto de vista de um usuário, simulando interações e checando se o *software* está produzindo o resultado esperado. São testes mais complexos de serem feitos e mais demorados de serem executados. Além disso, pequenas alterações na interface podem facilmente causar que os testes falhem.

O teste deve, em um ou mais momentos, verificar se o *software* produziu o resultado esperado. Para aplicações *web*, existem duas estratégias populares para fazer isso:

- Renderizar a página *web* e observar diretamente os *pixels* gerados, normalmente utilizando-se técnicas de OCR (*Optical Character Recognition*), ou comparando-a com uma versão de referência.
- Renderizar a página *web* e verificar a representação DOM (*Document Object Model*), que é uma estrutura de árvore em que cada nó representa uma parte do documento.

Testes de UI normalmente requerem uma configuração do estado inicial da aplicação, e existem duas estratégias para isso:

- Sequenciais: Cada teste deve ser executado de maneira independente.
- Paralelos: O estado final de um teste é o estado inicial para o próximo teste.

Por exemplo, se quisermos realizar um teste que adiciona uma nota na partitura e outro que remove uma nota na partitura, a maneira sequencial seria executar primeiro o teste de adição, onde seu estado final contém uma nota na partitura, e então executar o teste de remoção, que espera a existência dessa nota para poder removê-la. Na maneira paralela, o teste de remoção não pode depender de um estado anterior, então ele deve abrir uma partitura que já contém uma nota.

Optou-se por implementar os teste que validam os dados utilizando o DOM, e são executados de maneira paralela. Os seguintes testes foram implementados:

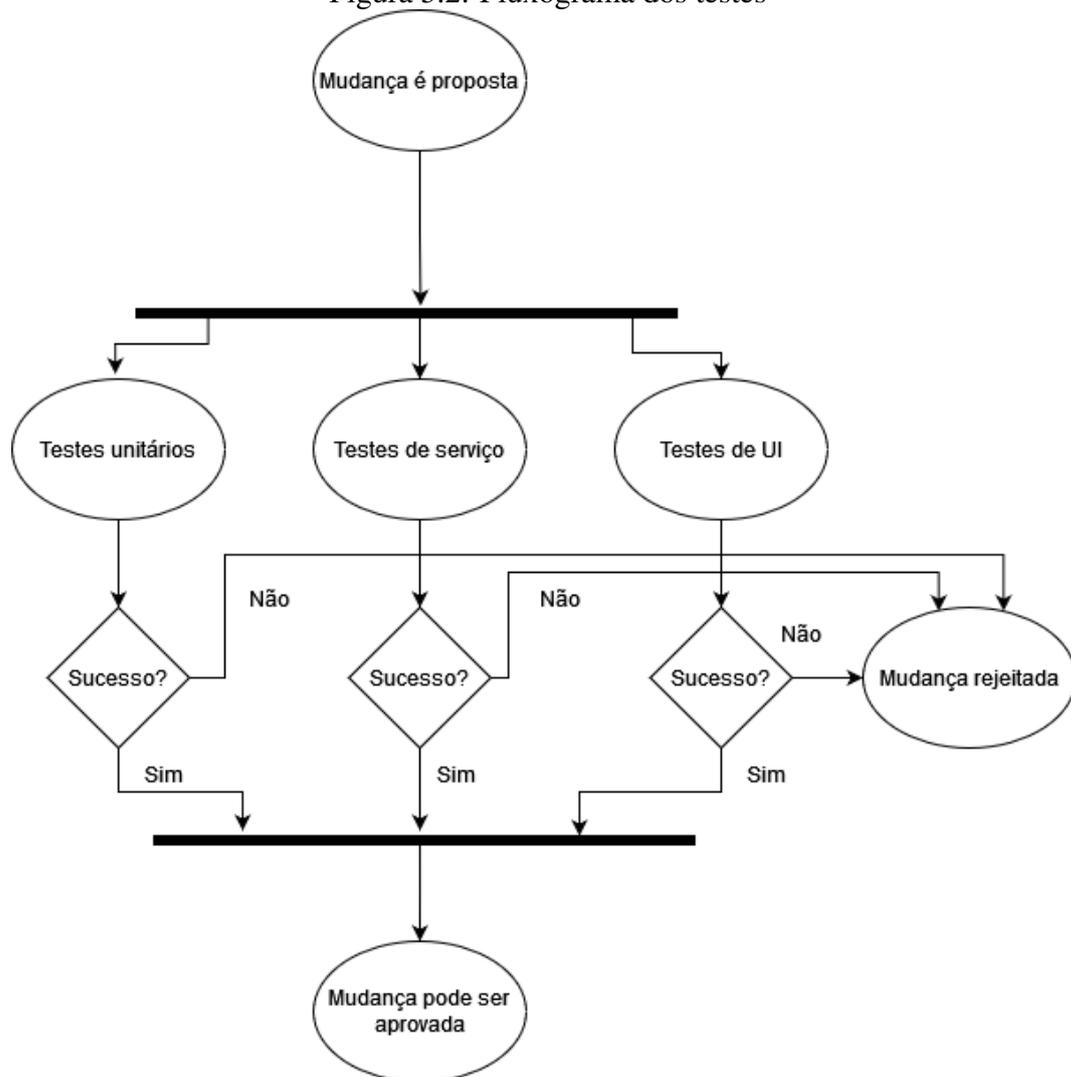
- Abrir uma partitura a partir de um arquivo no sistema
- Alterar das informações da partitura
- Alterar do texto de uma batida
- Alterar da casa de uma nota
- Criar e editar um acorde
- Desfazer e refazer a alteração da casa de uma nota

### 5.3 Integração com o controle de versão

Testes de regressão consistem em aplicar os mesmos testes em uma versão nova da aplicação, após uma alteração de código ou atualização de uma dependência. Isso se mostrou muito útil para garantir que novas versões da biblioteca AlphaTab continuaram funcionando corretamente durante o desenvolvimento da aplicação, e que continuem funcionando no futuro, minimizando o tempo necessário de executar os testes manualmente.

Para garantir que os testes sejam executados com sucesso antes de uma alteração ser aprovada, seja ela de código ou de uma dependência, utilizou-se de técnicas de CI/CD (*Continuous Integration/Continuous Delivery*) disponibilizadas pela ferramenta de hospedagem de código. Foram criados três *workflows*, um para cada categoria de teste. Caso todos os testes tenham sucesso, a alteração pode ser aprovada. O fluxograma da Figura 5.2 demonstra esse processo.

Figura 5.2: Fluxograma dos testes



Fonte: Autor

## 6 FUNCIONALIDADES

Esse capítulo descreve as funcionalidades implementadas na aplicação Escala. Uma visão geral pode ser vista na Figura 6.1, com a barra de ferramentas no topo, a listagem de faixas na esquerda, a renderização da partitura no meio e os controles do reprodutor na parte inferior.

Figura 6.1: Visão geral da aplicação

The screenshot displays the Escala application interface. On the left, a sidebar lists instrument tracks: Rhythm Gui, Solos, Bass, Bass, and Drum, each with a dropdown arrow. The main window features a menu bar with options: DOCUMENTO, EFEITOS, BATIDA, DURAÇÃO, and DINÂMICA, along with icons for settings, dark mode, and help. Below the menu is a toolbar with icons for file operations and playback. The central area shows the score for 'Back in Black' by AC/DC, with music and lyrics by Young, Young, Johnson. It specifies 'Guitar Standard Tuning' and a tempo of 96. The score includes a 'Rhythm Gui' part with a treble clef, 4/4 time signature, and a dynamic marking of *f*. The guitar part is shown with a treble clef and a bass clef, with fret numbers and string indicators (T, A, B) for the bass line. The bottom of the interface contains a playback control bar with a play button, a volume slider, and a speed control set to 1x.

Fonte: Autor

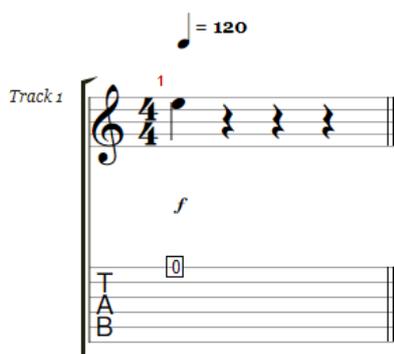
## 6.1 Cursor

O cursor no contexto desse editor de partituras é a seleção de uma nota ou uma posição de uma nota dentro de uma batida. Ele pode ser selecionado através do clique do mouse e movimentado utilizando-se o teclado.

Ele pode estar em um de três estados possíveis. O estado corrente do cursor influencia o resultado de diferentes operações do editor.

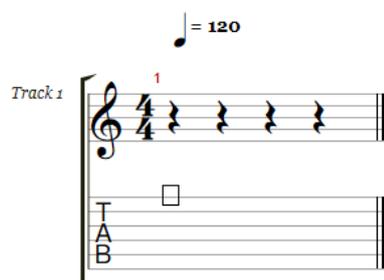
- Uma nota está selecionada, conforme mostra a Figura 6.2.
- Uma posição válida de uma nota está selecionada, conforme mostra a Figura 6.3.
- Nada está selecionado, conforme mostra a Figura 6.4.

Figura 6.2: Cursor selecionando uma nota



Fonte: Autor

Figura 6.3: Cursor selecionando uma posição válida para uma nota



Fonte: Autor

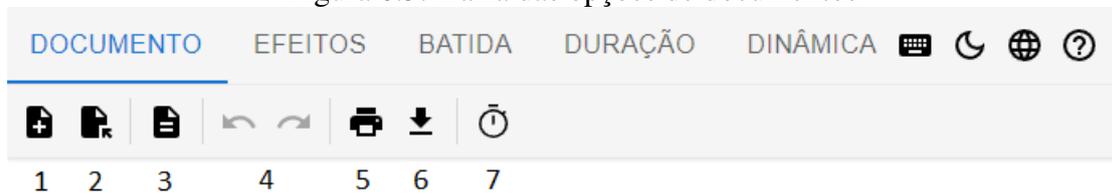
Figura 6.4: Cursor sem seleção

The image displays a musical score for 'Track 1' in 4/4 time. The tempo is marked as 120 BPM. The notation consists of two staves: a standard musical staff with a treble clef and a TAB staff below it. The musical staff contains four quarter notes, with a red cursor positioned above the first note. The TAB staff is currently empty. The tempo marking is shown as a quarter note followed by '= 120'.

Fonte: Autor

## 6.2 Operações no documento

Figura 6.5: Barra das opções de documentos



Fonte: Autor

### 6.2.1 Novo arquivo

A partitura corrente da aplicação é descartada e é gerada uma nova partitura vazia. Pode ser executado via o botão indicado pelo número 1 da Figura 6.5.

### 6.2.2 Abrir Arquivo

Utiliza-se do navegador para solicitar que o usuário selecione um arquivo a ser aberto pela aplicação. Pode ser acessado via o botão indicado pelo número 2 da Figura 6.5. São suportadas as seguintes extensões de arquivos:

- .gp
- .gp3
- .gp4
- .gp5
- .gpx
- .musicxml
- .mxml
- .xml
- .capx

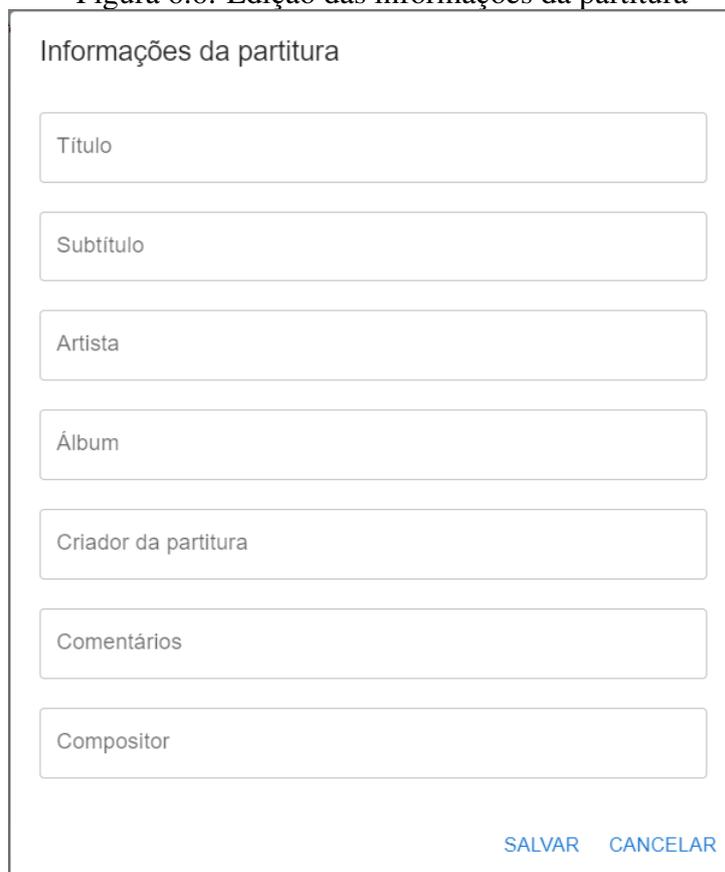
Após a escolha do arquivo, seus dados são importados e sua primeira faixa é selecionada e renderizada por padrão. O histórico das operações de desfazer e refazer também são limpos.

### 6.2.3 Informações da partitura

Possibilita a leitura e edição de informações sobre a partitura. Pode ser acessado via o botão indicado pelo número 3 da Figura 6.5. Conforme demonstra a Figura 6.6, os seguintes campos são disponibilizados:

- Título
- Subtítulo
- Artista
- Álbum
- Criador da tablatura
- Comentários
- Compositor

Figura 6.6: Edição das informações da partitura



Informações da partitura

Título

Subtítulo

Artista

Álbum

Criador da partitura

Comentários

Compositor

[SALVAR](#) [CANCELAR](#)

Fonte: Autor

### **6.2.4 Desfazer e Refazer**

A aplicação implementa um sistema de desfazer e refazer, comum em programas de edição de dados e detalhado no capítulo 4.4.3. Pode ser utilizado pelos botões indicados pelo número 4 da Figura 6.5. Quando não for possível desfazer ou refazer a última operação, o botão é desabilitado. Também é possível realizar tais ações via atalhos no teclado.

### **6.2.5 Imprimir**

A impressão utiliza a API da biblioteca de AlphaTab de renderização, que invoca a impressão da página pelo navegador. Nota-se que a página impressa não inclui os componentes do editor, apenas a tablatura. Pode ser acessado via o botão indicado pelo número 5 da Figura 6.5.

### **6.2.6 Download**

É possível exportar a tablatura no formato do guitarpro (.gp) ou em formato MIDI (.mid). Pode ser acessado via o botão indicado pelo número 6 da Figura 6.5.

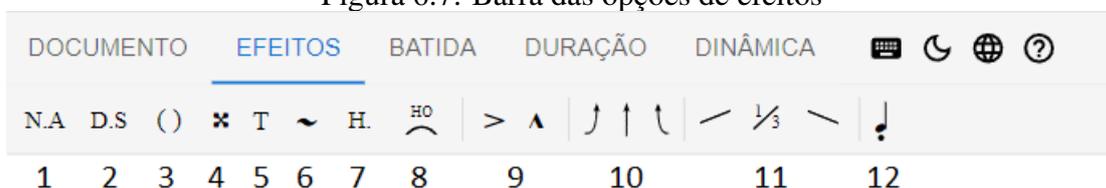
### **6.2.7 Tempo**

Possibilita a alteração do tempo da partitura em batidas por minuto. Pode ser alterado via o botão indicado pelo número 7 da Figura 6.5.

### 6.3 Operações de efeito sobre uma nota

As operações nessa seção são aplicadas a uma nota individual, conforme definido na seção 4.1.1. As opções só serão aplicáveis caso o cursor esteja selecionando uma nota, conforme explicado na seção 6.1. Caso o efeito já esteja aplicado sobre a nota selecionada, a cor do botão será alterada para informar ao usuário tal estado da nota, e caso nenhuma nota esteja selecionada, os botões vão ser desabilitados.

Figura 6.7: Barra das opções de efeitos



Fonte: Autor

#### 6.3.1 Nota Abafada

Abreviado como 'N.A' ou 'P.M' (*Palm mute* em inglês), é o efeito de se encostar a corda com a palma da mão, gerando um efeito de um som abafado. Pode ser alternado entre ligado e desligado no botão indicado pelo número 1 na Figura 6.7.

#### 6.3.2 Deixe Soar

Abreviado como 'D.S', ou 'L.R' (*Let it Ring* em inglês), consiste em deixar a nota reproduzir sem silenciá-la. Em instrumentos de corda, normalmente significa que a nota deve continuar sendo pressionada até que a vibração da corda atenuar, que a próxima nota deva ser tocada na mesma corda ou o músico precise tocar outra nota com o mesmo dedo. Pode ser alternado entre ligado e desligado pelo botão indicado pelo número 2 na Figura 6.7.

### **6.3.3 Nota Fantasma**

É uma nota com função rítmica e sem um tom discernível. Em instrumentos de corda, pode não ser tocada, mas se for, o músico deve dar ênfase na percussão, e abafar a corda para atenuar o tom. Pode ser alternado entre ligado e desligado pelo botão indicado pelo número 3 na Figura 6.7.

### **6.3.4 Nota Morta**

É uma técnica de utilizar a mão esquerda sobre a corda, mas sem pressionar uma casa específica, e tocar a corda. Recebe esse nome pois gera um som ‘morto’. Pode ser alternado entre ligado e desligado pelo botão indicado pelo número 4 na Figura 6.7.

### **6.3.5 Tapping**

Consiste em utilizar a mão direita para pressionar a casa no braço do instrumento, similar a um *hammer on*. Pode ser alternado entre ligado e desligado no botão indicado pelo número 5 na Figura 6.7.

### **6.3.6 Vibrato**

Em instrumentos de corda, é a variação de tom produzida pelo movimento de oscilação da mão esquerda enquanto a nota está sendo sustentada (APEL, 1950). Pode ser alternado entre ligado e desligado pelo botão indicado pelo número 6 na Figura 6.7.

### **6.3.7 Harmônica**

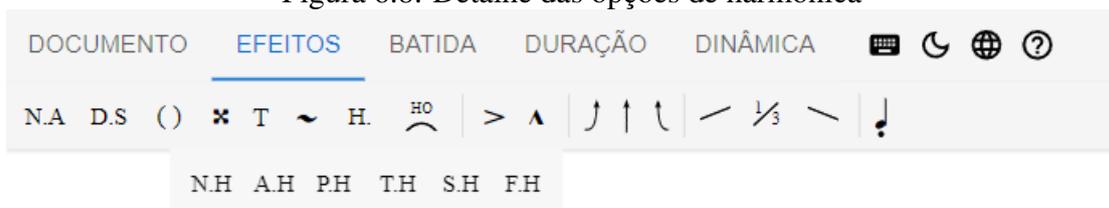
Uma corda ao ser tocada, gera uma vibração. "Enquanto a corda vibra por inteiro, simultaneamente, ela vibra também dividindo-se em duas metades. [. . .]. Ela vibra também dividindo-se em terços, em quartos, em quintos, etc., produzindo as vibrações secundárias."(MED, 1996, p. 92). A técnica de harmônica significa isolar esses sobretons secundários, que por serem em uma frequência maior, geram tons mais agudos. Existem

algumas maneiras diferentes de se isolar essas harmônicas, comuns em instrumentos de corda:

1. Harmônica Natural: É produzido encostando o dedo da mão esquerda levemente na corda enquanto ela vibra.
2. Harmônica Artificial: É produzido pressionando a corda com a mão esquerda e simultaneamente encostando a mão direita levemente na corda.
3. Harmônica *Pinch*: É produzido usando a mão direita para dar um ‘beliscão’ na corda no braço da guitarra, 12 casas acima da nota que está sendo pressionada com a mão esquerda.
4. Harmônica *Tap*: É produzido usando a mão direita para pressionar a corda no braço da guitarra, 12 casas acima da nota que está sendo pressionada com a mão esquerda.
5. Harmônica *Semi*: Parecido com a harmônica artificial, mas encosta-se a mão direita extremamente leve, de forma que não atenua tanto a harmônica principal.
6. Harmônica *Feedback*: Utiliza-se a saída do amplificador para gerar uma ressonância com a corda que está sendo tocada.

Uma relação dos tipos de harmônica implementados está disponível na tabela 6.3. O menu das opções de harmônica pode ser aberto ao clicar-se no botão indicado pelo número 7 na Figura 6.7, e a harmônica pode ser selecionada nesse menu conforme demonstrado na Figura 6.8.

Figura 6.8: Detalhe das opções de harmônica



Fonte: Autor

### 6.3.8 *Hammer On/Pull off*

São duas técnicas agrupadas em uma única ação, pois são mutuamente exclusivas e dependem do contexto.

- *Hammer On*: Consiste pressionar com a mão esquerda uma nota em alguma casa acima da atual, na mesma corda, sem utilizar a mão direita para tocar a corda.

- *Pull off*: Consiste em soltar uma corda com a mão esquerda, ao mesmo tempo que uma casa abaixo está sendo pressionada, sem utilizar a mão direita para tocar a corda.

Pode ser alternado entre ligado e desligado no botão indicado pelo número 8 na Figura 6.7.

### 6.3.9 Nota acentuada & Nota muito acentuada

Indica que a nota deve ser executada com uma intensidade maior do que a normal. Pode ser alternado entre ligado e desligado pelos botões indicados pelo número 9 na Figura 6.7.

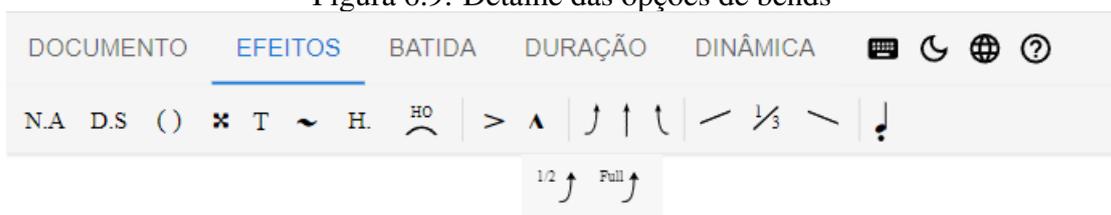
### 6.3.10 *Bend*

É uma técnica comum em guitarra onde a corda pressionada é movimentada verticalmente enquanto é tocada, produzindo uma variação do tom. Pode ser visualizada como uma função, onde o eixo horizontal é o tempo, e o eixo vertical é a movimentação vertical da corda. Divide-se o *bend* em três etapas:

- *Pre-bend*: Posição que a corda deve estar antes dela ser tocada
- *Bend*: Variação da corda enquanto ela está sendo tocada
- *Release*: Retorno da corda à posição original.

Cada etapa pode ser configurada como desligada, ativada com meio tom e ativada com um tom inteiro. Baseado nas opções selecionadas, o editor utiliza uma tabela com os valores pré-computados para gerar a discretização da função de *bend*. Cada etapa pode ser configurada utilizando-se os botões indicados pelo número 10 na Figura 6.7.

Figura 6.9: Detalhe das opções de bends



### 6.3.11 *Slide*

Técnica aonde se mantém a mão esquerda pressionando alguma nota no braço do instrumento, ao mesmo tempo que se movimenta horizontalmente, alterando a casa que está sendo executada. Diferencia-se os slides em três possíveis tipos:

- *Slide in*: Executa-se o slide antes da execução da nota, gerando-se um tom crescente ou decrescente. Similar a um fade-in. Não especifica-se qual a nota de origem do slide, ficando a critério do leitor.
  - *Slide in from above*: A origem do slide vem de cima, ou seja, de uma nota mais aguda.
  - *Slide in from below*: A origem do slide vem de baixo, ou seja, de uma nota mais grave.
- *Slide*: Executa-se o slide entre duas notas, gerando-se um tom contínuo entre elas.
  - *Shift Slide*: A primeira nota é executada, então o dedo que está pressionando a corda desliza para a próxima nota, e a corda é tocada novamente.
  - *Legato Slide*: Semelhante ao *Shift Slide*, mas a diferença é que a corda não é tocada novamente na segunda nota.
- *Slide out*: Executa-se o slide após a reprodução da nota, gerando-se um tom crescente ou decrescente. Similar a um fade-out. Não se especifica qual a nota de destino do slide, ficando a critério do leitor.
  - *Slide out up*: O destino do slide deve ser em cima, ou seja, em uma nota mais aguda.
  - *Slide out down*: O destino do slide deve ser abaixo, ou seja, em uma nota mais grave.

Nesse editor, optou-se por implementar o *Shift Slide*, *Slide in from below* e o *Slide out up*, que são as variações mais comuns. Cada etapa do bend pode ser configurada utilizando-se os botões indicados pelo número 12 na Figura 6.7. Um resumo dessas opções e quais delas são implementadas por diferentes editores pode ser vista na tabela 6.2.

### 6.3.12 *Staccato*

Também conhecido como ‘ponto de diminuição’, é um sinal indicado por um ponto em cima da nota, que sinaliza uma redução do tempo de execução da nota, seguido por uma breve pausa (APEL, 1950).

Existem diferentes tipos de *staccato* (MED, 1996):

- Simples: Divide a duração em duas metades, uma de som e outra de silêncio.
- Alongado: Divide a duração em quatro partes, uma de som e três de silêncio.
- Ligado ou Brando: Divide a duração em quatro partes, três de som e uma de silêncio.

Optou-se pela implementação do *Staccato* Simples, que pode ser alternado entre ligado e desligado no botão indicado pelo número 12 na Figura 6.7.

Tabela 6.1: Comparativo dos efeitos sobre uma nota

<i>Efeito</i>	<i>Flat.io free</i>	<i>Flat.io Premium</i>	<i>Tux Guitar</i>	<i>Escala</i>
<i>Bends</i>	Sim	Sim	Sim	Sim
Deixe soar	Sim	Sim	Sim	Sim
<i>Detached Legato</i>	Sim	Sim	Não	Não
Dinâmica	Sim	Sim	Sim	Sim
Duração da nota	Sim	Sim	Sim	Sim
<i>Fade in</i>	Não	Não	Sim	Não
<i>Grace Note</i>	Sim	Sim	Sim	Não
<i>Hammer On</i>	Sim	Sim	Sim	Sim
Harmônica <i>Feedback</i>	Não	Não	Não	Sim
Harmônica <i>Pinch</i>	Não	Não	Sim	Sim
Harmônica <i>Semi</i>	Não	Não	Sim	Sim
Harmônica <i>Tap</i>	Não	Não	Sim	Sim
Harmônica artificial	Sim	Sim	Sim	Sim
Harmônica natural	Sim	Sim	Sim	Sim
<i>Legato Slide</i>	Não	Não	Não	Não
Nota abafada	Sim	Sim	Sim	Sim
Nota acentuada	Sim	Sim	Sim	Sim
Nota fantasma	Sim	Sim	Sim	Sim
Nota morta	Sim	Sim	Sim	Sim
Nota muito acentuada	Sim	Sim	Sim	Sim
<i>Popping</i>	Não	Não	Sim	Não
<i>Pull off</i>	Sim	Sim	Sim	Sim
<i>Shift Slide</i>	Sim	Sim	Sim	Sim
<i>Slapping</i>	Não	Não	Sim	Não
<i>Slide In from above</i>	Não	Não	Não	Não
<i>Slide In from below</i>	Não	Não	Não	Sim
<i>Slide Out down</i>	Não	Não	Não	Não
<i>Slide Out up</i>	Não	Não	Não	Sim
<i>Staccatissimo</i>	Sim	Sim	Não	Não
<i>Staccato</i>	Sim	Sim	Sim	Sim
<i>Tapping</i>	Não	Não	Sim	Sim
<i>Tenuto</i>	Sim	Sim	Não	Não
<i>Ties</i>	Sim	Sim	Sim	Sim
<i>Tremolo Bar</i>	Não	Não	Sim	Não
<i>Tremolo Picking</i>	Não	Não	Sim	Sim
<i>Trill</i>	Sim	Sim	Sim	Não
<i>Vibrato</i>	Não	Não	Sim	Sim

Fonte: Autor

Tabela 6.2: Comparativo dos efeitos de slide

<i>Efeito</i>	<i>Flat.io free</i>	<i>Flat.io Premium</i>	<i>Tux Guitar</i>	<i>Escala</i>
<i>Slide In from above</i>	Não	Não	Não	Não
<i>Slide In from below</i>	Não	Não	Não	Sim
<i>Slide Out up</i>	Não	Não	Não	Sim
<i>Slide Out down</i>	Não	Não	Não	Não
<i>Shift Slide</i>	Sim	Sim	Sim	Sim
<i>Legato Slide</i>	Não	Não	Não	Não

Fonte: Autor

Tabela 6.3: Comparativo dos efeitos de harmônica

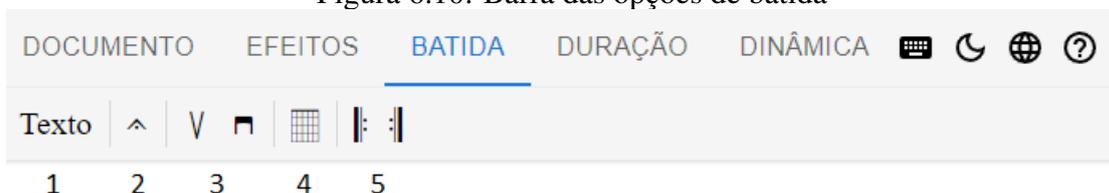
<i>Efeito</i>	<i>Flat.io free</i>	<i>Flat.io Premium</i>	<i>Tux Guitar</i>	<i>Escala</i>
Harmônica Natural	Sim	Sim	Sim	Sim
Harmônica Artificial	Sim	Sim	Sim	Sim
Harmônica <i>Pinch</i>	Não	Não	Sim	Sim
Harmônica <i>Tap</i>	Não	Não	Sim	Sim
Harmônica <i>Semi</i>	Não	Não	Sim	Sim
Harmônica <i>Feedback</i>	Não	Não	Não	Sim

Fonte: Autor

## 6.4 Operações sobre uma batida

As operações nesta seção são aplicadas a uma batida, definida na seção 4.1.2. As opções só serão aplicáveis caso o cursor esteja selecionando uma nota ou uma posição válida para uma nota. Caso o efeito já esteja aplicado sobre a batida selecionada, a cor do botão será alterada para informar ao usuário tal estado da batida, e um novo clique no ícone removerá o efeito.

Figura 6.10: Barra das opções de batida

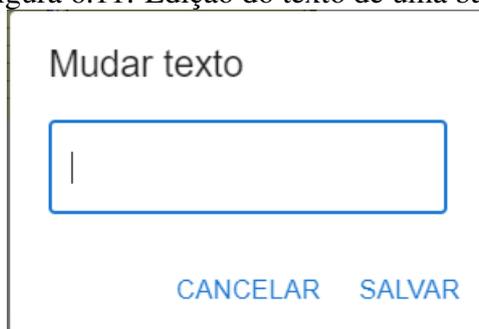


Fonte: Autor

### 6.4.1 Texto

Para uma determinada batida, é possível configurar um curto texto que será exibido logo acima das notas da batida na partitura. Tal texto pode ser a letra da música, alguma orientação sobre como o trecho deve ser executado, ou alguma outra informação qualquer. A edição do texto pode ser realizada utilizando-se o botão indicado pelo número 1 da Figura 6.10. Ao ser clicado, será aberta uma janela com o campo textural conforme mostra a Figura 6.11.

Figura 6.11: Edição do texto de uma batida



Fonte: Autor

### 6.4.2 *Fermata*

A *fermata*, também conhecida como suspensão, vem do italiano e significa ‘parada’. Indica uma pequena pausa na execução, realizada pelo prolongamento da duração das notas. Pode se alternado entre ligado e desligado utilizando o botão indicado pelo número 2 da Figura 6.10.

### 6.4.3 *Upstroke e Downstroke*

Ao se tocar um conjunto de notas em uma mesma batida em um instrumento de cordas, é possível fazer de cima para baixo (cordas mais graves para mais agudas) ou de baixo para cima (cordas mais agudas para mais graves). Essa notação é importante para informar padrões de batida. É possível alterar entre *Upstroke* e *Downstroke* utilizando os botões indicados pelo número 3 da Figura 6.10.

### 6.4.4 **Acorde**

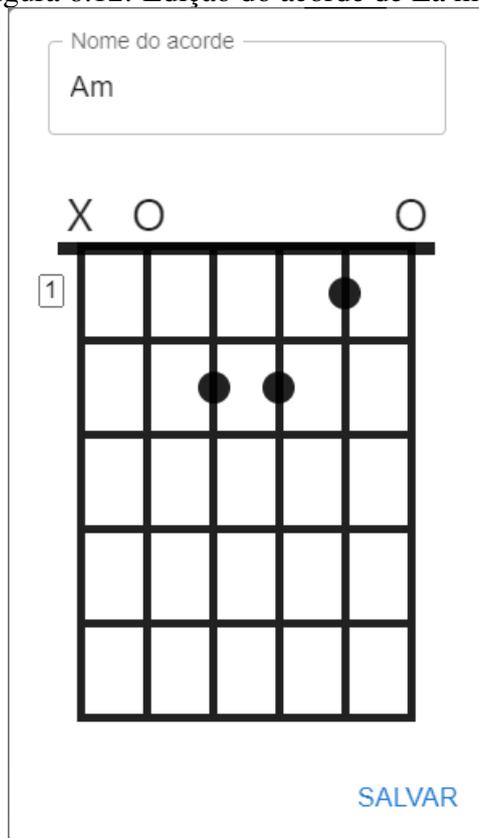
Um acorde é a ocorrência simultânea de múltiplos tons, geralmente três ou mais (APEL, 1950). Ao clicar no botão de acorde, indicado pelo número 4 da Figura 6.10, é aberto uma janela onde é possível informar o acorde que se deseja adicionar, conforme demonstrado na Figura 6.12. Os componentes principais são:

- Nome: O nome do acorde. Normalmente se usa uma notação simplificada. (Por exemplo, ‘Am’ em vez de ‘Lá menor’).
- Deslocamento: Para não ser preciso representar o braço inteiro do instrumento, opta-se por exibir apenas 5 casas mais o deslocamento inicial.
- Notas: Um ‘X’ representa que a corda não deve ser tocada. ‘O’ Representa que a corda deve ser tocada na casa zero, e quando existir um círculo preto na linha vertical, indica que a corda deve ser tocada na casa indicada, levando-se em conta o deslocamento.

Ao salvar o acorde, o editor irá criar as notas necessárias e associará a batida ao acorde, exibindo seu nome em cima das notas. O editor também manterá todos os acordes usados em uma seção especial da barra de ferramentas, permitindo que o usuário adicione

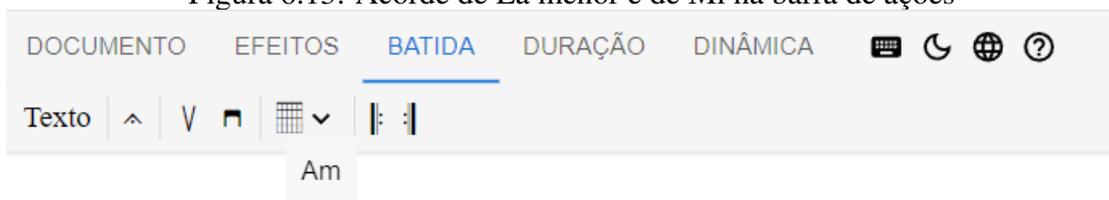
o acorde novamente com apenas um clique, conforme exibido na Figura 6.13.

Figura 6.12: Edição do acorde de Lá menor



Fonte: Autor

Figura 6.13: Acorde de Lá menor e de Mi na barra de ações



Fonte: Autor

#### 6.4.5 Repetição

É utilizado o *ritornelo*, sinal que determina a repetição de um trecho musical. Consiste em uma barra dupla, sendo uma grossa e outra fina, com dois pontos (MED, 1996). Para o funcionamento correto, é necessário que o comando de abertura de repetição e o de fechamento de repetição sejam utilizados em conjunto, abrangendo as batidas que devem ser repetidas. Também é possível configurar a quantidade de repetições. Pode ser configurado utilizando os botões indicados pelo número 5 da Figura 6.10.

Tabela 6.4: Comparativo dos efeitos sobre uma batida

<i>Efeito</i>	<i>Flat.io free</i>	<i>Flat.io Premium</i>	<i>Tux Guitar</i>	<i>Escala</i>
Repetição	Sim	Sim	Sim	Sim
Número de repetições	Apenas 2x	Sim	Sim	Sim
Acordes	Sim	Sim	Sim	Sim
<i>Fermata</i>	Sim	Sim	Sim	Sim

Fonte: Autor

## 6.5 Operações na duração

### 6.5.1 Duração

Figura 6.14: Barra das opções de duração



Fonte: Autor

A duração do tempo em que uma nota deve ser reproduzida pode ser ajustada, partindo-se de uma unidade de tempo, baseada no BPM (Batidas por minuto), e diminuindo pela metade:

- Semibreve: Uma unidade de tempo
- Mínima: 1/2 unidade de tempo
- Semínima: 1/4 unidade de tempo
- Colcheia: 1/8 unidade de tempo
- Semicolcheia: 1/16 unidade de tempo
- Fusa: 1/32 unidade de tempo
- Semifusa: 1/64 unidade de tempo

Essas opções são disponibilizadas, respectivamente, pelo conjunto de botões indicados pelo número 1 na Figura 6.14.

Tabela 6.5: Comparativo dos efeitos de duração

<i>Efeito</i>	<i>Flat.io free</i>	<i>Flat.io Premium</i>	<i>Tux Guitar</i>	<i>Escala</i>
Semibreve	Sim	Sim	Sim	Sim
Mínima	Sim	Sim	Sim	Sim
Semínima	Sim	Sim	Sim	Sim
Colcheia	Sim	Sim	Sim	Sim
Semicolcheia	Sim	Sim	Sim	Sim
Fusa	Sim	Sim	Sim	Sim
Semifusa	Sim	Sim	Sim	Sim

Fonte: Autor

### **6.5.2 Nota ligada**

Também chamado de ‘ligadura de prolongamento’, conecta duas notas de mesma altura, somando-se a duração. Somente a primeira nota é articulada. Pode ser alternado entre ligado e desligado no botão indicado pelo número 2 na Figura 6.14.

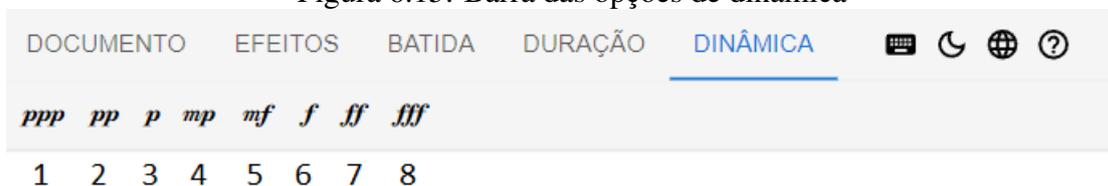
## 6.6 Operações na dinâmica

A dinâmica "É o grau de intensidade com que o som é emitido ou articulado"(MED, 1996, p.213). Os níveis de intensidade são geralmente abreviados por termos italianos colocados sob a pauta. (MED, 1996). As seguintes graduações são utilizadas:

- ppp: Extremamente suave
- pp: Muito suave
- p: Suave
- mp: Meio suave
- mf: Meio forte
- f: Forte
- ff: Muito forte
- fff: Extremamente forte

Essas opções se encontram conforme a Figura 6.15 indica. Um comparativo das implementações pode ser encontrado na tabela 6.6.

Figura 6.15: Barra das opções de dinâmica



Fonte: Autor

Tabela 6.6: Comparativo dos efeitos de dinâmica

Dinâmica	Flat.io free	Flat.io Premium	Tux Guitar	Escala
<i>ppp</i>	Sim	Sim	Sim	Sim
<i>pp</i>	Sim	Sim	Sim	Sim
<i>p</i>	Sim	Sim	Sim	Sim
<i>mp</i>	Sim	Sim	Sim	Sim
<i>mf</i>	Sim	Sim	Sim	Sim
<i>f</i>	Sim	Sim	Sim	Sim
<i>ff</i>	Sim	Sim	Sim	Sim
<i>fff</i>	Sim	Sim	Sim	Sim

Fonte: Autor

## 6.7 Faixas de música

Uma faixa, conforme definido no capítulo 4.1.6, representa um instrumento executado na música. Uma única partitura pode conter múltiplas faixas, como demonstrado na Figura 6.16. O usuário pode criar, editar ou remover faixas. A janela de edição de uma faixa pode ser vista na figura 6.17.



Fonte: Autor

### 6.7.1 Nome da faixa

Cada faixa pode ser nomeada, facilitando sua identificação e propósito. Normalmente indica o tipo do instrumento, ou quando existem mais de um instrumento do mesmo tipo, qual parte ou músico a executa.

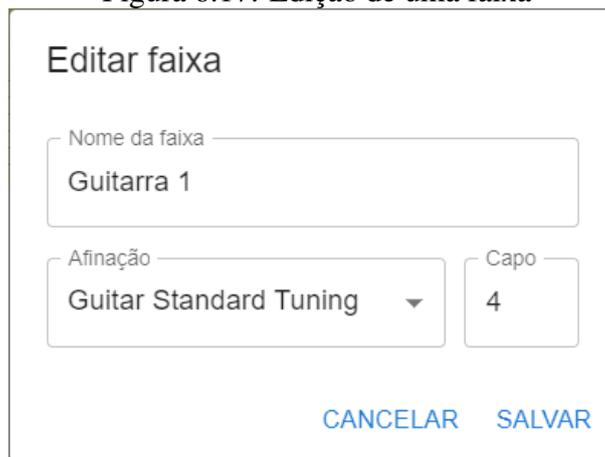
### 6.7.2 Afinação

É possível escolher uma afinação pré-definida para um instrumento de seis cordas para cada faixa.

### 6.7.3 Capotraste

Também chamado de capo, o capotraste é uma ferramenta que permite facilmente alterar a afinação do instrumento. Normalmente se utiliza uma barra de borracha que segura as cordas em uma determinada casa do braço do instrumento. É possível definir qual casa que o capotraste deve ser utilizado.

Figura 6.17: Edição de uma faixa



Editar faixa

Nome da faixa

Afinação

Capo

[CANCELAR](#) [SALVAR](#)

Fonte: Autor

## 6.8 Outros

Figura 6.18: Outras opções



Fonte: Autor

### 6.8.1 Atalhos no teclado

O menu com informações dos atalhos existentes, conforme pode ser visto na Figura 6.19, pode ser visualizados pelo usuário ao clicar no ícone de teclado na barra de ferramentas superior, indicado pelo número 1 na Figura 6.18. Os seguintes atalhos foram implementados:

- Alterar casa da nota
- Remover nota
- Desfazer
- Refazer
- Mover cursor para esquerda
- Mover cursor para direita
- Mover cursor para cima
- Mover cursor para baixo
- Mover cursor para a próxima barra
- Mover cursor para a barra anterior

### 6.8.2 Modo escuro

O usuário pode alternar entre o modo escuro e o modo claro utilizando o botão indicado pelo número 2 na Figura 6.18 da barra de ferramentas superior. A tabela 6.7 mostra o essa funcionalidade entre as aplicações comparadas.

Figura 6.19: Janela de atalhos

Atalhos do teclado	
Alterar casa da nota	Teclado numérico
Remover nota	Botão Delete
Desfazer	Ctrl + Z
Refazer	Ctrl + Y
Mover cursor para esquerda	Seta para esquerda
Mover cursor para direita	Seta para direita
Mover cursor para cima	Seta para cima
Mover cursor para baixo	Seta para baixo
Mover cursor para próxima barra	Ctrl + Setas para direita
Mover cursor para a barra anterior	Ctrl + Setas para esquerda

Fonte: Autor

Tabela 6.7: Comparativo do suporte do modo escuro

<i>Flat.io free</i>	<i>Flat.io Premium</i>	<i>Tux Guitar</i>	<i>Escala</i>
Não	Não	Não	Sim

Fonte: Autor

### 6.8.3 Seleção de Idioma

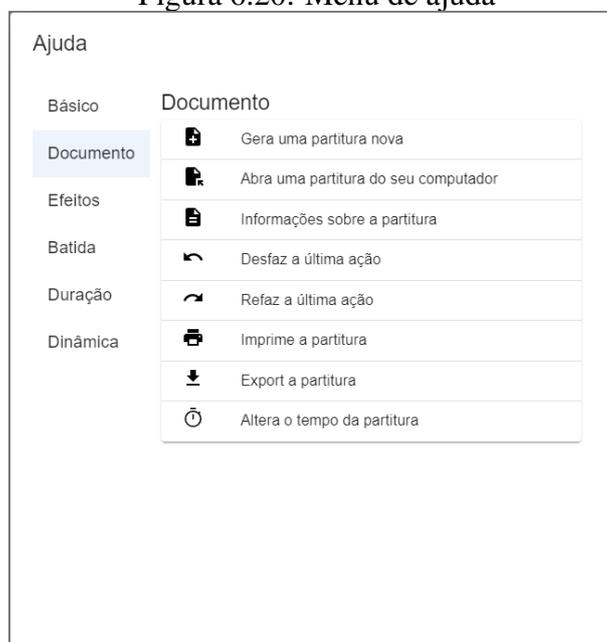
É possível alterar o idioma da aplicação utilizando o botão indicado pelo número 3 na Figura 6.18.

### 6.8.4 Menu de Ajuda

Um menu de ajuda foi implementado com instruções básicas e explicações das funcionalidades 6.20. Ele pode ser acessado utilizando a barra de ferramenta superior, utilizando o botão indicado pelo número 4 na Figura 6.18. Dividiu-se em seis categorias de informações:

- Básico: Navegação do cursor, alteração de traste e remoção de nota. 6.1
- Documento: Opções relacionadas ao menu de Documento 6.2
- Efeitos: Opções relacionadas ao menu de Efeitos 6.3
- Batida: Opções relacionadas ao menu de Batida 6.4
- Duração: Opções relacionadas ao menu de Duração 6.5
- Dinâmica: Opções relacionadas ao menu de Dinâmica 6.6

Figura 6.20: Menu de ajuda



Fonte: Autor

## 6.9 Controle do reprodutor

As funcionalidades de controle do reprodutor MIDI foram agrupadas na região inferior da aplicação.

Figura 6.21: Controles do reprodutor



Fonte: Autor

### 6.9.1 Tocar/Parar

Alterna-se entre o estado de reprodução e de pausa. Indicado pelo botão número 1 na Figura 6.21.

### 6.9.2 *Count In*

Adiciona alguns segundos antes do início da reprodução após o usuário clicar no botão de 'tocar/parar'. Útil caso o usuário esteja praticando a execução da partitura junto com a sua reprodução, pois proporciona tempo para sua preparação. Pode ser alternado entre ligado e desligado no botão indicado pelo número 2 na Figura 6.21.

### 6.9.3 Metrônomo

O metrônomo é uma ferramenta utilizada para ajudar o músico a manter o tempo correto, produzindo um som de acordo com a frequência configurada. Uma vantagem de ter o metrônomo integrado ao sintetizador é a configuração automática dessa frequência, de acordo com o BPM da partitura. Pode ser alternado entre ligado e desligado no botão indicado pelo número 3 na Figura 6.21.

#### **6.9.4 Repetir**

Quando ativado, a reprodução recomeça após chegar ao fim. Pode ser alternado entre ligado e desligado no botão indicado pelo número 4 na Figura 6.21.

#### **6.9.5 Volume**

Possibilita-se o ajuste do volume da reprodução. Pode ser ajustado utilizando-se a barra horizontal indicada pelo número 5 na Figura 6.21.

#### **6.9.6 Velocidade de reprodução**

É possível ajustar a velocidade de reprodução entre 25% e 200%, em incrementos e decrementos de 25%. Pode ser alterado no botão indicado pelo número 6 na Figura 6.21.

## 7 CONCLUSÃO

Foi possível desenvolver uma aplicação para edição e reprodução de tablaturas para a *web*, que pode ser utilizada em diversos sistemas operacionais como Windows, MacOS e distribuições Linux. Com foco em músicos amadores e estudantes, contém um conjunto de funcionalidades robusto e, em alguns pontos, mais completo que aplicações já estabelecidas.

Com o uso da linguagem Typescript para a programação e bibliotecas de *software* como React e MUI para a criação e composição dos componentes visuais, foi possível criar um grande número de funcionalidades em pouco tempo. O uso de testes automatizados facilitou a detecção de problemas conforme a aplicação evoluiu e proporciona mais segurança para mudanças futuras.

A integração com a biblioteca de código AlphaTab se mostrou muito desafiadora. Apesar de ser possível, realizar modificações na tablatura após seu carregamento inicial requer um entendimento profundo da modelagem dos dados e funcionamento interno do seu código. Além disso, *bugs* existentes e a pouca documentação dificultaram o desenvolvimento deste trabalho, mas foi possível contribuir para a melhoria desses problemas, com algumas contribuições aceitas.<sup>1 2 3</sup>.

O código-fonte da aplicação Escala foi disponibilizado com a licença MIT e se encontra-se disponível no Github do autor<sup>4</sup>. A aplicação também foi hospedada e pode ser acessada gratuitamente<sup>5</sup>.

Por fim, para melhorar ainda mais a aplicação, testes com usuários são necessários para obtenção de feedbacks valiosos acerca da usabilidade da aplicação criada, bem como para o descobrimento de possíveis *bugs* desconhecidos até o presente momento. Para o futuro, um caminho possível para a monetização seria agregar valor com um serviço integrado a *Cloud*, oferecendo aos usuários uma solução completa para a hospedagem de suas partituras.

---

<sup>1</sup><<https://github.com/CoderLine/alphaTab/pull/933>>

<sup>2</sup><<https://github.com/CoderLine/alphaTab/pull/746>>

<sup>3</sup><<https://github.com/CoderLine/alphaTabWebsite/pull/60>>

<sup>4</sup><<https://github.com/gallegretti/escala>>

<sup>5</sup><<https://gallegretti.github.io/escala/>>

## REFERÊNCIAS

- ALPHATAB. 2022. Disponível na Internet: <<https://alphatab.net/>>.
- APEL, W. **The Harvard dictionary of music**. [S.l.]: Belknap Press: An Imprint of Harvard University Press, 1950.
- BEIZER, B. **Software Testing Techniques (2nd Ed.)**. USA: Van Nostrand Reinhold Co., 1990. ISBN 0442206720.
- BYRAM-WIGFIELD, B. **A brief history of music notation on computers**. 2021. Disponível na Internet: <<https://www.scoringnotes.com/opinion/a-brief-history-of-music-notation-on-computers/>>.
- COHN, M. **Succeeding with Agile: Software Development Using Scrum**. [S.l.]: Addison-Wesley Professional, 2009. ISBN 978-0321579362.
- DASH, P.; HU, Y. C. How much battery does dark mode save? an accurate oled display power profiler for modern smartphones. In: **Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services**. New York, NY, USA: Association for Computing Machinery, 2021. (MobiSys '21), p. 323–335. ISBN 9781450384438. Disponível na Internet: <<https://doi.org/10.1145/3458864.3467682>>.
- EISFELD, H.; KRISTALLOVICH, F. **The Rise of Dark Mode : A qualitative study of an emerging user interface design trend**. 2020.
- ELSEA, P. **Music Notation with Computers**. <<http://artsites.ucsc.edu/ems/music/equipment/computers/notation/notation.html>>. Acessado 11/09/2022.
- FINALE Music Notation Software. 2022. Disponível na Internet: <<https://www.finalemusic.com/>>.
- FLAT.IO. 2022. Disponível na Internet: <<https://flat.io/>>.
- GAMMA, E. et al. **Padrões de Projeto: Soluções reutilizáveis de software orientado a objetos**. [S.l.]: Bookman, 2000. ISBN 9788573076103.
- GOOGLE. **Material is a design system created by Google to help teams build high-quality digital experiences for Android, iOS, Flutter, and the web**. 2022. Disponível na Internet: <<https://material.io/design/introduction#principles>>.
- MED, B. **Teoria da música**. 4. ed. [S.l.]: Musimed, 1996.
- MICROSOFT. **Typescript: Javascript with Sintax for Types**. 2022. Disponível na Internet: <<https://www.typescriptlang.org/>>.
- MUSICXML Software. 2022. Disponível na Internet: <<https://www.musicxml.com/software/>>.
- PCMAGAZINE. Score review. v. 7, n. 20, p. 264, 1988. Disponível na Internet: <<https://archive.org/details/PC-Mag-1988-11-29/page/n267/mode/2up>>.

PETERSON, J.; SCHUBERT, E. Music notation software: Some observations on its effects on composer creativity. **Proceedings of ICoMCS December**, Citeseer, v. 127, 2007.

SIBELUS Music Notation Software. 2022. Disponível na Internet: <<https://www.avid.com/sibelius>>.

SMITH, L. **The SCORE Music Publishing System**. <<https://web.archive.org/web/20190602040203/http://scoremus.com/score.html>>. Acessado 11/09/2022.

STACKOVERFLOW. **2022 Developer Survey**. 2022. Disponível na Internet: <<https://survey.stackoverflow.co/2022/#most-popular-technologies-webframe>>.

TECHNOLOGIES, S. M. **Standard Music Font Layout**. 2022. Disponível na Internet: <<https://www.smufi.org/>>.

TUXGUITAR. 2022. Disponível na Internet: <<https://sourceforge.net/projects/tuxguitar/>>.

WIKIPEDIA. **Comparison of scorewriters — Wikipedia, The Free Encyclopedia**. 2022. Disponível na Internet: <<https://en.wikipedia.org/w/index.php?title=Comparison%20of%20scorewriters&oldid=1121449556>>.

WIKIPEDIA. **List of scorewriters — Wikipedia, The Free Encyclopedia**. 2022. Disponível na Internet: <<https://en.wikipedia.org/w/index.php?title=List%20of%20scorewriters&oldid=1103862112>>.