

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

ANA LÍVIA KERBER

**Protótipo de um conversor MusicXML para  
MIDI usando Android**

Trabalho de Graduação.

Prof. Dr. Marcelo Soares Pimenta  
Orientador

Porto Alegre, dezembro de 2012.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## **AGRADECIMENTOS**

Agradeço a todos que, de alguma forma, me ajudaram na realização deste trabalho:

Aos meus pais, Derli Elói Kerber e Cláudia Kerber, e à minha irmã Luiza Kerber, por todo o apoio e carinho durante toda a minha graduação, que comemoraram comigo todas as minhas conquistas e que me deram força nos momentos difíceis.

Ao meu namorado Fernando Ferrary, que me fez acreditar no meu potencial, e que me ofereceu ajuda, carinho e muita paciência nessa fase complicada de final de curso.

Às minhas amigas Thais Krischer e Lívia Freire, que me acompanharam durante praticamente todo o curso e que sempre estiveram presentes quando precisei.

A todos os amigos e familiares próximos que não estão citados aqui, mas que me ajudaram de muitas formas durante a minha graduação.

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>5</b>
<b>LISTA DE FIGURAS .....</b>	<b>6</b>
<b>LISTA DE TABELAS .....</b>	<b>7</b>
<b>RESUMO.....</b>	<b>8</b>
<b>ABSTRACT.....</b>	<b>9</b>
<b>1 INTRODUÇÃO .....</b>	<b>10</b>
<b>1.1 Motivação .....</b>	<b>10</b>
<b>1.2 Objetivo .....</b>	<b>11</b>
<b>1.3 Estrutura do trabalho .....</b>	<b>11</b>
<b>2 CONCEITOS IMPORTANTES .....</b>	<b>12</b>
<b>2.1 MIDI .....</b>	<b>12</b>
2.1.1 O cabeçalho de arquivo (header chunk) .....	12
2.1.2 O cabeçalho de trilha (track chunk).....	13
2.1.3 As mensagens MIDI .....	13
<b>2.2 MusicXML .....</b>	<b>14</b>
2.2.1 MusicXML 3.0 .....	18
<b>2.3 Android.....</b>	<b>19</b>
<b>3 TRABALHOS RELACIONADOS .....</b>	<b>19</b>
<b>4 IMPLEMENTAÇÃO .....</b>	<b>21</b>
<b>4.1 O escopo.....</b>	<b>21</b>
<b>4.2 A plataforma .....</b>	<b>21</b>
<b>4.3 A interface .....</b>	<b>22</b>
<b>4.4 O parser .....</b>	<b>23</b>
<b>4.5 A estrutura do projeto.....</b>	<b>24</b>
<b>5. RESULTADOS .....</b>	<b>28</b>
<b>6. CONCLUSÃO .....</b>	<b>31</b>
<b>REFERÊNCIAS .....</b>	<b>33</b>
<b>ANEXO: TABELA DE TAGS E ELEMENTOS MUSICXML .....</b>	<b>34</b>

## **LISTA DE ABREVIATURAS E SIGLAS**

DOM	Document Object Model
DTD	Document Type Definition
MIDI	Musical Instrument Digital Interface
SAX	Simple API for XML
SMF	Standard MIDI File
XML	EXtensible Markup Language
XSD	XML Schema Definition

## LISTA DE FIGURAS

Figura 2.1: Exemplo de arquivo MusicXML .....	17
Figura 4.1: Tela inicial do protótipo .....	22
Figura 4.2: Navegador de arquivos.....	23
Figura 4.3: A estrutura de pacotes e classes do projeto.....	24
Figura 4.4: Implementação da geração de código da nota MIDI .....	25

## **LISTA DE TABELAS**

Tabela 2.1: Principais mensagens MIDI de canal.....	14
Tabela 4.1: Principais conversões feitas entre os elementos de MusicXML e MIDI ....	26
Tabela 5.1: Comparação entre os arquivos de exemplo convertidos pelo Zong Player e pelo Player em desenvolvimento .....	28

## RESUMO

Uma área importante dentro da computação musical é a representação digital de partituras, que compreende o armazenamento e a manipulação de elementos musicais. O formato que tem se tornado padrão para representação de partituras digitais é o MusicXML (MXL2012), que é baseado no já consolidado padrão XML.

Este trabalho propõe um protótipo de tocador do formato MusicXML através da conversão para o formato MIDI, em plataforma Android. Desta forma, o player MIDI que já vem embutido na plataforma Android pode ser aproveitado para executar peças em MusicXML, que não são tipicamente suportadas pelos tocadores tradicionais.

Primeiramente, os conceitos fundamentais de MusicXML e MIDI são resumidos, e uma descrição das características principais deste tocador e do seu funcionamento é a seguir apresentada, assim como exemplos de uso.

**Palavras-Chave:** MusicXML, Android, MIDI.



# **Prototype of a MusicXML player in Android using MIDI**

## **ABSTRACT**

A very important area in computer music is the representation of digital sheet, which includes the storage and the handle of musical elements. A format that became standard to represent musical digital sheets is MusicXML, which is based in the consolidated standard XML.

This work proposes a prototype of MusicXML player, which converts into MIDI format, using Android. This way the MIDI Player that came with Android can be used to play MusicXML files that are not supported by traditional players. First, the main concepts of MusicXML and MIDI are briefly described and a description of the main characteristics of this player and its functioning is presented, with examples of use.

**Keywords:** MusicXML, MIDI, Android.

# 1 INTRODUÇÃO

A computação está cada vez mais presente na música. Há inúmeras formas de interação com computadores para produzir música. Os celulares, *tablets* e *smartphones* vêm ganhando espaço nessa área, uma vez que já possuem bons recursos de áudio e vídeo, estão presentes no cotidiano das pessoas e possuem formas muito interessantes de interação, como o acelerômetro e telas sensíveis ao toque.

A venda e uso de celulares com Android vem crescendo muito e, com isso, a demanda por novos aplicativos também cresce. Muitos aplicativos para a área musical vêm sendo desenvolvidos, para estimular a criação e o aprendizado. Tanto músicos leigos como profissionais estão cada vez mais buscando a tecnologia para auxiliar no processo criativo e formal da música.

## 1.1 Motivação

Uma área muito importante na computação musical é a área de representação digital musical. O uso de programas de edição e visualização de partituras é cada vez mais comum entre músicos profissionais e leigos. O formato MusicXML vem sendo adotado como novo padrão para representação de partituras, onde por muito tempo não havia padrão e muitos programas usavam formatos diferentes e complexos.

Porém, o formato MusicXML não possui tocador nativo, tanto em sistemas *desktop* como em sistemas web ou móveis. O padrão MIDI, apesar de suas limitações, possui tocador nativo nas placas de som de computadores, celulares e outros dispositivos móveis, bem como programas que reconhecem os comandos MIDI.

Já existem algumas implementações de tocadores MusicXML através da conversão para o formato MIDI e do uso de bibliotecas de som voltadas para este formato, mas a maioria para sistemas *desktop*. Isso acaba impossibilitando que peças musicais criadas e editadas em um programa que usa o formato MusicXML possam ser tocadas diretamente em celulares, *smartphones* e outros dispositivos móveis que usam Android.

## 1.2 Objetivo

Como aplicativos móveis geralmente apresentam restrições de recursos de hardware e de sistema operacional é interessante manter uma interoperabilidade entre sistemas móveis e sistemas *desktop*.

O objetivo deste trabalho de conclusão de curso é o estudo dos formatos musicais MusicXML e MIDI para o desenvolvimento de um protótipo básico de um tocador de MusicXML, utilizando a plataforma Android. Para isso será utilizado o tocador padrão do Android que já possui suporte ao formato MIDI.

Este protótipo permitirá que usuários de programas *desktop* de manipulação de partituras que utilizam MusicXML possam executar as peças em um dispositivo móvel com Android, sem necessitar de conversão via outro sistema.

## 1.3 Estrutura do trabalho

No capítulo 1 foi descrita a motivação deste trabalho, bem como o contexto da área de computação musical.

No segundo capítulo serão resumidos conceitos importantes para o entendimento deste trabalho, os formatos MIDI e MusicXML.

Alguns trabalhos relacionados com a área estão citados no capítulo 3.

No capítulo 4 estão detalhados o projeto e a implementação do protótipo, bem como algumas decisões de projeto.

No capítulo 5 estão descritos alguns resultados comparativos.

As conclusões deste trabalho, bem como sugestões para trabalhos futuros estão no capítulo 6.

## 2 CONCEITOS IMPORTANTES

Para a implementação deste protótipo, bem como para o bom entendimento deste trabalho, alguns conceitos importantes devem ser previamente apresentados.

### 2.1 MIDI

O MIDI é tanto um padrão para interfaces digitais musicais quanto para um formato de arquivo musical digital. É um formato amplamente utilizado que possui aplicativos capazes de executar as peças embutidos na maioria dos dispositivos. Também tem amplo uso em sintetizadores e outros instrumentos musicais eletrônicos.

O MIDI foi anunciado em 1982, mas foi utilizado pela primeira vez em um instrumento em 1983. Em 1983 a especificação do formato possuía apenas oito páginas para definir as instruções básicas. Logo houve uma demanda por novos tipos de mensagens MIDI para garantir um melhor controle dos sintetizadores. O formato padrão de arquivos MIDI foi definido em 1991.

Neste trabalho não focaremos nas interfaces, mas sim no formato de arquivo padrão MIDI, o SMF (*Standard MIDI File*). Os arquivos MIDI são definidos através de mensagens binárias, que são enviadas sequencialmente para o sintetizador do instrumento ou da placa de som.

Um arquivo MIDI é composto por um cabeçalho padrão geral, cabeçalhos individuais para cada trilha e sequências de mensagens sequenciais para cada trilha. As mensagens MIDI podem ser de três tipos: eventos MIDI, eventos exclusivos do sistema e meta-eventos.

#### 2.1.1 O cabeçalho de arquivo (header chunk)

O cabeçalho de arquivo aparece obrigatoriamente no início de cada arquivo MIDI e segue o seguinte padrão de sequência de *bytes*:

```
4D 54 68 64 00 00 00 06 ff ff nn nn dd dd
```

Onde os quatro primeiros *bytes* correspondem aos caracteres ASCII MThd, e marcam o início do arquivo. Os próximos quatro *bytes* indicam o tamanho total do

cabeçalho e vem sempre com o valor seis, que é o atual tamanho padrão para este cabeçalho.

Os dois bytes seguintes, marcados como “ff ff” representam o formato do arquivo. Há três possíveis formatos:

- 00 00: trilha única
- 00 01: múltiplas trilhas, síncrono
- 00 02: múltiplas trilhas, assíncrono

Neste trabalho não serão considerados os arquivos do tipo 00 02.

Os próximos dois *bytes*, marcados como “nn nn” informam o número de trilhas do arquivo. Os dois últimos *bytes* representam o número de *delta-time ticks* por semínima.

### 2.1.2 O cabeçalho de trilha (track chunk)

Cada trilha deve iniciar com um cabeçalho antes das suas respectivas mensagens e segue o seguinte padrão de sequência de *bytes*:

```
4D 54 72 6B xx xx xx xx
```

Onde os quatro primeiros *bytes* representam os caracteres ASCII MTrk, que indicam o início de uma nova trilha. Os próximos quatro *bytes*, denotados com “xx xx xx xx”, indicam a quantidade de *bytes* que esta trilha ocupa no arquivo, sem considerar o tamanho do próprio cabeçalho.

### 2.1.3 As mensagens MIDI

As trilhas MIDI são compostas por mensagens MIDI. Cada mensagem, seja ela mensagem de sistema, meta-mensagem ou evento MIDI, é precedida de um *delta-time*, que é o tempo que deve decorrer até que esta mensagem seja executada. Mensagens que não necessitam de marcações temporais, como mensagens de escrita de parte da letra da música, devem ter *delta-time* zero, bem como mensagens que são executadas imediatamente ou mensagens executadas simultaneamente com a mensagem anterior.

Para economizar espaço na representação destes *delta-times*, o padrão MIDI usa o formato de tamanho variável para esta representação. Os valores são representados em sete *bits* por *byte*, com o *byte* mais significativo primeiro. O último *bit* de cada *byte* serve para indicar se o próximo *byte* ainda faz parte da informação de *delta-time* ou se é início de uma nova mensagem. Todos os *bytes* que fazem parte do *delta-time* possuem o último bit habilitado, exceto o último *byte*, que possui o último *bit* em zero, para indicar o final desta informação.

Algumas das principais mensagens MIDI de canal estão detalhadas na tabela 2.1.

Tabela 2.1: Principais mensagens MIDI de canal

<i>Código da mensagem e número do canal MIDI (nnnn)</i>	<i>Parâmetros da mensagem</i>	<i>Descrição</i>
1000nnnn	0kkkkkkk 0vvvvvvv	Evento <i>Note Off</i> . Esta mensagem é enviada quando uma nota é liberada (desligada) (kkkkkkk) representa o número da nota e (vvvvvvv) a velocidade.
1001nnnn	0kkkkkkk 0vvvvvvv	Evento <i>Note On</i> . Esta mensagem é enviada quando uma nota é pressionada (ligada) (kkkkkkk) representa o número da nota e (vvvvvvv) a velocidade.
1010nnnn	0kkkkkkk 0vvvvvvv	Pressão de nota polifônica ( <i>Aftertouch</i> ). Esta mensagem é enviada na maioria das vezes pressionando a tecla após ela "chegar ao fundo". (kkkkkkk) é o número da nota e (vvvvvvv) é o valor da pressão.
1100nnnn	1100nnnn	Mudança de timbre. (ppppppp) é o novo valor de timbre conforme a tabela de timbres do MIDI.

## 2.2 MusicXML

O MusicXML (MXL2012) é um formato digital para representação de partituras musicais interativas em notação ocidental, bem como seu aspecto sonoro. É um formato aberto, sob licença “royalty-free”.

O Music XML foi baseado primariamente em dois formatos acadêmicos: O MuseData, desenvolvido por Walter Hewlett no Centro de Pesquisas para Computação Assistida na Humanidade (CCARH), localizada na Universidade de Stanford; e o

Humdrum, desenvolvido por David Huron, com base na Universidade do Estado de Ohio.

O formato foi desenvolvido pela Recordare, LLC, cujos direitos sobre o MusicXML foram comprados pela MakeMusic.

O formato foi planejado para ser o mais completo possível, para ser extensível e para ser simples, de forma que qualquer aplicação possa utilizá-lo sem prejuízos de função ou necessidade de adicionar funcionalidades que não possam ser representadas pelo formato.

O MusicXML surgiu em contrapartida a criação de formatos proprietários e dependentes da aplicação como uma alternativa para exibição, edição, performance e análise de partituras digitais. Diferentemente do formato MIDI, que é muito empregado para representar informações sobre os elementos musicais da peça não garantindo precisão no aspecto de representação visual, o formato MusicXML busca expressar cada partitura em toda a sua riqueza de detalhes. Um formato muito simples, com altíssimo poder de expressão e que vem sendo amplamente utilizado e aprimorado.

Sua utilização inclui mais de 150 aplicativos, como Finale, Sibelius, Capella, e programas que digitalizam partituras. Isso permite que músicos que usam diferentes aplicações possam compartilhar partituras e trabalhar de forma colaborativa, bem como um músico usar diferentes aplicativos para editar uma única partitura.

É baseado no padrão XML, que é um padrão consolidado em termos de especificação, aumentando ainda mais a sua aceitação. Com o uso do XML a preocupação com sintaxe é praticamente eliminada, aumentando a ênfase da semântica. Arquivos XML podem facilmente ser lidos e manipulados com o uso de um *parser*, que já vem embutido na maioria dos compiladores ou que pode ser facilmente encontrado em alguma biblioteca para praticamente qualquer linguagem de programação.

Está disponível para o formato MusicXML a definição tanto em XSD quanto em DTD, que são dois padrões aceitos e definidos pela W3C.

A preferência é que seja utilizada a definição em XSD, que provê uma definição muito mais precisa do formato MusicXML 3.0 que o DTD, e que é o padrão utilizado atualmente. O uso da definição em XSD para o desenvolvimento de *softwares* que utilizam o MusicXML os torna muito mais confiáveis, pelas garantias de validação que o XSD tem e que não estão presentes no DTD e possibilita o uso em conjunto com as últimas tecnologias em termos de XML.

O XSD do formato MusicXML é dividido em vários arquivos:

- Partwise/Timewise : define todos os elementos tanto de partituras organizadas por compassos em relação as trilhas quanto trilhas em relação aos compassos.
- Opus: uma coleção de partituras organizadas em uma entidade maior
- XLink namespace
- XML namespace
- Open Score Format Piano/Vocal/Guitar Profile
- Open Score Format Container

O MusicXML (na versão 3.0) também dispõe de seis arquivos de estilo XSLT para diferentes tipo de transformação musical. São eles:

- Parttime.xsl: converte arquivos partwise MusicXML em arquivos timewise.
- Timepart.xsl: converte arquivos timewise MusicXML em arquivos partwise.
- To20.xsl: converte arquivos MusicXML 3.0 files em arquivos MusicXML 2.0.
- To11.xsl: converte arquivos MusicXML 2.0 em arquivos MusicXML 1.1.
- To10.xsl: converte arquivos MusicXML 1.1 em arquivos MusicXML 1.0.
- MidiXml.xsl: normaliza os arquivos MIDI XML em relação aos valores absolutos de *timestamps*.

Os elementos do MusicXML seguem as mesmas regras dos elementos de um arquivo XML comum.

Por decisão de projeto, características de elementos são geralmente representadas por caracteres entre *tags*, como por exemplo, a altura da nota. Atributos de *tag* servem, na maioria das vezes, para representar identificadores referentes ao elemento da *tag*, como um identificador para a atual trilha.

Na figura 2.1 é apresentado um exemplo simples de como é estruturado um arquivo MusicXML. A definição de todas as possíveis tags encontra-se na página oficial, tanto no formato DTD quanto XSD.



```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
  "-//Recordare//DTD MusicXML 3.0 Partwise//EN"
  "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>

```

Figura 2.1: Exemplo de arquivo MusicXML (MXL2012).

Esta figura representa uma partitura extremamente simples, com apenas um compasso, com uma semibreve de uma nota dó central, 4/4, na clave de sol.

Como pode ser percebido na figura, o formato é simples e as informações podem ser facilmente extraídas, porém é um formato bastante verboso.

Pela figura 2.1 podemos perceber algumas características básicas dos arquivos MusicXML. Todos os arquivos devem começar com o cabeçalho padrão do formato XML. Logo depois temos a *tag* de <score-partwise>. Há duas formas de gerar um arquivo MusicXML: score-partwise e score-timewise. O formato score-partwise indica que o arquivo será estruturado em trilhas e que cada trilha será estruturada em compassos. O formato score-timewise é estruturado por compasso, onde cada compasso pode ter várias trilhas correspondentes.

Como o arquivo de exemplo é score-partwise, segue a lista das trilhas que, neste caso, é apenas uma trilha. Depois cada trilha é detalhada e separada em seus compassos (*measures*).

Algumas *tags* servem apenas para agrupar elementos, como a *tag* `<time>`, que indica que dentro dela será definida a marcação de tempo, com duas *tags* aninhadas, cada uma com seu parâmetro.

Algumas *tags* não possuem abertura e fechamento padrão, como é o caso da *tag* `</chord>`, que vem aninhada dentro da *tag* `<note>` quando há um acorde. Ela indica que a nota atual deve ser tocada simultaneamente com a nota anterior na sequência do arquivo, mas não tem parâmetros próprios.

Algumas *tags* servem para representar apenas aspectos visuais da partitura (como a *tag* `<lyrics>`) e outras também representam o aspecto sonoro, como a altura tonal e o timbre.

Todas as *tags* que são reconhecidas pelo padrão MusicXML, suas definições, possíveis parâmetros e regras de formação estão definidas nos arquivos DTD e XSD.

Um arquivo MIDI pode ser representado por um arquivo MusicXML, mas o contrário não acontece. O arquivo MusicXML pode representar os aspectos sonoros e visuais presente no MIDI. Porém, o arquivo MIDI é muito mais restrito, não representando visualmente repetições na partitura ou múltiplas vozes em uma mesma trilha, por exemplo.

Como um arquivo MusicXML é um gravado em formato de texto ASCII, ele pode acabar ocupando uma quantidade considerável a mais de memória que um arquivo MIDI, que é puramente binário. Por isso existem os arquivos com extensão `.mxl`, que são arquivos MusicXML comprimidos. Estes arquivos conseguem uma boa taxa de compressão.

Geralmente atributos de *tags* representam algum índice ou identificador, como em `<measure number="1">`, e grande parte dos parâmetros são representados como caracteres entre *tags*, como em `<beats>4</beats>`.

Algumas *tags* representam apenas aspectos visuais da peça, como a *tag* `<clef>`, que representa a clave. Outras também guardam informação sobre os aspectos sonoros da peça musical, como a *tag* `<step>`, que define qual a nota a ser exibida ou tocada.

### 2.2.1 MusicXML 3.0

O formato 3.0 foi lançado em agosto de 2011 e é a atual versão até a data de escrita deste artigo. Esta versão inclui tanto o formato de DTD (Document Type Definition) quanto o SXD (XSL Schema Definition).

As novas versões do formato apenas complementam as versões anteriores, sem criar mudanças incompatíveis.

As mudanças em relação ao formato 2.0 incluem três áreas principais:

- Uma taxonomia padrão de 886 timbres de instrumentos, para permitir uma melhor integração entre as aplicações;
- Cobertura de mais repertórios de partituras, incluindo notações diferentes das ocidentais, como a notação Chinesa, Persa, Turca e Iraquiana;
- Suporte para descrições novas e mais detalhadas de notação musical.

## 2.3 Android

Android é um sistema operacional baseado em Linux desenvolvido para dispositivos móveis como *tablets* e *smartphones*, principalmente com tela sensível ao toque. Android é *open source* e a Google detém seus direitos, que antes pertenciam à Android Inc.

Por ser *open source* o sistema permite que o software possa ser livremente modificado e distribuído por fabricantes de dispositivos, operadoras sem fio e desenvolvedores entusiastas. Além disso, o Android tem uma grande comunidade de desenvolvedores de aplicativos ("apps") que ampliam a funcionalidade dos dispositivos. A linguagem é uma espécie de versão personalizada da linguagem Java.

Como foi desenvolvido principalmente para dispositivos com tela sensível ao toque, possui várias formas de interação e manipulação. Pode responder bem aos chamados de *hardware* em relação ao acelerômetro, giroscópio, sensor de proximidade e aos toques na tela segundo as diferentes interações, como arrastar o dedo sobre a tela.

A tela inicial é muito semelhante aos sistemas operacionais convencionais para *desktop*, com ícones e ações associadas a eles.

O desenvolvimento de aplicativos é feito na linguagem Java adaptada, utilizando o SDK (software development kit) do Android. O SDK inclui um conjunto abrangente de ferramentas de desenvolvimento, incluindo um depurador, bibliotecas de *software*, um emulador de terminal com base no QEMU, documentação de código de amostra, e tutoriais. O ambiente de desenvolvimento suportado oficialmente integrado (IDE) é o Eclipse usando o Android Development Tools (ADT) *plugin*.

## 3 TRABALHOS RELACIONADOS

O formato MusicXML é um formato recente. Os programas de edição de partitura e programas sequenciadores que utilizam MusicXML já apresentam um tocador embutido, mas a maioria deles é de código fechado. Um conversor *desktop* que vem sendo desenvolvido com licença GNU General Public License é o Zong Player (ZON2012). Este programa tem tanto a opção de gerar um novo arquivo MIDI que converte o MusicXML de entrada quanto apenas tocar um arquivo MusicXML através do envio de mensagens MIDI à placa de som.

O programa foi desenvolvido utilizando a linguagem Java. O programa é complexo e tenta realizar a conversão de forma precisa, com cobertura de todos os elementos MusicXML.

Porém, pouco pôde ser aproveitado das classes e do algoritmo utilizado para a implementação do protótipo em Android. O Zong foi implementado utilizando um *parser* do tipo DOM (Document Object Model), que cria uma árvore em memória com todas as *tags* e possui métodos específicos para o acesso delas. O uso deste modelo de *parser* é inviável em celulares, pois estes possuem recursos bastante limitados em relação à memória.

Outro programa disponível com código aberto é o MuseScore (MUS2012). Ele é desenvolvido em C++ e inclui muitas outras funções, como edição e visualização de partituras. O programa é altamente complexo, possui várias dependências para a sua compilação e dificilmente serviria para os propósitos deste trabalho.

Recentemente foi desenvolvido o aplicativo MuseScore para Android. Isso permite que os usuários possam fazer suas edições, visualizações e ouvir o resultado das suas composições tanto no seu *desktop* quanto no seu dispositivo móvel. Porém, o aplicativo não é grátis e, diferentemente da versão *desktop*, ainda não possui código aberto.

## 4 IMPLEMENTAÇÃO

Nas próximas seções serão esclarecidos os detalhes de projeto e implementação do protótipo proposto. O código fonte está disponível no site: <https://github.com/alkerber/Converter>.

### 4.1 O escopo

Nem todas as *tags* MusicXML estão tratadas neste protótipo. O foco do projeto é criar um tocador de MusicXML, portanto muitas *tags* MusicXML que são compatíveis com o padrão MIDI que não estão relacionadas ao aspecto sonoro (como a parte de letras da partitura) não foram tratadas neste projeto.

Outras *tags* também não foram tratadas pela limitação de tempo do projeto, uma vez que o formato é extenso e complexo, e nem todas as conversões necessárias de MusicXML para MIDI estão explícitas nos manuais.

A interface é simples e funcional, uma vez que serve apenas para auxiliar na depuração do projeto. Não está no escopo deste trabalho pesquisar melhores formas de interação com o usuário final.

### 4.2 A plataforma

Este protótipo foi implementado utilizando a plataforma Android. O emulador de dispositivo utilizado foi o MOTODEV Studio, da Motorola.

Ele é um pacote completo de desenvolvimento. O instalador já inclui o ambiente de desenvolvimento Eclipse com as ferramentas de desenvolvimento Android ADT (Android Development Tools) e ainda faz o *download* automático da SDK do Android.

Infelizmente a Motorola não fornece mais esse pacote. A Motorola ainda manteve a seção para desenvolvedores ([http://www.motorola.com/sites/motodev/us-en/motodev\\_lp.html](http://www.motorola.com/sites/motodev/us-en/motodev_lp.html)), mas a recomendação é que os componentes necessários para o desenvolvimento em Android sejam instalados manualmente.

### 4.3 A interface

A interface do protótipo foi desenvolvida em Android, utilizando o ambiente MOTODEV. A interface do projeto não é o foco do projeto, portanto foi mantida simples e funcional.

A navegação pode ser feita entre duas telas, uma com um tocador e a outra com um seletor de arquivo.

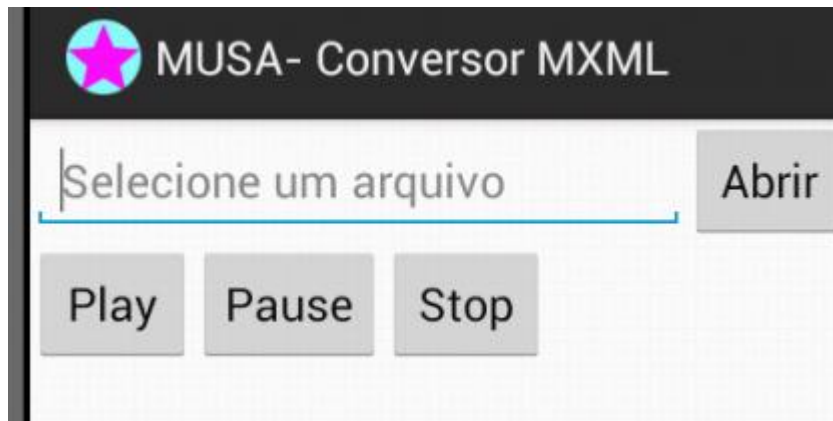


Figura 4.1: Tela inicial do protótipo.

A figura 4.1 mostra a tela inicial do aplicativo, com os botões que chamam as funções *do media player* embutido no Android e o botão de abrir, que acessa a tela de navegador de arquivos.

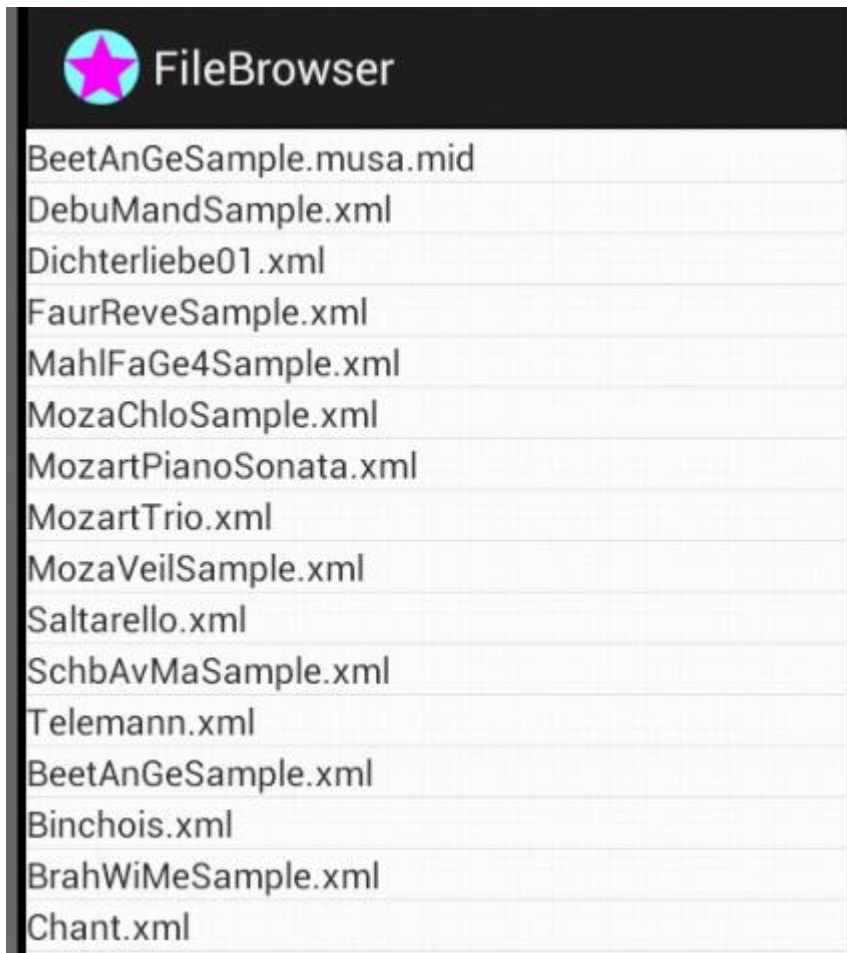


Figura 4.2: Navegador de arquivos.

Na figura 4.2 está o navegador de arquivos. Ao selecionar um arquivo com extensão .xml a função de conversão é chamada, e o arquivo convertido é passado ao tocador, que se encontra na tela principal.

#### 4.4 O parser

Diferentemente da plataforma JavaME (Java para dispositivos móveis, que foi pouco utilizada), a plataforma Android possui suporte ao *parser* do tipo DOM. Porém, não é recomendado que se use o modelo DOM, que acaba por gerar uma árvore em memória com todos os elementos do arquivo XML a ser trabalhado. Para pequenos arquivos e arquivos que necessitam uma navegação maior entre os elementos o *parser* do tipo DOM pode ser muito útil e muito bem utilizado. Porém, para arquivos muito grandes e onde a leitura dos elementos é feita uma só vez, ele se torna custoso em termos de memória e processamento.

Por isso foi utilizado o *parser* do tipo SAX nesta implementação. Parsers SAX são orientados a eventos. Os principais eventos tratados são: início de arquivo, final de arquivo, início de *tag*, fim de *tag* e aparecimento de caracteres entre as *tags*. Sua manipulação é muito simples e não há a necessidade de manter a árvore de *tags* XML em memória.

## 4.5 A estrutura do projeto

O projeto foi estruturado conforme a figura 4.3. O projeto foi implementado usando o paradigma de orientação a objetos. Várias classes foram criadas e separadas em pacotes com funcionalidades relacionadas. Algumas classes servem para representar meta-objetos, apenas armazenando os parâmetros para posterior utilização.

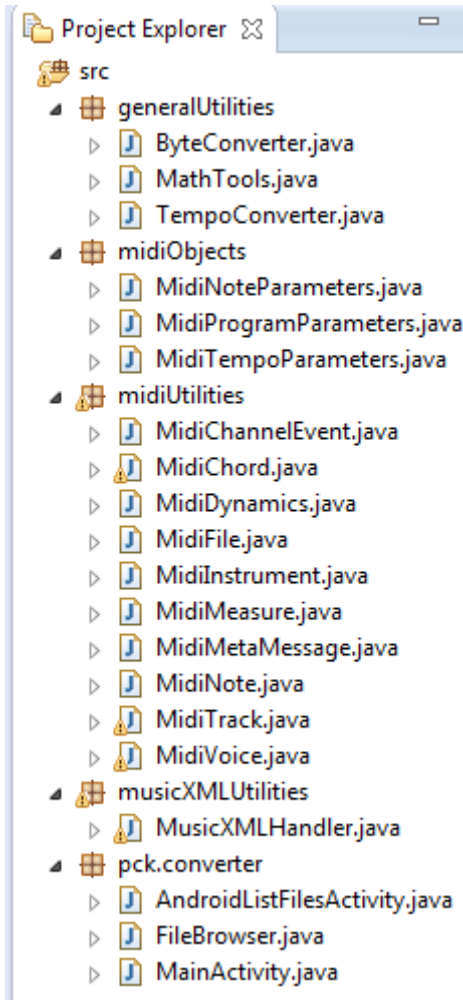


Figura 4.3: A estrutura de pacotes e classes do projeto.

O pacote `generalUtilities` contém classes responsáveis por conversões e operações indiretamente relacionadas ao trabalho, como operações matemáticas e deslocamento de *bits* em um *byte*.

O pacote `midiObjects` contém classes responsáveis pelo armazenamento de parâmetros de alguns elementos. Não possuem métodos associados, portanto não realizam nenhuma operação, e a manipulação dos objetos gerados deve ser feita por outra classe.

No pacote `midiUtilities` estão as classes de conversão dos elementos de MusicXML para MIDI. A maioria dos métodos recebe parâmetros diretamente do *parser* ou manipula os objetos das classes do pacote `midiObjects`, criados pelo *parser*.



O *parser* de arquivo se encontra no pacote `musicXMLUtilities`. Foi utilizado um *parser* SAX, já presente no Android. Este tipo de *parser* é orientado a eventos. Os métodos que tratam eventos e que vem por padrão no *parser* devem ser sobrescritos para gerar os resultados esperados. Para cada um dos métodos do *parser* foi atribuída uma função:

- Início de arquivo: criar um novo arquivo MIDI
- Início de *tag*: criar objetos para armazenar os parâmetros a serem lidos nos caracteres ou que foram lidos como atributos da *tag*.
- Final de *tag*: armazenar na trilha correta os eventos MIDI correspondentes à *tag*.
- Caracteres entre *tags*: ler os parâmetros e armazenar no objeto correspondente criado pelo início da *tag*.
- Final de arquivo: escrever as *tags* armazenadas no arquivo MIDI criado.

Os métodos descritos na classe `MusicXMLHandler` (responsável pelo *parser*) determinam o fluxo da conversão, chamam os métodos das classes de `midiUtilities` ou criam objetos das classes de `midiObjects`.

O código representado pela figura 4.4 mostra como foi feita a conversão dos valores de parâmetros de nota MusicXML para um valor hexadecimal MIDI.

```
public byte returnNoteNumber() {
    byte note = 0x00;
    switch (this.step) {
        case 'C':
            note = 0x0C;
            break;
        case 'D':
            note = 0x0E;
            break;
        case 'E':
            note = 0x10;
            break;
        case 'F':
            note = 0x11;
            break;
        case 'G':
            note = 0x13;
            break;
        case 'A':
            note = 0x15;
            break;
        case 'B':
            note = 0x17;
            break;
    }
    note = (byte) (note + (byte) (this.octave * 12 + alter));
    return note;
}
```

Figura 4.4: Implementação da geração de código da nota MIDI.

O valor de nota retornado pela implementação da figura 4.4 é usado nas mensagens MIDI de ligar e desligar nota.

A tabela 4.1 representa algumas das principais conversões feitas, detalhando a correspondência entre os eventos MIDI e os eventos MusicXML.

Tabela 4.1: Principais conversões feitas entre os elementos de MusicXML e MIDI

<i>Tags e atributos MusicXML</i>	<i>Códigos e parâmetros MIDI</i>
<note>	Indica que um novo evento MIDI note on vai acontecer.
<step>	MIDI note on. Parâmetro é usado no cálculo do valor da nota MIDI
<alter>	MIDI note on. Parâmetro é usado no cálculo do valor da nota MIDI
<octave>	MIDI note on. Parâmetro é usado no cálculo do valor da nota MIDI
<duration>	Usado para calcular o delta time para o evento MIDI note off
<chord/>	Indica a presença de notas simultâneas. Essas notas terão o delta time inicial = 0x00, indicando que ocorrem ao mesmo tempo do evento anterior de ligar nota.
<sound tempo="x">	MIDI set tempo -> $60.000.000 / x$
<sound dynamics="x">	O valor x em MusicXML é a porcentagem sobre 90, que representa o <i>forte</i> em MIDI. Valor MIDI-> $x/100 * 90$
<midi-channel>	O primeiro canal MusicXML é o 1, enquanto no MIDI é o 0. MIDI channel = MusicXML channel - 1
<midi-program>	O primeiro programa MusicXML é o 1, enquanto no MIDI é o 0. MIDI program = MusicXML program - 1
</rest>	MIDI note on com velocidade 0 ou soma do delta time até o próximo evento

Como pode ser visto na tabela 4.1, os valores de canal e programa variam no MusicXML e no MIDI. Os canais MIDI são numerados a partir do canal 0, enquanto os canais MusicXML começam no índice 1.

Os arquivos necessários para a interface estão no pacote `pck.converter`.

Como o valor final do *delta-time ticks* é calculado baseado no valor final do mínimo múltiplo comum entre as *divisions* de cada trilha há três alternativas possíveis:

- armazenar todas as informações sobre as trilhas em memória, para posterior escrita no arquivo MIDI
- ler o arquivo mais de uma vez, armazenando apenas as *divisions* na primeira leitura. Na segunda leitura os compassos já podem ser escritos no arquivo MIDI na medida em que vão sendo lidos e processados pelo *parser*.
- sobrescrever trechos do arquivo MIDI gerado, alterando o delta-time relativo de cada nota para o valor absoluto, que foi calculado no final do arquivo.

Este protótipo foi implementado armazenando as trilhas em memória, para facilitar também a adição de novas funcionalidades depois, como *multi-part* e acordes. A alternativa de sobrescrever o arquivo MIDI foi descartada pela complexidade desnecessária. Seria necessário armazenar a posição no arquivo MIDI de todos os delta-times, calcular o valor absoluto e substituir os valores relativos gravados no arquivo.

As principais dificuldades encontradas são referentes a arquivos com mais de uma voz, arquivos com repetição e com acordes. Estas dificuldades não foram tratadas no protótipo, estando descritas com mais detalhes na conclusão, como base para trabalhos futuros.

O formato MIDI não representa múltiplas vozes na mesma trilha de forma separada, como o MusicXML. No MusicXML primeiro é detalhado todo o compasso da primeira voz e depois é usado uma *tag* de `<backup>` com o tempo correspondente, para indicar que se está voltando o contador de tempo para detalhar a outra voz que deve ser executada no mesmo compasso. Para representar isso no MIDI é necessário armazenar temporariamente os compassos de todas as vozes da mesma trilha e calcular quais notas serão tocadas simultaneamente na forma de acorde e calcular os devidos *delta-times*.

O formato MIDI também não tem suporte à repetição. Seria necessário verificar onde começa o trecho a ser repetido, guardar todas as mensagens MIDI relativas a esse trecho na memória e depois duplicar este código no arquivo MIDI.

Quanto aos acordes, é necessário armazenar temporariamente todas as notas do acorde, ordená-las de forma crescente em relação à duração, gerar o evento MIDI de ligar cada uma delas, com *delta-time* zero e ir desligando conforme as notas de duração menor.

## 5. RESULTADOS

O protótipo desenvolvido ainda é um modelo restrito, porém já possui funcionalidades básicas. Foram feitos testes usando os arquivos MusicXML de modelo do site do detentor dos direitos e usando os arquivos convertidos pelo Zong player e pelo protótipo. Foi utilizado um leitor de arquivos em hexadecimal, para avaliar as mensagens MIDI. Todos os testes foram realizados no emulador que foi utilizado na implementação.

O objetivo dos testes foi verificar o quanto os arquivos gerados pelo protótipo eram próximos dos arquivos gerados por um programa já consolidado. Os resultados estão na tabela 5.1, exibindo algumas características dos arquivos testados e o quanto eles se aproximaram do tocador de exemplo tomado como base. A coluna dificuldade se baseia em um critério subjetivo, onde arquivos sem acordes, sem múltiplas trilhas ou vozes são considerados fáceis, arquivos com alguma característica marcante como acordes ou múltiplas vozes são considerados médios e arquivos com algumas dessas características marcantes exibidas frequentemente são considerados difíceis.

Tabela 5.1: Verificação entre os arquivos de exemplo convertidos pelo Zong Player e pelo Player em desenvolvimento.

Verificação entre os arquivos de exemplo convertidos pelo Zong Player e pelo Player em desenvolvimento									
arquivo xml	n° de tracks	multi-part	multi-voice	program change	chords	dificuldade	comparação	observações	percepção auditiva
BeetAnGeSample	2	sim	sim	sim	sim	médio	binários com algumas semelhanças	problemas com multi-part tocado sequencial	similar
Bichois	2	sim	sim	não	não	difícil	mensagens midi semelhantes	problema com o início assíncrono	similar, mas prejudicada pelas trilhas não-sincronizadas
BrahWiMeSample	2	sim	sim	sim	sim	médio	binário inicia igual	multi-part tocado sequencial	harmonioso, porém algumas partes paralelas são tocadas sequencialmente
BrookWestSample	2	não	não	sim	sim	fácil	binários com algumas semelhanças	trocas com nro de canal	trechos podem ser reconhecidos
Chant	1	não	não	não	não	fácil	arquivos binários idênticos	-	idêntica

DebuMandSample	2	sim	sim	sim	sim	difícil	mensagens midi semelhantes	problemas com contagem do tempo	harmonioso, mas muito prejudicado pelas partes paralelas tocadas sequencialmente
Dichterliebe	2	sim	sim	sim	sim	difícil	mensagens midi semelhantes	multi-part tocado sequencial	semelhantes, porém algumas partes paralelas são tocadas sequencialmente
Echigo-Jishi	1	não	não	sim	não	fácil	arquivos binários idênticos	-	idêntica
FaurReveSample	2	sim	sim	sim	sim	difícil	mensagens midi semelhantes	multi-part tocado sequencial	semelhantes, porém algumas partes paralelas são tocadas sequencialmente
MahlFaGe4Sample	2	sim	sim	sim	sim	difícil	mensagens midi semelhantes	multi-part tocado sequencial	semelhantes, porém algumas partes paralelas são tocadas sequencialmente
MozaChloSample	2	sim	sim	sim	sim	difícil	mensagens midi semelhantes	multi-part tocado sequencial	harmonioso, porém algumas partes paralelas são tocadas sequencialmente
MozartPianoSonata	1	sim	sim	sim	sim	difícil	mensagens midi semelhantes	multi-part tocado sequencial	semelhantes, porém algumas partes paralelas são tocadas sequencialmente
MozartTrio	5	não	não	sim	sim	difícil	binário inicia igual	problemas com contagem do tempo	diferente e desarmonioso
MozaVeilSample	2	sim	sim	sim	sim	difícil	mensagens midi semelhantes	multi-part tocado sequencial	semelhantes, porém algumas partes paralelas são tocadas sequencialmente
Saltarello	1	não	não	não	não	fácil	arquivos binários idênticos	-	idêntica
SchbAvMaSample	2	sim	sim	sim	sim	difícil	mensagens midi semelhantes	multi-part tocado sequencial	semelhantes, porém algumas partes paralelas são tocadas sequencialmente
Telemann	2	sim	sim	sim	sim	difícil	mensagens midi semelhantes	multi-part tocado sequencial	semelhantes, porém algumas partes paralelas são tocadas sequencialmente

Analisando a tabela 5.1 é possível notar que o protótipo realiza bem a conversão nos arquivos simples, com apenas uma trilha e sem acordes. Nos arquivos mais complexos, com *multi-part* e acordes, a conversão continua sendo feita, porém as notas são gravadas sequencialmente. Algumas peças com variação de parâmetros de tempo foram convertidas com os valores corretos de altura tonal e timbres, porém com problemas ao atribuir os corretos parâmetros de tempo às notas.

Quando a peça apresenta mais de uma voz, as vozes são tocadas sequencialmente, mas mantendo a marcação temporal. Quando há acordes, muitas vezes a marcação temporal fica comprometida, uma vez que alguma das vozes pode apresentar duração maior que a outra em função das notas do acorde serem tocadas sequencialmente. Geralmente a marcação temporal errada é mais sentida em peças que utilizam mais de um canal, pois os canais são tocados simultaneamente e acabam ficando não sincronizados.

## 6. CONCLUSÃO

O formato MusicXML, apesar de simples, é extenso e completo. Nem toda documentação está bem detalhada e nem todas as conversões são explícitas.

O protótipo desenvolvido ainda é um modelo restrito, porém já possui funcionalidades básicas.

Este protótipo pode facilmente ser utilizado de base para trabalhos futuros, por ser desenvolvido usando os princípios da orientação a objetos, podendo ser facilmente complementado.

Uma melhoria que pode ser feita é armazenar apenas uma trilha em memória por vez. Isso será possível se o arquivo MusicXML for lido mais de uma vez, sendo que a primeira leitura seria apenas para ler os parâmetros de *divisions*.

O protótipo só tem suporte a arquivo com uma voz, sem acordes, e no modelo *score-partwise*.

Algumas sugestões para trabalhos futuros são detalhadas durante esta conclusão.

É possível implementar as funcionalidades faltantes através de algumas modificações. Acordes podem ser facilmente incluídos, através de uma classe que trate os acordes e gere o código MIDI correspondente. O algoritmo é simples e consiste em ordenar o vetor de durações das notas, ligar todas as notas MIDI com *delta time* zero (indicando que serão tocadas simultaneamente) e calcular a duração faltante para enviar os eventos de desligar nota.

Arquivos com mais de uma voz também podem ser implementados, através da análise das *tags* `<backup></backup>` e `<forward></forward>`. É necessário armazenar temporariamente cada compasso (*measure*), para calcular os *delta-times* correspondentes. O cálculo dos *delta-times* fica um pouco mais complicado, pois o MIDI não tem uma representação direta de mais de uma voz. Uma alternativa simples e rápida seria alocar mais um canal MIDI para a outra voz. Mais isso poderia gerar conflitos com outros canais. O ideal seria fazer o cálculo dos *delta-times*, analisando onde as notas das diferentes vozes da mesma trilha se intercalam.

Para oferecer suporte a arquivos *score-timewise* seriam muitas necessárias muitas mudanças estruturais, principalmente pelo fato de que o *parser* utilizado é orientado a eventos. O comportamento do *parser* teria que ser alterado, adicionando uma cláusula de teste para cada *tag* relativa envolvida, como *tags* de `<measure>` e `<part>`, por exemplo.

Também é possível acrescentar funções para converter partes não sonoras, mas que possuem correspondência em MIDI adicionando novas cláusulas na estrutura do parser.



## REFERÊNCIAS

- (EXA2012) **Revista Exame: Brasil é o terceiro que mais cresce em Android e iOS**  
(<http://exame.abril.com.br/tecnologia/noticias/brasil-e-o-terceiro-que-mais-cresce-em-android-e-ios>) última visita em 18/12/2012
- (GOO2006) GOOD,M. Lessons from the Adoption of MusicXML as an Interchange Standard, **Recordare LLC**, 2006
- (MAN2012) **MuseScore Manual** (<http://musescore.org/en/developers-handbook/references/design-implementation>) última visita em 18/12/2012
- (MID2012) **MIDI Manufacturers Association**  
(<http://www.midi.org/techspecs/midimessages.php>) última visita em 18/12/2012
- (MUS2012) **MuseScore para Android** (<http://musescore.com/android>) última visita em 18/12/2012
- (MXL2012) **MusicXML** (<http://www.makemusic.com/musicxml/specification>) última visita em 18/12/2012
- (PIM2009) PIMENTA,M. S.; FLORES, L.V.; CAPASSO, A.; TINAJERO,P.; KELLER, D. Ubiquitous Music: Concepts and Metaphors, **Simpósio Brasileiro de Computação Musical**, 2009
- (TER2012) **Tecnologia Terra: Pesquisa: 14% dos brasileiros têm smartphone; Android domina**  
(<http://tecnologia.terra.com.br/noticias/0,,OI5776362EI15606,00-Pesquisa+dos+brasileiros+tem+smartphone+Android+domina.html>) última visita em 18/12/2012
- (ZON2012) **Zong MusicXML Player**  
(<http://zongmusic.wordpress.com/category/musicxml/>) última visita em 18/12/2012

## ANEXO: TABELA DE TAGS E ELEMENTOS MUSICXML

Name	File	Type	Si nc e	Description
accelerate	comm on.mo d	Attri but e	1, 0	Accelerate playback at end? Values: yes, no
accent	note.m od	Ele me nt	1, 0	Horizontal accent
accidental	note.m od	Ele me nt	1, 0	Notated accidental. Values: sharp, natural, flat, double-sharp, sharp-sharp, flat-flat, natural-sharp, natural-flat, quarter-flat, quarter-sharp, three-quarters-flat, three-quarters-sharp; sharp-down, sharp-up, natural-down, natural-up, flat-down, flat-up, triple-sharp, triple-flat, slash-quarter-sharp, slash-sharp, slash-flat, double-slash-flat, sharp-1, sharp-2, sharp-3, sharp-5, flat-1, flat-2, flat-3, flat-4, sori, and koron added in 3.0
accidental- mark	note.m od	Ele me nt	1, 0	Accidental mark modifying a notation or ornament, such as turns or trills. Values: same as for accidental
accidental- text	comm on.mo d	Ele me nt	2, 0	Formatted accidental used by part-name-display and similar elements. Values: same as for accidental
accord	directi on.mo d	Ele me nt	1, 1	Individual string tuning for scordatura
accordion- high	directi on.mo d	Ele me nt	2, 0	Is high section of accordion registration present?
accordion- low	directi on.mo d	Ele me nt	2, 0	Is low section of accordion registration present?
accordion- middle	directi on.mo d	Ele me nt	2, 0	Is middle section of accordion registration present? Values: 1, 2, 3
accordion- registration	directi on.mo d	Ele me nt	2, 0	Accordion registration symbol
actual-notes	comm on.mo d	Ele me nt	1, 0	How many notes played in given time
additional	attribut es.mo d	Attri but e	2, 0	If "yes", indicates that this is an added clef (e.g. for cues), not a substitution
after-barline	attribut es.mo d	Attri but e	3, 0	Does a clef at the start of a measure appear after the barline? Values: yes, no
alter	note.m od	Ele me nt	1, 0	Chromatic alteration in semitones. May use decimal values for microtones.
alternate	comm on.mo	Attri but	1, 0	Alternate fingering? Values: yes, no

	d	e		
any	sound.s.dtd	Element	3,0	Mapping of any instrument sound, solo or ensemble, between its MusicXML ID and an application or library ID
appearance	layout.mod	Element	2,0	General graphical settings for music's final form appearance
approach	note.mod	Attribute	3,0	How the beginning of a compound ornament looks relative to the main part. Values: above, below
arpeggiate	note.mod	Element	1,0	Part of arpeggiated chord
arrow	note.mod	Element	3,0	Arrow used for a musical technical indication
arrow-direction	note.mod	Element	3,0	Direction for straight arrow. Values: left, up, right, down, northwest, northeast, southeast, southwest, left right, up down, northwest southeast, northeast southwest, other
arrow-style	note.mod	Element	3,0	Arrow visual style. Values: single, double, filled, hollow, paired, combined, other
articulations	note.mod	Element	1,0	Articulations and accents
artificial	note.mod	Element	1,0	Artificial harmonic
attack	note.mod	Attribute	1,0	Offset from starting time for playback, in divisions
attribute	identity.mod	Attribute	2,0	Which attribute of an element is supported or not
attributes	attributes.mod	Element	1,0	Musical information that typically changes at measure boundaries
backup	note.mod	Element	1,0	Move backward to coordinate multiple voices in one part
barline	barline.mod	Element	1,0	Barline information
barre	direction.mod	Element	1,1	Indicate barre chords in a frame
bar-style	barline.mod	Element	1,0	Values: regular, dotted, dashed, heavy, light-light, light-heavy, heavy-light, heavy-heavy, none; tick and short added in 2.0
base-pitch	note.mod	Element	1,0	Base pitch for harmonic, not what sounds
bass	direction.mod	Element	1,0	Altered bass as in pop music (as opposed to using inversions)
bass-alter	direction.mod	Element	1,0	Pitch alteration for bass

bass-step	direction.mod	Element	1,0	Pitch step for bass
beam	note.mod	Element	1,0	Beam type. Values: begin, continue, end, forward hook, backward hook.
beam-level	common.mod	Entity	1,0	Distinguishes eighth to 1024th beams Values: 1, 2, 3, 4, 5, 6; 7 and 8 added in 3.0
beater	direction.mod	Element	3,0	Pictograms for beaters that do not distinguish materials. Values: bow, chime hammer, coin, finger, fingernail, fist, guiro scraper, hammer, hand, jazz stick, knitting needle, metal hammer, snare stick, spoon mallet, triangle beater, triangle beater plain, wire brush
beat-repeat	attributes.mod	Element	1,0	Beat repeat mark
beats	attributes.mod	Element	1,0	Time signature numerator
beats	common.mod	Attribute	1,0	How many added beats in the trill / bend
beat-type	attributes.mod	Element	1,0	Time signature denominator
beat-unit	direction.mod	Element	1,0	Metronome beat unit; same values as type element
beat-unit-dot	direction.mod	Element	1,0	Metronome beat unit augmentation dot
bend	note.mod	Element	1,0	String bends, as in guitar and tab notation
bend-alter	note.mod	Element	1,0	Number of semitones in bend; decimal values for microtones
bend-sound	common.mod	Entity	1,0	Collection of bend playback attributes
bezier	common.mod	Entity	1,0	Collection of Bezier curve attributes
bezier-offset	common.mod	Attribute	1,0	Bezier point: horizontal in divisions
bezier-offset2	common.mod	Attribute	1,0	Outgoing Bezier point for continue type: horizontal in divisions
bezier-x	common.mod	Attribute	1,0	Bezier point: horizontal in tenths
bezier-x2	common.mod	Attribute	1,0	Outgoing Bezier point for continue type: horizontal in tenths
bezier-y	common.mod	Attribute	1,0	Bezier point: vertical in tenths

	d	e		
bezier-y2	comm on.mo d	Attri but e	1, 0	Outgoing Bezier point for continue type: vertical in tenths
blank-page	directi on.mo d	Attri but e	2, 0	Number of blank pages to insert before current measure
bookmark	link.mo d	Ele me nt	1, 0	Well-defined target for incoming XLink
bottom- margin	layout. mod	Ele me nt	1, 1	Bottom margin in tenths
bottom-staff	attribut es.mo d	Attri but e	2, 0	Bottom staff for part-symbol element
bracket	comm on.mo d	Attri but e	1, 0	Show level with brackets? Values: yes, no
bracket	directi on.mo d	Ele me nt	1, 0	Flexible line and bracket definition, often combined with words
bracket	directi on.mo d	Attri but e	2, 0	Tuplet bracket displayed in metric relationship? Values: yes, no
bracket	note.m od	Attri but e	1, 0	Tuplet bracket displayed? Values: yes, no
bracket- degrees	directi on.mo d	Attri but e	1, 1	All harmony degrees should be in a bracket
breath-mark	note.m od	Ele me nt	1, 0	Breath mark. Values: comma, tick, empty string. Text values added in 3.0.
caesura	note.m od	Ele me nt	1, 0	Caesura or railroad tracks
cancel	attribut es.mo d	Ele me nt	1, 0	Key signature cancellation
cancel	attribut es.mo d	Attri but e	2, 0	Does key-octave refer to key signature cancellation? Values: yes, no
capo	attribut es.mo d	Ele me nt	1, 0	Capo fret for tablature
cautionary	note.m od	Attri but e	1, 0	Cautionary accidental? Values: yes, no
chord	note.m od	Ele me nt	1, 0	Indicates note is an additional chord tone with preceding note
chromatic	attribut es.mo d	Ele me nt	1, 0	Chromatic steps for transposing written to sounding pitch
circular- arrow	note.m od	Ele me nt	3, 0	Circular arrow direction. Values: clockwise, anticlockwise

clef	attribut es.mod	Ele me nt	1, 0	Clef
clef-octave- change	attribut es.mod	Ele me nt	1, 0	Octave difference (e.g. -1 for treble clef used by tenor voice)
coda	comm on.mod	Ele me nt	1, 0	Coda sign
coda	barline .mod	Attri but e	2, 0	Place to jump forwards from tocoda with same value
coda	directi on.mod	Attri but e	1, 0	Place to jump forwards from tocoda with same value
color	comm on.mod	Attri but e	1, 1	Color. Values: hexadecimal RGB triple or ARGB tuple in sRGB color space.
color	comm on.mod	Enti ty	1, 1	Collection of color attributes
container	contai ner.dtd	Ele me nt	2, 0	Document element for MusicXML container file
creator	identity .mod	Ele me nt	1, 0	Creator, similar to Dublin Core.
credit	score. mod	Ele me nt	1, 1	Credit that appears on first page of a score
credit-image	score. mod	Ele me nt	2, 0	Graphical image to be included in credit
credit-type	score. mod	Ele me nt	3, 0	Purpose behind a credit. Standard values: page number, title, subtitle, composer, arranger, lyricist, rights
credit-words	score. mod	Ele me nt	1, 1	Text portion of a credit, including formatting
cue	note.m od	Ele me nt	1, 0	Indicates a cue note, written but not played
dacapo	directi on.mod	Attri but e	1, 0	Go back to beginning of movement. Value: yes
dalsegno	directi on.mod	Attri but e	1, 0	Jump backwards to segno with same value
damp	directi on.mod	Ele me nt	1, 0	Harp damping
damp-all	directi on.mod	Ele me nt	1, 0	Harp damping for all strings
damper- pedal	directi on.mod	Attri but e	1, 0	Damper pedal control. Values: yes, no. Yes is depressed, no is released. Numeric values added in 2.0.
dashed-	comm	Enti	3,	Length of dashes and spaces in a dashed line, in tenths

formatting	on.mod	ty	0	
dashes	direction.mod	Element	1, 0	Dashed line, often used with <i>cresc.</i> , <i>dim.</i> , and other words
dash-length	common.mod	Attribute	3, 0	Length of dashes in a dashed line, in tenths
defaults	score.mod	Element	1, 1	Collect score-wide defaults, including layout and fonts
default-x	common.mod	Attribute	1, 0	Change origin of default horizontal position
default-y	common.mod	Attribute	1, 0	Change origin of default vertical position
degree	direction.mod	Element	1, 0	Add, alter, or subtract individual notes from a harmony
degree-alter	direction.mod	Element	1, 0	Alteration for degree, relative to dominant if adding: 1 for sharp, -1 for flat, etc.
degree-type	direction.mod	Element	1, 0	Type of degree alteration. Values: add, alter, subtract
degree-value	direction.mod	Element	1, 0	Degree of chord being affected: 1 for root, 3 for third, etc.
delayed-inverted-turn	note.mod	Element	3, 0	Inverted turn towards end of a note
delayed-turn	note.mod	Element	1, 0	Turn towards end of a note
departure	note.mod	Attribute	3, 0	How the ending of a compound ornament looks relative to the main part. Values: above, below
detached-legato	note.mod	Element	1, 0	Detached legato
diatonic	attributes.mod	Element	1, 0	Diatonic steps for transposing written to sounding pitch
dir	common.mod	Attribute	2, 0	Text directionality, similar to W3C Internationalization Tag Set. Values: ltr, rtl, lro, rlo
direction	barline.mod	Attribute	1, 0	Repeat direction. Values: backward, forward
direction	direction.mod	Element	1, 0	Musical indication not tied to a particular note
direction	note.mod	Attribute	1, 0	Direction of arpeggiation
direction-type	direction.mod	Element	1, 0	Type of direction; may be combined

	d	nt		
directive	comm on.mod	Attri but e	2, 0	If yes, use directive-style default-x position, aligned with start of time signature
directive	comm on.mod	Enti ty	2, 0	Collection of directive attributes
directive	attribut es.mod	Ele me nt	1, 0	Like directions, but at start of measure. Deprecated in Version 2.0; use directive attribute instead.
display- octave	note.m od	Ele me nt	1, 0	What octave would this be if pitched? (If percussion clef, treat as if treble clef)
display-step	note.m od	Ele me nt	1, 0	What step would this be if pitched? (If percussion clef, treat as if treble clef)
display-text	comm on.mod	Ele me nt	2, 0	Portion of formatted text used by part-name-display and similar elements
distance	layout. mod	Ele me nt	3, 0	Standard distance between notation elements, in tenths
divisions	attribut es.mod	Ele me nt	1, 0	How many divisions per quarter note for a duration
divisions	barline .mod	Attri but e	2, 0	New divisions per quarter note for use with segno and coda
divisions	directi on.mod	Attri but e	1, 0	New divisions per quarter note for use with segno and coda
document- attributes	comm on.mod	Enti ty	2, 0	Collection of attributes for MusicXML document elements
doit	note.m od	Ele me nt	1, 0	Indeterminate slide up from pitch
dot	note.m od	Ele me nt	1, 0	Dot of prolongation / augmentation dot
double	attribut es.mod	Ele me nt	1, 0	Transposition involves doubling one octave down
double- tongue	note.m od	Ele me nt	1, 0	Double tongue
down-bow	note.m od	Ele me nt	1, 0	Down bow
duration	note.m od	Ele me nt	1, 0	Intended nominal (not performance-specific) duration, in divisions
dynamics	comm on.mod	Ele me nt	1, 0	Dynamics marking
dynamics	directi on.mod	Attri but e	1, 0	Dynamics or MIDI velocity in percentage of default forte value (90 for MIDI)



dynamics	note.mod	Attribute	1,0	MIDI Note On velocity as percentage of default MIDI forte volume of 90
editorial	comment.mod	Entity	1,0	Footnote and level elements
editorial	note.mod	Attribute	1,0	Editorial accidental? Values: yes, no
editorial-voice	comment.mod	Entity	1,0	Footnote, level, and voice elements
effect	direction.mod	Element	3,0	Pictograms for sound effect percussion instruments. Values: anvil, auto horn, bird whistle, cannon, duck call, gun shot, klaxon horn, lions roar, police whistle, siren, slide whistle, thunder sheet, wind machine, wind whistle
element	identity.mod	Attribute	1,0	Element that has explicit indication of support or no support
element	link.mod	Attribute	2,0	Descendant of next sibling element to use for bookmark or link
elevation	direction.mod	Attribute	1,0	3D elevation. Deprecated in 2.0; use elevation element instead.
elevation	comment.mod	Element	2,0	3D elevation from -180 to 180; 0 is level, 90 directly above, -90 directly below
elision	note.mod	Element	1,0	Elision symbol for multiple syllables on single note. Text values added in 2.0.
enclosure	comment.mod	Attribute	1,1	Enclosure for non-rehearsal mark text. Values: rectangle, oval, none; enclosure-shape entity values added in 3.0
enclosure	direction.mod	Attribute	1,1	Enclosure for rehearsal mark. Values: square, circle, none; enclosure-shape entity values added in 3.0
enclosure-shape	comment.mod	Entity	3,0	Text enclosures using the enclosure attribute. Values: rectangle, square, oval, circle, bracket, triangle, diamond, none
encoder	identity.mod	Element	1,0	Who did the encoding
encoding	identity.mod	Element	1,0	Information about this MusicXML encoding
encoding-date	identity.mod	Element	1,0	When the encoding was done, as yyyy-mm-dd date
encoding-description	identity.mod	Element	1,0	Description of encoding
end-dynamics	note.mod	Attribute	1,0	MIDI Note Off velocity as percentage of default MIDI forte volume of 90
ending	barline.mod	Element	1,0	Multiple endings (e.g. first and second)

end-length	barline.mod	Attribute	1, 1	Ending jog size in tenths
end-length	direction.mod	Attribute	1, 0	Bracket jog size in tenths
end-line	note.mod	Element	1, 0	End of line for karaoke-style applications
end-paragraph	note.mod	Element	1, 0	End of paragraph for karaoke-style applications
ensemble	score.mod	Element	2, 0	Performance is intended for a section. Optional numeric value indicates size of section.
ensemble	sounds.dtd	Element	3, 0	Mapping of ensemble instrument sound between its MusicXML ID and an application or library ID
extend	note.mod	Element	1, 0	Word extension
eyeglasses	direction.mod	Element	1, 0	Eyeglasses mark
f	common.mod	Element	1, 0	f dynamic
falloff	note.mod	Element	1, 0	Indeterminate slide down from pitch
fan	note.mod	Attribute	1, 1	Fanned beams. Values: accel, rit, none
feature	direction.mod	Element	1, 0	Feature of this grouping
fermata	common.mod	Element	1, 0	Fermata symbol. Text values added in 2.0: normal, angled, square
ff	common.mod	Element	1, 0	ff dynamic
fff	common.mod	Element	1, 0	fff dynamic
ffff	common.mod	Element	1, 0	ffff dynamic
fffff	common.mod	Element	1, 0	fffff dynamic
ffffff	common.mod	Element	1, 0	ffffff dynamic
fifths	attributes.mod	Element	1, 0	Number of flats or sharps in traditional key signature
figure	note.mod	Element	1,	Figured bass figure

	od	me nt	0	
figured-bass	note.m od	Ele me nt	1, 0	Figured bass
figure- number	note.m od	Ele me nt	1, 0	Figured bass number
filled	note.m od	Attri but e	1, 0	Notehead filled? Values: yes, no.
fine	directi on.mo d	Attri but e	1, 0	Final note or rest value: duration if numeric, or "yes" if no duration change
fingering	comm on.mo d	Ele me nt	1, 0	Fingering indication, typically 1, 2, 3, 4, 5
fingernails	note.m od	Ele me nt	1, 0	Fingernails for harp notation
first-beat	comm on.mo d	Attri but e	1, 0	Percentage of direction for starting a bend
first-fret	directi on.mo d	Ele me nt	1, 0	Fret shown in top space of frame
font	comm on.mo d	Enti ty	1, 0	Collection of font attributes
font-family	comm on.mo d	Attri but e	1, 0	Font family, like CSS. Generic styles are music, serif, sans-serif.
font-size	comm on.mo d	Attri but e	1, 0	Font size, like CSS - either number or CSS size
font-style	comm on.mo d	Attri but e	1, 0	Font style, like CSS. Values: normal, italic
font-weight	comm on.mo d	Attri but e	1, 0	Font weight, like CSS. Values: normal, bold
footnote	comm on.mo d	Ele me nt	1, 0	Editorial footnote text
forward	note.m od	Ele me nt	1, 0	Move forward to coordinate multiple voices in one part, as in invisible rest
forward- repeat	directi on.mo d	Attri but e	1, 0	Forward repeat sign is implied. Value: yes
fp	comm on.mo d	Ele me nt	1, 0	fp dynamic
frame	directi on.mo d	Ele me nt	1, 0	Chord frame or fretboard diagram
frame-frets	directi on.mo	Ele me	1, 0	Size of frame in horizontal spaces

	d	nt		
frame-note	direction.mod	Element	1,0	Note to include in frame
frame-strings	direction.mod	Element	1,0	Size of frame in vertical lines
fret	common.mod	Element	1,0	Fret for tablature / chord symbol: open string is 0
full-note	note.mod	Entity	1,0	Common note elements between cue/grace and regular notes
full-path	container.dtd	Attribute	2,0	Path relative to root folder of the zip file
function	direction.mod	Element	1,0	Harmony function (e.g. I, V)
fz	common.mod	Element	1,0	fz dynamic
glass	direction.mod	Element	3,0	Pictograms for glass percussion instruments. Value: wind chimes
glissando	note.mod	Element	1,0	Rapidly moving from one pitch to another, sounding discrete half steps
grace	note.mod	Element	1,0	Indicates a grace note
group	score.mod	Element	1,0	Indicate purposes for part, e.g. score, parts, sound, data
group-abbreviation	score.mod	Element	1,0	Abbreviated version of group name
group-abbreviation-display	score.mod	Element	2,0	Formatted version of group-abbreviation
group-barline	score.mod	Element	1,0	Common barline for group? Values: yes, no, Mensurstrich
grouping	direction.mod	Element	1,0	Grouping for musical analysis
group-name	score.mod	Element	1,0	Full name of group
group-name-display	score.mod	Element	2,0	Formatted version of group-name
group-symbol	score.mod	Element	1,0	How group is indicated in score. Values: none, brace, line, bracket; square added in 3.0
group-time	score.mod	Element	2,0	Displayed time signature should stretch across all parts and staves in group
halign	comm	Attri	1,	Horizontal alignment. Values: left, center, right

	on.mod	but e	1	
halign	comm on.mod	Enti ty	1, 1	Collection of horizontal alignment attributes
hammer-on	note.mod	Ele me nt	1, 0	Hammer on (slur up the fretboard)
handbell	note.mod	Ele me nt	3, 0	Techniques used in handbell and handchime music. Values: damp, echo, gyro, hand martellato, mallet lift, mallet table, martellato, martellato lift, muted martellato, pluck lift, swing
harmonic	note.mod	Ele me nt	1, 0	Natural and artificial harmonics
harmony	directi on.mod	Ele me nt	1, 0	Harmony data, used for chord symbols or analysis
harmony-chord	directi on.mod	Enti ty	1, 0	Harmony chord information; may be stacked in a single harmony element
harp-pedals	directi on.mod	Ele me nt	1, 1	Harp pedal diagram
heel	note.mod	Ele me nt	1, 0	Heel indication for organ pedals
height	directi on.mod	Attri but e	1, 1	Frame height in tenths
hole	note.mod	Ele me nt	3, 0	Symbols used for woodwind/brass fingerings and other notations
hole-closed	note.mod	Ele me nt	3, 0	Is the hole closed, open, or half-open? Values: yes, no, half
hole-shape	note.mod	Ele me nt	3, 0	Shape of the hole symbol; circle by default
hole-type	note.mod	Ele me nt	3, 0	What the hole symbol represents in terms of instrument fingering or other techniques
humming	note.mod	Ele me nt	1, 0	Humming
id	comm on.mod	Attri but e	1, 0	IDREF to score-instrument for midi-instrument element; added to midi-device and play elements in 3.0
id	link.mod	Attri but e	1, 0	Unique ID for bookmark
id	note.mod	Attri but e	1, 0	IDREF to score-instrument for instrument element
id	score.mod	Attri but e	1, 0	Unique ID for score-part or score-instrument
id	score.mod	Attri but e	1, 0	IDREF to score-part for part element

		e		
id	sound.s.dtd	Attribute	3,0	Unique ID for sound
identification	identity.mod	Element	1,0	Basic score metadata
image	direction.mod	Element	2,0	Graphical image to be included in score
implicit	score.mod	Attribute	1,0	Show measure number never appear here? Values: yes, no
instrument	note.mod	Element	1,0	Which score-instrument to use for this note
instrument-abbreviation	score.mod	Element	1,0	Abbreviated version of instrument name
instrument-name	score.mod	Element	1,0	Full name of instrument
instruments	attributes.mod	Element	1,0	Number of instruments per part
instrument-sound	score.mod	Element	3,0	Default timbre for playback; the standard sounds listed in sounds.xml may be used
interchangeable	attributes.mod	Element	3,0	Second in a pair of dual time signatures
inversion	direction.mod	Element	1,0	Inversion for harmony: 0 for root position, 1 for first inversion, etc.
inverted-mordent	note.mod	Element	1,0	Inverted mordent
inverted-turn	note.mod	Element	2,0	Inverted turn
ipa	common.mod	Element	3,0	International Phonetic Alphabet sounds for vocal music
isolat1	common.mod	Entity	1,0	Character entities for ISO Latin-1
isolat2	common.mod	Entity	1,0	Character entities for ISO Latin-2
justify	common.mod	Attribute	1,0	Justification. Values: left, center, right. In direction.mod in 1.0.
justify	common.mod	Entity	1,1	Collection of justification attributes
key	attributes.mod	Element	1,0	Key signature

key-accidental	attribut es.mod	Ele me nt	3, 0	Accidental in non-traditional key signature. Values: same as for accidental
key-alter	attribut es.mod	Ele me nt	1, 0	Amount of alteration in non-traditional key signature
key-octave	attribut es.mod	Ele me nt	2, 0	Which octave does each element of a key signature appear?
key-step	attribut es.mod	Ele me nt	1, 0	Altered tone in non-traditional key signature
kind	directi on.mod	Ele me nt	1, 0	Type of chord. Many values from triads through 13ths.
last-beat	comm on.mod	Attri but e	1, 0	Percentage of trill where last beat falls, or percentage of direction for ending a bend
laughing	note.m od	Ele me nt	1, 0	Laughing
layout-tenths	comm on.mod	Enti ty	1, 1	Positioning unit in tenths of interline space, used for elements.
left-divider	layout. mod	Ele me nt	3, 0	System separator mark on the left side of the page
left-margin	layout. mod	Ele me nt	1, 1	Left margin in tenths
left-right	comm on.mod	Enti ty	2, 0	Does one element appear to the left or right of another element? Values: left, right
letter-spacing	comm on.mod	Attri but e	2, 0	Text spacing. Values: normal or a number representing ems to add
letter-spacing	comm on.mod	Enti ty	2, 0	Collection of text spacing attributes
level	comm on.mod	Ele me nt	1, 0	Indication of editorial level
level-display	comm on.mod	Enti ty	1, 1	Collection of parentheses, bracket, and size attributes
line	attribut es.mod	Attri but e	1, 0	Staff line numbered from bottom to top for staff-tuning
line	attribut es.mod	Ele me nt	1, 0	Staff line numbered from bottom to top for clefs
line	directi on.mod	Attri but e	1, 0	Use line instead of Ped / * signs? Values: yes, no
line-end	directi on.mod	Attri but e	1, 0	Bracket jog up, down, neither, or arrow? Values: up, down, both, arrow, none
line-height	comm	Attri	2,	Text leading. Values: normal or a number representing

	on.mod	but e	0	percent of font height to add
line-height	comm on.mod	Enti ty	2, 0	Collection of text leading attributes
line-shape	comm on.mod	Attri but e	1, 0	Values: straight, curved
line-shape	comm on.mod	Enti ty	1, 0	Collection of line shape attributes
line-through	comm on.mod	Attri but e	2, 0	Number of text strike-through lines. Values: 0, 1, 2, 3
line-type	comm on.mod	Attri but e	1, 0	Values: solid, dashed, dotted, wavy
line-type	comm on.mod	Enti ty	1, 0	Collection of line type attributes
line-width	layout.mod	Ele me nt	2, 0	Width of a line type in tenths
link	link.mod	Ele me nt	1, 0	Outgoing XLink
link-attributes	link.mod	Enti ty	1, 0	Collection of XLink attributes supported in MusicXML files
location	attribut es.mod	Attri but e	2, 0	Location of key signature cancellation relative to new key signature. Values: left, right; before-barline added in 3.0
location	barline .mod	Attri but e	1, 0	Barline location. Values: left, right, middle
location	directi on.mod	Attri but e	2, 0	Location of first-fret text. Values: left, right
location	note.m od	Attri but e	3, 0	Which portion of hole is filled in when hole-closed is half? Values: right, bottom, left, top
long	note.m od	Attri but e	1, 0	Longer mordent appearance? Values: yes, no.
lyric	note.m od	Ele me nt	1, 0	Text underlay for lyrics
lyric-font	score.mod	Ele me nt	1, 1	Lyric font
lyric-language	score.mod	Ele me nt	1, 1	Lyric language for a verse/section/chorus
make-time	note.m od	Attri but e	1, 0	Divisions to add in order to make time for grace note playback
measure	note.m od	Attri but e	3, 0	Complete measure rest? Values: yes, no



measure	score.mod	Element	1,0	Higher level element in timewise, lower level in partwise
measure-distance	layout.mod	Element	1,1	Horizontal distance from previous measure, as for coda
measure-layout	layout.mod	Element	1,1	Measure layout including distance
measure-numbering	direction.mod	Element	2,0	How measure numbers are displayed on this part. Values: none, measure, system
measure-repeat	attributes.mod	Element	1,0	Single or multiple measure repeat mark
measure-style	attributes.mod	Element	1,0	Special way to print
media-type	container.dtd	Attribute	2,0	Type of different top-level root files
member-of	direction.mod	Attribute	1,0	Indicate hierarchy within grouping
membrane	direction.mod	Element	3,0	Pictograms for membrane percussion instruments. Value: bass drum, bass drum on side, bongos, conga drum, goblet drum, military drum, snare drum, snare drum snares off, tambourine, tenor drum, timbales, tomtom
metal	direction.mod	Element	3,0	Pictograms for metal percussion instruments. Values: almglocken, bell, bell plate, brake drum, Chinese cymbal, cowbell, crash cymbals, crotale, cymbal tongs, domed gong, finger cymbals, flexatone, gong, hi-hat, high-hat cymbals, handbell, sistrum, sizzle cymbal, sleigh bells, suspended cymbal, tam tam, triangle, Vietnamese hat
metronome	direction.mod	Element	1,0	Standard metronome marks
metronome-beam	direction.mod	Element	2,0	Beaming of notes displayed in metric relationship
metronome-dot	direction.mod	Element	2,0	Dot displayed in metric relationship
metronome-note	direction.mod	Element	2,0	Note displayed in metric relationship
metronome-relation	direction.mod	Element	2,0	Relationship symbol between two sets of metronome-note elements. Values: equals
metronome-tuplet	direction.mod	Element	2,0	Tuplet displayed in metric relationship
metronome-type	direction.mod	Element	2,0	Type of note displayed in metric relationship. Values: same as type element
mf	command.mod	Element	1,0	mf dynamic

midi-bank	comm on.mod	Ele me nt	1, 0	MIDI bank from 1 to 16, 384
midi-channel	comm on.mod	Ele me nt	1, 0	MIDI channel from 1 to 16
midi-device	score. mod	Ele me nt	1, 0	MIDI DeviceName meta event
midi- instrument	comm on.mod	Ele me nt	1, 0	Specify MIDI instrument for score-instrument or sound
midi-name	comm on.mod	Ele me nt	1, 0	MIDI ProgramName meta event
midi-program	comm on.mod	Ele me nt	1, 0	MIDI program number from 1 to 128
midi- unpitched	comm on.mod	Ele me nt	1, 0	For unpitched instruments, MIDI note number from 1 to 128
millimeters	layout. mod	Ele me nt	1, 1	Number of millimeters in scaling ratio
miscellaneous	identity .mod	Ele me nt	1, 0	Collection of metadata not yet specified in the MusicXML format
miscellaneous-field	identity .mod	Ele me nt	1, 0	Individual metadata field
mode	attribut es.mod	Ele me nt	1, 0	Mode for key signature
mordent	note.m od	Ele me nt	1, 0	Mordent
movement- number	score. mod	Ele me nt	1, 0	Number of this movement
movement- title	score. mod	Ele me nt	1, 0	Title of this movement
mp	comm on.mod	Ele me nt	1, 0	mp dynamic
multiple-rest	attribut es.mod	Ele me nt	1, 0	Multimeasure rest
music-data	score. mod	Enti ty	1, 0	Basic musical data in a MusicXML score
music-font	score. mod	Ele me nt	1, 1	Music font
mute	comm on.mod	Ele me nt	3, 0	Muting for different instruments. Values: on, off, straight, cup, harmon-no-stem, harmon-stem, bucket, plunger, hat, solotone, practice, stop-mute, stop-hand, echo, palm
name	identity .mod	Attri but	1, 0	Name of metadata field

		e		
name	link.mod	Attribute	1,0	Name of bookmark or link
name	note.mod	Attribute	1,0	Type of lyric
name	score.mod	Attribute	1,1	Lyric type for lyric font or language
natural	note.mod	Element	1,0	Natural harmonic
new-page	direction.mod	Attribute	1,0	Force a page break? Values: yes, no
new-page	opus.dtd	Attribute	2,0	Is first page of this score different than last page of previous score? Values: yes, no
new-system	direction.mod	Attribute	1,0	Force a system break? Values: yes, no
niente	direction.mod	Attribute	3,0	Circle at point of the wedge? Values: yes, no
non-arpeggiate	note.mod	Element	1,0	Note is at top or bottom of a bracket indicating not to arpeggiate
non-controlling	score.mod	Attribute	1,0	Does this measure not synchronize with measures in other parts? Values: yes, no
normal-dot	common.mod	Element	1,0	Augmentation dot for normal-type value
normal-notes	common.mod	Element	1,0	How many notes usually occupy this time
normal-type	common.mod	Element	1,0	Type of note for normal-notes value
notations	note.mod	Element	1,0	Musical notations for a note
note	note.mod	Element	1,0	Individual note or rest
notehead	note.mod	Element	1,0	Notehead. Values: slash, triangle, diamond, square, cross, x, circle-x, normal, none; inverted triangle, arrow down, arrow up, slashed, back slashed, do, re, mi, fa, so, la, and ti values added in 1.1; cluster added in 2.0; fa up, circle dot, left triangle, and rectangle added in 3.0
notehead-text	note.mod	Element	3,0	Text displayed inside a notehead. Not needed for TAB or jianpu clefs.
note-size	layout.mod	Element	2,0	Percentage of regular note size to use for cue, grace, or large notes
number	attribut	Attribute	1,	Staff number for attributes. Added to key and time in 1.1;

	es.mod	but e	0	added to transpose in 3.0
number	barline.mod	Attribute	1, 0	What goes under ending line (e.g. "1", "1, 2")
number	common.mod	Attribute	1, 0	A number-level value for the wavy-line element
number	direction.mod	Attribute	1, 0	Used in many elements, usually as a number-level entity
number	layout.mod	Attribute	1, 1	Staff number for staff-layout.
number	note.mod	Attribute	1, 0	Used in many elements, usually as a number-level entity
number	score.mod	Attribute	1, 1	Lyric number for font or language
number	sounds.dtd	Attribute	3, 0	Size of ensemble
number-level	common.mod	Entity	1, 0	Distinguishes 6 concurrent objects. Values: 1, 2, 3, 4, 5, 6
number-of-lines	common.mod	Entity	2, 0	Number of lines in text decoration. Values: 0, 1, 2, 3
octave	note.mod	Element	1, 0	Octave number: Values: 0 through 9. Octave 4 is octave started by middle C.
octave-change	attributes.mod	Element	1, 0	Octaves for transposing written to sounding pitch
octave-shift	direction.mod	Element	1, 0	Octave shift up or down from sounding pitch positions
offset	direction.mod	Element	1, 0	Visual offset from current position, in divisions
open-string	note.mod	Element	1, 0	Open string
opus	opus.dtd	Element	1, 0	Document element for opus DTD
opus	score.mod	Element	1, 0	Link to a parent opus document
opus-link	opus.dtd	Element	1, 0	Link to another opus document
orientation	common.mod	Attribute	1, 0	Slur / ties overhand vs. underhand. Values: over, under
orientation	common.mod	Entity	1, 0	Collection of orientation attributes

	d			
ornaments	note.mod	Element	1,0	Collection of ornaments for a note
other-appearance	layout.mod	Element	2,0	Other appearance not yet specified in the MusicXML format
other-articulation	note.mod	Element	1,0	Other articulation not yet specified in the MusicXML format
other-direction	direction.mod	Element	1,0	Other direction not yet specified in the MusicXML format
other-dynamics	common.mod	Element	1,0	Text dynamic
other-notation	note.mod	Element	1,0	Notation not yet specified in the MusicXML format
other-ornament	note.mod	Element	1,0	Ornament not yet specified in the MusicXML format
other-percussion	direction.mod	Element	3,0	Percussion pictogram not yet specified in the MusicXML format
other-play	common.mod	Element	3,0	Playback technique not yet specified in the MusicXML format
other-technical	note.mod	Element	1,0	Technical indication not yet specified in the MusicXML format
overline	common.mod	Attribute	2,0	Number of text overlines. Values: 0, 1, 2, 3
p	common.mod	Element	1,0	p dynamic
page	score.mod	Attribute	2,0	Page number where credit should appear. First page is 1 and does not take page-number attributes into account
page-height	layout.mod	Element	1,1	Page height in tenths
page-layout	layout.mod	Element	1,1	Page layout including height, width, and margins
page-margins	layout.mod	Element	1,1	Page margins
page-number	direction.mod	Attribute	1,1	Sets the number of a new page
page-width	layout.mod	Element	1,1	Page width in tenths
pan	common.mod	Element	2,0	Horizontal pan from -180 to 180; 0 is straight ahead, -90 hard left, 90 hard right

pan	direction.mod	Attribute	1,0	Horizontal pan. Deprecated in 2.0; use pan element instead.
parentheses	common.mod	Attribute	1,0	Show level with parentheses? Values: yes, no
parentheses	direction.mod	Attribute	1,0	Is metronome mark parenthesized? Values: yes, no
parentheses	note.mod	Attribute	1,0	Notehead or figured bass in parentheses? Values: yes, no
parentheses-degrees	direction.mod	Attribute	1,1	All harmony degrees should be in parentheses
part	score.mod	Element	1,0	Higher level element in partwise, lower level in timewise
part-abbreviation	score.mod	Element	1,0	Abbreviated version of musical part name
part-abbreviation-display	common.mod	Element	2,0	Formatted version of part-abbreviation
part-group	score.mod	Element	1,0	Grouping of parts in a score
part-list	score.mod	Element	1,0	List of all the different musical parts in this movement
part-name	score.mod	Element	1,0	Full name of the musical part
part-name-display	common.mod	Element	2,0	Formatted version of part-name
part-symbol	attributes.mod	Element	2,0	Symbol for a multi-staff part. Values: none, brace, line, bracket; square added in 3.0
pedal	direction.mod	Element	1,0	Piano pedal marks
pedal-alter	direction.mod	Element	1,1	Pitch alteration for harp pedal tuning
pedal-step	direction.mod	Element	1,1	Pitch step for harp pedal tuning
pedal-tuning	direction.mod	Element	1,1	Harp pedal tuning for pedal diagram
percussion	direction.mod	Element	3,0	Percussion pictogram symbols
per-minute	direction.mod	Element	1,0	Per minute value, either number or text including numbers
pitch	note.mod	Element	1,	Pitch data: combination of step, alteration, and octave

	od	ment	0	
pitched	direction.mod	Element	3,0	Pictograms for pitched percussion instruments. Values: chimes, glockenspiel, mallet, marimba, tubular chimes, vibraphone, xylophone
pizzicato	direction.mod	Attribute	1,0	Within sound element, start pizzicato for all notes. Values: yes, no
pizzicato	note.mod	Attribute	1,0	Is this single note pizzicato? Values: yes, no
placement	comm.on.mod	Attribute	1,0	Placement relative to another symbol. Values: above, below
placement	comm.on.mod	Entity	1,0	Collection of placement attributes
play	comm.on.mod	Element	3,0	Playback techniques
plop	note.mod	Element	1,0	Indeterminate slide down to pitch
pluck	note.mod	Element	1,1	Plucking fingering on a fretted instrument. Text values added in 2.0. Typical values: p, i, m, a
plus-minus	direction.mod	Attribute	2,0	Use plus and minus instead of flat and sharp for harmony degree alteration. Values: yes, no
port	score.mod	Attribute	1,0	Unofficial MIDI 1.0 port / cable number from 1 to 16
position	comm.on.mod	Entity	1,0	Collection of positioning attributes
position	link.mod	Attribute	2,0	Position of descendant element for link or bookmark; first position is 1. Values: numbers
pp	comm.on.mod	Element	1,0	pp dynamic
ppp	comm.on.mod	Element	1,0	ppp dynamic
pppp	comm.on.mod	Element	1,0	pppp dynamic
ppppp	comm.on.mod	Element	1,0	ppppp dynamic
pppppp	comm.on.mod	Element	1,0	pppppp dynamic
pre-bend	note.mod	Element	1,0	Negative bend-alter indicates a pre-bend
prefix	note.mod	Element	1,0	Figured bass prefix

		nt		
primary	sound.s.dtd	Attribute	3,0	Primary choice when multiple mappings exist between MusicXML and application/library sound IDs
principal-voice	direction.mod	Element	3,0	Principal and secondary voices
print	direction.mod	Element	1,0	Print suggestions
print-dot	common.mod	Attribute	1,0	Should an augmentation dot be printed? Values: yes, no
print-frame	direction.mod	Attribute	1,0	Print frame for harmony? Values: yes, no
print-lyric	common.mod	Attribute	1,1	Should a lyric be printed? Values: yes, no
print-object	common.mod	Attribute	1,0	Should an object be printed? Values: yes, no
print-object	common.mod	Entity	1,1	Collection of print object attributes
printout	common.mod	Entity	1,0	Collection of print / no print attributes
print-spacing	common.mod	Attribute	1,0	Should spacing be left? Values: yes, no
print-spacing	common.mod	Entity	1,1	Collection of print spacing attributes
print-style	common.mod	Entity	1,1	Collection of position, font, and color entities
pull-off	note.mod	Element	1,0	Pull off (slur down the fretboard)
reference	common.mod	Attribute	1,1	Display-only editorial information? Values: yes, no
rehearsal	direction.mod	Element	1,0	Rehearsal mark
relation	identity.mod	Element	2,0	Related resource for the music that is encoded. similar to Dublin Core
relative-x	common.mod	Attribute	1,0	Change horizontal position relative to default
relative-y	common.mod	Attribute	1,0	Change vertical position relative to default
release	note.mod	Attribute	1,0	Offset from stopping time for playback, in divisions



release	note.mod	Element	1,0	Negative bend-alter indicates a release
repeat	barline.mod	Element	1,0	Repeat marks
repeater	note.mod	Attribute	1,0	Tremolo beams? Values: yes, no; deprecated in 3.0
rest	note.mod	Element	1,0	Rest or silence
rf	common.mod	Element	1,0	rf dynamic
rfz	common.mod	Element	1,0	rfz dynamic
right-divider	layout.mod	Element	3,0	System separator mark on the right side of the page
right-margin	layout.mod	Element	1,1	Right margin in tenths
rights	identity.mod	Element	1,0	Rights, similar to Dublin Core, including copyright and other notices
root	direction.mod	Element	1,0	Root of harmony in terms of pitch, not function
root-alter	direction.mod	Element	1,0	Pitch alteration for root
rootfile	container.dtd	Element	2,0	Top-level file in a MusicXML container
rootfiles	container.dtd	Element	2,0	Starting points for representations of MusicXML score
root-step	direction.mod	Element	1,0	Pitch step for root
rotation	common.mod	Attribute	2,0	Rotate text around halign/valign point. Values: numbers from -180 to 180
scaling	layout.mod	Element	1,1	Specify how many millimeters are equal to how many tenths
schleifer	note.mod	Element	1,0	German name to avoid conflict with slide element
scoop	note.mod	Element	1,0	Indeterminate slide up to pitch
scordatura	direction.mod	Element	1,1	Scordatura string tuning
score	opus.d	Element	1,	Link to a score within the opus

	td	me nt	0	
score-header	score. mod	Enti ty	1, 0	Basic score metadata plus the part-list
score-instrument	score. mod	Ele me nt	1, 0	How instruments are contained within parts
score-part	score. mod	Ele me nt	1, 0	How a part is contained in the score
score-partwise	score. mod	Ele me nt	1, 0	Document element for partwise.dtd
score-timewise	score. mod	Ele me nt	1, 0	Document element for timewise.dtd
second-beat	comm on.mo d	Attri but e	1, 0	Percentage of trill where second beat falls
segno	comm on.mo d	Ele me nt	1, 0	Segno sign
segno	barline .mod	Attri but e	2, 0	Place to jump backwards from dalsegno with same value
segno	directi on.mo d	Attri but e	1, 0	Place to jump backwards from dalsegno with same value
semi-pitched	comm on.mo d	Ele me nt	3, 0	Categories of indefinite pitch for percussion. Values: high, medium-high, medium, medium-low, low, very-low
senza-misura	attribut es.mo d	Ele me nt	1, 0	Indicates no time signature; optional element content indicating symbol added in 3.0
separator	attribut es.mo d	Attri but e	3, 0	Arrangement between beats and beat-type values in a time signature. Values: none, horizontal, diagonal, vertical, adjacent
sf	comm on.mo d	Ele me nt	1, 0	sf dynamic
sffz	comm on.mo d	Ele me nt	1, 0	sffz dynamic
sfp	comm on.mo d	Ele me nt	1, 0	sfp dynamic
sfpp	comm on.mo d	Ele me nt	1, 0	sfpp dynamic
sfz	comm on.mo d	Ele me nt	1, 0	sfz dynamic
shake	note.m od	Ele me nt	1, 0	Shake
show-frets	attribut es.mo d	Attri but e	1, 0	How to display tablature frets. Values: letters, numbers

show-number	direction.mod	Attribute	2, 0	Tuplet number(s) displayed in metric relationship? Values: actual, both, none
show-number	note.mod	Attribute	1, 0	Tuplet number(s) displayed? Values: actual, both, none
show-type	note.mod	Attribute	1, 0	Tuplet type(s) displayed? Values: actual, both, none.
sign	attributes.mod	Element	1, 0	Clef sign. Values: G, F, C, percussion, TAB, none; jianpu added in 3.0
sign	direction.mod	Attribute	3, 0	Ped and * signs are used for pedal? Values: yes, no
size	attributes.mod	Attribute	2, 0	Additional clef symbol size. Values: full, cue, large
size	common.mod	Attribute	1, 0	Show level with size? Values: full, cue, large
size	direction.mod	Attribute	1, 0	Octave shift size: 8 for one octave, 15 for two octaves
size	note.mod	Attribute	1, 0	Note size. Values: full, cue; large added in 1.1
slash	attributes.mod	Element	1, 0	Slash notation
slash	note.mod	Attribute	1, 0	Grace note slashed? Values: yes, no
slash	note.mod	Attribute	3, 0	Turn slashed by a vertical line? Values: yes, no
slash-dot	attributes.mod	Element	2, 0	Definition of beat for beat-repeat and slash
slashes	attributes.mod	Attribute	1, 0	Number of slashes in repeat sign
slash-type	attributes.mod	Element	2, 0	Definition of beat for beat-repeat and slash. Values: same as type element
slide	note.mod	Element	1, 0	Rapidly moving from one pitch to another continuously
slur	note.mod	Element	1, 0	Note is at beginning, end, or inflection point of slur
snap-pizzicato	note.mod	Element	1, 0	Snap pizzicato
soft-pedal	direction.mod	Attribute	1, 0	Soft pedal control. Values: yes, no. Yes is depressed, no is released. Numeric values added in 2.0.
software	identity	Element	1,	Software used for encoding

	.mod	me nt	0	
solo	score. mod	Ele me nt	2, 0	Performance is intended by a solo instrument
solo	sound s.dtd	Ele me nt	3, 0	Mapping of solo instrument sound between its MusicXML ID and an application or library ID
sostenuto- pedal	directi on.mo d	Attri but e	1, 0	Sostenuto pedal control. Values: yes, no. Yes is depressed, no is released. Numeric values added in 2.0.
sound	directi on.mo d	Attri but e	2, 0	Does offset affect playback as well as display? Values: yes, no
sound	directi on.mo d	Ele me nt	1, 0	Playback suggestions
sound	sound s.dtd	Ele me nt	3, 0	MusicXML standard sound
sounding- pitch	note.m od	Ele me nt	1, 0	Sounding pitch for harmonic
sounds	sound s.dtd	Ele me nt	3, 0	Document element for MusicXML instrument sounds
source	directi on.mo d	Attri but e	2, 0	URL for image file to be included in score direction
source	score. mod	Attri but e	2, 0	URL for image file to be included in score credit
source	identity .mod	Ele me nt	1, 0	Source for music that is encoded, similar to Dublin Core
space-length	comm on.mo d	Attri but e	3, 0	Length of spaces in a dashed line, in tenths
spiccato	note.m od	Ele me nt	1, 0	Spiccato / stroke articulation
spread	directi on.mo d	Attri but e	1, 0	Wedge spread in tenths
staccatissim o	note.m od	Ele me nt	1, 0	Staccatissimo
staccato	note.m od	Ele me nt	1, 0	Staccato
stack- degrees	directi on.mo d	Attri but e	1, 1	Stack harmony degree elements above each other
staff	comm on.mo d	Ele me nt	1, 0	Staff assignment for multi-staff parts
staff-details	attribut es.mo	Ele me	1, 0	Details for different types of staves

	d	nt		
staff-distance	layout.mod	Element	1, 1	Staff distance within a system
staff-layout	layout.mod	Element	1, 1	Staff distances for this part
staff-lines	attributes.mod	Element	1, 0	Number of lines in this staff
staff-size	attributes.mod	Element	1, 1	Size of staff space, in percent of default scaling.
staff-spacing	direction.mod	Attribute	1, 0	Spacing between part staves, in tenths; deprecated in 1.1
staff-tuning	attributes.mod	Element	1, 0	Staff tuning for tablature
staff-type	attributes.mod	Element	1, 0	Values: ossia, cue, editorial, regular, alternate
start-note	common.mod	Attribute	1, 0	Starting note for playback. Values: upper, main, below
start-stop	common.mod	Entity	1, 0	Typically used for type attribute. Values: start, stop
start-stop-continue	common.mod	Entity	1, 0	Values: start, stop, continue
start-stop-single	common.mod	Entity	2, 0	For notations that can be multi- or single-note. Values: start, stop, single
staves	attributes.mod	Element	1, 0	Number of staves per part
steal-time-following	note.mod	Attribute	1, 0	Percentage of time to steal from following note for grace note playback
steal-time-previous	note.mod	Attribute	1, 0	Percentage of time to steal from previous note for grace note playback
stem	note.mod	Element	1, 0	Stem direction. Values: down, up, none, double.
step	note.mod	Element	1, 0	Diatonic pitch step. Values: A, B, C, D, E, F, G
stick	direction.mod	Element	3, 0	Pictograms where the material in the stick is included
stick-location	direction.mod	Element	3, 0	Pictograms for location of sticks on cymbals and other instruments. Values: center, rim, cymbal bell, cymbal edge
stick-material	direction.mod	Element	3, 0	Material for stick pictograms. Values: soft, medium, hard, shaded, x

stick-type	direction.mod	Element	3,0	Type of stick for material-specific pictograms. Values: bass drum, double bass drum, timpani, xylophone, yarn
stopped	note.mod	Element	1,0	Stopped
stress	note.mod	Element	2,0	Stress articulation
string	direction.mod	Attribute	1,1	String in scordatura accord element: highest string is 1
string	common.mod	Element	1,0	String for tablature / chord symbol: highest string is 1
string-mute	direction.mod	Element	3,0	String mute on / string mute off symbols
strong-accent	note.mod	Element	1,0	Vertical accent
substitution	common.mod	Attribute	1,0	Substitute fingering for middle of note? Values: yes, no
substitution	note.mod	Attribute	1,1	Substitute heel or toe in middle of note? Values: yes, no
suffix	note.mod	Element	1,0	Figured bass suffix
supports	identity.mod	Element	1,0	Are certain MusicXML elements supported in this encoding? (e.g. stem, beam)
syllabic	note.mod	Element	1,0	Syllable type for hyphenation. Values: single, begin, end, middle
symbol	attributes.mod	Attribute	1,0	Time signature symbol. Values: common, cut, single-number, normal; note and dotted-note added in 3.0
symbol	direction.mod	Attribute	3,0	Symbol to use to specify a degree. Values: major, minor, augmented, diminished, half-diminished
symbol	direction.mod	Attribute	3,0	Symbol for principal voice. Values: Hauptstimme, Nebenstimme, plain, none
symbol-size	common.mod	Entity	1,0	Symbol size, such as for notes. Values: full, cue; large added in 1.1.
system-distance	layout.mod	Element	1,1	Vertical distance from previous system
system-dividers	layout.mod	Element	3,0	System separator marks on left and right sides of the page
system-layout	layout.mod	Element	1,1	Left and right margins plus vertical distance
system-	layout.	Element	1,	System margins relative to page margins, in tenths

margins	mod	me nt	1	
tap	note.m od	Ele me nt	1, 0	Tap on fretboard
technical	note.m od	Ele me nt	1, 0	Performance indications for individual instruments
tempo	directi on.mo d	Attri but e	1, 0	Tempo in quarter notes per minute
tenths	comm on.mo d	Enti ty	1, 0	Positioning unit in tenths of interline space, used for attributes.
tenths	layout. mod	Ele me nt	1, 1	Number of tenths in scaling ratio
tenuto	note.m od	Ele me nt	1, 0	Tenuto
text	directi on.mo d	Attri but e	1, 1	How harmony kind should be displayed if not using symbols
text	directi on.mo d	Attri but e	2, 0	How other parts of harmony elements should be displayed if not using symbols
text	note.m od	Ele me nt	1, 0	Lyric text
text- decoration	comm on.mo d	Enti ty	2, 0	Collection of text decoration attributes
text-direction	comm on.mo d	Enti ty	2, 0	Collection of text direction attributes, similar to W3C Internationalization Tag Set
text- formatting	comm on.mo d	Enti ty	2, 0	Common formatting attributes for text elements
text-rotation	comm on.mo d	Enti ty	2, 0	Collection of text rotation attributes
text-x	barline .mod	Attri but e	2, 0	Offset for start of ending text relative to start of the ending line
text-y	barline .mod	Attri but e	2, 0	Offset for baseline of ending text relative to start of the ending line
thumb- position	note.m od	Ele me nt	1, 0	Thumb position
tie	note.m od	Ele me nt	1, 0	Note is part of a tie for performance purposes
tied	note.m od	Ele me nt	1, 0	Note appears graphically as part of a tie
time	attribut es.mo	Ele me	1, 0	Time signature

	d	nt		
time-modification	note.mod	Element	1,0	Time modification for tuplets and other durational changes
time-only	direction.mod	Attribute	1,0	Which time in repeated section to apply the sound element
time-only	note.mod	Attribute	1,0	Which time to play a note or tie in a repeated section. Added to tie element in 3.0.
time-relation	attributes.mod	Element	3,0	Symbol that represents interchangeable aspect of dual time signatures. Values: parentheses, bracket, equals, slash, space, hyphen.
times	barline.mod	Attribute	1,0	Number of times repeated section is played
time-separator	attributes.mod	Entity	3,0	Arrangement between beats and beat-type values in a time signature using the separator attribute
time-symbol	attributes.mod	Entity	3,0	Time signature display using the symbol attribute
timpani	direction.mod	Element	3,0	Timpani pictogram
title	opus.dtd	Element	1,0	Title of this opus
tocoda	direction.mod	Attribute	1,0	Jump forwards to coda with same value
toe	note.mod	Element	1,0	Toe indication for organ pedals
top-bottom	common.mod	Entity	1,0	For vertical shapes. Values: top, bottom
top-margin	layout.mod	Element	1,1	Top margin in tenths
top-staff	attributes.mod	Attribute	2,0	Top staff for part-symbol element
top-system-distance	layout.mod	Element	1,1	Vertical distance from top page margin
touching-pitch	note.mod	Element	1,0	Pitch where string is touched for artificial harmonic
transpose	attributes.mod	Element	1,0	Transposition from written to sounding pitch
tremolo	note.mod	Element	1,1	Single- or double-note tremolo ornament. Double-note added in 2.0.
trill-mark	note.mod	Element	1,0	Trill mark



trill-sound	comm on.mod	Enti ty	1, 0	Collection of trill playback attributes
trill-step	comm on.mod	Attri but e	1, 0	Size of trill playback step. Values: whole, half, unison
triple-tongue	note.m od	Ele me nt	1, 0	Triple tongue
tuning-alter	comm on.mod	Ele me nt	1, 0	Pitch alteration for a staff line. In attributes.mod in 1.0.
tuning- octave	comm on.mod	Ele me nt	1, 0	Octave for a staff line. In attributes.mod in 1.0.
tuning-step	comm on.mod	Ele me nt	1, 0	Pitch step for a staff line. In attributes.mod in 1.0.
tuplet	note.m od	Ele me nt	1, 0	Note is at beginning or end of a tuplet
tuplet-actual	note.m od	Ele me nt	1, 0	Indicates notes actually played
tuplet-dot	note.m od	Ele me nt	1, 0	Augmentation dot for tuplet-type value
tuplet-normal	note.m od	Ele me nt	1, 0	Indicates notes usually occupying this time
tuplet- number	note.m od	Ele me nt	1, 0	Number of tuplet notes
tuplet-type	note.m od	Ele me nt	1, 0	Type of tuplet notes
turn	note.m od	Ele me nt	1, 0	Turn
two-note-turn	comm on.mod	Attri but e	1, 0	Size of turn at end of trill. Values: whole, half, none
type	attribut es.mod	Attri but e	1, 0	Measure style type. Values: start, stop
type	barline .mod	Attri but e	1, 0	Ending type. Values: start, stop, discontinue
type	comm on.mod	Attri but e	1, 0	Fermata type. Values: upright, inverted
type	comm on.mod	Attri but e	3, 0	Playback technique type for other-play element
type	directi on.mod	Attri but e	1, 0	Wedge type. Values: crescendo, diminuendo, stop
type	directi	Attri	2,	Metronome tuplet type. Values: start, stop

	on.mod	but e	0	
type	directi on.mod	Attri but e	1, 0	Pedal type. Values: start, stop; change added in 1.1
type	directi on.mod	Attri but e	1, 0	Used in many elements, usually as start-stop, start-stop-continue, or start-stop-single entity
type	directi on.mod	Attri but e	1, 0	Octave shift type. Values: up, down, stop
type	directi on.mod	Attri but e	3, 0	String mute type. Values: on, off
type	directi on.mod	Attri but e	1, 0	Harmony type. Values: explicit, implied, alternate
type	directi on.mod	Attri but e	2, 0	Image MIME type. Typical values: application/postscript, image/gif, image/jpeg, image/png, image/tiff
type	identity .mod	Attri but e	1, 0	Creator type. Standard values added in 2.0: composer, lyricist, arranger
type	identity .mod	Attri but e	1, 0	Rights type. Standard values added in 2.0: music, words, arrangement
type	identity .mod	Attri but e	2, 0	Relation type. Standard values: music, words, arrangement
type	layout. mod	Attri but e	1, 1	Page numbers where layout applies. Values: odd, even, both
type	layout. mod	Attri but e	2, 0	Line width type. Values: beam, bracket, dashes, enclosure, ending, extend, heavy barline, leger, light barline, octave shift, pedal, slur middle, slur tip, staff, stem, tie middle, tie tip, tuplet bracket, wedge
type	layout. mod	Attri but e	2, 0	Note size type. Values: cue, grace, large
type	layout. mod	Attri but e	3, 0	Distance type. Values: hyphen, beam
type	note.m od	Ele me nt	1, 0	Values: 256th, 128th, 64th, 32nd, 16th, eighth, quarter, half, whole, breve, long; 1024th, 512th, and maxima added in 3.0
type	note.m od	Attri but e	1, 0	Used in many elements, usually as start-stop, start-stop-continue, or start-stop-single entity
type	note.m od	Attri but e	1, 0	Strong accent direction. Values: up, down
type	note.m od	Attri but e	1, 0	Non-arpeggiate type. Values: top, bottom
type	score. mod	Attri but e	1, 0	Part-group type. Values: start, stop
type	score.	Attri	2,	Credit image MIME type. Typical values:

	mod	but e	0	application/postscript, image/gif, image/jpeg, image/png, image/tiff
underline	comm on.mod	Attri but e	2, 0	Number of text underlines. Values: 0, 1, 2, 3
unpitched	note.m od	Ele me nt	1, 0	Musical notes notated on the staff but lacking definite pitch
unplayed	directi on.mod	Attri but e	3, 0	What to display above an unplayed string in a frame
unstress	note.m od	Ele me nt	2, 0	Unstress articulation
up-bow	note.m od	Ele me nt	1, 0	Up bow
up-down	comm on.mod	Enti ty	1, 0	For arrow direction. Values: up, down
use-dots	attribut es.mod	Attri but e	1, 0	Use dots with beat-repeat or slash
use-stems	attribut es.mod	Attri but e	1, 0	Use stems with slash notation
use-symbols	attribut es.mod	Attri but e	1, 0	Use 1-bar / 2-bar / 4-bar symbols for multimeasure rests
use-symbols	directi on.mod	Attri but e	1, 1	Use harmony symbols rather than letters and numbers
valign	comm on.mod	Attri but e	1, 1	Vertical alignment for text or images
valign	comm on.mod	Enti ty	1, 1	Vertical alignment for text. Values: top, middle, bottom, baseline
valign-image	comm on.mod	Enti ty	2, 0	Vertical alignment for images. Values: top, middle, bottom
value	identity .mod	Attri but e	2, 0	Which value of an element or attribute is supported or not
version	comm on.mod	Attri but e	1, 1	MusicXML format version. Added to opus element in 2.0.
vertical-turn	note.m od	Ele me nt	3, 0	Turn with shape arranged vertically from upper left to lower right
virtual-instrument	score. mod	Ele me nt	3, 0	Specific virtual instrument used for an instrument sound
virtual-library	score. mod	Ele me nt	3, 0	Virtual instrument library name
virtual-name	score. mod	Ele me nt	3, 0	Library-specific name for virtual instrument

		nt		
voice	comm on.mod	Ele me nt	1, 0	Distinguish musical voice
volume	comm on.mod	Ele me nt	2, 0	MIDI channel volume as percentage of maximum. Values: 0 to 100
wavy-line	comm on.mod	Ele me nt	1, 0	Wavy line symbol
wedge	directi on.mod	Ele me nt	1, 0	Crescendo / diminuendo wedges
width	directi on.mod	Attri but e	1, 1	Frame width in tenths
width	score. mod	Attri but e	1, 1	Measure width in tenths
winged	barline .mod	Attri but e	3, 0	Does repeat have winged extensions? Values: none, straight, curved, double-straight, double-curved
with-bar	note.m od	Ele me nt	1, 0	Bend is done at bridge with whammy or vibrato bar
wood	directi on.mod	Ele me nt	3, 0	Pictograms for wood percussion instruments. Values: board clapper, cabasa, castanets, claves, guiro, log drum, maraca, maracas, ratchet, sandpaper blocks, slit drum, temple block, vibraslap, wood block
word-font	score. mod	Ele me nt	1, 1	Font for words and other non-lyric text
words	directi on.mod	Ele me nt	1, 0	Text direction
work	score. mod	Ele me nt	1, 0	Information regarding the overall work
work-number	score. mod	Ele me nt	1, 0	Number of work
work-title	score. mod	Ele me nt	1, 0	Title of work
xlink.ns	link.mo d	Enti ty	1, 0	XLink namespace
xlink:actuate	link.mo d	Attri but e	1, 0	XLink 1.0 actuate. Values: onRequest, onLoad, other, none
xlink:href	link.mo d	Attri but e	1, 0	XLink 1.0 href
xlink:role	link.mo d	Attri but e	1, 0	XLink 1.0 role
xlink:show	link.mo d	Attri but e	1, 0	XLink 1.0 show. Values: new, replace, embed, other, none

xlink:title	link.mod	Attribute	1,0	XLink 1.0 title
xlink:type	link.mod	Attribute	1,0	XLink 1.0 type. Value: simple
xml:lang	attributes.mod	Attribute	1,0	Language for directives
xml:lang	common.mod	Attribute	1,0	Language for words, credit-words, and other text elements
xml:lang	direction.mod	Attribute	1,1	Language for rehearsal marks
xml:lang	note.mod	Attribute	1,0	Language for lyric text
xml:lang	score.mod	Attribute	1,1	Default lyric language
xml:space	common.mod	Attribute	3,0	Whitespace handling for formatted text
xmlns:xlink	link.mod	Attribute	1,0	XLink 1.0 namespace. Value: %xlink.ns;
yes-no	common.mod	Entity	1,0	For boolean-like attributes. Values: yes, no
yes-no-number	common.mod	Entity	2,0	For attributes that can be boolean or numeric. Values: yes, no, and numbers
yyyy-mm-dd	common.mod	Entity	1,0	Calendar dates in ISO 8601 format