



UNIVERSIDADE DE SÃO PAULO
ECA - ESCOLA DE COMUNICAÇÃO E ARTES.
DEPARTAMENTO DE MÚSICA

ROBSON DIAS

ANALYSIS PROJECT - ANÁLISE DE MÚSICA SIMBÓLICA POR PADRÕES.
(Versão Corrigida (versão original disponível na Biblioteca da ECA/USP))

TESE DE DOUTORADO

SÃO PAULO
2020

ROBSON DIAS

**ANALYSIS PROJECT - ANÁLISE DE MÚSICA SIMBÓLICA POR PADRÕES.
(Versão Corrigida (versão original disponível na Biblioteca da ECA/USP))**

Tese de doutorado apresentado ao Departamento de Música da Escola de Comunicação e Artes da Universidade de São Paulo como parte dos requisitos necessários à obtenção do título de doutor em musicologia.

Orientador: Diósnio Machado Neto

**SÃO PAULO
2020**

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo na Publicação
Serviço de Biblioteca e Documentação
Escola de Comunicações e Artes da Universidade de São Paulo
Dados inseridos pelo(a) autor(a)

DIAS, ROBSON
Analysis Project - Análise de Música Simbólica por Padrões
/ ROBSON DIAS; orientador, Diósnio Machado Neto. -- São Paulo,
2020.
258 p.: il.

Tese (Doutorado) - Programa de Pós-Graduação em Música
Escola de Comunicações e Artes / Universidade de São Paulo.
Bibliografia
Versão corrigida

1. Análise Musical Computacional 2. Música Simbólica 3.
Padrões e Esquemas Musicais 4. Figuras de Retórica 5.
Programação em Python I. Machado Neto, Diósnio II. Título.
CDD 21.ed. - 780

FOLHA DE APROVAÇÃO

DIAS, Robson. *Analysis Project - Análise de Música Simbólica por Padrões*. 237 p. Tese (Doutorado) - Programa de Pós-Graduação em Música Escola de Comunicações e Artes / Universidade de São Paulo, 2020.

Aprovado em: 19/06/2020

Professor (a) Doutor (a): Diósnió Machado Neto
Instituição: EACH - Universidade de São Paulo (USP)
Julgamento: _____
Assinatura: _____

Professor (a) Doutor (a): Regis Rossi Alves Faria
Instituição: EACH – Universidade de São Paulo (USP)
Julgamento: _____
Assinatura: _____

Professor (a) Doutor (a): Renato Fabbri
Instituição: Externo
Julgamento: _____
Assinatura: _____

Professor (a) Doutor (a): Anselmo Guerra de Almeida
Instituição: Universidade Federal de Goiás (UFGoiás) – Externo
Julgamento: _____
Assinatura: _____

Professor (a) Doutor (a): Maria Cristina Ferreira de Oliveira
Instituição: ICMC - Universidade de São Paulo (USP)
Julgamento: _____
Assinatura: _____

Agradecimentos

Ao meu orientador que mesmo reconhecendo minhas limitações acreditou em mim e foi muito generoso ao me conduzir na difícil realização desta tarefa.

A minha mãe, uma das figuras mais importantes na construção do meu caráter. Obrigado por me cobrar tanto.

Theo lembre-se: You and me, together forever.

Waniaceli Dias, companheira e amiga, obrigado por me acompanhar mesmo me minas “loucuras”.

Ronaldo Novaes, sem você esta tese não teria acontecido. Obrigado por acolher em sua casa. Aquela “economia” com o hotel viabilizou meus estudos. Amigos e irmãos para sempre.

Todos que direta ou indiretamente fazem parte da construção do que sou: agradeço a fidelidade e o companheirismo.

Quem não sabe o que procura não
entende o que encontra” (Hans Selye)

RESUMO

O presente trabalho realizou o estudo de música simbólica tendo o como objetivo a de busca padrões e esquemas pré-determinados em um grupo extenso de obras. Foram escolhidas algumas figuras retóricas musicais e esquemas tratados na obra *Music in The Galant Style* de Gjerdingen(2007). Para a realização da pesquisa fez-se necessário o desenvolvimento de algoritmos de buscas bem como um método capaz de organizar as fases do trabalho analítico que foi realizado em um ambiente computacional. O Analysis Project (nome dado a ferramenta de suporte a este trabalho) realizou todo o processo. Verificamos se o quantitativo de repetições de determinado padrão encontradas em um conjunto de obras nos facilitaria a compreensão sobre os processos de estilo e recepção daquele contexto. Esta era nossa hipótese inicial. Os resultados nos mostraram que o processamento computacional inerentes ao modelo de representação do conteúdo de interesse possui limitações. Realizamos uma análise baseada na identificação de padrões sintáticos que facilmente poderiam ser reconhecidos no texto musical contudo estes padrões não capturam a semântica do texto.

Palavras-chave: Musicologia; Análise musical computacional; Tecnologia; Python.

ABSTRACT

The present work carried out the study of symbolic music with the aim of searching for pre-determined patterns and schemes in an extensive group of works. Some rhetorical musical figures and schemes dealt with in the work *Music in The Galant Style* by Gjerdingen (2007) were chosen. In order to carry out the research, it was necessary to develop search algorithms as well as a method capable of organizing the phases of the analytical work that was carried out in a computational environment. The Analysis Project (the name given to the support tool for this work) performed the entire process. We verified whether the number of repetitions of a given pattern found in a set of works would facilitate our understanding of the style and reception processes in that context. This was our initial hypothesis. The results showed that the computational processing inherent to the model of representation of the content of interest has limitations. We performed an analysis based on the identification of syntactic patterns that could easily be recognized in the musical text, however these patterns do not capture the semantics of the text.

Keywords: Musicology; Computational musical analysis; Technology; Software, Python.

LISTA DE ILUSTRAÇÕES

Figura 1 – Analista X Obra Musical	23
Figura 2 - Analista X Computador X obra	23
Figura 3 - Página Inicial do MelodicMatch.....	26
Figura 4 - Visualização do site oficial do Music21	27
Figura 5 - Nota impressa	28
Figura 6 - Music21 - Pequena Melodia.....	28
Figura 7 - Visualização do site oficial do ELVIS	29
Figura 8 - Visualização do site oficial do MusicXML Analyser.....	30
Figura 9 - – Frequências em um grupo de obras no formato MusicXML.....	31
Figura 10 - Interface para inserir padrões para pesquisa nos arquivos MusicXML. ...	32
Figura 11 - O padrão pesquisado aparece em vermelho.	33
Figura 12 - Modelo de Metodologia Analítica	37
Figura 13 - Metodologia fase 01	38
Figura 14 - Cadastro de Projetos	38
Figura 15 - Cadastro de projetos - obras musicais.....	39
Figura 16 - Lista das estâncias de um projeto.....	40
Figura 17 - Cadastro de instâncias de projetos.....	40
Figura 18 - Representação visual da primeira fase de pesquisa.....	41
Figura 19 - Metodologia fase 02.....	42
Figura 20 - Lista parcial dos módulos e associações com as instâncias.....	43
Figura 21 - Cadastro de módulos.....	43
Figura 22 - Cadastro de categorias	44
Figura 23 - Categoria com a opção de padrões do Gjerdingen.....	45
Figura 24 - Cadastro de categorias com a opção de padrão do usuário.....	46
Figura 25 - Cadastro de categorias com opção de padrão do usuário 02.....	46
Figura 26 - Representação visual da primeira e segunda fase de pesquisa	47
Figura 27 - Metodologia fase 03.....	48
Figura 28 - Lista dos projetos a pesquisar	49
Figura 29 - Realizar a pesquisa.....	49
Figura 30 - Visualização em forma de dendograma	50
Figura 31 - Exemplo de verificação de registro realizada.....	51
Figura 32 - Listagem dos projetos a serem verificados	51
Figura 33 - Verificação dos resultados	52
Figura 34 - Representação visual da metodologia proposta pelo Analysis Project ...	53
Figura 35 - Metodologia fase 04.....	54
Figura 36 - Lista de projetos concluídos.....	54
Figura 37 - Extrato parcial de um modelo de conclusão de projeto.....	55
Figura 38 - Exemplo de Romanesca	57
Figura 39 - Um esquema da Romanesca com saltos no baixo	59
Figura 40 - Exemplo de Cláusula	60
Figura 41 - Padrão Dó - Ré - Mi na voz aguda.....	60
Figura 42 - Exemplo do padrão Dó - Ré - Mi.....	61
Figura 43 - Exemplo do esquema Prinner.....	61
Figura 44 - Locatelli, Opus 6, no. 11, mvt. i, adagio, m. 16 (Amsterdam, 1737).....	62
Figura 45 - Durante, Studio no. i, mvt. i, m. i (ca. 1747)	63
Figura 46 - Tritto, a Fonte bass (ca. 17905)	63
Figura 47 - Daube's "three chords" (1756) set as a cadence	64

Figura 48 - Riepel's prototype of the Ponte melody with a likely bass.....	64
Figura 49 - Gossec, Missa pro defunctis, mvt. 15, [Andante], m. 26 (Paris, 1760)....	65
Figura 50 - Classificação das Figuras de Retórica	66
Figura 51 - MusicXML - Exemplo 01	71
Figura 52 - Tela Inicial do Analysis Project.....	79
Figura 53 - Tela de acesso ao painel de controle.....	80
Figura 54 - Visão geral do painel de controle	81
Figura 55 - Opções de cadastro de figuras de retórica	82
Figura 56 - Organização das Figuras de Retórica no Analysis Project	82
Figura 57 - Recursos de inserção e filtros de busca	83
Figura 58 - – Exemplo de Cadastro de Figuras de Retórica	84
Figura 59 - Tratamento de Inclusão de Registros	85
Figura 60 - Exemplo de Inserção de Textos e Mídia	85
Figura 61 - Cadastro de Padrões e Esquemas propostos por Gjerdingen	86
Figura 62 - Listagem dos Padrões Cadastrados	86
Figura 63 - Opções de Cadastro para os Padrões de Gjerdingen.....	87
Figura 64 - Listagem parcial de instrumentos cadastrados	88
Figura 65 - Cadastro de Instrumentos.....	89
Figura 66 - Listagem das Famílias de Instrumentos Cadastradas	89
Figura 67 - Cadastro de Famílias de Instrumentos	90
Figura 68 - Opções de cadastro de obras	90
Figura 69 - Lista parcial de autores cadastrados.....	91
Figura 70 - Cadastro de autores.....	91
Figura 71 - Lista parcial de obras cadastradas.....	92
Figura 72 - Cadastro de obras.....	93
Figura 73 - Opções de Cadastro de estruturas musicais.	94
Figura 74 - Cadastro de materiais utilizados nas pesquisas	94
Figura 75 - Lista dos refinamentos de pesquisa.....	95
Figura 76 - Lista das transformações melódicas	96
Figura 77 - Lista das transformações rítmicas.....	97
Figura 78 - Lista de transformações da tipologia.....	98
Figura 79 - Lista das transposições.....	99
Figura 80 - Cadastro de ferramentas administrativas.....	100
Figura 81 - Lista dos grupos de pesquisa cadastrados	101
Figura 82 - Cadastro de grupos de pesquisa	101
Figura 83 - Lista de usuários do sistema.....	104
Figura 84 - Cadastro de Usuários	104
Figura 85 - Seleção de permissões para o usuário	105
Figura 86 - Lista de relatórios diversos	106
Figura 87 - Relatório de pesquisas a serem verificadas.....	106
Figura 88 - Lista parcial dos trabalhos concluídos	107
Figura 89 - Lista parcial das conclusões disponíveis para impressão.....	108
Figura 90 - Suporte da aplicação	108
Figura 91 - Últimas Ações Realizadas	109
Figura 92 - Visualização da página oficial do python	110
Figura 93 - Representação Artística da Máquina de Turing	111
Figura 94 - Painel de Controle do WampServer.....	114
Figura 95 - Visualização da página oficial do WampServer	115
Figura 96 - Verificação da versão do python.....	116
Figura 97 - Comando para instalação da Virtualenv	117

Figura 98 - Comando para a instalação do virtualenvwrapper-win.....	117
Figura 99 - Comando para a criação do ambiente virtual.....	118
Figura 100 - Ativando o ambiente virtual.....	118
Figura 101 - Comando de instalação do Django.....	119
Figura 102 - Comando para criar uma aplicação em Django.....	120
Figura 103 - Estrutura de um projeto Django.....	120
Figura 104 - Arquivo _init_.py.....	121
Figura 105 - Arquivo settings.py.....	122
Figura 106 - Arquivo views.py.....	122
Figura 107 - Arquivo urls.py.....	123
Figura 108 - Diretório das aplicações pessoais.....	123
Figura 109 - Comando para a criação de apps.....	124
Figura 110 - Exemplo de arquivo models.py.....	124
Figura 111 - Exemplo de arquivo admin.py.....	125
Figura 112 - Exemplo de arquivo test.py.....	125
Figura 113 - Comando makemigrations.....	126
Figura 114 - Comando Migrate.....	126
Figura 115 - Comando runserver.....	126
Figura 116 - Conexões no servidor do domínio analysisproject.org.....	129
Figura 117 - Primeiro teste no servidor.....	131
Figura 118 - Primeira coleta de dados cena 01.....	134
Figura 119 - Segunda coleta de dados cena 01.....	134
Figura 120 - Terceira coleta de dados cena 01.....	135
Figura 121 - Quarta coleta de dados cena 01.....	135
Figura 122 - Quinta coleta de dados cena 01.....	136
Figura 123 - Primeira coleta de dados cena 02.....	136
Figura 124 - Segunda coleta de dados cena 02.....	137
Figura 125 - Terceira coleta de dados cena 02.....	137
Figura 126 - Quarta coleta de dados cena 02.....	138
Figura 127 - Quinta coleta de dados cena 02.....	138
Figura 128 - Requisições para uma resposta de 30 segundos.....	139
Figura 129 - Dados resultantes das coletas nas cenas 01 e 02.....	140
Figura 130 - Tempo Total X Número de requisições.....	141
Figura 131 - Requisições por segundo X total das requisições.....	142
Figura 132 - Tempo de utilização X número de requisições.....	142
Figura 133 - Tipos de Atributos.....	144
Figura 134 - Visão do Usuário do estilo Árvore.....	147
Figura 135 - Prelúdio 09 - Cravo bem temperado compasso 21.....	149
Figura 136 - Sonata III em Fá menor - 01 movimento - Telemann.....	149
Figura 137 - Concerto Grosso em Ré maior Op.6 no. 5 - Handel.....	150
Figura 138 - Romanesca com baixo progressivo.....	151
Figura 139 - Resultado da pesquisa com o Prinner.....	151
Figura 140 - Resultado da pesquisa Do - Ré - Mi.....	152
Figura 141 - Padrão de pesquisa aleatório escolhido pelo usuário.....	152
Figura 142 - Padrão encontrado em Bach.....	153
Figura 143 - Padrão encontrado em Schumann.....	154
Figura 144 - Um exemplo de construção melódica.....	155
Figura 145 - Área de desenvolvimento.....	156

LISTA DE TABELAS

Tabela 1 - Passos a serem realizados na fase 01.....	41
Tabela 2 - Passos a serem realizados na fase 02.....	47
Tabela 3 - Passos a serem realizados na etapa 03	53
Tabela 4 - Graus da escala X notas correspondentes	57
Tabela 5 - Estrutura de grupos da Romanesca.....	58
Tabela 6 - Parâmetros para a coleta de dados	133

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CPU	Unidade Central de Processamento
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
D3	Data-Driven Documents
DBMS:	Data Base Management System
GNU	General Public License
GPL	General Public License
HTML	HyperText Markup Language
HTTP	Hyper Text Transport Protocol
ICMC	Instituto de Ciências Matemáticas e Computação da USP
JS	Java Script
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MIT	Massachusetts Institute of Technology
MTV	Model / Template / View
MusicXML	Music Extensible Markup Language
MXL	Music XML compress
MySQL	Sistema de banco de dados baseado em SQL
ORM	Object-relational mapping
OS	Operacional System
PCA	Portland Cement Association
PDF	Portable Document Format
PHP	Personal Home Page
POO	Object-Oriented Programming
PY	Python
SGDB	Sistema de Gerenciamento de Banco de Dados
SQL	Estrutured Query Language
SSL	Secure Socket Layer
USP	Universidade de São Paulo
VHLL	Very High Level Language
XML	Extensible Markup Language

Sumário

1. INTRODUÇÃO	17
2. ANÁLISE MUSICAL E MÚSICA SIMBÓLICA (CONCEITOS INICIAIS)	21
2.1. Música Simbólica.....	22
2.2. Análise musical e a utilização de ferramentas tecnológicas	22
2.3. Análise musical realizada com recursos computacionais	24
2.4. Principais pesquisas e suas aplicações	25
2.4.1. O MelodicMatch.....	25
2.4.2. O MUSIC21	27
2.4.3. O ELVIS PROJECT e suas funcionalidades.....	29
2.4.4. MusicXML Analyser	30
3. ANÁLISE MUSICAL COMPUTACIONAL E METODOLOGIA	35
3.1. ANALYSIS PROJECT: UM ORGANIZADOR DO MÉTODO	37
3.1.1. Fase 01 - Coleta de Dados.....	38
3.1.2. Fase 02 - Preparação da Pesquisa	42
3.1.3. Fase 03 - Realização das buscas e análise dos resultados	48
3.1.4. Fase 04 – Conclusões	54
4. PADRÕES E ESQUEMAS PREDEFINIDOS NO ANALYSIS PROJECT	56
4.1. “Music in the Galant Style” - Robert Gjerdingen	56
4.1.1. Romanesca e um exemplo de lógica de programação.....	56
4.1.2. Clausulae.....	59
4.1.3. Dó - Ré - Mi	60
4.1.4. Prinner	61
4.1.5. Fenaroli.....	62
4.1.6. Fonte	63
4.1.7. Indugio.....	63
4.1.8. Ponte	64
4.1.9. Quiescenza.....	65

4.2.	Figuras retórico-musicais.	65
4.2.1.	Figuras que afetam a melodia	66
4.2.2.	Figuras que afetam a harmonia	68
4.2.3.	Elementos diversos	69
5.	PROTOCOLO MUSICXML.	71
5.1.	Manipulação das tags nos arquivos MusicXML.....	71
5.2.	Arquivos no formato texto e python.	77
6.	ANALYSIS PROJECT	79
6.1.	Recursos e funcionalidades da aplicação	79
6.1.1.	Tela de Abertura	79
6.1.2.	Tela de Acesso	80
6.1.3.	Painel de Controle (Visão Geral)	80
6.1.4.	Modelos de busca	81
6.1.5.	Metodologia Aplicada	87
6.1.6.	Cadastros Gerais.....	88
6.1.7.	Estruturas Musicais	94
6.1.8.	Ferramentas Administrativas	100
6.1.9.	Relatórios Diversos.....	106
6.1.10.	Suporte da Aplicação (Help).....	108
6.1.11.	Ações Recentes	109
7.	REQUISITOS TÉCNICOS.....	110
7.1.	Python – Principais características e usabilidade.....	110
7.2.	Ambiente de desenvolvimento.....	112
7.2.1.	Sistema operacional e instalações	113
7.2.2.	Configurações do ambiente de desenvolvimento	116
7.3.	Framework Django e suas aplicações.....	119
7.4.	Análise de desempenho	127
7.4.1.	Ferramentas utilizadas para análise de desempenho	128
7.4.2.	Etapas para a realização da análise do servidor	130

7.4.3. Coleta de dados das cenas propostas:.....	133
7.4.4. Requisições necessárias para resposta em cerca de 30 segundos	139
7.4.5. Construção das tabelas para verificação dos dados	139
7.4.6. Considerações Finais	143
7.5. Visualização Computacional.....	143
7.5.1. Sobre os Dados coletados.....	144
7.5.2. Apresentação de dados no Analysis Project	145
8. RESULTADOS E DISCUSSÃO	148
8.1. Resultados.....	148
8.1.1. Pesquisas com figuras de retórica.....	148
8.1.2. Pesquisas com padrões e esquemas de Gjerdingen.....	150
8.1.3. Pesquisa com padrão determinado pelo usuário.....	152
8.1.4. Exemplo interessante	154
8.2. Discussão	155
9. Conclusão	159
Referências.....	161
APÊNDICE	
APÊNDICE A - Arquitetura de programação do Analysis Project	167
APÊNDICE B - Opções de permissões para grupos do Analyis Project	184
APÊNDICE C - Lista de Figuras de Retórica cadastradas no Analysis Project em ordem alfabética.....	191
APÊNDICE D - Lista de autores relacionados a Retórica Musical cadastrados no Analysis Project (Referencias primarias e secundárias)	200
ANEXOS	
ANEXO A - Recursos do Music21.....	205
ANEXO B- Listas as obras castradas no Analysis Project	208
ANEXO C - Lista de softwares compatíveis com o padrão MusicXML	235
ANEXO D - MusicXML (Arquivos DTD)	245

1. INTRODUÇÃO

Segundo Duckles(2014) a musicologia é o estudo acadêmico da música. Os trabalhos realizados em seu domínio podem ser classificados em nove grandes eixos de pesquisa: 1) Método histórico; 2) Método teórico e analítico; 3) Crítica textual; 4) Pesquisa arquivística; 5) Lexicografia e terminologia; 6) Organologia e iconografia; 7) Práticas interpretativas; 8) Estética e crítica; 9) Dança e história da dança.

O ramo da musicologia classificado como método teórico e analítico, procura compreender a música através de suas estruturas internas de construção. O presente trabalho se insere nesse cenário, apontando exclusivamente para a análise de música simbólica.

A análise musical, como a compreendemos atualmente, é a apropriação de uma peça musical por um pesquisador que teria a habilidade de descrever suas estruturas de construção. A capacidade de observar como os elementos de determinada peça musical se articulam é a resultante de um investigativo, que tem como objetivo a geração de um conhecimento sobre o objeto de estudo.

O analista necessita conhecer diversas ferramentas de análise musical para escolher qual é a mais adequada para a realização do seu projeto. A escolha da ferramenta de análise está diretamente relacionada com a postura que o musicólogo se coloca em relação à obra. Segundo (TREITLER, 1989) podemos distinguir quatro situações analíticas:

1. Observador dos eventos musicais: (trata do que a obra é). A análise focaliza a peça musical na esfera de si mesma e/ou do seu compositor.
2. Interpretação da obra/análise crítica: (trata do que a obra deve ser). O foco analítico já se estende à esfera do intérprete
3. Explicar existência da obra: (causa da existência da obra). A obra é colocada na esfera do conjunto de obras ao qual ela se relaciona por traços estilísticos.
4. Coordena a obra musical com o mundo do qual ela é produto: (a obra está na esfera máxima da cultura geral de sua época). Foco na análise do discurso.

Um analista habitual utiliza Hoffmann, Kretzschmar, Riemann, Tovey, Schoenberg, Reti, Keller, Babbitt, Perle, Forte, Lewin, Schenker entre outros como instrumental para suas análises. Agrega a estes pesquisadores conceitos da sociologia, linguística, história, semiótica e diversas disciplinas que irão atender suas necessidades musicológicas; esta diversidade de objetivos foi claramente observada por TREITLER (1989).

Paralelo a este turbilhão de necessidades, a tecnologia chega avassaladora, oferecendo possibilidades de verificações até então desconhecidas na musicologia. Realizar um estudo comparativo com um escopo superior a 6000 obras é uma realidade recente. Os resultados obtidos, podem em alguns aspectos, serem extremamente esclarecedores. A esta “nova possibilidade” a musicologia se curva, com alguns justificáveis receios, mas com grandes esperanças porque vislumbra as potencialidades de tal assertiva.

“... planteo que la realización de un análisis musical asistido por computadoras no es un fin en sí mismo, sino una herramienta a disposición que se inserta en un plan de investigación más amplio; ofrece alternativas que están ausentes de los análisis tradicionales; acelera el estudio de los datos; permite abordar cuestiones relativas a los procesos cognitivos de la interpretación y la percepción; provee recursos innovadores para la descripción y presentación del análisis”. (Sammartino, 2015)

Segundo Sammartino, a análise musical realizada com o auxílio de uma ferramenta tecnológica, abre caminhos e possibilidades até então desconhecidas pela musicologia. O musicólogo que nós cognominamos de “um analista habitual” realiza suas análises de forma manual, ficando limitado a um conjunto restrito de obras. O trabalho é meticuloso e o esforço é intenso, consumindo muito tempo do pesquisador. O processamento computacional e aceleração de dados modifica radicalmente este cenário.

Exemplificando o modelo adotado por um “analista habitual”, o processo geralmente é realizado em três etapas: 1) Busca de determinado evento musical na partitura; 2) Classificação do evento; 3) Oferta de sentido as análises realizadas. A utilização da tecnologia nos permite realizar as duas primeiras etapas deste processo com maior rapidez e menor margem de erros. Estas duas fases de pesquisa são partes integrantes, até então indissociáveis, da análise musical, inclusive sendo

valoradas como um diferencial do especialista. A mudança é profunda e não se restringe ao ambiente de trabalho. É necessário o desenvolvimento de novas habilidades e métodos.

Segundo (CASTANHA, 2008) existe diferença entre norma, técnica e método. Normatização são convenções que utilizamos para que nossos trabalhos possam ser inteligíveis, técnica são procedimentos utilizados para recolher os dados necessários à pesquisa e método é a utilização destes dados aferindo-lhes sentido.

Normas são convenções, utilizadas para poupar o tempo dos leitores e para garantir modelos próximos de organização para os trabalhos científicos, variando de lugar para lugar, de época para época e, muitas vezes, de pesquisador para pesquisador. Técnicas, por outro lado, são os meios práticos para se realizar uma determinada pesquisa, para se recolher, organizar e interpretar o material que interessa ao tema e que, obviamente, dependem do método que será empregado. Métodos, no entanto, são as grandes concepções de trabalho do pesquisador, refletindo sua ideologia a respeito do significado das obras de arte no passado e no presente e do significado da própria pesquisa para a sua compreensão, dependendo de ideologias dominantes em sua época e de uma opção pessoal do próprio pesquisador.(CASTANHA, 2008)

A utilização de procedimentos tecnológicos como meio prático de se organizar a pesquisa musicológica deve ter a capacidade de recolher, organizar e interpretar o material relativo ao nosso objeto de estudo.

Nosso objetivo inicial foi verificar a incidência de padrões retóricos e alguns esquemas propostos pelo pesquisador Gjerdingen(2007) em uma grande quantidade de obras. A ferramenta capaz de realizar essa pesquisa (recolher, organizar e interpretar) não existe.

Criamos!

O Analysis Project foi desenvolvido com o objetivo de preencher essa lacuna, oferecendo aos musicólogos a capacidade de realizar pesquisas de padrões em música simbólica associada grandes coleções. A aplicação não é, em si, o objetivo desta tese, contudo sem ela este trabalho não poderia acontecer. Ela é o resultado prático de uma necessidade, um ponto de apoio para o desenvolvimento de novas tecnologias voltadas para a análise musical.

A maior dificuldade encontrada em todo o percurso foi a conversa entre a musicologia (com seus afetos sublimes) e a tecnologia (com suas maravilhosas certezas). Mundos muito diferentes com linguagens distintas. Este trabalho em sua estrutura pode sinalizar o que se deve (ou não) fazer em contextos dessa natureza¹.

Desenvolvemos um método capaz de conversar com essas duas realidades. O pesquisador terá de estar diante do computador para realizar sua análise. O conhecimento prévio de recursos computacionais não é um pré-requisito para o desenvolvimento de suas habilidades como especialista. Diante desse quadro optamos por alguns procedimentos:

- 1) O processo deve ser simples e direto.
- 2) Vertical com etapas bem definidas; um modelo “passo a passo”.
- 3) Hierárquico com valores diferenciados das ações principais

O Analysis Project é um validador desse método que pretende ser simples em seus procedimentos gerando aprendizado rápido e alto desempenho. Foram realizados, contudo, poucos testes de verificação da metodologia devido a aplicação estar estacionada no computador pessoal do proponente deste trabalho. Teríamos que colocar a ferramenta em um ambiente de produção para chegarmos a resultados mais conclusivos.

Realizamos pesquisas em cenários variados, alterando o volume de obras analisadas, bem como períodos históricos, compositores e instrumentação. Os resultados gerados mostraram-se consistentes e organizados de maneira sólida e transparente. A clareza na apresentação dos argumentos escolhidos pelo pesquisador está relacionada as técnicas de visualização computacional empregadas na aplicação.

O Analysis Project pode ser classificado como um sistema especialista. O pré-processamento se realiza através da extração dos registros em arquivos MusicXml que são alojados e indexados em um banco MySQL. A base de dados é gerenciada pelo framework Django. O sistema de pesquisa até o presente momento é fixado em funções que realizam a busca nesta base de dados. A linguagem de programação utilizada é o Python.

¹ A “tecnologia do futuro” aprende com seus erros.

2. ANÁLISE MUSICAL E MÚSICA SIMBÓLICA (CONCEITOS INICIAIS)

Segundo Bent e Drabkin (1990) a análise musical tem seus primórdios no Sec. XVII com estudo das formas, repetições e sequências musicais.

Mattheson(Sec. XVIII) desenvolve o trabalho de Burmeister (Sec.XVII) e em seu último escrito *Der Vollkommene Capellmeister ("O mestre-de-capela Perfeito")* ressalta a importância de conhecimentos extrínsecos à técnica musical para atingir o ideal de perfeição.

Ein ungenannter Französischer Schriftsteller meldet ausdrücklich: ein Componist werde nimmer in seiner Kunst hervorragen, falls er keine Gelehrsamkeit besitze. Ein Mahler kann wol ein Künstler seyn; ist er aber kein Historicus, so wird er zwar ein künstliches Bild, doch nicht die Gemüths-Bewegungen, welche es, nach dem Inhalt der Geschichte, haben sollte, ausdrücken. Ein gleiches verstehe man von einem Componisten; seine Arbeit kan endlich das Stück eines fleissigen Meisters heissen; weil es ihm aber an der Gelehrsamkeit mangelt, hat er die Natur des Textes, wie der Mahler die Leidenschafften seines Bildes, nicht in Acht nehmen Können.²

A estudo científico da música dissociado à prática é uma proposta inovadora. Este “novo pensamento” metodológico oferece a musicologia um campo de trabalho, mais voltado ao fazer científico enquanto a prática musical fica no escopo do campo artístico.

Como decorrer do estudo de princípios lógicos relacionados a música, surge em 1722 a publicação do livro *Traité de l'harmonie* (Tratado de Harmonia) do teórico Jean-Philippe Rameau. O estudo da música passa do campo da audição e apreciação para a compreensão de fenômenos utilizando a partitura musical (música simbólica).

Sauveur(1701) desenvolve seus estudos baseando suas análises em princípios da física e acústica seguido por Stumpf(1890).

No início do século XX o músico e teórico Heinrich Schenker desenvolve uma metodologia analítica baseado na premissa de que o processo composicional se

² Um escritor francês anônimo relata explicitamente que um compositor nunca se destacaria em sua arte se não tivesse erudição. Um pintor pode ser um artista; mas se não for um historiador, realizará uma imagem artificial(no sentido de perfeição técnica), mas não os movimentos emocionais que, de acordo com o conteúdo da história, devem expressá-la. O mesmo é entendido por um compositor; seu trabalho pode finalmente ser chamado de peça de um mestre trabalhador; mas como ele não possui erudição, ele não conseguirá compreender à natureza do texto, igualmente com o pintor, os afetos de seu quadro. (Tradução do Autor)

realiza através da ornamentação dos acordes. Simplificando as estruturas chegaríamos aos princípios fundamentais da composição. Este modelo analítico ficou conhecido como Análise Schenkeriana.

Outros pesquisadores importantes influenciaram a análise musical, cada qual oferecendo uma visão diferente Réti(1951) propôs em seu trabalho *The Thematic Process in Music* , um modelo no qual temas ou motivos temáticos seriam a base de desenvolvimento das obras musicais.

A análise musical baseada em equivalência foi proposta por Ruwet(1972). Este modelo é utilizado em linguística e se baseia em repetições. Qualquer repetição pode ser considerada uma unidade. Aqui todas as equivalências são validadas (notas/motivo/frase). Todos os trabalhos existentes de análise de grande quantidade de obras musicais tiveram aí a sua origem. O presente trabalho, pelo momento histórico, está mais próximo aos modelos analíticos utilizados em inteligência artificial do que os processos aplicados por Ruwet, contudo, deixamos aqui registrado a grande admiração pelo trabalho do linguista. Em futuro não muito distante O Analysis Project utilizará algoritmos de PNL (processamento de linguagem natural) as necessidades musicológicas.

2.1. Música Simbólica

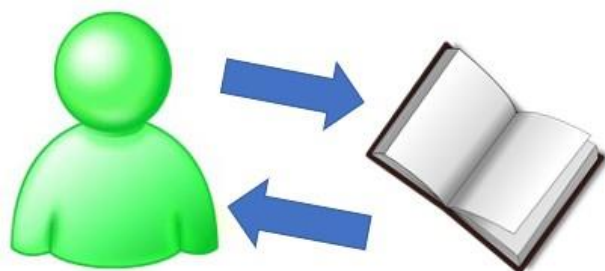
Música simbólica é àquela armazenada em sistema de notação. O símbolo traduz todas as suas características audíveis como frequência, duração dos eventos, dinâmica, timbre. Modelos computacionais foram desenvolvidos para realizar esta “tradução” do som. Os editores de partituras bem como o sistema MIDI e o padrão MusicXML são exemplos de casos bem sucedidos.

Nosso projeto é um sistema de busca de padrões em música simbólica utilizando o protocolo MusicXML.

2.2. Análise musical e a utilização de ferramentas tecnológicas

A análise musical está baseada em um relacionamento do analista e seu objeto de estudo. Em um processo convencional (independente da teoria analítica adotada pelo musicólogo) teremos um espaço ocupado por dois objetos como descrito na figura abaixo.

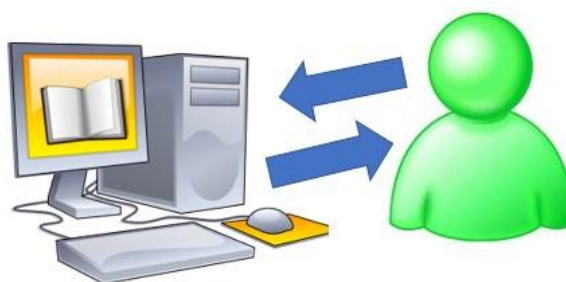
Figura 1 – Analista X Obra Musical



Fonte: Do Autor

Ao utilizarmos ferramental tecnológico observamos a presença de uma ferramenta mediadora:

Figura 2 - Analista X Computador X obra



Fonte: Do Autor

A figura 2 procura descrever o processo de análise computacional. Existe uma relação do analista com os resultados que o computador oferece. Pode haver um relacionamento entre o estudioso e peça musical diretamente, mas não é imprescindível para que o movimento de análise aconteça. Os resultados oferecidos pela ferramenta devem ser analisados e, em caso de inconsistência descartados pelo pesquisador.

A figura 1 ilustra um comportamento analítico que classificamos como tradicional. Nele qualquer intervenção externa é desnecessária. As descobertas ocorrem no exercício do processo não havendo necessidade de nenhum argumento pré-definido. A base do conhecimento analítico se encontra na mente do especialista.

Comparando os dois processos, a utilização de uma ferramenta computacional irá modificar drasticamente o processo metodológico. Seu papel não se resume a um

facilitador por oferecer como resultado um conjunto considerável de variáveis e possibilidades até então inacessíveis.

Para a realização da análise musical computacional precisamos de um método que descreve e direciona o pesquisador neste novo ambiente de trabalho, uma base de dados consistente e um argumento que aparece na forma de padrões ou esquemas

“O que significa se referir a um padrão musical como um esquema? O termo em si tem uma longa história e foi utilizado pela primeira vez em filosofia e, em seguida, na psicologia. “Esquema” (Kant) refere-se ao que é amplamente chamado de uma representação mental ou categoria, e, portanto, compartilha significados de termos como “ideia” ou “forma” (Platão), “tipo ideal” (Weber), “semelhança de família” (Wittgenstein), “Arquétipo” (Frye), “protótipo” (Posner), “essência” (Putnam), “tipo natural” (Rosch), e assim por diante.”

(Gjerdingen 2007, p. 10)³

Precisamos ter alguns protótipos definidos ou alguns padrões. Deixemos claro que sem definir anteriormente quais são seus “modelos de busca”, o pesquisador não conseguiria iniciar o seu trabalho. Sem a consciência do que estes modelos significam sua pesquisa será ineficiente

2.3. Análise musical realizada com recursos computacionais.

Entre as ferramentas desenvolvidas para auxílio da musicologia utilizando recursos computacionais identificamos basicamente duas vertentes:

1. Criação e edições digitais (cf. Kepper et al., 2014; Veit, 2015)
2. Estudos quantitativos para análises musicológicas. (cf. Müllensiefen e Frieler, 2004; Vigilanti, 2007).

As análises de cunho quantitativo utilizam sistemas de recuperação de informações musicais (MIR). Estes sistemas se prestam a pesquisa de coleções musicais (geralmente grande quantitativo de obras) de acordo com os parâmetros musicológicos determinados pelo pesquisador.

Dentre os sistemas desenvolvidos apontamos alguns aspectos:

³ Tradução: (do Autor)

1. Positivos: Ferramentas desenvolvidas com reconhecimento óptico como o Peachnote 5 (Viro, 2011) utilizando o (OMR) para indexar mais de um milhão de folhas da Biblioteca de Música Petrucci .
2. Negativos: Dificuldade de usabilidade (Interfaces de linhas de comando requerendo do usuários conhecimentos de programação⁴).Consultas e pesquisas restritas somente às coleções disponibilizadas no aplicativo⁵.

O Analysis Project utiliza como padrão o formato MusicXML e existem hoje algumas bases de dados que disponibilizam grande quantidade de música neste formato (Good, 2001)⁶.

2.4. Principais pesquisas e suas aplicações

As pesquisas e aplicativos aqui descritos foram aqueles que tivemos acesso. Em todos os casos utilizam o padrão MusicXML como alicerce a sua base de dados.

2.4.1. O MelodicMatch

O MelodicMatch é um software de análise musical que utiliza padrões para a composição dos seus relatórios de pesquisa. É uma ferramenta que permite a procura de frases musicais em um conjunto obras podendo o usuário utilizar filtros de seleção na melodia e no ritmo de um padrão pré-definido. O formato de arquivo utilizado nas pesquisas é o MusicXML.

Basicamente oferece 3 funcionalidades:

1. Criação de um resumo gráfico da música.
2. Localização de padrões musicais.
3. Utilização de recursos gráficos para comparar as formas musicais.

O MelodicMatch foi utilizado para o estudo comparativo dos padrões existentes na Ladainha de Nossa Senhora de Faustino Xavier do Prado e um grupo de compositores portugueses da corte de D. João V. Os resultados da pesquisa, bem

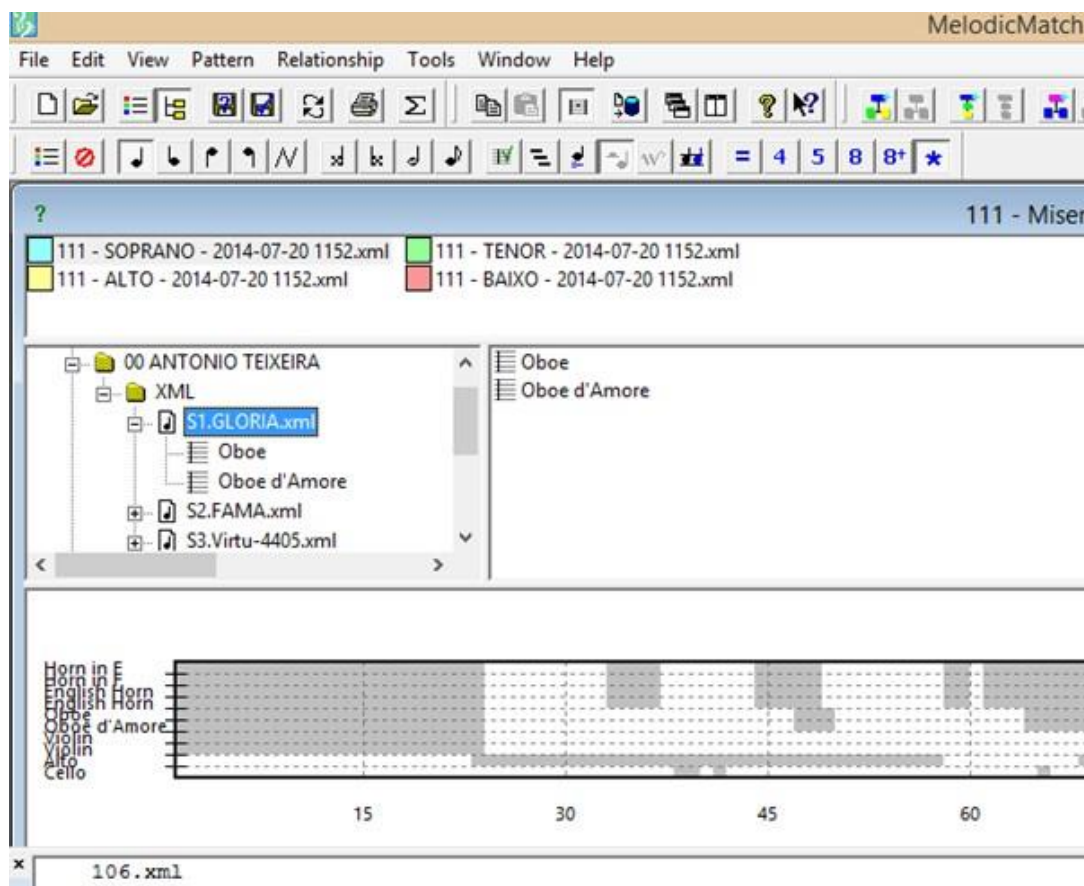
⁴ Como exemplo podemos citar o Hundrum 2.0 e também o Music21.

⁵ Como exemplo podemos citar o RISM 3.0 e o HymnQuest 4.0.

⁶ O site <https://www.musicxml.com/music-in-musicxml/> oferece uma grande variedade de páginas que disponibilizam coleções no formato MusicXML.

como todo o processo analítico descritivo se encontra publicado em minha dissertação de Mestrado⁷.

Figura 3 - Página Inicial do MelodicMatch



Fonte: Do Autor

A figura 03 é um exemplo de utilização do MelodicMatch. Na parte superior esquerda encontramos os padrões que foram pesquisados. (quatro linhas melódicas superpostas) A estrutura de comparação (Padrão de número 111) possui as melodias existentes nas linhas do soprano, contralto tenor e baixo. Logo abaixo encontramos o resultado de similaridades nas obras de Antônio Teixeira (Glória nas partes do oboé o oboé d'Amore). O gráfico na parte inferior do aplicativo indica os instrumentos relacionados aos compassos da obra selecionada.

⁷ Utilização de Ferramentas Tecnológicas para Análise Musical: A Ladainha de Faustino Xavier do Prado na Visão de Um Descritor. Disponível em <https://teses.usp.br/teses/disponiveis/27/27157/tde-12012016-112114/publico/ROBSONDIAS.pdf>

2.4.2. O MUSIC21

Music21⁸ é um conjunto de ferramentas desenvolvidas para auxílio a pesquisa musicológica. Ele é focado na resolução rápida de questões quantitativas e qualitativas sobre determinado grupo de obras que possui em sua base de dados. Tem uma área de atuação bem diversificada a saber:

- Ensino e prática dos fundamentos da teoria musical
- Edição de notação musical
- Estudo geral da música e aplicações na psique humana
- Composição por algoritmos

Figura 4 - Visualização do site oficial do Music21

The screenshot shows the official website for Music21. The browser address bar indicates the URL is web.mit.edu/music21/. The page title is "music21: a toolkit for computer-aided musicology".

What is music21?
 Music21 is a set of tools for helping scholars and other active listeners answer questions about music quickly and simply. If you've ever asked yourself a question like, "I wonder how often Bach does *that*" or "I wish I knew which band was the first to use these chords in this order," or "I'll bet we'd know more about Renaissance counterpoint (or Indian ragas or post-tonal pitch structures or the form of minuets) if I could write a program to automatically write more of them," then music21 can help you with your work.

How simple is music21 to use?
 Extremely. After starting Python and typing "from music21 import *" you can do all of these things with only a single line of music21 code:

```

Display a short melody in musical notation:
converter.parse("tinynotation: 3/4 c4 d8 f g16 a g f#").show()

Print the twelve-tone matrix for a tone row (in this case the opening of Schoenberg's Fourth String Quartet):
print (serial.rowToMatrix([2,1,9,10,5,3,4,0,8,7,6,11]) )
  
```

A code snippet for the `closedPosition` method is shown on a sticky note:

```

def closedPosition(self):
    ...
    returns a new Chord object with :

    >>> chord1 = Chord(["C#4", "65",
    >>> chord2 = chord1.closedPosition()
    >>> print(chord2.lily.value)
    <cis' e' g'>4
    ...

    newChord = copy.deepcopy(self)
    tempChordNotes = newChord.pitches:
    chordBassPS = self.bass().ps
    for thisPitch in tempChordNotes:
        while thisPitch.ps > chordBa:
            thisPitch.octave = thisP:
    newChord.pitches = tempChordNote:
  
```

Links at the bottom of the page include:

- [Get Started with music21](#)
- [Browse the music21 documentation](#)
- [Download music21 from GitHub](#)

Fonte: <http://web.mit.edu/music21/>

É um "Kit de ferramentas" disponível desde 2008, criado no MIT (Massachusetts Institute of Technology) e se encontra em constante expansão.

2.4.2.1. Exemplo de utilização

Apesar de ser um exemplo que se encontra no sítio (web) do desenvolvedor, achei prudente transcrever, podendo assim, explicar os comandos detalhadamente

facilitando a compreensão da utilização da ferramenta. Uma das dificuldades de popularização do Music21 e a impossibilidade de acessar seus recursos sem conhecer a linguagem de programação Python.

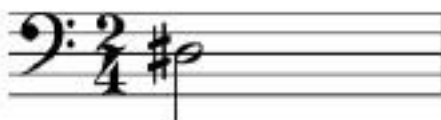
Com o Python 3 ou superior ativo (em um servidor ou em sua máquina local) abra o prompt de comando e digite:

- `pip install music21` (Instala o Music21) em seguida digite:
- `from music21 import *` (Importa as bibliotecas para serem utilizadas)

Imprimindo uma nota na tela

- `n = note.Note("D#3")` - Escolheu a nota que deseja imprimir
- `n.duration.type = 'half'` - Escolheu a figura rítmica
- `n.show()` - imprime ela no aplicativo instalado de música.

Figura 5 - Nota impressa



Fonte: <https://web.mit.edu/music21/doc/about/what.html>

Um exemplo mais elaborado de impressão na tela de uma pequena frase musical:

Figura 6 - Music21 - Pequena Melodia

```
littleMelody = converter.parse("tinynotation: 3/4 c4 d8 f g16 a g f#")
littleMelody.show()
```



Fonte: <https://web.mit.edu/music21/doc/about/what.htm>

|

Vamos analisar a escrita do comando acima:

3/4 (compasso)

c4 (Nota dó central com a figura rítmica semínima representada pelo número

4)

- d8 (Nota ré com a figura rítmica colcheia representada pelo número 8)
- f (Nota fá mantendo a figura anterior que é uma colcheia)
- g16 (Nota sol com a figura rítmica semicolcheia representada pelo número 16)
- a g # (Notas lá, sol e fá sustenido mantendo a figura rítmica anterior)

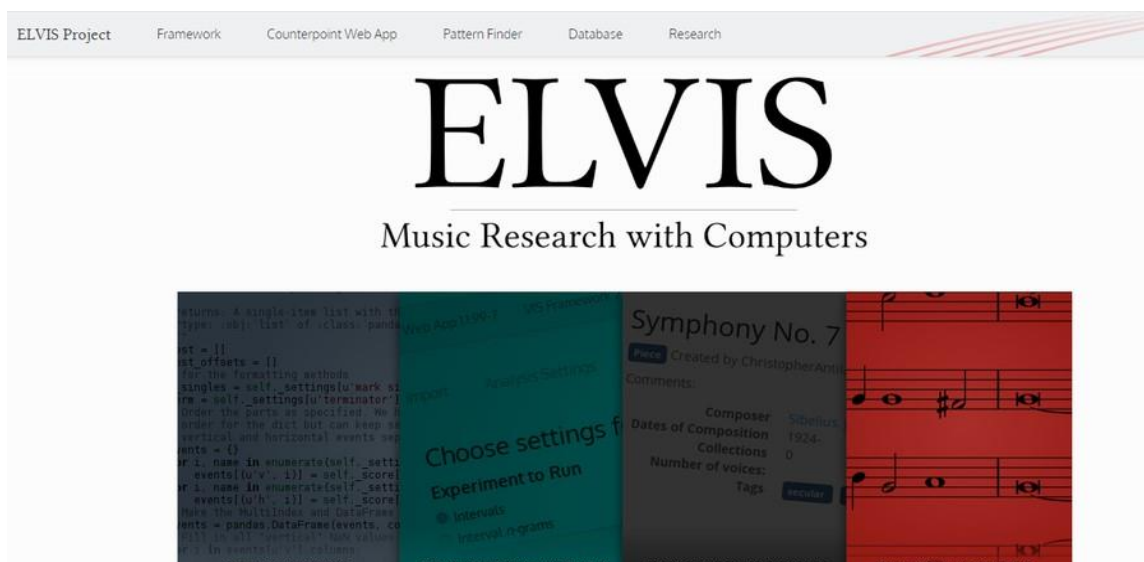
2.4.2.2. Funcionalidades do Music21

O Music21 é uma aplicação com diversas funcionalidades abordando ramos diferenciados da musicologia. Recomendando a home-page oficial do desenvolvedor⁹ para maiores esclarecimentos.

2.4.3. O ELVIS PROJECT e suas funcionalidades

O ELVIS é um projeto que possui diversos aplicativos dos quais destacamos o VIS, especialmente desenvolvido para análise musical.

Figura 7 - Visualização do site oficial do ELVIS



Fonte: <https://elvisproject.ca/>

O VIS é um software disponibilizado como software livre de acordo com os termos da Affero General Public License.

⁹ <https://web.mit.edu/music21/doc/index.html>

VIS é um pacote Python que usa as bibliotecas music21 e PANDAS sendo um sistema flexível que ajuda a desenvolver programas de análise musical por computador.

O VIS foi produzido pela equipe da Universidade McGill responsável pelo Projeto ELVIS e sua documentação se encontra disponível no site <https://vis-framework.readthedocs.io/en/v3.0.5/>

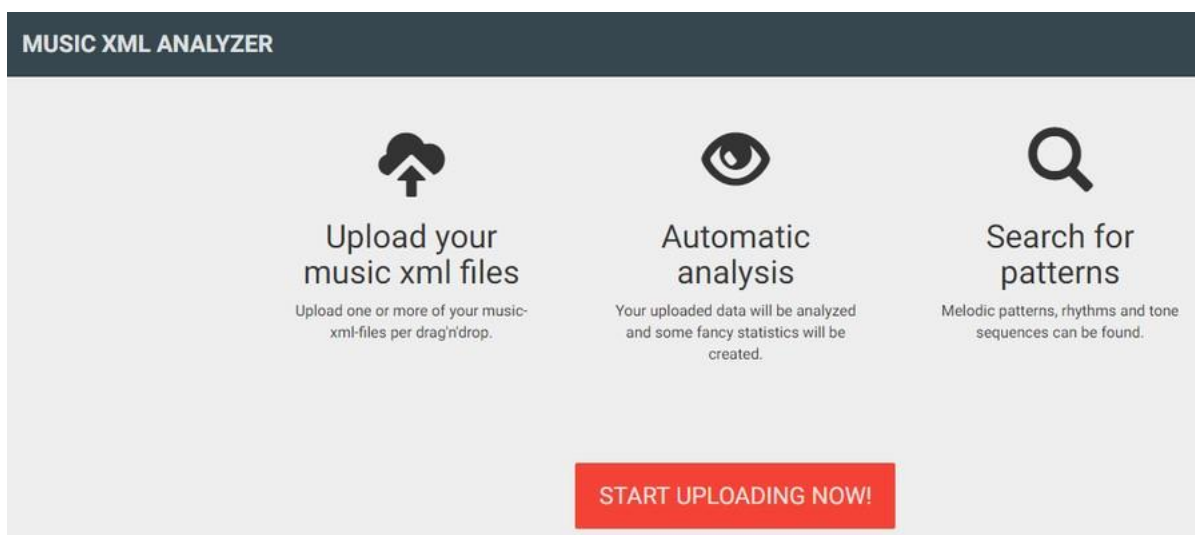
O projeto ELVIS possui em seu domínio¹⁰ um aplicativo para ser utilizado na web para análise de obras contrapontísticas.

Apesar da proposta do VIS ser auxiliar o desenvolvimento de softwares para análise de música simbólica, não utilizamos suas bibliotecas. Preferimos criar nossas soluções visto o Analysis Project necessitar de particularidades que serão descritas posteriormente.

2.4.4. MusicXML Analyser

O MusicXML Analyser é um software que foi idealizado para analisar padrões melódicos de maneira quantitativa em uma música ou um conjunto de obras.

Figura 8 - Visualização do site oficial do MusicXML Analyser



Fonte: <http://music-xml-analyzer.herokuapp.com/>

As pesquisas são realizadas pela web e os resultados são oferecidos aos usuários em diversos formatos como áudio, e arquivos no formato (.pdf) e (.csv).

¹⁰ <http://elvisproject.ca/> (Nota do Autor)

A ferramenta é composta de três componentes a saber:

Função de Upload

Função de Análise

Função de pesquisa

Primeiramente o usuário faz um Upload dos arquivos que deseja analisar. Logo em seguida estes arquivos são analisados pela ferramenta que calcula as frequências básicas. Os resultados são guardados em um banco de dados SQL e podem ser exibidos.

Figura 9 - – Frequências em um grupo de obras no formato MusicXML



Fonte: <http://dh2016.adho.org/static/data/458.html>

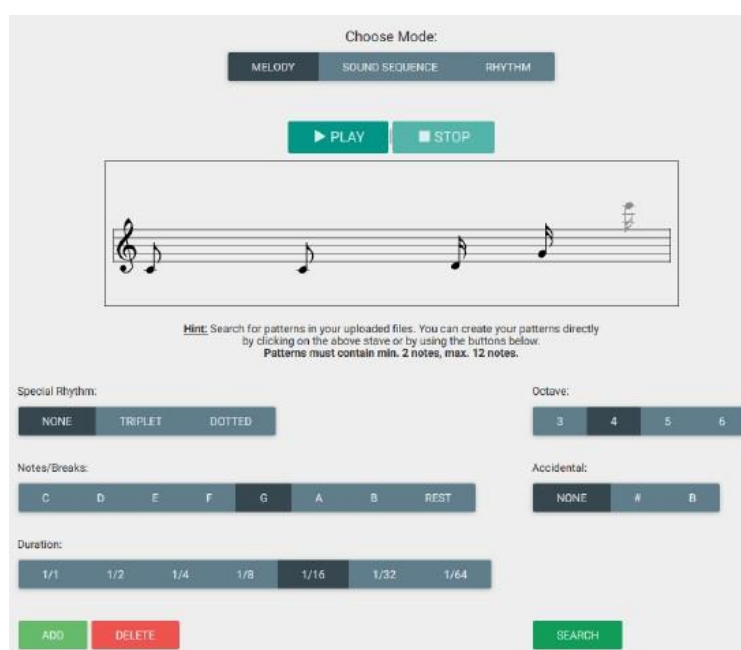
A figura 9 apresenta um resultado de busca efetuado pela aplicação. Ela nos oferece estatísticas gerais de notas (aparecem 254 notas Si), quais os instrumentos utilizados¹¹, intervalos, duração das notas. Caso o musicólogo deseje pesquisar por

¹¹ No caso dos instrumentos eles devem estar descritos no arquivo MusicXML para que possam aparecer na estatística. (Nota do Autor)

um padrão rítmico (colcheia, colcheia, semínima etc.) ou melódico (do, do mi, ré) ele também obterá resultado satisfatório. para uma busca mais refinada onde tenha um sequência de notas e ritmos ele deve escrever da seguinte forma: (a ' / colcheia, c' / colcheia, e ' / semínima, f' / semínima) e irá obter também bons resultados.

Podem ser igualmente realizada pesquisas utilizando a biblioteca virtual VexFlow¹². Lembrando igualmente que os padrões de pesquisa realizados podem ser executados porque o software utiliza uma biblioteca de sons MIDI (Midi.js¹³).

Figura 10 - Interface para inserir padrões para pesquisa nos arquivos MusicXML.



Fonte: <http://dh2016.adho.org/static/data/458.html>

A figura 10 representa a tela para a inserção de padrões a serem pesquisados nos arquivos. O Usuário pode inclusive ouvir o padrão selecionado. Depois de realizada a pesquisa os resultados são exibidos.

¹² Para maiores informações consultar o site <http://www.vexflow.com/> (Nota do Autor)

¹³ Para maiores informações consultar o site <http://mudcu.be/midi-js> (Nota do Autor)

Figura 11 - O padrão pesquisado aparece em vermelho.

Lee Actor (2003) - Prelude to a Tragedy
(ActorPreludeSample.xml)

▶ PLAY
■ STOP

About the finding:

Part name (Instrument):	Clarinet in Bb	Key: C major
Part ID:	P5	Measures: 35 - 38
Voice:	1	

Fonte: <http://dh2016.adho.org/static/data/458.html>

Como mostra a figura 11, o resultado apresenta do padrão pesquisado bem como o número total de ocorrências. Toda a melodia pode ser reproduzida no navegador ou baixada em um arquivo (.pdf¹⁴).

O MusicXML Analyser utilizou para sua construção as seguintes tecnologias:

- HTML¹⁵
- CSS¹⁶
- JavaScript¹⁷
- PHP¹⁸

com as seguintes bibliotecas:

- Laravel¹⁹

¹⁴ Para a exportação no formato PDF o desenvolvedor do software utilizou a biblioteca jsPDF. Maiores informações no site <https://parall.ax/products/jspdf> (Nota do Autor)

¹⁵ HTML é uma linguagem de marcação que pode ser interpretada por navegadores. É utilizada para desenvolvimento de web-sites. (Nota do Autor)

¹⁶ CSS tem como principal função adicionar estilos a um documento web. (Nota do Autor)

¹⁷ "JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multi-paradigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web". Texto extraído do site <https://pt.wikipedia.org/wiki/JavaScript>

¹⁸ PHP é uma linguagem de programação muito utilizada na web capaz de gerar código dinâmico. É uma linguagem interpretada. (Nota do Autor).

¹⁹ Maiores informações no site do desenvolvedor <http://laravel.com/> (Nota do Autor)

- [jQuery](#)²⁰
- [D3.js](#)²¹
- [Bootstrap](#)²²
- [Typed.js](#)²³
- [Dropzone.js](#)²⁴

O presente capítulo, inicialmente, realizou um pequeno resumo sobre a história da análise musical, tentou demonstrar como um analista habitual se coloca diante do seu objeto de estudo e descreveu algumas possibilidades de um novo modelo: A análise musical computacional.

Sobre análise musical utilizando ferramentas tecnológicas, utilizamos exemplos abordados em três níveis:

- Questões conceituais (O que o sistema realiza, porque e a quem se destina)
- Questões metodológicas (Como ele realiza)
- Questões tecnológicas (Quais as tecnologias que ele utiliza)

Reconhecemos que são níveis diferentes de abstração e que podem ser desinteressantes para alguns leitores. Se para o musicólogo explicar o conceito de frase musical ou motivo seria completamente desnecessário, é de fundamental importância para um programador. A recíproca é verdadeira.

²⁰ Maiores informações no site do desenvolvedor <https://jquery.com/> (Nota do Autor)

²¹ Maiores informações no site do desenvolvedor <http://d3js.org/> (Nota do Autor)

²² Maiores informações no site do desenvolvedor <http://getbootstrap.com/> (Nota do Autor)

²³ Maiores informações no site do desenvolvedor <http://www.mattdboldt.com/demos/typed-js/> (Nota do Autor)

²⁴ Maiores informações no site do desenvolvedor <http://www.dropzonejs.com/> (Nota do Autor)

3. ANÁLISE MUSICAL COMPUTACIONAL E METODOLOGIA

Como observado anteriormente, existem diferentes tipos de análise: a fraseológica, a harmônica, a schenkeriana, entre outras. Todos os modelos observados realizam o processo analítico de maneira manual. O volume de obras analisadas utilizando estes métodos geralmente é pequeno; porque demanda tempo e atenção de um especialista. Qual a necessidade que se realizar pesquisas musicológicas em grande quantidade de obras? Para chegarmos a conclusões sobre as características musicais existentes em determinado período histórico devemos analisar o maior número de obras possíveis relacionadas, de outra sorte, realizaremos estudos de casos com generalizações dos resultados o que em nossa opinião é um procedimento frágil.

No contexto musicológico a análise musical de grande volume de obras teve seu início com Ruwet(1972). Para o pesquisador a música obedecia aos mesmos princípios de sintaxe da linguística sendo construída a partir de elementos que se repetem. Validou seus experimentos utilizando ferramentas computacionais e algoritmos que analisavam as notas, os motivos e as frases repetidas nas obras musicais²⁵.

A estes “elementos que se repetem” damos o nome de padrões ou esquemas. A diferença fundamental entre as análises propostas por Ruwet e as que desenvolvemos nesta tese está nos objetos de verificação de similaridades. Os algoritmos de Ruwet selecionavam as notas ou as estruturas contidas nas obras (motivos, frases) e realizava uma combinação indicando a quantidade de repetições existentes de determinado evento. A partir dos valores quantitativos recolhidos o musicólogo chegaria a conclusões sobre determinado conjunto de obras. Nosso interesse é verificar a existência de padrões pré-determinados (Figuras retórico-musicais e esquemas propostos por Gjerdingen(2007) em sua obra *Music In The Galant Style*) em um grupo de obras. Estes padrões ou esquemas possuem um “valor” ou um sentido intrínseco. Verificar se uma figura de retórica existe em um determinado grupo de obras é um indicativo dos afetos envolvidos nas composições estudadas. O

²⁵ O musicólogo Knud Jeppesen, uma das maiores autoridades na obra de Palestrina do século XX utilizou esta metodologia analítica em seus trabalhos.

resultado da pesquisa será quantitativo (número de vezes que aparece determinado evento) mas também qualitativo (o sentido do aparecimento do evento no contexto)

As variáveis de pesquisa que possuímos que necessitam ser tratadas são as seguintes:

1. Contexto (Obras musicais em grande volume), para determinar as características de um determinado estilo musical ou período histórico devemos analisar o maior número possível de obras relacionadas ao nosso objeto de estudo.
2. Formato de dados (Música simbólica) construiremos nossa base de dados utilizando o padrão MusicXML.
3. Objetos de verificação de similaridades (Padrões e esquemas) figuras retórico musicais e esquemas propostos por Gjerdingen(2007) em sua obra Music in The Galant Style.

Possuímos o contexto (as obras musicais), o formato de dados (Musicxml) e os objetos de pesquisa (padrões e esquemas) agora necessitamos do método (como buscar estes padrões neste contexto?) e da ferramenta computacional capaz de realizar o procedimento.

Os projetos descritos neste trabalho como o Music21, o ELVIS, o MelodicMatch e o MusicXML Analyser são projetos experimentais com menos de uma década de existência. Não conseguiremos realizar as análises propostas utilizando estas ferramentas.

Em nossa dissertação de mestrado²⁶, propusemos uma metodologia analítica que iremos adotar e ampliar. Realiza-se basicamente em três etapas:

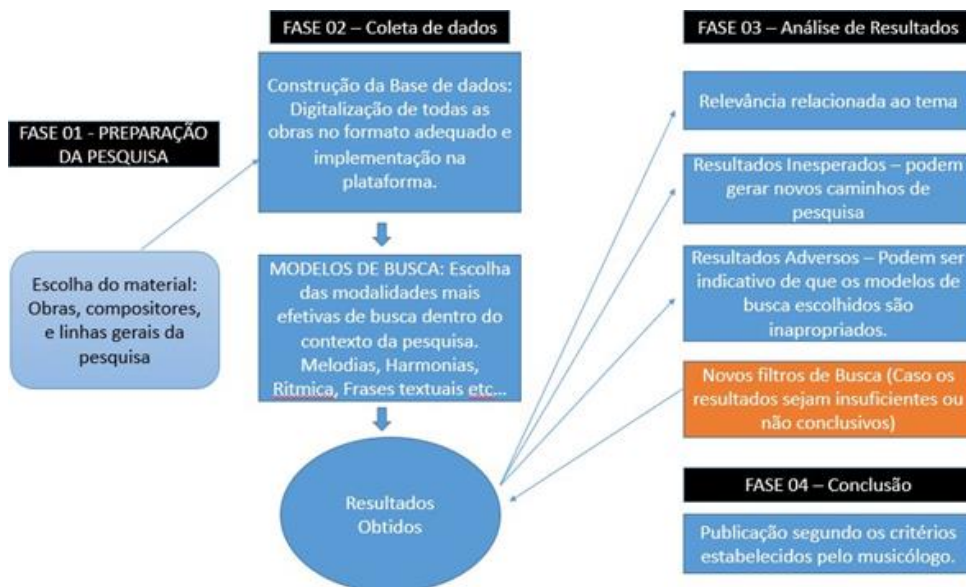
- Observação dos fenômenos
- Descoberta da relação entre eles.
- Generalização das relações.

E em quatro fases:

- Preparação da pesquisa
- Coleta de dados
- Análise de resultados
- Conclusão

²⁶ <https://www.teses.usp.br/teses/disponiveis/27/27157/tde-12012016-112114/publico/ROBSONDIAS.pdf>

Figura 12 - Modelo de Metodologia Analítica



Fonte: Do Autor

A figura 12 representa de forma gráfica todas as etapas do processo analítico proposto. O fluxograma indica as fases iniciais de preparação da pesquisa, coleta de dados, análise dos resultados e a conclusão.

3.1. ANALYSIS PROJECT: UM ORGANIZADOR DO MÉTODO

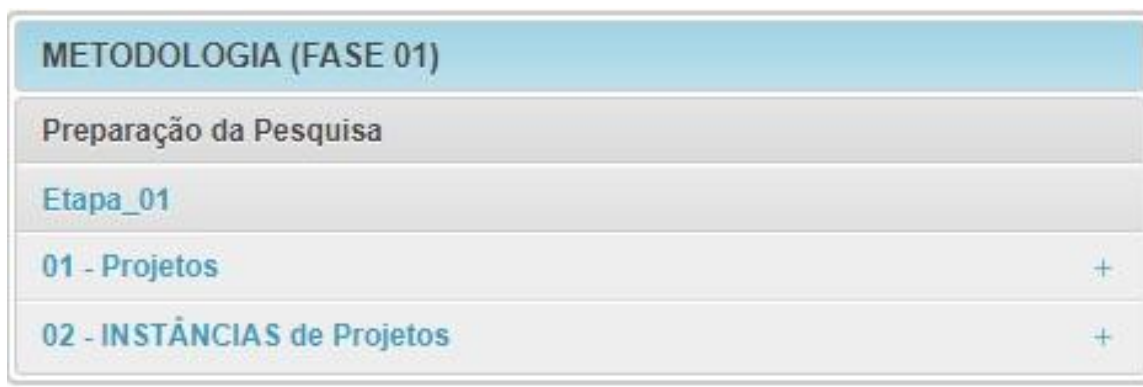
O modelo de análise que propomos foi desenvolvido no intuito de organizar por etapas (fases) a pesquisa musicológica. O foco que anteriormente estava voltado aos objetos de verificação (processo analítico escolhido pelo pesquisador) agora está direcionado ao contexto (corpus onde se realizará a pesquisa). O Analysis Project é uma aplicação que se propõe a conduzir de forma clara o usuário em suas pesquisas. O aplicativo é um artefato construído para validar o método proposto.

A seguir mostraremos na prática a utilização do método, descrevendo as etapas do processo dentro da ferramenta. Apesar da aparência de “manual de utilização” do software, aqui encontraremos uma sugestão de como realizar pesquisas de padrões em grande volume de obras que pode ser adaptada a outras realidades.

3.1.1. Fase 01 - Coleta de Dados

A primeira fase da pesquisa se resume a coleta de dados.

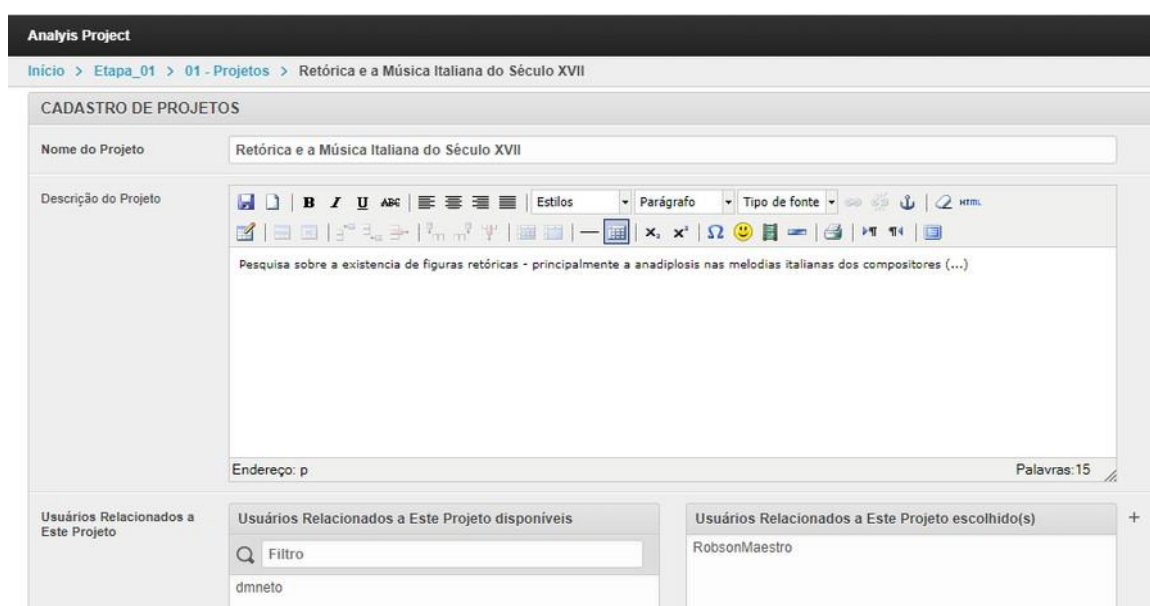
Figura 13 - Metodologia fase 01



Fonte: Do Autor

Na fase de preparação da pesquisa selecionaremos as principais características que vamos utilizar no trabalho. Definimos as características básicas do nosso projeto e suas instâncias²⁷.

Figura 14 - Cadastro de Projetos



Fonte: Do Autor

²⁷ O exemplo que vamos cadastrar é hipotético. Foi construído apenas para ilustrar a metodologia. (Nota do Autor)

Primeiramente definimos os seguintes campos:

Nome do projeto

Descrição (A pesquisa está relacionada a que? Objetivos? Perguntas problemas e tudo aquilo que poderia descrever o objeto de estudo)

Usuários que farão parte do projeto (observe que nesta opção temos a nossa esquerda os pesquisadores que podem participar que são definidos anteriormente nos grupos de pesquisa e a direita os que participarão de maneira ativa do projeto).

Figura 15 - Cadastro de projetos - obras musicais

The screenshot displays a web interface for selecting musical works for a project. It is divided into three main sections:

- Left Panel:** Titled "Selecione as OBRAS vinculadas a este PROJETO". It contains a search filter labeled "Filtro" and a list of available works: BWV 101.7, BWV 103.6, BWV 104.6, BWV 10.7, BWV 110.7, BWV 111.6, BWV 112.5, BWV 113.8, BWV 114.7, BWV 115.6, BWV 11.6, BWV 116.6, and BWV 117.4. A blue link "Escolher todos" is at the bottom.
- Middle Panel:** Titled "Selecione as OBRAS vinculadas a este PROJETO disponíveis". It contains a search filter and a list of available works: BWV 101.7, BWV 103.6, BWV 104.6, BWV 10.7, BWV 110.7, BWV 111.6, BWV 112.5, BWV 113.8, BWV 114.7, BWV 115.6, BWV 11.6, BWV 116.6, and BWV 117.4. A blue link "Escolher todos" is at the bottom.
- Right Panel:** Titled "Selecione as OBRAS vinculadas a este PROJETO escolhido(s)". It contains a list of selected works: BWV 102.7, BWV 108.6, BWV 112.5-SC, and BWV 1.6. A blue link "Remover todos" is at the bottom.

At the bottom of the interface, there is a grey bar with the text "TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO".

Fonte: Do Autor

Finalizando nosso cadastro de projetos possuímos os seguintes campos:

Obras vinculadas a esta pesquisa. (A esquerda temos as obras disponíveis no banco de dados²⁸ e a direita as obras que foram escolhidas). Selecione aqui **todas** as obras que farão parte da pesquisa.

Tratamento sobre a Inclusão do registro (obrigatório)

primeira fase finaliza com as escolhas das instâncias²⁹ do projeto.

²⁸ Caso o musicólogo deseje realizar suas pesquisas em uma base própria ele poderá enviar as músicas para a plataforma. As partituras enviadas podem ser disponibilizada para todos os usuários ou ser classificada como um registro privado. Para maiores esclarecimentos sobre o assunto indicamos o capítulo 6 item 6.1.7.2 deste trabalho onde tratamos detalhadamente sobre o cadastro de obras e privacidade. (Nota do Autor)

²⁹ Substantivo feminino - Qualidade daquilo que é iminente, do que pode acontecer a qualquer momento, do que está prestes a acontecer; iminência. Extraído do dicionário aurélio on-line em <https://www.dicio.com.br/instancia/> - acesso em 13/07/2019.

Figura 16 - Lista das estâncias de um projeto

02 - INSTÂNCIAS de Projetos

3 resultados		6 total	
<input type="checkbox"/>	Nome da Instancia:	^	Selecione o PROJETO vinculado a esta INSTÂNCIA
<input type="checkbox"/>	Obras de Bach		Retórica e a Música do Século XVII
<input type="checkbox"/>	Obras de Haendel		Retórica e a Música do Século XVII
<input type="checkbox"/>	Opera Italiana do período		Retórica e a Música do Século XVII

Fonte: Do Autor

A instância de um projeto é a seleção de determinado material sonoro que ilustra o contexto geral de pesquisa. Para o contexto proposto (Retórica e a música do século XVII) estas três instâncias (Obras de Bach / Obras de Handel e Ópera Italiana do período) ilustram corretamente segundo o critério do musicólogo.

Figura 17 - Cadastro de instâncias de projetos

Analysis Project

[Início](#) > [Etapa_01](#) > [02 - INSTÂNCIAS de Projetos](#) > [Obras de Bach](#)

Modificar 02 - INSTÂNCIAS de Projetos

CADASTRO DE INSTANCIAS DE PROJETOS

Nome da Instancia:

Descrição da Instancia do Projeto:

Estilos Parágrafo Tipo de fonte

Utilizar o compositor por ser um ícone da música do período

Endereço: p Palavras:11

Selecione o PROJETO vinculado a esta INSTÂNCIA:

Mecanismo de Busca: ←

TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO

Fonte: Do Autor

Podemos cadastrar um número ilimitado de instâncias para um projeto. O Cadastro de instâncias de projeto possui as seguintes opções:

Nome da instância

Descrição

Projeto vinculado

Mecanismo de busca (O usuário decide entre padrões do Gjerdingen / figuras de retórica / ou padrão que ele vai criar).

Tratamento sobre a Inclusão do registro (obrigatório)

O método aqui proposto define como primeira fase de pesquisa os seguintes passos:

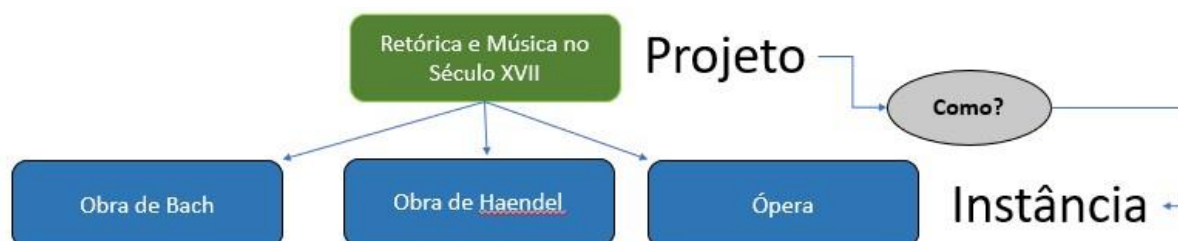
Tabela 1 - Passos a serem realizados na fase 01

Passos	Ação
01	Definir o nome do projeto
02	Descrever o projeto (O que ele propõe?)
03	Selecionar os musicólogos participantes
04	Selecionar as obras
05	Definir a Instância (a partir de agora vai acontecer isto)
06	Definir a modalidade de busca (figura de retórica / padrão do Gjerdingen / padrão do usuário)

Fonte: Do Autor

A tabela 1 sintetiza os todos passos necessários para a realização da primeira fase de um projeto.

Figura 18 - Representação visual da primeira fase de pesquisa



Fonte: Do Autor

Na figura 18 temos a representação gráfica da primeira fase de um projeto. As instancias de um projeto é uma ilustração e devem responder à questão: Como? No exemplo pretende-se realizar uma pesquisa sobre Retórica e música no século XVII Iremos buscar estas informações nas obras de Bach, Handel e óperas do período. Nesta fase coletamos os dados necessários para a realização do projeto e definimos a direção que irá seguir.

3.1.2. Fase 02 - Preparação da Pesquisa

A segunda fase é a de preparação do material para a realização da pesquisa. Selecionar os melhores padrões de busca e as especificidades de cada instancia deve ser realizado agora.

Figura 19 - Metodologia fase 02

METODOLOGIA (FASE 02)	
Coleta de Dados	
Etapa_02	
01 - MÓDULOS	+
02 - CATEGORIAS	+

Fonte: Do Autor

A figura 19 representa a segunda fase da pesquisa prevê duas etapas:

- Escolhas de Módulos
- Definição das Categorias de pesquisa

Figura 20 - Lista parcial dos módulos e associações com as instâncias

Analysis Project

Início > Etapa_02 > 01 - MÓDULOS

01 - MÓDULOS

4 total

<input type="checkbox"/>	Módulo:(O QUE vamos verificar da instância?)	2 ^	Selecione a Instancia:	1 ^
<input type="checkbox"/>	Cravo bem temperado		Obras de Bach	
<input type="checkbox"/>	Partitas		Obras de Bach	
<input type="checkbox"/>	Quarteto de Cordas		Obras de Haendel	
<input type="checkbox"/>	óperas de Giovanni Legrenzi		Opera Italiana do período	

Fonte: Do Autor

Podemos visualizar na figura 20 os módulos que são as especificações do conteúdo a ser pesquisado em cada instancia. É possível observar que na instância “obras de Bach” iremos pesquisar somente as obras: O cravo bem temperado e as partitas.

Figura 21 - Cadastro de módulos

Analysis Project

Início > Etapa_02 > 01 - MÓDULOS > óperas de Giovanni Legrenzi


Modificar 01 - MÓDULOS

MÓDULOS

Módulo:(O QUE vamos verificar da instância?)

Selecione a Instancia → ✕

Observações Relativas a módulo



Foi um dos maiores compositores italianos do seu tempo.

Endereço: p Palavras:9

TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO

Fonte: Do Autor

A figura 21 nos mostra a tela de cadastro de módulos. Podemos cadastrar um número ilimitado para cada instancia de projeto. O Cadastro de módulos possui as seguintes opções:

- Módulo (O que irá ser verificado na instância)
- Seleção da instância
- Descrição
- Tratamento sobre a inclusão do registro (obrigatório)

A segunda fase finaliza-se com o cadastro das categorias de pesquisa. As ferramentas de busca serão especificadas (figuras e padrões) bem como a inserção de padrões do usuário.

Figura 22 - Cadastro de categorias

Fonte: Do Autor

A figura 22 representa a tela inicial de cadastro de categorias. Cada módulo pode receber um número ilimitado de categorias. As opções de cadastro de categorias são:

- Seleção do módulo relacionado
- Nome da categoria
- Observações

- Padrões escolhidos para busca (Observe que no caso definimos que nossa instância trabalharia com figuras de retórica lá na primeira fase, assim, a seleção possível será somente com figuras de retórica ³⁰).
- Seleção das obras específicas (No cadastro do PROJETO o musicólogo informou todas as obras que irá utilizar em sua pesquisa - No cadastro de CATEGORIAS ele irá informar dentro do corpus anteriormente selecionado qual a obra necessária para satisfazer este tópico de pesquisa).
- Tratamento sobre a inclusão do registro (obrigatório)

Figura 23 - Categoria com a opção de padrões do Gjerdingen



Fonte: Do Autor

A figura 23 demonstra as opções que o usuário encontraria se desejasse realizar uma pesquisa com os padrões do Gjerdingen. A pesquisa que iremos realizar como modelo de demonstração da utilização do Analysis Project neste capítulo não utilizará estes padrões.

Caso o usuário tenha configurado em sua instância que utilizará padrão próprio, ao cadastrar a categoria ele deverá enviar o padrão no formato MusicXML e escolher como ele irá se comportar em relação às estruturas musicais.

³⁰ Caso o musicólogo deseje trabalhar no mesmo contexto com um padrões diferenciados (Gjerdingen ou criado por ele) deve criar uma nova instância. A medida que as necessidades vão surgindo vamos adicionando possibilidades ao nosso projeto sem perder os resultados anteriores.

Figura 24 - Cadastro de categorias com a opção de padrão do usuário

Analysis Project

Início > Etapa_02 > 02 - CATEGORIAS > Pesquisa de figuras que afetam a melodia de forma descendente

CATEGORIAS

CATEGORIA RELACIONADA AO MODULO: Cravo bem temperado

NOME DA CATEGORIA: Pesquisa de figuras que afetam a melodia de forma descendente

OBSERVAÇÕES: As figuras que afetam a melodia de forma descendente podem traduzir um sentido de melancolia segundo Matterson

Endereço: p Palavras: 17

PADRÕES ESCOLHIDOS PARA ESTA PESQUISA

Fazer o Upload do Pattern utilizado Escolher arquivo Nenhum arquivo selecionado

Fonte: Do Autor

A figura 24 demonstra a opção onde o usuário poderá realizar o upload do arquivo para a base de dados do Analysis Project. O musicólogo deve escolher as opções de transformação relacionadas ao seu padrão:

Figura 25 - Cadastro de categorias com opção de padrão do usuário 02

PADRÕES ESCOLHIDOS PARA ESTA PESQUISA

Fazer o Upload do Pattern utilizado Escolher arquivo Nenhum arquivo selecionado

Escolha o MATERIAL UTILIZADO na Pesquisa

Escolha o MATERIAL UTILIZADO na Pesquisa: disponíveis

Filtro

Arquivos
Escrita Manual de Intervalos
Linguagem poética (letra das obras)

Fonte: Do Autor

A figura 25 nos mostra as opções de escolha de Material Utilizado nas pesquisas que são:

- Refinamentos
- Transformações melódicas
- Transformações rítmicas
- Transposições

- Tipografia (caso para pesquisas de letras das músicas)

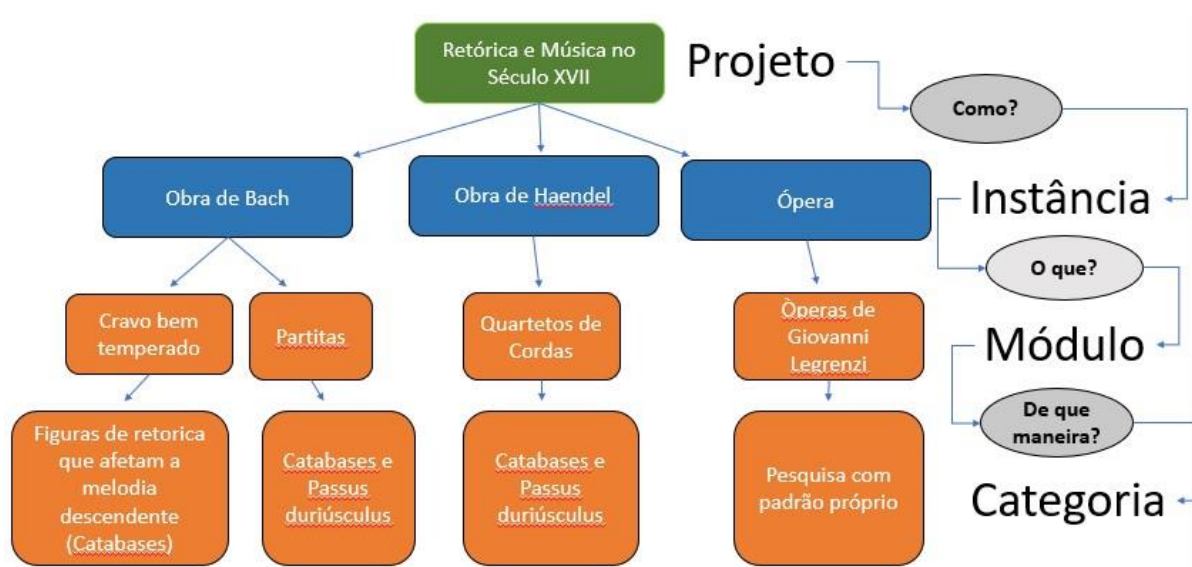
Tabela 2 - Passos a serem realizados na fase 02

Passos	Ação
01	Cadastrar os Módulos
02	Descrever a necessidade de cada módulo
03	Associar os módulos as categorias
04	Definir o material de busca (Selecionar figuras de retórica / padrões do Gjerdingen / padrão do usuário)
05	Selecionar as obras específicas para a categoria

Fonte: Do Autor

A tabela 2 descreve os passos a serem realizados na segunda fase da pesquisa. Nesta fase preparamos todo o material para a realização das buscas. O Objetivo inicial do projeto define o que deverá ser realizado.

Figura 26 - Representação visual da primeira e segunda fase de pesquisa



Fonte: Do Autor

A figura 26 ilustra graficamente a primeira e segunda fase da pesquisa. Definida as instancias que iremos trabalhar, é necessário selecionar que obras iremos pesquisar. Da instancia “obras de Bach” iremos pesquisar o que? Obras completas? Os corais? As partitas? Definimos assim nossos módulos. A partir deste momento precisamos definir nossas categorias de pesquisa, ou seja, de que maneira vamos obter os dados que precisamos: Da **instancia** (Obras de Bach) vamos pesquisar no **módulo** (O cravo bem temperado) a **categoria** (Figura de retórica - Catabase).

Realizada estas duas etapas da pesquisa estamos prontos para coleta dos dados que serão analisados.

3.1.3. Fase 03 - Realização das buscas e análise dos resultados

Na terceira fase da pesquisa vamos utilizar o Analysis Project para realizar as buscas dos padrões escolhidos nas obras.

Figura 27 - Metodologia fase 03

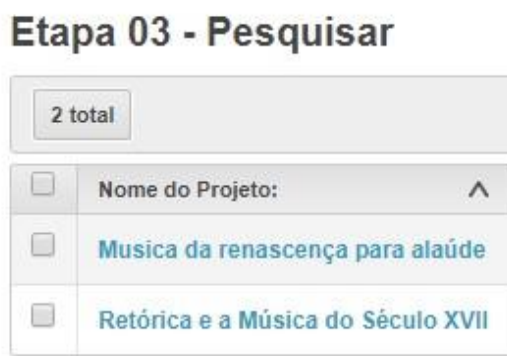
METODOLOGIA (FASE 03)	
Análise dos Resultados Obtidos	
Etapa_03	
01 - PESQUISAR	+
02 - VERIFICAÇÃO DOS RESULTADOS	+

Fonte: Do Autor

A figura número 27 ilustra as opções que iremos encontrar na terceira fase da nossa pesquisa que está dividida dois tópicos:

- Pesquisa (Busca dos padrões nas obras escolhidas)
- Verificação dos dados. (Análise dos resultados obtidos)

Figura 28 - Lista dos projetos a pesquisar



Fonte: Do Autor

A figura 28 é uma listagem dos projetos ativos para o usuário. O Analysis Project permite que um usuário possa manter um número ilimitado de projetos.

Figura 29 - Realizar a pesquisa

← → ↻ 🏠 ⓘ 127.0.0.1:8000/pesquisar/

Metodologia fase 03/01 - Realizar a pesquisa.
Usuário: RobsonMaestro

PROJETO	INSTANCIA	MÓDULO	CATEGORIA	PESQUISAR	CONCLUÍDO
Retórica e a Música do Século XVII	OBRAS DE BACH	Cravo Bem Temperado	Figuras de retórica que afetam a melodia descendente	PESQUISAR	OK
		Partitas	Catábases e Passus duriúsculus	PESQUISAR	
	OBRAS DE HAENDEL	Quartetos de cordas	Catábases e Passus duriúsculus	PESQUISAR	
	ÓPERA DO PERÍODO	Óperas de Giovanni Legrenzi	Pesquisa com padrão próprio	PESQUISAR	

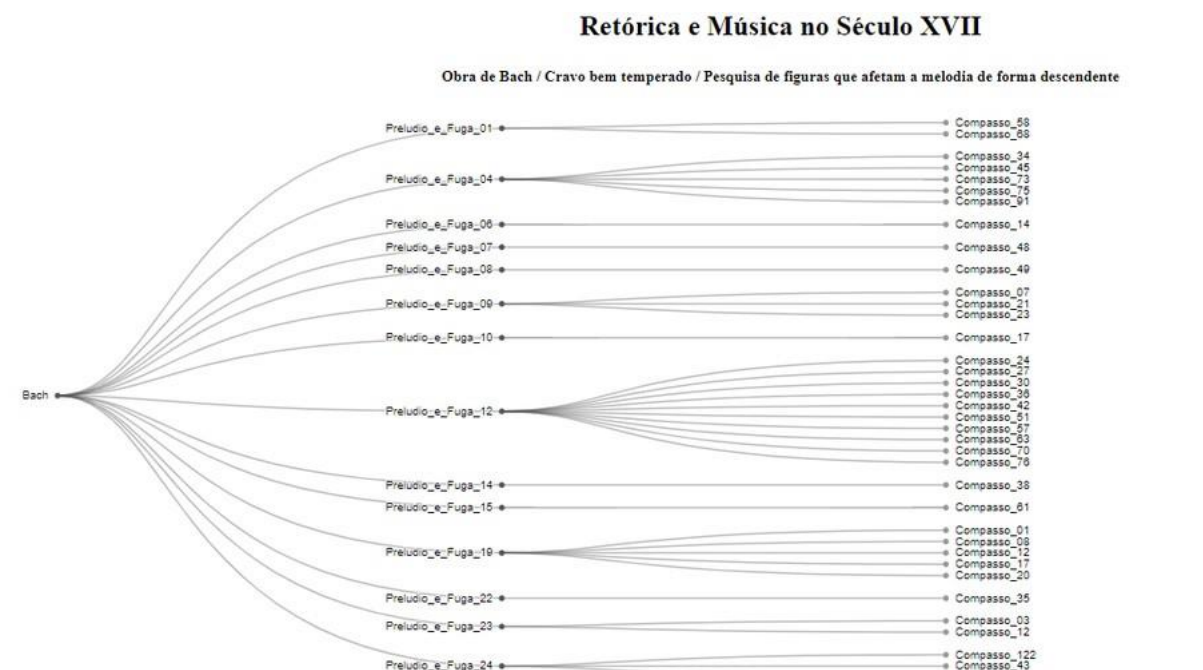
Copyright © 2013–20 todos os direitos reservados.

Fonte: Do Autor

Ao acessar o projeto escolhido, o usuário visualizará uma tabela (figura 29) com o projeto, as instancias, os módulo e as categorias. O link pesquisar é o responsável por realizar as buscas com os padrões selecionados anteriormente na opção

Categoria de pesquisa. Assim que finalizar aparece o hiperlink (em vermelho) com a expressão “Ok”. Clicando conseguirá visualizar os resultados³¹.

Figura 30 - Visualização em forma de dendograma



Fonte: Do Autor

A figura 30 mostra uma visualização computacional em forma de dendograma. A **instancia** são as obras de Bach, os **módulos** são as obras pesquisadas e a **categoria** são as figuras de retórica que afetam a melodia de forma descendente. Observe que com esta visualização podemos rapidamente perceber que no prelúdio e fuga número 10 temos o maior número de ocorrências desta categoria (12 ocorrências) em contraposição a outras obras que aparece somente uma vez. Estes dados dispostos desta maneira são importantes para o pesquisador que, a uma primeira vista, pode concluir que determinada obra tem maior ou menor possibilidade de validar sua pesquisa.

A verificação dos resultados deve ser realizada de forma manual. Não foi implementado no Analysis Project a visualização da partitura musical com os resultados podendo ser uma possibilidade de implementação para a próxima versão.

³¹ A visualização de resultados será discutida em capítulo posterior. (Nota do Autor)

Figura 31 - Exemplo de verificação de registro realizada

PRAELUDIUM IX.

Fonte: Das wohltemperierte Clavier - Edições Heinle pag.36

A figura 31 mostra um exemplo de similaridade encontrado pelo Analysis Project. As impressões do pesquisador quanto aos resultados obtidos devem ser cadastradas na plataforma. Estas impressões são impressas na conclusão.

Figura 32 - Listagem dos projetos a serem verificados

Etapa 03 - Verificação dos resultados

2 total		
<input type="checkbox"/>	Nome do Projeto: ^	Data da Última Atualização
<input type="checkbox"/>	Musica da renascença para alaúde	25 de Fevereiro de
<input type="checkbox"/>	Retórica e a Música do Século XVII	25 de Fevereiro de

Fonte: Do Autor

A figura 32 é uma listagem dos projetos a serem verificados. Isto indica para o usuário que determinado projeto possui registros não tratados, impossibilitando a geração das conclusões (Etapa 4)

Figura 33 - Verificação dos resultados

The screenshot shows a web application interface for adding research results. At the top, there is a breadcrumb trail: 'Início > Etapa_03 > 02 - VERIFICAÇÃO DOS RESULTADOS > Adicionar 02 - VERIFICAÇÃO DOS RESULTADOS'. The main title is 'Adicionar 02 - VERIFICAÇÃO DOS RESULTADOS'. Below this, the interface is titled 'VERIFICAÇÃO DOS RESULTADOS DA PESQUISA'. It features several form elements: 'SELECIONE A CATEGORIA' with a dropdown menu showing 'Pesquisa de figuras que afetam a', 'SELECIONE O RESULTADO' with a dropdown menu showing 'Pretudio e fuga 01 - compasso 58', and 'Relevancia deste Resultado' with a dropdown menu showing 'Baixa'. There is a section for 'Observações das Verificações' with a rich text editor containing the text: 'Este resultado não é um passus durissimus e sim apenas uma progressão cromática de quatro notas em descendência.' Below the editor, there are fields for 'Endereço: p' and 'Palavras: 17'. At the bottom, there are two dropdown menus: 'Resultado Verificado?' with 'Sim' selected, and 'Imprime este Resultado na Conclusao?' with 'Não' selected. The footer of the interface reads 'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO'.

Fonte: Do Autor

A figura 33 mostra as opções de verificação dos resultados:

- Selecione a categoria
- Selecione o resultado
- Relevância do resultado (Nenhuma / Baixa / Média / Alta / Resultado inesperado)
- Observações sobre este resultado
- Resultado verificado (Sim / Não)
- Imprime este resultado na conclusão (Sim / Não)
- Tratamento sobre a inclusão do registro (obrigatório)

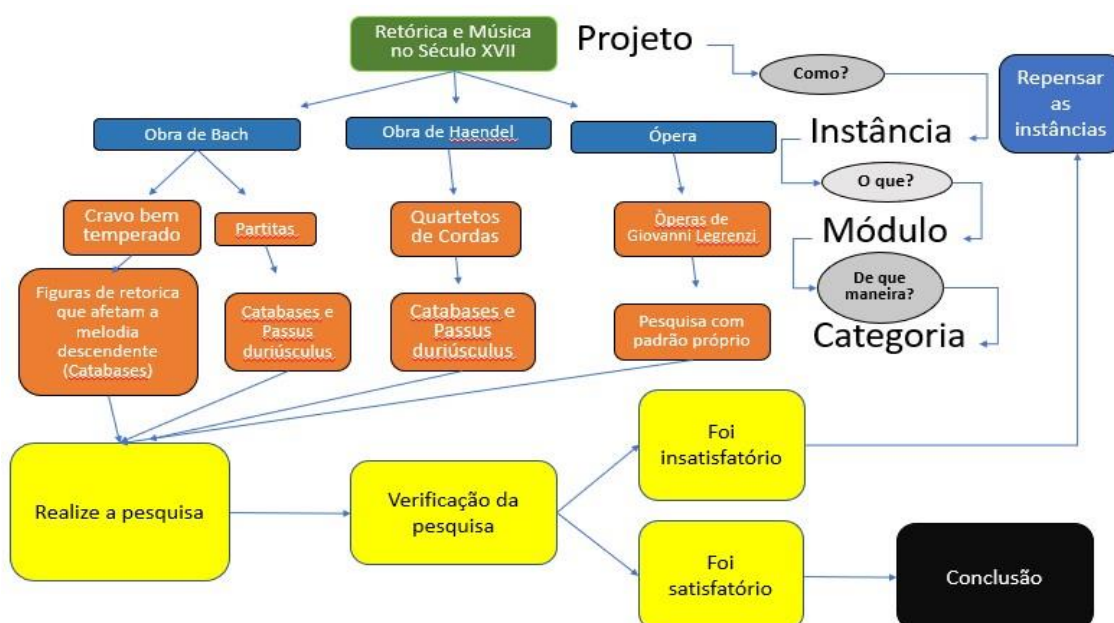
Tabela 3 - Passos a serem realizados na etapa 03

Passos	Ações
01	Realizar a pesquisa no aplicativo
02	Analisar os dados coletados
03	Se os resultados foram satisfatórios concluir
04	Se os dados forem insuficientes retornar a pesquisa para o tópico “instancias” e reformular.

Fonte: Do Autor

A tabela 3 descreve os passos a serem realizados na fase 03 do projeto. Esta etapa indicará os rumos que o pesquisador deve seguir. Caso os dados coletados não satisfaçam aos objetivos propostos no projeto, devemos reconsiderar as *instâncias* propostas.

Figura 34 - Representação visual da metodologia proposta pelo Analysis Project



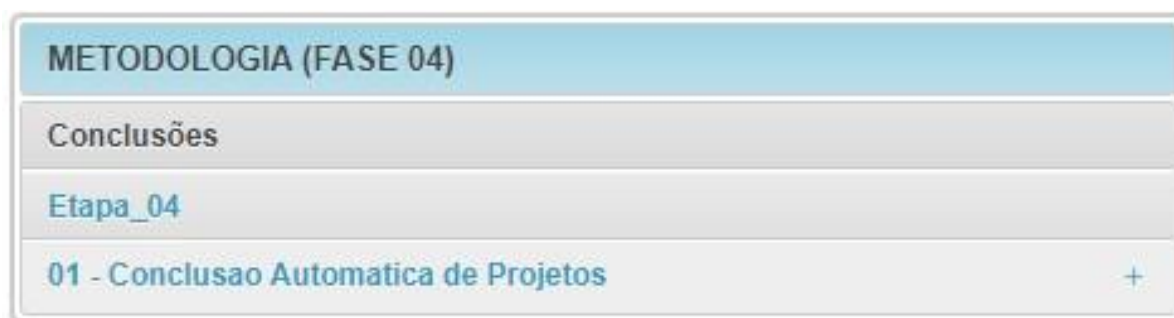
Fonte: Do Autor

A figura 34 é um resumo de toda a metodologia proposta nesta tese. Se para o musicólogo, os resultados obtidos são satisfatórios, deve passar para a última gerando as conclusões.

3.1.4. Fase 04 – Conclusões

A fase final da pesquisa, a aplicação vai gerar automaticamente uma conclusão. Todos os dados coletados em todas as fases do processo serão organizados e poderão ser visualizados pelo usuário

Figura 35 - Metodologia fase 04



Fonte: Do Autor

A figura 35 mostra o acesso do usuário aos seus projetos.

Figura 36 - Lista de projetos concluídos



Fonte: Do Autor

O projeto somente estará disponível para gerar a conclusão se todos os registros forem verificados pelo pesquisador. A figura 36 mostra os projetos finalizados pelo usuário.

Figura 37 - Extrato parcial de um modelo de conclusão de projeto.



Fonte: Do Autor

A figura 37 mostra um exemplo parcial de uma conclusão gerada pelo Analysis Project. Todos os dados que estão entre colchetes(ex: [Projeto Retórica e a Música do Século XVII], [RobsonMaestro], etc...) foram inserções que o usuário realizou no banco de dados no período em que a pesquisa esteve ativa. Na conclusão estes dados são rerepresentados para que possa avaliar se os seus objetivos foram alcançados. Em qualquer momento da pesquisa, inclusive depois de finalizada ela pode ser modificada ou reaproveitada em outros contextos.

O Analysis Project é um aplicativo que realiza duas funções:

1. Buscar padrões e esquemas em um corpus de obras definido pelo usuário
2. Organiza a pesquisa realizada. (Validação do método)

4. PADRÕES E ESQUEMAS PREDEFINIDOS NO ANALYSIS PROJECT

Para a realização de pesquisas de música simbólica com recursos tecnológicos precisamos definir padrões de busca.

O Analysis Project possui duas linhas de pensamento relativo a padrões e esquemas: Figuras de retórica e alguns padrões definidos pelo pesquisador Robert Gjerdingen³².

4.1. “Music in the Galant Style” - Robert Gjerdingen

O termo “Galant” foi utilizado por Gjerdingen para descrever um estilo musical do século XVIII de característica “cortês”. Criando uma visualização poética imaginemos jovens aristocratas conversando e dançando um minueto. O termo “Galant” como barroco, clássico ou romântico são frutos da racionalização ocorrida no século XX emprestadas dos livros de história.

A abordagem de Gjerdingen baseia-se em estudos estilísticos seguindo escolas de pesquisa musicológica onde características comuns nas obras são consideradas determinantes de um estilo.

Nos capítulos do livro Gjerdingen apresenta um esquema (um estereótipo) composicional padrão para o estilo “Galant”³³. Os padrões seriam uma “forma de etiqueta” que o compositor podia (ou não) seguir, dependendo da ocasião e de suas intenções.

4.1.1. Romanesca e um exemplo de lógica de programação

O primeiro exemplo de Gjerdingen foi a *romanesca*. Vamos sugerir neste subcapítulo uma “lógica” que pode ser seguida para construirmos um algoritmo que traduza (em linguagem de programação) de forma eficiente o que seria este esquema.

³² Neste capítulo vamos realizar uma abordagem superficial da obra de Robert Gjerdingen (*Music in the galant style*) e na obra de Rubén Cano (*Música y retórica en el Barroco*). Figuras, padrões e esquemas serão apenas ilustrados. Abordaremos o que for essencial para a construção do Analysis Project sugerindo um aprofundamento das referidas obras.

³³ Gjerdingen tem a preocupação sempre de contextualizar seus padrões como formas referenciais para determinado período histórico.

Figura 38 - Exemplo de Romanesca

ROMANESCA

The image shows a musical score for a piece titled 'ROMANESCA'. It is written in 12/8 time and consists of two staves: a treble clef (top) and a bass clef (bottom). The key signature has two flats (B-flat and E-flat). The score is divided into three measures. The treble staff contains a melody with notes and rests, with circled numbers 1 through 7 indicating fingerings. The bass staff contains a bass line with notes and rests, with circled numbers 1 through 5 indicating fingerings. The entire piece is enclosed in a large bracket.

Fonte: Music in the Galant Style - Robert Gjerdingen pag28

Passo 01 - Selecionar as vozes extremas (agudo e grave)

Passo 02 - Olhar armadura de clave e estabelecer o tom (Caso acima Si bemol).

Este procedimento estabelece as notas que devemos procurar.

Tabela 4 - Graus da escala X notas correspondentes

Graus da Escala	Notas Correspondentes
01	Sib
02	Dó
03	Ré
04	Mib
05	Fá
06	Sol
07	Lá

Fonte: Do Autor

Passo 03 - Precisamos encontrar seis grupos de ocorrências (em sequência). Cada grupo possui duas notas. Buscamos o primeiro grupo que seriam as notas ré na voz mais aguda e Sib na mais grave³⁴. Ao encontrar passamos para o segundo grupo e assim por diante.

Tabela 5 - Estrutura de grupos da Romanesca

Grupos intervalares Notas correspondentes	
3 - 1	Ré - Sib
2 - 5	Dó - Fá
1 - 6	Sib - Sol
7 - 3	La - Ré
6 - 4	Sol - Mib
5 - 1	Fá - Sib

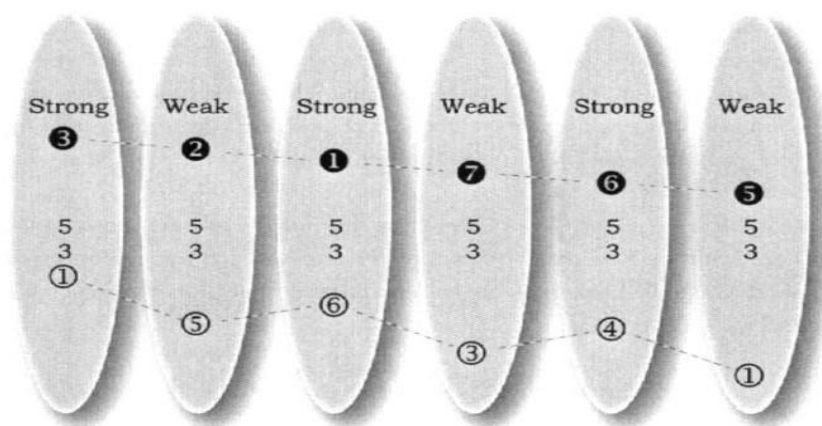
Fonte: Do Autor

Caso encontremos os grupos de notas nesta sequência teríamos um positivo do esquema Romanesca.

Segundo Gjerdingen a Romanesca tem diversas variações.

³⁴ O arquivo MusicXML possui uma formatação com tags muito específicas para cada acontecimento musical, assim fica relativamente simples encontrar um evento musical em seu corpo. No capítulo 6 deste trabalho descrevo cada uma detalhadamente. (Nota do autor)

Figura 39 - Um esquema da Romanesca com saltos no baixo



Fonte: Music in the Galant Style - Robert Gjerdingen pag.29

O exemplo aqui proposto seria de uma Romanesca com saltos no baixo. Cada estrutura ou variação da Romanesca terá de ser tratada de forma individualizada em nossa base de dados, contudo, a lógica de programação será a mesma.

4.1.2. Clausulae

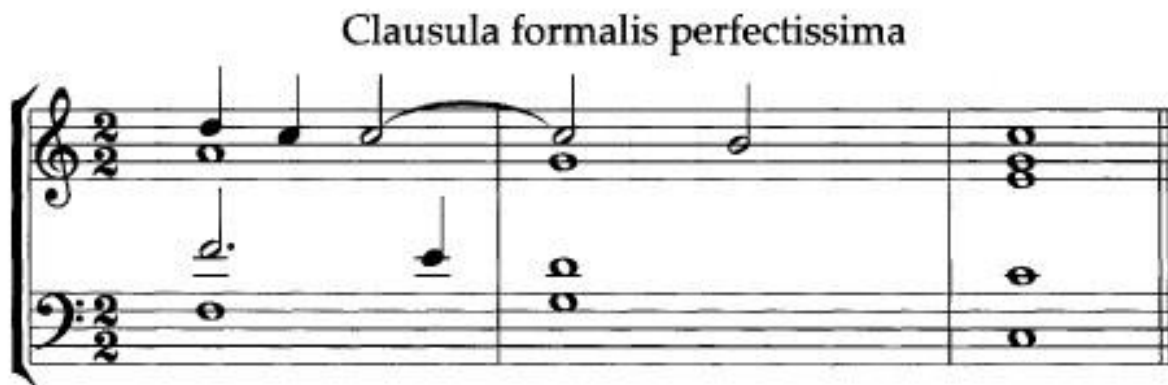
A Clausulae é um esquema cadencial de fechamento das obras musicais.

O TERMO LATINO ESCOLHIDO PELOS ESCRITORES MEDIEVAIS para descrever o sentido percebido de fechamento e finalização provocada por certas figuras melódicas foi a cláusula (pi. clausulae; fim, conclusão ou fim) .1 Com o tempo, esse sentido foi transferido para fórmulas envolvendo duas vozes contrapontísticas e, muito mais tarde, sucessões fórmulas de acordes multivoice. O exemplo abaixo mostra uma fórmula de quatro vozes que Johann Gottfried.

Walther (1684-1748), organista de Weimar, maestro do jovem príncipe Johann Ernst e amigo de J. S. Bach, descrito em 1708 como uma cláusula formalis perfectissima. 2 - O que é comumente descrito hoje como uma “cadência autêntica perfeita”. Perfectissima convida a tradução como “a mais perfeita”, mas o significado pretendido estava mais próximo do “mais completo”, referindo-se ao grau de fechamento. Conseqüentemente o título do exemplo pode ser renderizado como “um fechamento da forma mais completa³⁵.”(Gjerdingen, R. pag139)

³⁵ Tradução do Autor

Figura 40 - Exemplo de Cláusula



Fonte: Music in the Galant Style - Robert Gjerdingen pag.139

Este movimento cadencial é estudado largamente nos livros de harmonia tradicionais do século XX, contudo, dentro do estilo Galant devemos nos atentar para os movimentos melódicos.

4.1.3. Dó - Ré - Mi

Este padrão é um padrão simples mais muito utilizado no estilo Galant. N

Figura 41 - Padrão Dó - Ré - Mi na voz aguda



Fonte: Music in the Galant Style - Robert Gjerdingen pag.77

Na figura acima o padrão é acompanhado pela progressão 1 - 7 - 1 no baixo.

Figura 42 - Exemplo do padrão Dó - Ré - Mi

DO-RE-MI

Fonte: Music in Galant Style - Robert Gjerdingen

O exemplo acima é exatamente o contrário do anterior. O padrão Dó - Ré - Mi está no baixo e a progressão 1 - 7 - 1 na linha mais aguda.

4.1.4. Prinner

O Prinner seria uma “resposta elegante”. Abaixo um exemplo desta figura musical:

Figura 43 - Exemplo do esquema Prinner

PRINNER

(6 5 4 3)

Partitura

(4 3 2 1)

Fonte: Music in the Galant Style - Robert Gjerdingen Pag.46

Segundo o autor, uma abertura como a romanesca pede uma resposta elegante como um Prinner.

Figura 44 - Locatelli, Opus 6, no. 11, mvt. i, adagio, m. 16 (Amsterdam, 1737)

Fonte: Música in Galant Style - Robert Gjerdingen pag.49

O autor oferece vários exemplos onde demonstra a proximidade destes dois esquemas.

4.1.5. Fenaroli

Fenaroli foi um importante teórico. Sobre ele Gjerdingen escreveu:

Fenaroli escreveu o famosa coleção de partimenti que, depois de 1790, foi usada por gerações de italianos e músicos franceses. Já em 1871, os partimenti de Fenaroli receberam elogios de não menos figura que Giuseppe Verdi. Fenaroli, portanto, serviu de ponte entre os séculos XVIII e séculos XIX, transmitindo o repertório napolitano de esquemas galantes. Reconhecer seu papel na inculcação e transmissão dessa importante tradição.(Gjerdingen, 2007).

Figura 45 - Durante, Studio no. i, mvt. i, m. i (ca. 1747)

The image shows a musical score for two sections: 'DO-RE-MI' and 'FENAROLI'. The score is written for a single melodic line in a treble clef, with a bass line below it. The key signature has one flat (B-flat) and the time signature is 6/8. The 'DO-RE-MI' section consists of three notes: D, E, and F, each with a trill (tr) and a fingering of 1, 2, and 3 respectively. The 'FENAROLI' section consists of seven notes: F, E, D, C, B, A, and G, with fingerings 5, 4, 3, 1, 7, and 1 respectively. The bass line has fingerings 1, 7, 1, 7, 1, 2, and 3 for the corresponding notes.

Fonte: Music in the Galant Style - Robert Gjerdingen pag. 225

A figura 45 mostra um esquema nomeado por Gjerdingen em sua homenagem.

4.1.6. Fonte

Fonte é um esquema que tem basicamente duas seções:

Figura 46 - Tritto, a Fonte bass (ca. 17905)

The image shows a musical score for 'Fonte bass' in a bass clef, with a 2/2 time signature. The score is divided into two sections: 'minor' and 'major'. The 'minor' section consists of six notes: G, F, E, D, C, and B, with a fingering of 7, 7, 1, 7, 7, and 1 respectively. The 'major' section consists of six notes: C, D, E, F, G, and A, with a fingering of 7, 7, 1, 7, 7, and 1 respectively. The key signature changes from one flat (B-flat) to no flats (C major) between the two sections.

Fonte: Music in the Galant Style - Robert Gjerdingen

A primeira em modo menor (Ré menor) e a segunda em modo maior (Dó maior). Observe que as tonalidades possuem a distância de um tom.

4.1.7. Indugio

Indugio é uma cadência galante típica: um acorde de 6/5 acima no baixo, um sétima de dominante finalizando com uma tríade.

Figura 47 - Daube's "three chords" (1756) set as a cadence

Fonte: Music in the Galant Style - Robert Gjerdingen pag.273

O autor chama de "Indugio" um esquema que "sinaliza uma brincadeira, que atrasa a chegada de uma cadência".

4.1.8. Ponte

Segundo Gjerdingen "Ponte pode ser caracterizada como uma extensão e um foco no segundo tipo de sonoridade" (Gjerdingen, 2007).

Figura 48 - Riepel's prototype of the Ponte melody with a likely bass

Fonte: Music in Galant Style - Robert Gjerdingen pag.198

A figura 48 demonstra um esquema de ponte com uma sugestão de baixos.

4.1.9. Quiescenza

Um exemplo de Quiescenza encontramos a linha do soprano realizando o movimento melódico (7 - 6 - 7 - 1)

Figura 49 - Gossec, Missa pro defunctis, mvt. 15, [Andante], m. 26 (Paris, 1760)

The image shows a musical score for a soprano part. The key signature has one sharp (F#) and the time signature is 3/4. The score is divided into two sections, each labeled 'QUIESCENZA' with a bracket above. The soprano line features a melodic sequence of notes: G4 (labeled 7), F#4 (labeled 6), G4 (labeled 7), and A4 (labeled 1). The bass line consists of a steady eighth-note pattern. There are various musical ornaments and dynamics throughout the piece.

Fonte: Music in Galant Style - Robert Gjerdingen pag.184

Na figura 49 ilustra um exemplo onde a tônica é um baixo pedal.

4.2. Figuras retórico-musicais.

A retórica é um sistema de regras com o objetivo de organizar o discurso. Procura em sua raiz unir o logos (razão, lógica argumentativa), ethos (sentido de si) e pathos (emoção e paixão). Cícero³⁶ afirmava que o orador completo deveria estar apto a unir estas três virtudes e ter discernimento.

As figuras retórico-musicais³⁷ são ferramentas utilizadas pelo compositor para convencer o ouvinte da coerência do seu discurso.

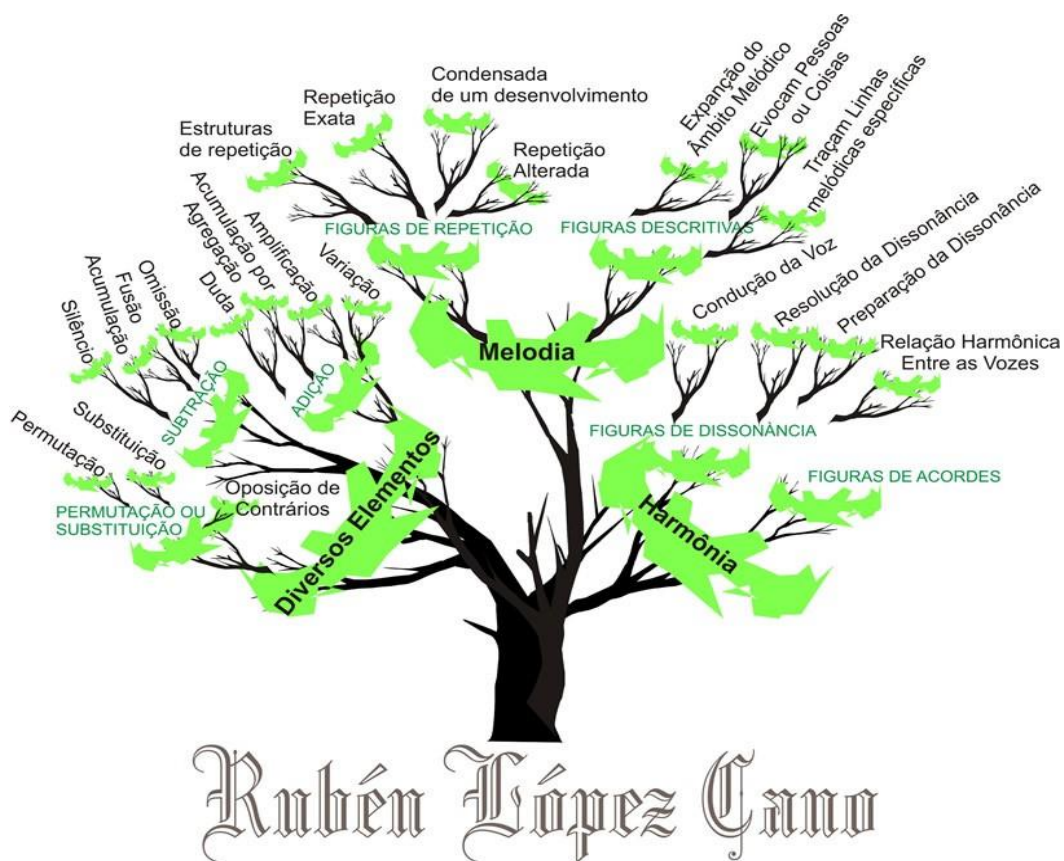
O Analysis Project desenvolveu alguns algoritmos para identificar figuras retóricas dentro do discurso. Como uma boa parte das figuras requer interpretação e significado, as ferramentas computacionais que trabalham com música simbólica devem ser utilizadas com cautela o que demonstraremos posteriormente.

³⁶ A listagem dos autores (referências primárias e secundárias) cadastradas no Analysis Project se encontra no APENDICE D. (Nota do autor)

³⁷ A lista das figuras retórico-musicais cadastradas no banco de dados do Analysis Project se encontra no APÊNDICE C. (Nota do Autor)

Rubén Cano, em seu trabalho *Música y retórica en el barroco* propôs um sistema para classificar as figuras retórico musicais.

Figura 50 - Classificação das Figuras de Retórica



Fonte: Do autor

Na figura 50 temos a classificação das figuras de retórica proposta por Cano(2000) divididas em classes e subclasses. Abaixo vou nominar as figuras existentes em cada subclasse apresentada. Esta classificação será a mesma encontrada no Analysis Project.

4.2.1. Figuras que afetam a melodia

A classificação das figuras já aponta para seu uso dispensando assim comentários mais extensos. Figuras que afetam a melodia são divididas em classes e subclasses a saber:

Figuras de repetição

Estruturas de repetição

Anaphora

Epistrophe

Anadiplosis

Epizeuxis

Complexio

Epanalepsis

Epanadiplosis

Repetição exata

* *Palilogia*

Condensada de um desenvolvimento

Percursio

Distributio Recaptulativa

Synecdoche-Metonymie

Repetição alterada

* *Traductio*

Figuras Descritivas

Expansão do âmbito melódico

Exclamatio

Ecphonesis

Hyperbole

Hypobole

Evocam pessoas ou coisas

Prosopopoeia

Assimilatio

Apostrophe

Paragoge

Traçam linhas melódicas específicas

Anabasis

Catabasis

Circulatio

Fuga Hypotiposis

Interrogatio

4.2.2. Figuras que afetam a harmonia

Figuras de dissonância

Condução de voz

Anticipatio Notae

Passus Duriusculus

Saltus Duriusculus

Superjectio

Subsumptio

Subsumptio Postpositiva M.

Syncope

*Transitus Notae * Quasi Transitus*

Transitus Inversus

Resolução da dissonância

Catachresis

Prolongatio

Symblema

Carachresica Syncopatio

Syncopatio Invertida

Preparação da dissonância

Ellipsis

Heterolepsis

Quaesitio Notae

Relação harmônica entre vozes

Inchoatio Imperfecta

Consonantiae Impropiae

Longinqua Distância

Tertia Deficiens

Quarta Superflua

Quinta Superflua

Sexta Superflua

Parrhesia

Cadentia Duriuscula

Figuras de acordes

Figuras de acordes

Noema

Analepsis

Mimesis Noema

Anadiplosis Noema

Anaploce

Faux Bourdon

4.2.3. Elementos diversos

Permutação ou substituição

Permutação

Antimetabole

*Atanacclasis * Hyperbaton*

Metabasis

Syncopatio

Substituição

Antistaechon

Mutatio Toni

Oposição de contrários

Antitheton

Hypallage

Silêncio

Aposiopesis

Articulus

Pausa

Suspiratio

Acumulação

* Polysyndeton

Fusão

Asyndeton

Synhaeresis

Omissão

Abruptio

Apocope

Ellipsis

Melalepsis

Adição

Duda

Suspensio

Dubitatio

Acumulação por agregação

Congeries

Enumeratio

Pleonasmus

Amplificação

Amplificatio

Transitus

Paragoge

Parenthesis

Metalepsis

Variação

*Apotomia * Diminutio*

Variatio

Emphasis

Epanalepsis

Schematoides

Parembole

r

A classificação das figuras de retórica por grupos de similaridades facilita o trabalho de pesquisa deixando mais clara sua atuação no contexto musical.

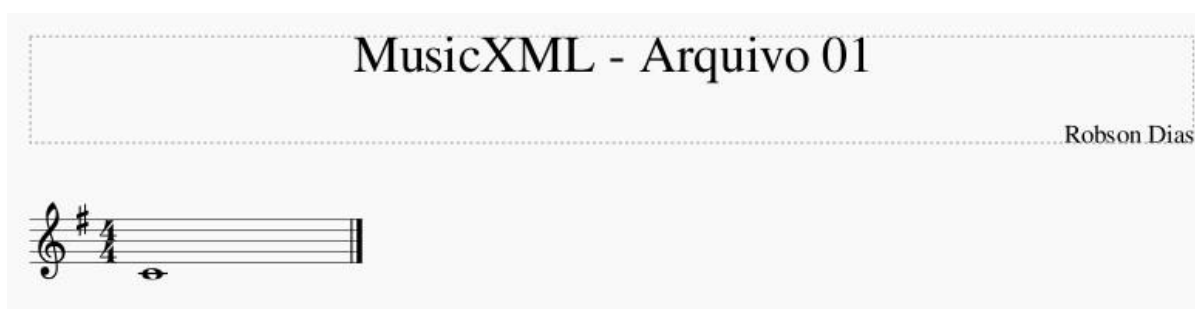
5. PROTOCOLO MUSICXML.

Neste capítulo vamos falar sobre o padrão MusicXML especificamente a versão 3.0. A decisão de adotar esta extensão como padrão para o Analysis Project está baseada na sua comunicabilidade com diversos “softwares” do mercado³⁸. Em sua concepção foi desenvolvido para o compartilhamento de arquivos de partituras entre programas de desenvolvedores diferentes³⁹.

5.1. Manipulação das tags nos arquivos MusicXML

Um arquivo MusicXML é um arquivo no formato texto que contém informações sobre música simbólica. Sua estrutura é simples e de fácil manipulação.

Figura 51 - MusicXML - Exemplo 01



Fonte: Do Autor

Exportamos esta partitura para o formato MusicXML. Vamos marcar no arquivo os elementos que vamos precisar para manipulação⁴⁰.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 3.1
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
  <work>
    <work-title>MusicXML - Arquivo 01</work-title>
  </work>
  <identification>
    <creator type="composer">Robson Dias
    </creator>
    <encoding>
```

³⁸ A lista destes software em ordem alfabética se encontra no ANEXO C. (Nota do Autor)

³⁹ Os exemplos de demonstração do padrão Musicxml aqui propostos foram desenvolvidos pelo autor utilizando o MUESCORE que pode ser baixado de forma gratuita no site do desenvolvedor. <https://musescore.com/dashboard> (Nota do Autor)

⁴⁰ Caso deseje maiores informações sobre o padrão acesse o site oficial em

```

    <software>MuseScore 3.4.2</software>
    <encoding-date>2020-02-27</encoding-date>
    <supports element="accidental" type="yes"/>
    <supports element="beam" type="yes"/>
    <supports element="print" attribute="new-page" type="yes"
value="yes"/>
    <supports element="print" attribute="new-system" type="yes"
value="yes"/>
    <supports element="stem" type="yes"/>
  </encoding>
</identification>
<defaults>
  <scaling>
    <millimeters>7.05556</millimeters>
    <tenths>40</tenths>
  </scaling>
  <page-layout>
    <page-height>1683.36</page-height>
    <page-width>1190.88</page-width>
    <page-margins type="even">
      <left-margin>56.6929</left-margin>
      <right-margin>56.6929</right-margin>
      <top-margin>56.6929</top-margin>
      <bottom-margin>113.386</bottom-margin>
    </page-margins>
    <page-margins type="odd">
      <left-margin>56.6929</left-margin>
      <right-margin>56.6929</right-margin>
      <top-margin>56.6929</top-margin>
      <bottom-margin>113.386</bottom-margin>
    </page-margins>
  </page-layout>
  <word-font font-family="FreeSerif" font-size="10"/>
  <lyric-font font-family="FreeSerif" font-size="11"/>
</defaults>
<credit page="1">
  <credit-words default-x="595.44" default-y="1626.67" justify="center"
valign="top" font-size="24">MusicXML - Arquivo 01</credit-words>
</credit>
<credit page="1">
  <credit-words default-x="1134.19" default-y="1526.67" justify="right"
valign="bottom" font-size="12">Robson Dias</credit-words>
</credit>
<part-list>
  <score-part id="P1">
    <part-name>Piano</part-name>
    <part-abbreviation>Pno.</part-abbreviation>
    <score-instrument id="P1-I1">
      <instrument-name>Piano</instrument-name>
    </score-instrument>
    <midi-device id="P1-I1" port="1"></midi-device>
    <midi-instrument id="P1-I1">
      <midi-channel>1</midi-channel>
      <midi-program>1</midi-program>
      <volume>78.7402</volume>
      <pan>0</pan>
    </midi-instrument>
  </score-part>
</part-list>
<part id="P1">
  <measure number="1" width="307.54">

```

```

<print>
  <system-layout>
    <system-margins>
      <left-margin>0.00</left-margin>
      <right-margin>769.95</right-margin>
    </system-margins>
    <top-system-distance>170.00</top-system-distance>
  </system-layout>
</print>
<attributes>
  <divisions>1</divisions>
  <key>
    <fifths>1</fifths>
  </key>
  <time>
    <beats>4</beats>
    <beat-type>4</beat-type>
  </time>
  <clef>
    <sign>G</sign>
    <line>2</line>
  </clef>
</attributes>
<note default-x="104.08" default-y="-50.00">
  <pitch>
    <step>C</step>
    <octave>4</octave>
  </pitch>
  <duration>4</duration>
  <voice>1</voice>
  <type>whole</type>
</note>
<barline location="right">
  <bar-style>light-heavy</bar-style>
</barline>
</measure>
</part>
</score-partwise>

```

Codificação A estrutura acima descreve um arquivo MusicXML simples. A partir dela vamos identificar todas as estruturas necessárias para a manipulação dos dados musicais.

<?xml version="1.0" encoding="UTF-8"?> Indica que a codificação utilizada é Unicode UTF-8⁴¹. Qualquer arquivo XML exige este modelo de declaração.

⁴¹ UTF-8 (8-bit Unicode Transformation Format- informática avançada) é um tipo de codificação binária (Unicode) de comprimento variável criado por Ken Thompson e Rob Pike. Pode representar qualquer carácter universal padrão do Unicode, sendo também compatível com o ASCII. Por esta razão, está lentamente a ser adaptado como tipo de codificação padrão para email, páginas web, e outros locais onde os caracteres são armazenados. Texto extraído de <https://pt.wikipedia.org/wiki/UTF-8> - acesso em 13/04/2019 (Nota do Autor)

Declaração que o arquivo é um MusicXML

`<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 3.1 Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">` Estamos afirmando que o arquivo é tipo PARTWISE⁴² (partitura escrita em partes). É utilizado uma declaração pública indicando o local onde o arquivo DTD é encontrado na net. - Geralmente os softwares instalam uma cópia local para as DTD's do MusicXML. (Nota do Autor)

Corpo do arquivo

`<score-partwise version="3.1"></score-partwise>` Vamos escrever tudo que é necessário entre estas tags. Indica igualmente a versão que estamos utilizando do MusicXML (3.1 no caso).

Título, subtítulo etc. . .

```
<work>
<work-title>MusicXML - Arquivo 01</work-title>
</work>
```

Identificação, encoding, data, layout

```
<identification>
<creator type="composer">Robson Dias</creator>
<encoding>
<software>MuseScore 3.4.2</software>
<encoding-date>2019-04-13</encoding-date>
<supports element="accidental" type="yes"/>
<supports element="beam" type="yes"/>
<supports element="print" attribute="new-page" type="yes"
value="yes"/>
<supports element="print" attribute="new-system" type="yes"
value="yes"/>
<supports element="stem" type="yes"/>
</encoding>
</identification>
```

⁴² Colocamos no ANEXO D os principais esquemas de DTD do MusicXML. (Nota do Autor)

Lista de partes (Funcionalidades midi e identificação do Instrumento)

```

<part-list>
<score-part id="P1">
<part-name>Piano</part-name>
<part-abbreviation>Pno.</part-abbreviation>
<score-instrument id="P1-I1">
<instrument-name>Piano</instrument-name>
</score-instrument>
<midi-device id="P1-I1" port="1"></midi-device>
<midi-instrument id="P1-I1">
<midi-channel>1</midi-channel>
<midi-program>1</midi-program>
<volume>78.7402</volume>
<pan>0</pan>
</midi-instrument>
</score-part>
</part-list>

```

Identificação da parte e seus atributos – A atenção especial a este tópico se deve a sua importância no desenvolvimento dos algoritmos. Na prática, o que precisamos manipular se encontra na <part></part> da música.

```

<part id="P1">
<measure number="1" width="307.54">
<print>
<system-layout>
<system-margins>
<left-margin>0.00</left-margin>
<right-margin>769.95</right-margin>
</system-margins>
<top-system-distance>170.00</top-system-distance>
</system-layout>
</print>

```

```

<attributes>
<divisions>1 </divisions> (NÚM. DO COMP.)
<key> (ARMADURA DE CLAVE / + sustenidos - bemois)
<fifths>1 </fifths>
</key>
<time> (FÓRMULA DE COMPASSO)
<beats>4 </beats>
<beat-type>4 </beat-type>
</time>
<clef> (CLAVE)
<sign>G </sign>
<line>2 </line>
</clef>
</attributes>
<note default-x="104.08" default-y="-50.00">
<pitch> (ALTURA DA NOTA)
<step>C </step>
<octave>4 </octave>
</pitch>
<duration>4 </duration> (DURAÇÃO)
<voice>1 </voice>
<type>whole </type> (FIGURA RÍTMICA)
</note>
<barline location="right">
<bar-style>light-heavy </bar-style>
</barline>
</measure>
</part>

```

Os elementos que iremos manipular constantemente na construção dos algoritmos dos padrões de busca se encontram declarados na tag <attributes></attributes> (Número do compasso, armadura de clave, fórmula de compasso, clave) e na tag <note></note> (Frequência da nota, duração e figura rítmica associada).

Um olhar mais atento facilmente poderá identificar outros elementos musicais descritos no arquivo MusicXML. Atualmente o padrão oferece opções avançadas de grafia: não conhecemos nenhum símbolo musical em notação tradicional que não possa ser representado no formato.

5.2. Arquivos no formato texto e python.

A manipulação de arquivos no formato texto será realizada com python. O Analysis Project precisa basicamente seguir os seguintes passos:

Abrir o arquivo no disco :

Existe basicamente duas opções de acesso ao conteúdo de um arquivo: Abrir com permissão de leitura ou escrita. O Analysis Project precisa somente da permissão de leitura do arquivo. Abaixo um exemplo de código para as opções:

- ler = open('nome-do-arquivo', 'r')
- editar = open('nome-do-arquivo', 'w')

Se não passamos o segundo parâmetro ele assume como norma o formato leitura (r). Podemos igualmente passar um terceiro parâmetro relativo à codificação, contudo, no nosso caso a codificação será sempre UTF-8.

```
arquivo = open(nome-do-arquivo, 'r', encoding="utf8")
```

Procurar o necessário

Para realizarmos nossas buscas precisamos encontrar dentro dos arquivos os locais onde se localizam a informações:

```
with open("arquivo.txt", "r") as arq:
    for linha in arq:
        if linha.find('<divisions>') > -1:
            print linha
```

No exemplo acima procurei o número do compasso e imprimi toda a linha: Teria uma saída de dados assim:

1. <divisions>1</divisions>
2. <divisions>2</divisions>
3. <divisions>3</divisions> (...)

Através das Tags conseguimos selecionar a informação que desejamos (Frequência, claves, armadura etc. . .)

6. ANALYSIS PROJECT

O Analysis Project é uma ferramenta desenvolvida para esta tese por uma necessidade. É um auxílio para o musicólogo que se interessa em análise de música simbólica envolvendo uma grande quantidade de obras e buscas de padrões e esquemas musicais.

6.1. Recursos e funcionalidades da aplicação

Passaremos a descrever recursos e funcionalidades no Analysis Project com o objetivo de ilustrar casos e cenários mostrando como o software pode apoiar determinadas tarefas.

6.1.1. Tela de Abertura

Na tela inicial temos o nome do projeto, sua vinculação com esta tese e com o laboratório de musicologia e universidade a que pertence.

Figura 52 - Tela Inicial do Analysis Project



Copyright © 2013–20 todos os direitos reservados.

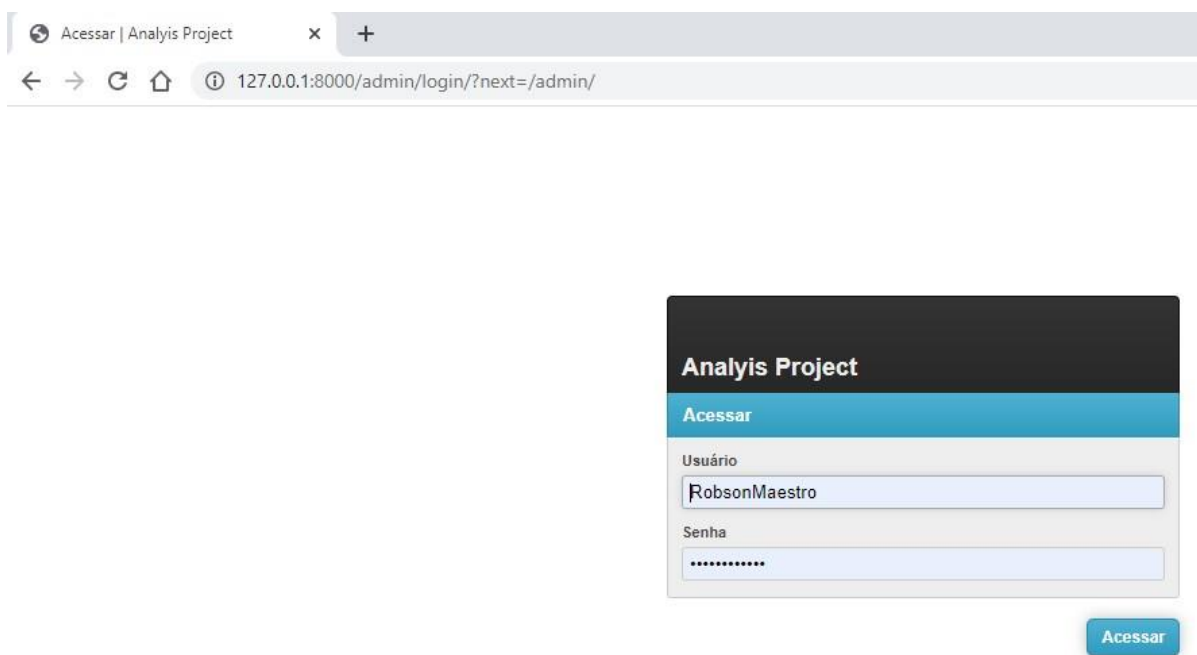
Fonte: Do Autor

Outra informação de relevância é sobre os direitos autorais e período de desenvolvimento que se iniciou em 2013.

6.1.2. Tela de Acesso

Clicando no botão de acesso presente na tela de abertura seguimos para a tela de validação de usuário e senha. O Analysis Project é uma aplicação de acesso restrito.

Figura 53 - Tela de acesso ao painel de controle



Fonte: Do Autor

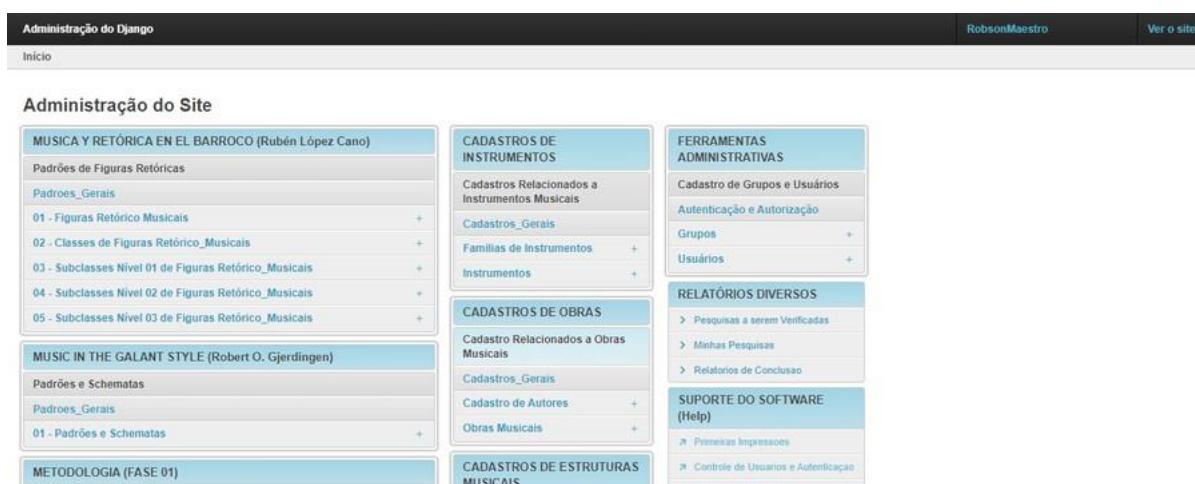
O endereço eletrônico descrito na figura acima (<http://127.0.0.1:8000/admin/login/?next=/admin/>) indica que a aplicação está hospedada em um servidor local.⁴³

6.1.3. Painel de Controle (Visão Geral)

O painel de controle nos oferece o acesso a todos os recursos da aplicação.

⁴³ Esta aplicação foi desenvolvida com o objetivo de pesquisa musicológica. Compreendendo seu potencial de utilização ainda não discutimos sobre os modelos de distribuição e licença que iremos adotar. (Nota do Autor)

Figura 54 - Visão geral do painel de controle



Fonte: Do Autor

A figura acima mostra uma visão geral do painel de controle. Nele o usuário pode manipular de forma efetiva os dados cadastrados, realizar pesquisas, obter visualizações de resultados e realizar todas as manutenções necessárias ao bom funcionamento da aplicação.

6.1.4. Modelos de busca

Basicamente o Analysis Project possui 2 modelos de busca e uma opção onde o usuário cadastra livremente o padrão ou esquema de deseja procurar.

Nas pesquisas que o software disponibiliza o usuário não pode cadastrar novos padrões podendo apenas alterar questões referentes a exemplos ou curiosidades sobre determinado registro. A programação foi implementada pelo desenvolvedor (Figuras de retórica e padrões e esquemas do Gjerdingen) são recursos não disponíveis para modificação pelo usuário.

A pesquisa de padrão determinado pelo usuário é realizada da seguinte maneira:

1. Criar um arquivo no formato MusicXML em um editor de partituras da sua escolha
2. Abrir o arquivo em um editor de texto
3. Copiar e colar dentro do campo destinado para esta finalidade
4. Escolher os tópicos mais relevantes para sua pesquisa.

6.1.4.1. Figuras de retórica

Nesta seção procuramos cadastrar diversas figuras de retórica. Utilizamos a sistematização proposta por Rubén López Cano em sua obra *Música Y Retórica en el barroco*.

Figura 55 - Opções de cadastro de figuras de retórica

MUSICA Y RETÓRICA EN EL BARROCO (Rubén López Cano)	
Padrões de Figuras Retóricas	
Padroes_Gerais	
01 - Figuras Retórico Musicais	+
02 - Classes de Figuras Retórico_Musicais	+
03 - Subclasses Nível 01 de Figuras Retórico_Musicais	+
04 - Subclasses Nível 02 de Figuras Retórico_Musicais	+
05 - Subclasses Nível 03 de Figuras Retórico_Musicais	+

Fonte: Do Autor

Segundo (Cano, 2001) as figuras de retórica podem ser organizadas em classes e subclasses.

Figura 56 - Organização das Figuras de Retórica no Analysis Project

Administração do Django							RobsonMaestro	Ver o site
Início > Padroes_Gerais > 01 - Figuras Retórico Musicais								
01 - Figuras Retórico Musicais							+ Adicionar 01 - Figura Retórico Musical	
10 total							Filtro	
<input type="checkbox"/>	Nome da Figura Retórico_Musical: ^	FIGURA É SINÔNIMO DE:	CLASSE vinculada a esta FIGURA	SUBCLASSE Nível 01 vinculada a esta CLASSE	SUBCLASSE NÍVEL 02 vinculada a esta SUBCLASSE NÍVEL 01	SUBCLASSE		
<input type="checkbox"/>	ANAFORA	ANAPHORA	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		
<input type="checkbox"/>	ANAPHORA	-	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		
<input type="checkbox"/>	ANÁSTROFE	EPISTROPHE	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		
<input type="checkbox"/>	CONVERSIÓN	EPISTROPHE	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		
<input type="checkbox"/>	EPISTROFE	EPISTROPHE	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		
<input type="checkbox"/>	EPISTROPHE	-	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		
<input type="checkbox"/>	MIMESIS (Thuringus)	ANAPHORA	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		
<input type="checkbox"/>	OMOIOPTOTON (Kircher)	EPISTROPHE	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		
<input type="checkbox"/>	REPETICIÓN	ANAPHORA	FIGURAS QUE AFETAM A MELODIA	FIGURAS DE REPETIÇÃO	ESTRUTURAS DE REPETIÇÃO	-		

Fonte: Do Autor

Observe a figura acima: A figura Anástrofe tem como sinônimo a Epistrophe. Esta figura pertence à CLASSE de figuras que afetam a melodia, a subclasse nível 01 figura de repetição e a subclasse nível 2 de estruturas de repetição.

Figura 57 - Recursos de inserção e filtros de busca

Um formulário de filtros de busca para a inserção de uma figura retórica musical. O formulário é exibido sobre uma interface de usuário com um botão azul no topo que diz '+ Adicionar 01 - Figura Retórico Musical'. O formulário contém os seguintes campos de filtro:

- Filtro** (dropdown)
- Nome da Figura Retórico_Musical:** Todos (dropdown)
- FIGURA É SINÔNIMO DE:** Todos (dropdown)
- CLASSE vinculada a esta FIGURA**: Todos (dropdown)
- SUBCLASSE Nivel 01 vinculada a esta CLASSE**: Todos (dropdown)
- SUBCLASSE NIVEL 02 vinculada a esta SUBCLASSE NIVEL 01**: Todos (dropdown)
- SUBCLASSE NIVEL 03 vinculada a esta SUBCLASSE NIVEL 02**: Todos (dropdown)
- Data da Ultima Alteração**: Qualquer data (dropdown)
- Usuário que Alterou**: Todos (dropdown)
- Data da Inserção** (campo não visível)

Fonte: Do Autor

Para Incluirmos novas figuras no banco de dados devemos clicar no botão (+ Adicionar 01 - Figura Retórico Musical)⁴⁴.

Foram adicionados filtros como facilitadores para a seleção

- Nome da Figura (Seleção única de figura específica)

⁴⁴ Esta opção somente está disponível para o desenvolvedor do sistema.

- Figura é Sinônimo de (Seleciona todas as figuras que são sinônimos)
- Classe vinculada a figura (Seleção de todas as figuras de uma classe específica)
- Subclasses - níveis 1, 2 e 3 (Seleciona todas as figuras de determinada subclasse)
- Filtros de controle:
 - Data de alteração
 - Usuário que alterou
 - Data da Inclusão daquela figura
 - Usuário responsável por aquele registro

Figura 58 - – Exemplo de Cadastro de Figuras de Retórica

Administração do Django	
Início > Padroes_Gerais > 01 - Figuras Retórico Musicais > MIMESIS (Thuringus)	
CADASTRO DE FIGURAS DE RETÓRICA	
Nome da Figura Retórico_Musical	MIMESIS (Thuringus)
FIGURA É SINÔNIMO DE	ANAPHORA → +
CLASSE vinculada a esta FIGURA	FIGURAS QUE AFETAM A MELODIA → +
SUBCLASSE Nivel 01 vinculada a esta CLASSE	FIGURAS DE REPETIÇÃO → +
SUBCLASSE NIVEL 02 vinculada a esta SUBCLASSE NIVEL 01	ESTRUTURAS DE REPETIÇÃO → +
SUBCLASSE NIVEL 03 vinculada a esta SUBCLASSE NIVEL 02	----- → +
Tratadistas	

Fonte: Do Autor

O cadastro das figuras retórico musicais tem as seguintes opções:

- Nome da figura
- Sinônimo

- Classe
- Subclasse 01, 02 e 03
- Tradadistas
- Definição geral
- Referências Bibliográficas
- Observações e Curiosidades
- Exemplos musicais
- Exemplos Textuais

Figura 59 - Tratamento de Inclusão de Registros

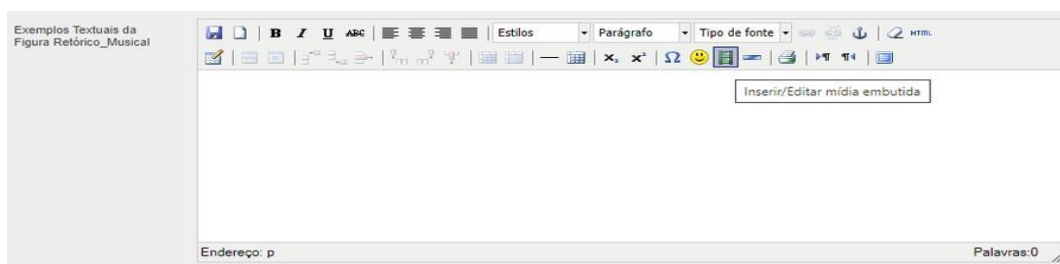
TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO			
Data da Última Alteração	3 de Março de 2019 às 19:10	Usuário que Alterou	RobsonMaestro → +
Data da Inserção	3 de Março de 2019 às 19:10	Usuário Responsável	RobsonMaestro → +
Este Registro é Público ou Privado?	Público	Este Registro pode ser editado?	Pode ser editado somente pelo

Fonte: Do Autor

Todos os registros que são incluídos no Analysis Project recebem um tratamento diferenciado de inclusão com as seguintes informações:

- Data em que se realizou alterações
- Usuário que alterou
- Data da inserção
- Usuário responsável
- Privacidade sobre o registro (nesta opção o usuário pode escolher se a inclusão que está realizando é de carácter público ou privado)
- Quem pode realizar alterações.

Figura 60 - Exemplo de Inserção de Textos e Mídia



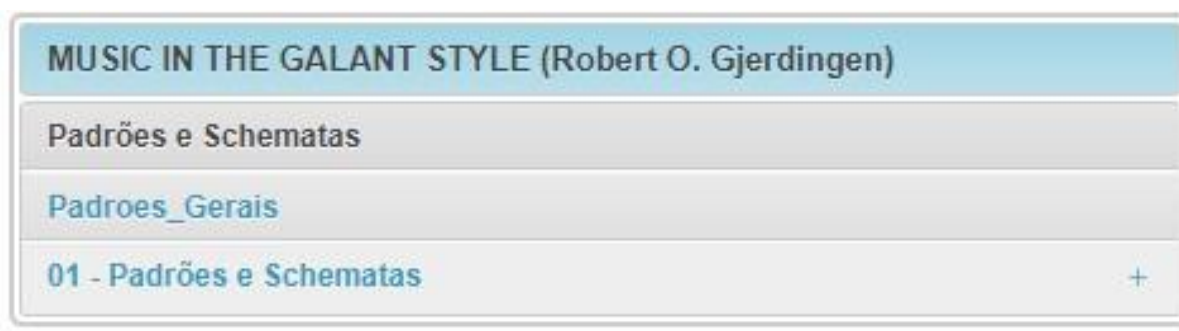
Fonte: Do Autor

O editor de texto e mídia do Analysis Project tem bons recursos de formatação permitindo ao usuário inserir imagens, sons e vídeos com o objetivo de facilitar a ilustração de determinado registro.

6.1.4.2. Esquemas Propostos por Robert O Gjerdingen

Como a opção anterior, este modelo de pesquisas já vem incluso na aplicação. Aqui estão cadastrados alguns esquemas e padrões propostos pelo pesquisador na obra Music in the Galant Style.

Figura 61 - Cadastro de Padrões e Esquemas propostos por Gjerdingen



Fonte: Do Autor

Acima a tela inicial no painel de controle para acesso às opções oferecidas para esta modalidade de pesquisa.

Figura 62 - Listagem dos Padrões Cadastrados

Início > Padroes_Gerais > 01 - Padrões e Schematas

01 - Padrões e Schematas + Adicionar 01 - Padrões e Schematas

12 total

<input type="checkbox"/>	Nome do Padrão ou Schemata:	Data da Última Alteração	Usuário que Alterou	Data da Inserção	Usuário Responsável
<input type="checkbox"/>	CLAUSULAE	4 de Março de 2019 às 21:13	RobsonMaestro	4 de Março de 2019 às 21:13	RobsonMaestro
<input type="checkbox"/>	DO-RÉ-MI	4 de Março de 2019 às 21:12	RobsonMaestro	4 de Março de 2019 às 21:12	RobsonMaestro
<input type="checkbox"/>	FENAROLI	4 de Março de 2019 às 21:15	RobsonMaestro	4 de Março de 2019 às 21:15	RobsonMaestro
<input type="checkbox"/>	FONTE	4 de Março de 2019 às 21:11	RobsonMaestro	4 de Março de 2019 às 21:11	RobsonMaestro
<input type="checkbox"/>	INDUGIO	4 de Março de 2019 às 21:21	RobsonMaestro	4 de Março de 2019 às 21:16	RobsonMaestro
<input type="checkbox"/>	MEYER	4 de Março de 2019 às 21:13	RobsonMaestro	4 de Março de 2019 às 21:13	RobsonMaestro
<input type="checkbox"/>	MONTE	4 de Março de 2019 às 21:13	RobsonMaestro	4 de Março de 2019 às 21:13	RobsonMaestro
<input type="checkbox"/>	PONTE	4 de Março de 2019 às 21:15	RobsonMaestro	4 de Março de 2019 às 21:15	RobsonMaestro
<input type="checkbox"/>	PRINNER	4 de Março de 2019 às 21:11	RobsonMaestro	4 de Março de 2019 às 21:11	RobsonMaestro

Filtro

Nome do Padrão ou Schemata: Todos

Data da Última Alteração: Qualquer data

Usuário que Alterou: Todos

Data da Inserção: Qualquer data

Usuário Responsável: Todos

Fonte: Do Autor

Na figura acima temos uma listagem dos padrões cadastrados, os filtros de busca e demais registros relacionados como as datas da última alteração, usuário que alterou, usuário que realizou o cadastro inicial e data.

Figura 63 - Opções de Cadastro para os Padrões de Gjerdingen

CADASTRO DE PADRÕES E SCHEMATAS

Nome do Padrão ou Schemata: ROMANESCA

Definição Geral do Padrão ou Schemata

FOR THE PATRONS OF GALANT MUSIC, making informed judgments about compositions and their performances required familiarity with the important schemata of the style. For the composers, making works worthy of praise required being able to produce exemplars of every schema correct in every detail. The more passive knowledge of patrons could be gleaned from frequent listening to the typical phrases of galant music. The active, operationalized knowledge of composers was carefully taught to them by music masters — maestros. The greatest maestros of the age worked in Italy, and they developed a unique method of instruction centered on the *partimento*—the instructional bass. A *partimento* resembled the bass part given to eighteenth-century accompanists, with the difference being the lack of any other players or their parts. The *partimento* was the bass to a virtual ensemble that played in the mind of the student and became sound through realization at the keyboard. In behavioral terms, the *partimento*, which often changed clefs temporarily to become any voice in the virtual ensemble, provided a series of stimuli to a series of schemata, and the learned responses of the student resulted in the multivoice fabric of a series of phrases and cadences. From seeing only one feature of a particular schema—any one of its characteristic parts—the student learned to complete the entire pattern, and in doing so committed every aspect of the schema to memory. The result was fluency in the style and the ability to “speak” this courtly language.

Fonte: Do Autor

Para os cadastros de padrões e esquemas propostos por Gjerdingen possuímos os seguintes campos:

- Nome do Padrão
- Descrição
- Observações e curiosidades
- Exemplos Musicais

Como no exemplo anterior o tratamento sobre a inclusão do registro (usuário responsável / data de inclusão / alterações e privacidade) são obrigatórios.

6.1.5. Metodologia Aplicada

Analysis Project é um organizador de metodologia, ou seja, ele fornece o ferramental para dar suporte à aplicação da metodologia proposta. Na aplicação existe quatro instancias descritas como fases de 01 a 04. Sem passar por estas etapas o musicólogo não conseguira realizar nenhuma pesquisa na aplicação⁴⁵.

⁴⁵ Maiores detalhes sobre os recursos metodológicos estão descritos no capítulo 03. (Nota do Autor)

6.1.6. Cadastros Gerais

Esta modalidade se refere a questões que estarão presentes em quase a totalidade das pesquisas a serem realizadas.

6.1.6.1. Instrumentos

Em arquivos MusicXML podemos como demonstrado anteriormente especificar os instrumentos utilizados em uma peça musical. Precisamos ter uma base de dados especificando-os para facilitar o resultado de nossas pesquisas.

Figura 64 - Listagem parcial de instrumentos cadastrados

Início > Cadastros_Gerais > Instrumentos

Instrumentos + Adicionar Instrumentos

23 total Filtro

<input type="checkbox"/>	Nome do Instrumento: ^	Selecione a Família deste Instrumento	Data da Última Alteração	Usuário que Alterou	Data da Inserção	Usuário Responsável
<input type="checkbox"/>	BAIXO	VOZES	3 de Março de 2019 às 23:09	RobsonMaestro	3 de Março de 2019 às 23:09	RobsonMaestro
<input type="checkbox"/>	CLARINETE	MADERAS	3 de Março de 2019 às 23:11	RobsonMaestro	3 de Março de 2019 às 23:11	RobsonMaestro
<input type="checkbox"/>	CONTRABAIXO ACÚSTICO	CORDAS	3 de Março de 2019 às 23:06	RobsonMaestro	3 de Março de 2019 às 23:06	RobsonMaestro
<input type="checkbox"/>	CONTRALTO	VOZES	3 de Março de 2019 às 23:07	RobsonMaestro	3 de Março de 2019 às 23:07	RobsonMaestro
<input type="checkbox"/>	FAGOTE	MADERAS	3 de Março de 2019 às 23:12	RobsonMaestro	3 de Março de 2019 às 23:12	RobsonMaestro
<input type="checkbox"/>	FLAUTA DOCE BAIXO	MADERAS	3 de Março de 2019 às 23:15	RobsonMaestro	3 de Março de 2019 às 23:15	RobsonMaestro
<input type="checkbox"/>	FLAUTA DOCE CONTRALTO	MADERAS	3 de Março de 2019 às 23:14	RobsonMaestro	3 de Março de 2019 às 23:14	RobsonMaestro
<input type="checkbox"/>	FLAUTA DOCE SOPRANO	MADERAS	3 de Março de 2019 às 23:13	RobsonMaestro	3 de Março de 2019 às 23:13	RobsonMaestro
<input type="checkbox"/>	FLAUTA DOCE TENOR	MADERAS	3 de Março de 2019 às 23:14	RobsonMaestro	3 de Março de 2019 às 23:14	RobsonMaestro
<input type="checkbox"/>	FLAUTA TRANSVERSAL 01	MADERAS	3 de Março de 2019 às 23:09	RobsonMaestro	3 de Março de 2019 às 23:09	RobsonMaestro

Fonte: Do Autor

Podemos organizar nossa listagem de instrumentos cadastrados por ordem alfabética, família, data de alteração entre outras opções.

Figura 65 - Cadastro de Instrumentos

CADASTRO DE INSTRUMENTOS

Nome do Instrumento:

Selecione a Família deste Instrumento: → + ×

História / Curiosidades sobre o instrumento



A flauta transversal, por vezes chamada de flauta transversa ou simplesmente de flauta, é um aerofone da família das madeiras. Apesar de atualmente ser fabricada em metal, em sua origem, esta era de madeira. Por esta razão, até hoje, a flauta transversal é classificada nas orquestras como um instrumento pertencente ao grupo das madeiras.

Endereço: p Palavras: 54

TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO

Fonte: Do Autor

O Cadastro de instrumentos possui as seguintes opções:

- Nome do Instrumento
- Família que pertence
- Histórico e descrição
- Tratamento sobre a Inclusão do registro (obrigatório)

6.1.6.2. Famílias de Instrumentos

Os instrumentos musicais são classificados em famílias ou grupos.

Figura 66 - Listagem das Famílias de Instrumentos Cadastradas

Famílias de Instrumentos + Adicionar Famílias de Instrumentos

5 total Filtro

<input type="checkbox"/>	Nome da Família: ▲	Data da Última Alteração	Usuário que Alterou	Data da Inserção	Usuário Responsável
<input type="checkbox"/>	CORDAS	3 de Março de 2019 às 22:51	RobsonMaestro	3 de Março de 2019 às 22:51	RobsonMaestro
<input type="checkbox"/>	MADEIRAS	3 de Março de 2019 às 22:52	RobsonMaestro	3 de Março de 2019 às 22:52	RobsonMaestro
<input type="checkbox"/>	METAIS	3 de Março de 2019 às 22:52	RobsonMaestro	3 de Março de 2019 às 22:52	RobsonMaestro
<input type="checkbox"/>	PERCUSSÃO	3 de Março de 2019 às 22:53	RobsonMaestro	3 de Março de 2019 às 22:53	RobsonMaestro
<input type="checkbox"/>	VOZES	3 de Março de 2019 às 22:53	RobsonMaestro	3 de Março de 2019 às 22:53	RobsonMaestro

5 total

Fonte: Do Autor

Usuário pode cadastrar livremente os grupos de instrumentos caso necessitem suas pesquisas de uma instrumentação não convencional.

Figura 67 - Cadastro de Famílias de Instrumentos

CADASTRO DE FAMÍLIAS DE INSTRUMENTOS

Nome da Família: CORDAS

Descrição: Instrumentos de cordas são instrumentos musicais cuja fonte primária de som é a vibração de uma corda tensionada quando beliscada, percutada ou friccionada. Na classificação Hornbostel-Sachs, bastante utilizada atualmente, estes instrumentos representam a classe dos cordofones identificada pelos códigos iniciados pelo número 3.

Endereço: p Palavras:42

TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO

Data da Última Alteração	3 de Março de 2019 às 22:51	Usuário que Alterou	RobsonMaestro → + ×
Data da Inserção	3 de Março de 2019 às 22:51	Usuário Responsável	RobsonMaestro → + ×

Fonte: Do Autor

As opções relacionadas ao cadastro de famílias de instrumentos são?

- Nome da família
- Descrição
- Tratamento de inclusão de registro (obrigatório)

6.1.6.3. Obras Musicais

O material básico que utilizamos em nossas pesquisas são as obras musicais. Elas devem estar cadastradas de forma correta em nossa base de dados para que os resultados das nossas pesquisas estejam corretos.

Figura 68 - Opções de cadastro de obras

CADASTROS DE OBRAS

- Cadastro Relacionados a Obras Musicais
- Cadastros_Gerais
- Cadastro de Autores +
- Obras Musicais +

Fonte: Do Autor

Relacionados a obras músicas temos o cadastro de autores e das músicas que compõe o corpus do Analysis Project.

6.1.6.4. Autores

Para a realização do cadastro de obras musicais no banco de dados é fundamental o cadastro de autores. Um dos problemas encontrados em bases públicas é que o mesmo autor é referenciado de forma diferente (J.S. Bach / Johann Sebastian Bach).

Figura 69 - Lista parcial de autores cadastrados

Cadastro de Autores

6 total

<input type="checkbox"/>	Nome do Autor: ^	Data de Nascimento:	Data de Falecimento:	Nacionalidade do Autor:
<input type="checkbox"/>	Arcangelo Corelli	Fusignano, 17/02/1643	Roma, 8/01/1713	Italiano
<input type="checkbox"/>	Fryderyk Franciszek Chopin	22 de fevereiro de 1810	Paris, 17 de Outubro de 1849	Polonês
<input type="checkbox"/>	George Frideric Handel	Halle an der Saale, 23/02/1685	Londres, 14/04/1759	Alemão
<input type="checkbox"/>	Johann Sebastian Bach	Eisenach, 21 de março de 1685	Leipzig, 28 de julho de 1750	Sacro Império Romano-Germânico, atual Alemanha.
<input type="checkbox"/>	Joseph Haydn	Rohrau, 31 de março de 1732	Viena, 31 de maio de 1809	Austríaco
<input type="checkbox"/>	Ludwig van Beethoven	Born, 17 de dezembro 1770	Viena, 26 de março de 1827	Alemão

Fonte: Do Autor

A padronização do cadastro evita uma série de problemas no resultado das pesquisas.

Figura 70 - Cadastro de autores

CADASTRO DE AUTORES


Nome do Autor:

Data de Nascimento:

Data de Falecimento:

Nacionalidade do Autor:

Biografia do Autor



Nascido numa família de longa tradição musical, cedo mostrou possuir talento e logo tornou-se um músico completo. Estudante incansável, adquiriu um vasto conhecimento da música europeia de sua época e das gerações anteriores. Desempenhou vários cargos em cortes e igrejas alemãs, mas suas funções mais destacadas foram a de Kantor da Igreja de São Tomás e Diretor Musical da cidade de Leipzig, onde desenvolveu a parte final e mais importante de sua carreira. Absorvendo inicialmente o grande repertório de música contrapontística germânica como base principal de seu estilo, recebeu mais tarde a influência italiana e francesa, através das quais sua obra se enriqueceu e transformou, realizando uma síntese original de uma multiplicidade de tendências. Praticou quase todos os gêneros musicais conhecidos em seu tempo, com a notável exceção da ópera, embora suas cantatas maduras revelem bastante influência desta que foi uma das formas mais populares do período Barroco.

Endereço: Palavras: 148

Fonte da Biografia do Autor:

Fonte: Do Autor

O cadastro de autores nos oferece as seguintes opções:

- Nome do autor
- Local / Data de Nascimento
- Local / Data de Falecimento
- Nacionalidade
- Biografia
- Fonte das referências bibliográficas
- Tratamento de inclusão de registro (obrigatório)

Algumas informações sobre o autor são de interesse do musicólogo que deseja selecionar obras por período (data) ou por nacionalidade dos compositores.

6.1.6.5. Obras

Possuímos hoje em nossa base de dados todas as músicas em formato MusicXML do Music21⁴⁶. Precisávamos de ter um corpus mínimo para a realização dos nossos testes de pesquisa. Os resultados aqui apresentados foram pesquisas realizadas somente nesta base de dados.

Figura 71 - Lista parcial de obras cadastradas

Analysis Project		
Início > Cadastros_Gerais > Obras Musicais		
Obras Musicais		
18 total		
<input type="checkbox"/>	Nome da Obra: ^	Autor da Obra:
<input type="checkbox"/>	BWV 101.7	Johann Sebastian Bach
<input type="checkbox"/>	BWV 102.7	Johann Sebastian Bach
<input type="checkbox"/>	BWV 103.6	Johann Sebastian Bach
<input type="checkbox"/>	BWV 104.6	Johann Sebastian Bach
<input type="checkbox"/>	BWV 10.7	Johann Sebastian Bach
<input type="checkbox"/>	BWV 108.6	Johann Sebastian Bach

Fonte: Do Autor

⁴⁶ A lista de obras cadastradas se encontra no Anexo B. (Nota do Autor)

Manter os nomes das partituras como está descrito no Music21 foi uma escolha por uma questão de compatibilidade.

Figura 72 - Cadastro de obras

The screenshot shows the 'Cadastro de Obras' form with the following details:

- Nome da Obra:** BWV 102.7
- Autor da Obra:** Johann Sebastian Bach
- Data em que a obra foi Composta:** (Empty field)
- Upload da Obra no formato MusicXML:** Escolher arquivo | Nenhum arquivo selecionado
- Instrumentação da Obra:**
 - Instrumentação da Obra: disponíveis:**
 - CLARINETE
 - CONTRABAIXO ACÚSTICO
 - FAGOTE
 - FLAUTA DOCE BAIXO
 - FLAUTA DOCE CONTRALTO
 - FLAUTA DOCE SOPRANO
 - FLAUTA DOCE TENOR
 - FLAUTA TRANSVERSAL 01
 - FLAUTA TRANSVERSAL 02
 - Instrumentação da Obra: escolhido(s):**
 - BAIXO
 - SOPRANO
 - TENOR
 - CONTRALTO

Fonte: Do Autor

Os usuários que têm permissão podem cadastrar em nosso banco de dados as obras que necessitam para a realização das suas pesquisas. Os campos que necessitam preencher são:

Nome da obra

Autor da obra (Deverá ser selecionado - caso não exista deve ser cadastrado primeiramente em nosso banco de autores)

Data de composição (Caso possua esta informação - registro não obrigatório)

Upload do arquivo MusicXML (Deve ser realizado e pode a qualquer momento ser alterado caso o usuário perceber qualquer incorreção)

Instrumentação da obra

Tratamento de inclusão de registro (obrigatório) - Aqui o usuário tem a opção de selecionar se esta obra é pública ou privada. No caso de ser uma obra pública ela constará em nossa base de dados. No caso de uma obra privada ela estará no diretório do grupo (se o grupo puder alterar) ou no diretório do usuário caso ele queira ter exclusividade do registro.

6.1.7. Estruturas Musicais

O cadastro de estruturas musicais está diretamente relacionado com as pesquisas de padrões livres propostos pelo usuário. A programação destas variáveis é realizada somente pelo administrador do sistema em arquivo restrito.

Figura 73 - Opções de Cadastro de estruturas musicais.

CADASTROS DE ESTRUTURAS MUSICAIS	
Cadastro Relacionados a Padrões Musicais	
Cadastros_Gerais	
Materiais Utilizados nas Pesquisas	+
Refinamentos	+
Transformacoes Melodicas	+
Transformacoes Rítmicas	+
Transformações da Tipografia	+
Transposições	+

Fonte: Do Autor

A inclusão de novas possibilidades depende do desenvolvimento de código de programação compatível.

6.1.7.1. Materiais utilizados nas pesquisas

Este tópico é de cadastro livre. Aqui o musicólogo pode adicionar qual o material que ele utilizou em determinada pesquisa para consulta futura.

Figura 74 - Cadastro de materiais utilizados nas pesquisas

Materiais Utilizados nas Pesquisas		
3 total		
<input type="checkbox"/>	Tipo de Material Utilizado na Pesquisa:	Da
<input type="checkbox"/>	Arquivos	24
<input type="checkbox"/>	Escrita Manual de Intervalos	24
<input type="checkbox"/>	Linguagem poética (letra das obras)	24

Fonte: Do Autor

Na figura acima temos algumas sugestões como arquivos de base específica, ou letra de música (uma frase musical ou movimento melódico quando se pronuncia a palavra kyrie por exemplo).

6.1.7.2. Refinamentos

Os refinamentos da pesquisa são algumas opções que o musicólogo escolhe relativo ao padrão que ele cadastrou. São responsáveis em gerar menor quantidade de registros que não condizem com suas buscas porque especificam exatamente o que ele deseja.

Figura 75 - Lista dos refinamentos de pesquisa



Fonte: Do Autor

A imagem acima é uma listagem parcial dos refinamentos que o Analysis Project oferece ao usuário. Em sua totalidade são estes:

Notas de apoio (ignorar / tratar / considerar como notas normais)

Correspondência entre o número de compassos (Se a pesquisa deve obedecer ao mesmo número de compassos do padrão)

Correspondência nas articulações

Correspondência na superposição de padrões (O pesquisador pode criar dois padrões - uma linha de baixo e tenor - e desejar um resultado positivo somente quando encontrar às duas linhas superpostas)

- Ignore alturas repetidas
- Ignore o ritmo da última nota padrão
- Ignore pausas a frente

6.1.7.3. Transformações Melódicas

As transformações melódicas são possibilidades oferecidas ao pesquisador onde encontrará como resposta não a melodia exata do padrão que ele escolheu, mais algo aproximado.

Figura 76 - Lista das transformações melódicas

Transformacoes Melodicas

6 total	
<input type="checkbox"/>	Transformacoes Melodicas Utilizadas na Pesquisa: ^
<input type="checkbox"/>	Alcance / Tessitura
<input type="checkbox"/>	Esboço
<input type="checkbox"/>	Forma original
<input type="checkbox"/>	Inversão
<input type="checkbox"/>	Retrógado
<input type="checkbox"/>	Retrógado Invertido

Fonte: Do Autor

A imagem acima é uma listagem parcial das transformações melódicas que o Analysis Project oferece ao usuário. Em sua totalidade são estes:

Alcance / Tessitura (Verificará o intervalo da melodia entre a nota mais grave e mais aguda)

Esboço (O padrão é considerado somente um esboço. O Analysis Project procurará algo “parecido” com ele - utilize esta função de forma cautelosa porque gerará muitos resultados sem significação⁴⁷.

- Forma Original (Verifica o padrão exatamente como escrito)
- Inversão
- Retrogradado
- Retrogradado invertido.

6.1.7.4. Transformações Rítmicas

Como as transformações melódicas, as rítmicas têm o objetivo de realizar buscas aproximadas ao padrão que o usuário selecionou.

Figura 77 - Lista das transformações rítmicas



Fonte: Do Autor

Segue abaixo as opções:

- Forma Original
- Retrogradado

⁴⁷ Esta função foi implementada como uma tentativa de conseguir que a aplicação reconheça variações de um tema. O resultado dos testes foi bem abaixo do esperado. (Nota do Autor)

- Aumentação
- Diminuição

6.1.7.5. Transformações de Tipologia

Um dos diversos recursos existentes no padrão MusicXML é a possibilidade de guardar o texto. Alguns pesquisadores utilizam a linguagem poética em seus trabalhos e temos possibilidades de manipular este contexto.

Figura 78 - Lista de transformações da tipografia

Transformações da Tipografia

5 total

<input type="checkbox"/>	Transformações da Tipografia Utilizadas na Pesquisa: ^
<input type="checkbox"/>	Expressão
<input type="checkbox"/>	Itálico
<input type="checkbox"/>	Língua
<input type="checkbox"/>	Negrito
<input type="checkbox"/>	Palavra chave

Fonte: Do Autor

As opções para transformações de tipografia são:

- Expressão (O pesquisador cadastra uma expressão como, por exemplo, kyrie eleison e o Analysis Project a seleciona nas obras)
- Itálico (texto em itálico)
- Língua (inglês, italiano etc.)
- Negrito (texto em negrito)
- Palavra-chave (O usuário cadastra a palavra que deseja buscar com, por exemplo, kyrie e o Analysis Project a seleciona nas obras).

6.1.7.6. Transposições

Ao realizar a pesquisa o musicólogo pode definir se ele deseja que busque o padrão somente na tonalidade original ou que encontre o padrão transposto.

Figura 79 - Lista das transposições



Fonte: Do Autor

A figura acima é uma lista parcial das transposições tratadas pelo Analysis Project. Em sua totalidade são:

- Unísono
- Segunda menor (ascendente / decendente)
- Segunda maior (ascendente / decendente)
- Terça diminuta (ascendente / decendente)
- Terça menor (ascendente / decendente)
- Terça maior (ascendente / decendente)
- Quarta justa (ascendente / decendente)

- Quarta aumentada (ascendente / descendente)
- Quinta diminuta (ascendente / descendente)
- Quinta justa (ascendente / descendente)
- Quinta aumentada (ascendente / descendente)
- Sexta menor (ascendente / descendente)
- Sexta maior (ascendente / descendente)
- Sétima menor (ascendente / descendente)
- Sétima maior (ascendente / descendente)
- Oitava (ascendente / descendente)
- Intervalos compostos
- Todas as transposições (Será considerado qualquer intervalo)⁴⁸

6.1.8. Ferramentas Administrativas

As ferramentas administrativas estão relacionadas ao acesso da aplicação. São extremamente úteis para o cadastro de grupos de pesquisa oferecendo a cada usuário do sistema as autorizações de acesso de acordo com sua função.

Figura 80 - Cadastro de ferramentas administrativas



Fonte: Do Autor

⁴⁸ No item transposições se o usuário selecionar alguma transposição (ex: segunda menor ascendente) somente ela será considerada. A seleção pode ser múltipla. Caso não deseje a forma original marque uníssono. (Nota do Autor)

As opções de nossa ferramenta de administração do Analysis Project são o cadastro de grupos pesquisa e usuários.

6.1.8.1. Cadastros de Grupos de Pesquisa

Ao cadastrarmos um grupo oferecemos a possibilidade da realização das pesquisas de forma individualizada.

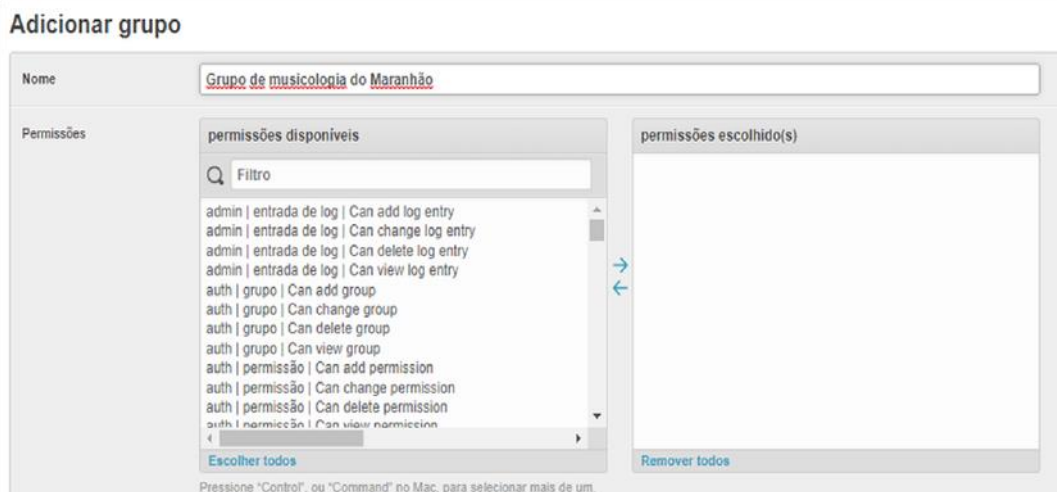
Figura 81 - Lista dos grupos de pesquisa cadastrados



Fonte: Do Autor

Por definição todos os registros do grupo são privados, contudo, caso o administrador do grupo queira disponibilizar alguma obra ou mesmo resultado de pesquisa pode publicar com facilidade.

Figura 82 - Cadastro de grupos de pesquisa



Fonte: Do Autor

Ao adicionarmos um grupo de pesquisa escolhemos quais as permissões de uso. Em relação aos logs gerados no sistema temos:

- admin | entrada de log | Can add log entry
- admin | entrada de log | Can change log entry
- admin | entrada de log | Can delete log entry
- admin | entrada de log | Can view log entry relacionados a grupos:
- auth | grupo | Can add group auth |
- grupo | Can change group auth |
- grupo | Can delete group auth |
- grupo | Can view group

O Laboratório de musicologia LAMUS por exemplo é o grupo principal. Ele pode adicionar grupos secundários para realizar trabalhos diferentes (pesquisa sobre retórica / música do sex XIV na Itália etc. . .). Ele coordena estes grupos que não podem, por exemplo adicionar novos grupos, mas que devem compartilhar seus trabalhos.

Opções relacionadas a permissão:

- auth | permissão | Can add permission
- auth | permissão | Can change permission
- auth | permissão | Can delete permission
- auth | permissão | Can view permission

O grupo pode modificar suas permissões de acesso? Observe que todos estes cuidados são para proteger os departamentos de musicologia que desejarem utilizar o aplicativo resguardando seus trabalhos.

Opções relacionadas a inclusão de usuários:

- auth | usuário | Can add user
- auth | usuário | Can change user
- auth | usuário | Can delete user
- auth | usuário | Can view user

Um grupo pode incluir/excluir usuários? Um grupo principal (LAMUS) poderia fazer uma inclusão mais um subgrupo (estudos de retórica) não. O administrador do laboratório tem controle integral dos seus grupos de pesquisa inclusive de todo o trabalho realizado por cada integrante através dos relatórios gerados (item 01 - logs).

Opções relacionadas aos módulos do sistema:

- Cadastros_Gerais | Autores | Can add Autores
- Cadastros_Gerais | Autores | Can change Autores
- Cadastros_Gerais | Autores | Can delete Autores
- Cadastros_Gerais | Autores | Can view Autores
- Cadastros_Gerais | Instrumentos | Can add Instrumentos
- Cadastros_Gerais | Instrumentos | Can change Instrumento
- Cadastros_Gerais | Instrumentos | Can delete Instrumentos
- Cadastros_Gerais | Instrumentos | Can view Instrumentos
- Cadastros_Gerais | Familias de Instrumentos | Can add Familias de Instrumentos
- Cadastros_Gerais | Familias de Instrumentos | Can change Familias de Instrumentos
- Cadastros_Gerais | Familias de Instrumentos | Can delete Familias de Instrumentos
- Cadastros_Gerais | Familias de Instrumentos | Can view Familias de Instrumentos
- Cadastros_Gerais | Obras Musicais | Can add Obras Musicais
- Cadastros_Gerais | Obras Musicais | Can change Obras Musicais
- Cadastros_Gerais | Obras Musicais | Can delete Obras Musicais
- Cadastros_Gerais | Obras Musicais | Can view Obras Musicais (. . .)⁴⁹

Observe que temos a possibilidade de liberar ou restringir o acesso a qualquer aplicação do sistema. A escolha das permissões de adicionar registro, alterar, deletar ou visualizar fica a critério do administrador. Desta maneira possuímos um controle completo da utilização do Analysis Project.

6.1.8.2. Cadastro de Usuários do Sistema

Cadastrar usuários no sistema é função do administrador (geral) ou do grupo de pesquisa. Não aconselhamos deixar esta função livre para qualquer usuário.

⁴⁹ Finalizamos aqui nossa lista. Caso deseje visualizar ela completa acesse o ANEXO C. (Nota do Autor)

Figura 83 - Lista de usuários do sistema

Analysis Project

Início > Autenticação e Autorização > Usuários

Usuários

2 total

<input type="checkbox"/>	Usuário ^	Endereço de email	Primeiro nome	Último nome	Membro da equipe
<input type="checkbox"/>	dmneto	dmneto@usp.br	Diósnio	Machado Neto	✖
<input type="checkbox"/>	RobsonMaestro	robson_maestro@usp.br			✔

2 total

Fonte: Do Autor

A funcionalidade de cadastro de usuários possui algumas especificidades, contudo, é muito similar a função de cadastro de grupos de pesquisa.

Figura 84 - Cadastro de Usuários

Analysis Project

Início > Autenticação e Autorização > Usuários > dmneto

Modificar usuário

Usuário:
Obrigatório. 150 caracteres ou menos. Letras, números e @/!+/_ apenas.

Senha:
Senhas brutas não são armazenadas, então não há como visualizar a senha desse usuário, porém você pode mudar a senha usando [esse form](#).

Informações pessoais

Primeiro nome:

Último nome:

Endereço de email:

Permissões

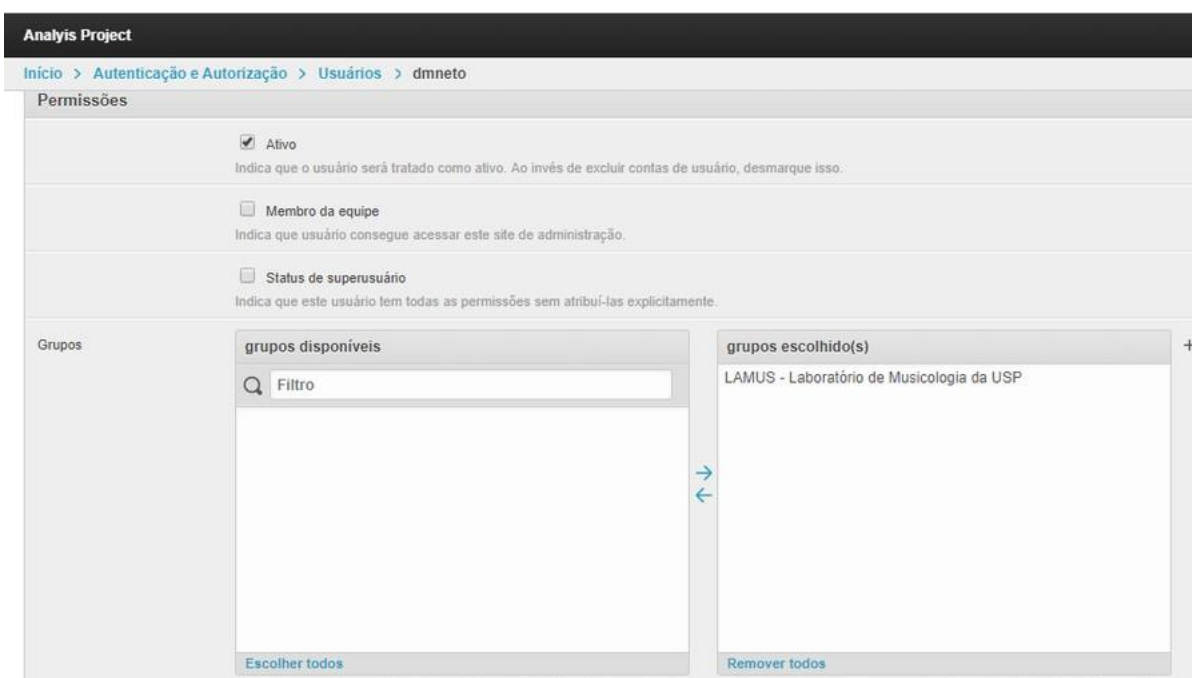
Ativo

Fonte: Do Autor

O Cadastro de usuários tem algumas opções que descrevemos a seguir

- Usuário (Escolha de um login para acesso a plataforma)
- Senha (Escolha de uma senha protegida por algoritmo - em nenhum local da aplicação podemos acessar a senha de forma bruta mais o usuário pode modificá-la quando desejar).
- Nome
- Último nome
- E-mail

Figura 85 - Seleção de permissões para o usuário



Fonte: Do Autor

Na seção de permissões temos as seguintes opções:

- Ativo
- Membro da Equipe (Caso habilitada poderá acessar o painel de controle).
- Super Usuário (Acesso irrestrito ao sistema)
- Grupos (Seleciona quais grupos ele pertence - todas as permissões e bloqueios do grupo serão herdados neste momento para este usuário.
- Permissões - Caso o administrador deseje oferecer uma permissão diferenciada para este usuário ele seleciona nesta aba.

- Data do último login
- Data quando o usuário foi inserido no sistema.

6.1.9. Relatórios Diversos

Esta opção permite a geração de diversos relatórios.

Figura 86 - Lista de relatórios diversos



Fonte: Do Auto

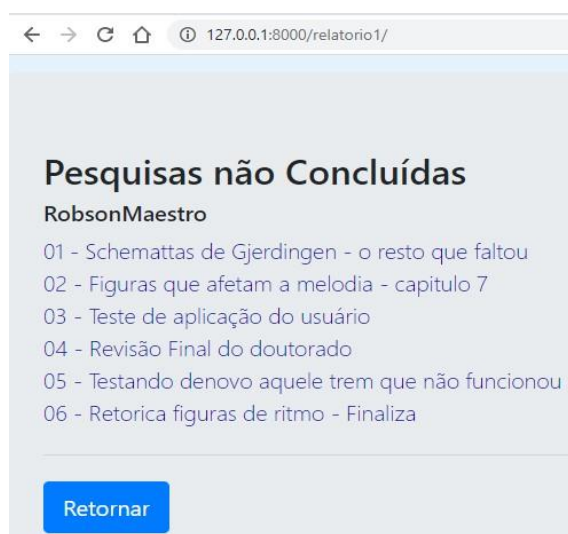
r

É uma opção em constante transformação. Deve ser flexível e se adaptar as necessidades que vão surgindo com a usabilidade do Analysis Project.

Pesquisas a serem verificadas

Com a utilização do Analysis Project sentimos a necessidade de um acesso rápido para as pesquisas que ainda não tinham sido concluídas.

Figura 87 - Relatório de pesquisas a serem verificadas



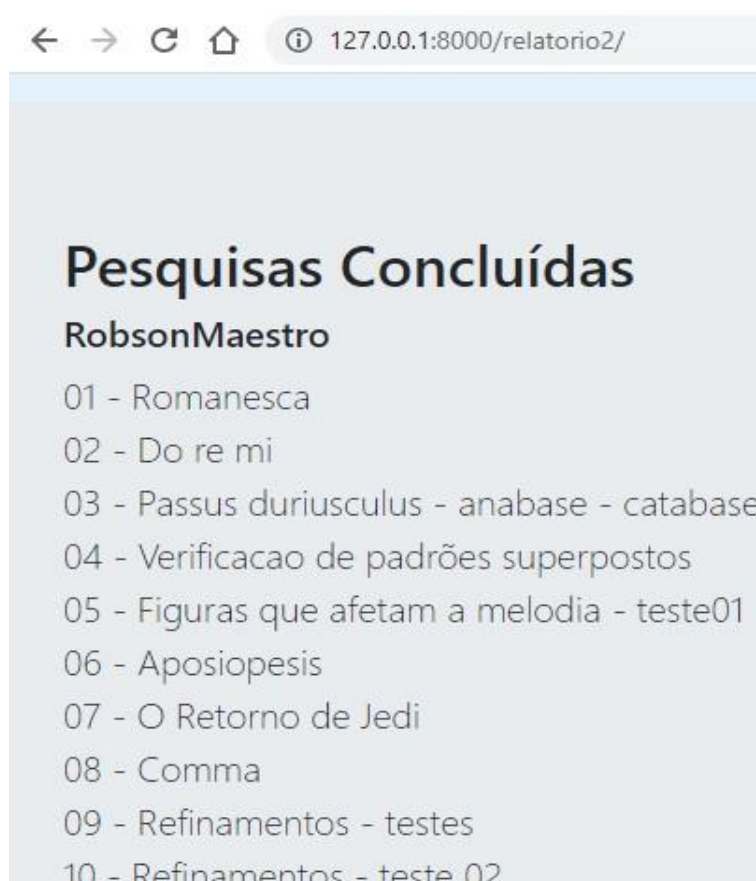
Fonte: Do Autor

Neste relatório é gerado uma lista com tópicos em formato de hipertexto (links) levando o musicólogo para a fase da pesquisa que está incompleta.

Pesquisas realizadas

Relatório que gera uma lista de trabalhos concluídos.

Figura 88 - Lista parcial dos trabalhos concluídos



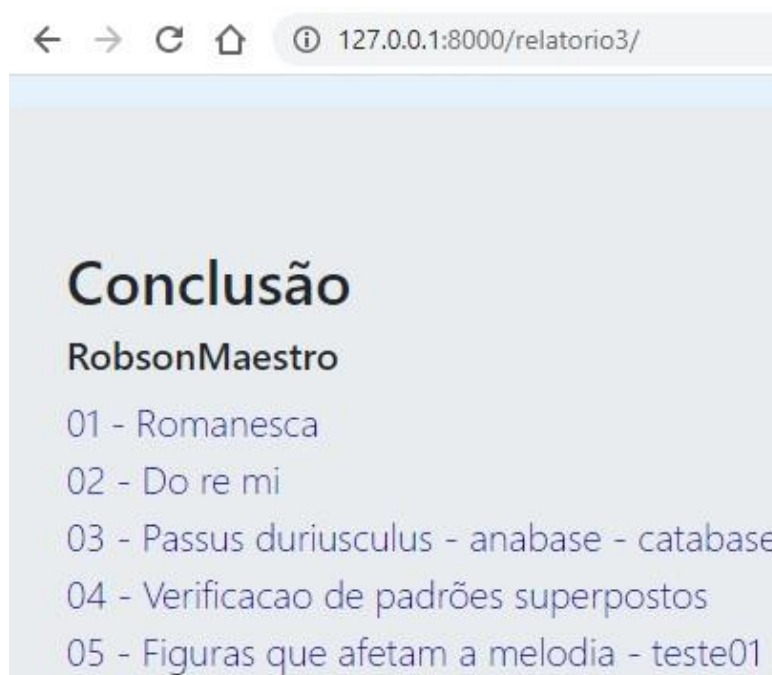
Fonte: Do Autor

Uma listagem simples sobre os trabalhos realizados na plataforma.

Relatórios de Conclusão

A mesma listagem anterior compõe este relatório, somente com o acréscimo de um hiperlink.

Figura 89 - Lista parcial das conclusões disponíveis para impressão



Fonte: Do Autor

Ao clicar no link o usuário é levado para o relatório de conclusão final⁵⁰. Finalizamos assim os relatórios que possuímos atualmente na plataforma.

6.1.10. Suporte da Aplicação (Help)

O Help é um descritivo de todos os recursos que existem na plataforma com dicas de utilização.

Figura 90 - Suporte da aplicação



Fonte: Do Autor

⁵⁰ Capítulo 03 item 3.1.4 (Nota do Autor)

Um recurso muito dinâmico e em constante transformação à medida que o aplicativo vai adicionando/subtraindo recursos⁵¹.

6.1.11. Ações Recentes

Neste tópico apresentado em tela temos as ações mais recentes realizadas na plataforma.

Figura 91 - Últimas Ações Realizadas



Fonte: Do Autor

Ao clicar nas ações o usuário é levado para a tela onde ocorreu as alterações facilitando assim o conhecimento do que foi modificado ou adicionado na plataforma por seu grupo de trabalho.

⁵¹ Não vamos adicionar o help neste trabalho, nem em seu corpo, nem como apêndice devido ao seu tamanho. Formatado tem um total de 307 páginas, aconselhamos sua leitura apenas on-line. (Nota do Autor)

7. REQUISITOS TÉCNICOS

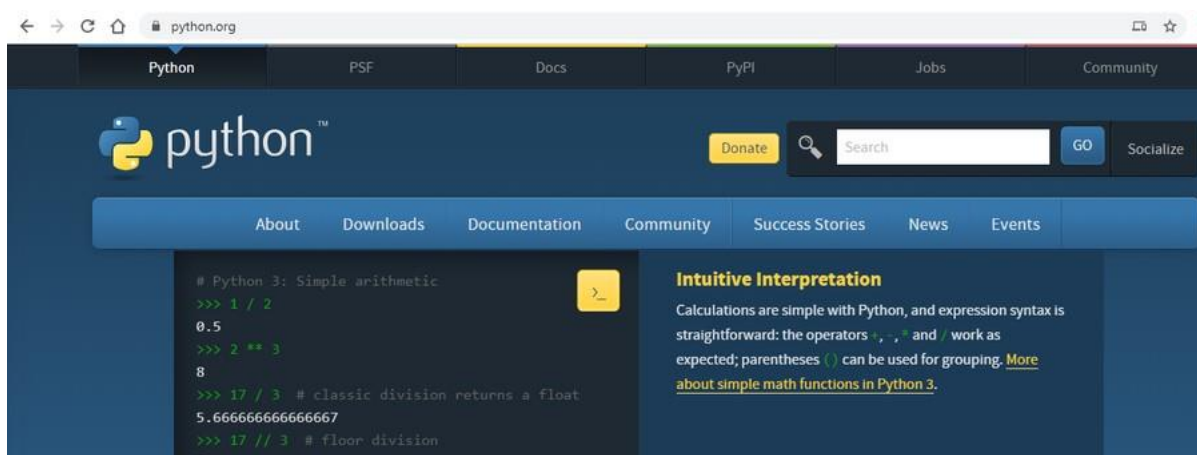
Sabemos que as páginas a seguir serão extremamente áridas para o musicólogo. Vamos discorrer sobre alguns requisitos técnicos utilizados para a implementação da aplicação e reconhecemos que o que descrevemos pertence a outra área do conhecimento. A musicologia atual requer a utilização de ferramentas tecnológicas e os laboratórios de musicologia precisarão de profissionais especialistas de outras áreas.

Aqui será exposto o mínimo necessário para a implantação do Analysis Project, as ferramentas e bibliotecas utilizadas.

7.1. Python – Principais características e usabilidade

Python⁵² é uma linguagem de programação lançada em 1991 por Guido Van Rossum. Sua execução se dá de forma interpretada via bytecode. Este recurso nos permite escrever todo o nosso projeto sem a preocupação com adaptações porquanto é executável em qualquer sistema operacional.

Figura 92 - Visualização da página oficial do python



Fonte: <https://www.python.org/>

A linguagem de programação possui 5 características básicas que passamos a transcrever:

⁵² Maiores informações acessar o site oficial da linguagem <https://www.python.org> (Nota do Autor)

¹¹ Sistema operacional (Nota do Autor)

1. “É uma linguagem de programação de alto nível (VHLL): possui um nível de abstração elevado. É uma linguagem mais próxima da linguagem humana.
2. Linguagem orientada a objetos: A POO foi desenvolvida na tentativa de aproximar o mundo real do virtual. O programador modela o que comumente chamamos de “mundo dos objetos” e explicita como estes objetos vão interagir. Assim ele determina quais mensagens cada objeto pode receber e qual a ação que aquele objeto terá de realizar.
3. Linguagem interpretada: necessita que o seu código fonte seja executado por um programa de computador (interpretador). Em seguida é executado pelo OS¹¹ ou pelo processador.
4. Linguagem procedural ou Imperativa: A programação imperativa descreve a computação como comandos que mudam o estado de um programa chamado de variáveis ⁵³⁵⁴. Em resumo um programa procedural é uma sequência de comandos para o computador executar. O nome “Imperativo” está ligado ao tempo verbal. A programação imperativa é baseada no conceito de Máquina de Alan Turing⁵⁵.

Figura 93 - Representação Artística da Máquina de Turing



Fonte: CEUAMI in Universidad Autónoma Metropolitana, México

Este conceito foi aprimorado por John Von Neumann e basicamente é a arquitetura dos computadores atuais. Aqui relatamos este fato, tão-somente para

⁵³ Acreditamos que o conceito de variáveis ficara bem claro quando gerarmos a nossa Conclusão (FASE

⁵⁴) do trabalho que é modificada na medida em que trocamos os parâmetros selecionados. (Nota do Autor)

⁵⁵ Abstração matemática que correspondem ao conjunto de funções computáveis. (nota do autor)

destacar a preferência dos programadores na escolha de linguagens procedurais. Neste modelo de arquitetura os dados como os programas são armazenados na mesma memória. A CPU que processa as instruções é separada da memória, assim os dados são transportados da memória para a CPU e os resultados devolvidos para a memória. As linguagens de programação imperativa contrastam com outras categorias de linguagem, tais como linguagens de programação declarativa, funcional e lógica. O presente trabalho não comporta maior discussão sobre o assunto que aqui colocado de forma ilustrativa, pois caracteriza a linguagem que estaremos utilizando e a maneira de construir o pensamento computacional.

5. Linguagem de tipagem dinâmica. A verificação dos tipos de dados é feita em tempo de execução que é feita por uma máquina virtual ou uma biblioteca. É uma linguagem de tipagem dinâmica o programador se preocupa mais com os tipos ⁵⁶.(Dias. R, 2015)

O motivo da escolha do desenvolvimento de nossa aplicação com python foi a comunicabilidade com o Music21. Este projeto desenvolvido no MIT em minha opinião é o aplicativo que possui as ferramentas mais importantes para a musicologia computacional atualmente e está em constante expansão.

7.2. Ambiente de desenvolvimento

O programador de forma geral trabalha com dois ambientes distintos:

Ambiente de desenvolvimento onde realiza os testes e verifica os erros de programação

Ambiente de produção onde a aplicação se encontra para acesso livre os usuários.

O Analysis Project passou por estas duas etapas. Hoje ele não está mais disponível em ambiente de produção por alguns motivos:

O projeto inicial utilizava o python em sua versão 2.7.3. Atualmente utilizamos a versão 3.7.2 deixando de ser compatível com o servidor anterior.

⁵⁶ Texto extraído de minha dissertação de mestrado. Julguei importante reproduzi-lo aqui em sua íntegra por ser simples e informativo na medida de nossa necessidade. Caso o leitor queira se aprofundar no assunto sugiro o acesso ao site oficial <https://www.python.org/> (Nota do autor)

Fizemos o deploy⁵⁷⁵⁸ de nossa aplicação no Heroku¹⁸. Ao realizarmos os testes iniciais, percebemos que os recursos existentes na plataforma não foram suficientes para que o Analysis Project rodasse corretamente. Detectamos uma enorme lentidão de resposta

Como solução acredito que a aplicação necessita de um servidor dedicado⁵⁹. Os altos custos do mercado inviabilizam agora a sua implantação.

No ambiente de desenvolvimento (minha máquina pessoal) existem ferramentas que auxiliam no controle dos erros do projeto. Demonstraremos aqui as opções que encontramos para desenvolver o projeto, sem entrarmos em questões relacionadas a juízos de valor. Temos a convicção que um programador mais hábil escolheria opções melhores.

7.2.1. Sistema operacional e instalações

Iniciamos o desenvolvimento do Analysis Project em 2013. Escolhemos como sistema operacional o Linux por sua facilidade de instalação, especificamente o Ubuntu⁶⁰ na versão 14.01. Por necessidades pessoais a aplicação atualmente se encontra estacionada em ambiente Windows (Windows 10) e passamos a descrever os passos utilizados para a implementação neste sistema operacional

⁵⁷ Colocamos o projeto em um servidor real (de produção) para acesso dos usuários. (Nota do Autor) ¹⁸ Heroku é uma plataforma em nuvem como um serviço que suporta várias linguagens de programação. É uma das primeiras plataformas em nuvem, o Heroku está em desenvolvimento desde junho de

⁵⁸, quando suportava apenas a linguagem de programação Ruby, mas agora suporta Java, Node.js, Scala, Clojure, Python, PHP e Go. (Tradução do Autor) Texto extraído de <https://en.wikipedia.org/wiki/Heroku> - acesso em 30/06/2019

⁵⁹ Em serviço de hospedagem dedicado o cliente aluga um servidor inteiro não compartilhado. Nele podemos implementar todos os recursos que são necessários para nossas necessidades. (Nota do Autor)

⁶⁰ “Ubuntu é um sistema operacional ou sistema operativo de código aberto, construído a partir do núcleo Linux, baseado no Debian e utiliza GNOME como ambiente de desktop de sua mais recente versão com suporte de longo prazo. É desenvolvido pela Canonical Ltd”. Texto extraído de <https://pt.wikipedia.org/wiki/Ubuntu> - acesso em 02/07/2019

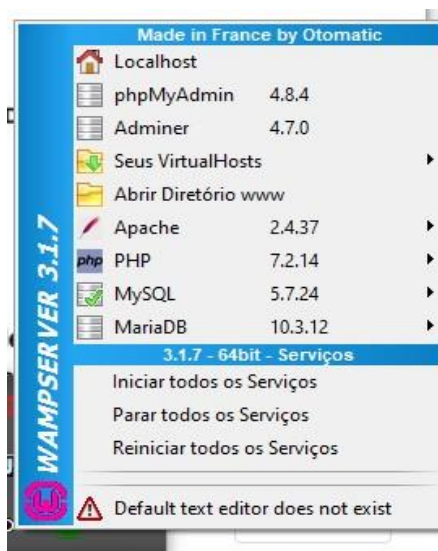
7.2.1.1. Instalação do WampServer

O WampServer⁶¹ é uma solução completa para desenvolvimento de aplicativos para web em um ambiente Windows. Possui integrado os seguintes recursos:

- Apache2
- PHP
- MySQL
- PhpMyAdmin (Gerenciador do banco de dados)

Sua instalação é extremamente simples. Baixe o arquivo de instalação disponível no site oficial na versão compatível com o seu sistema operacional. Instale normalmente como qualquer programa para Windows. Ao finalizar a instalação rode o programa.

Figura 94 - Painel de Controle do WampServer



Fonte: Do Autor

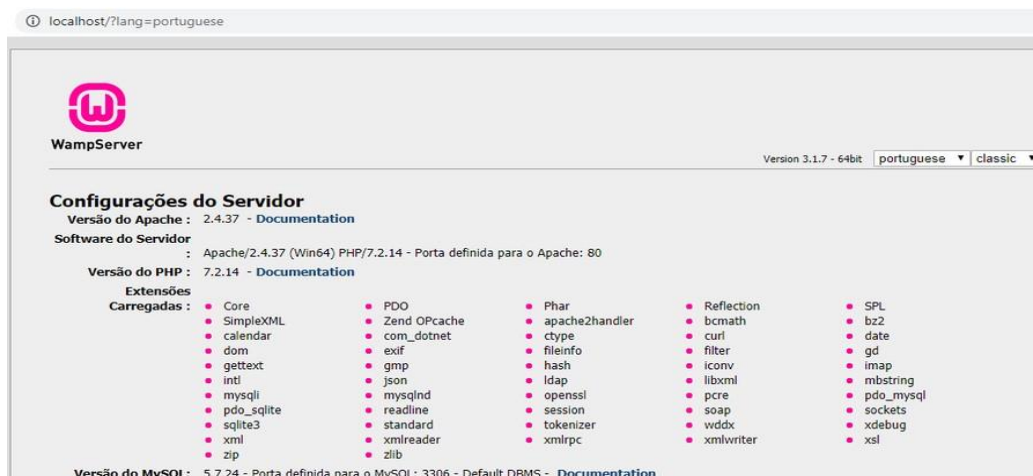
O WampServer quando roda corretamente inicializa 3 serviços:

- Servidor Apache
- PHP
- MySQL

⁶¹ Maires informações e download no site do desenvolvedor

<http://www.wampserver.com/en/downloadwampserver-64bits/> (Nota do Autor)

Figura 95 - Visualização da página oficial do WampServer



Fonte: Do Autor

Abra o navegador de sua preferência e digite localhost. Se o WampServer estiver rodando corretamente você visualizará a página acima⁶².

7.2.1.2. Instalação do Python

Igualmente a instalação do Python é simples. Baixe o arquivo de instalação disponível no site oficial na versão compatível com o seu sistema operacional. Instale normalmente como qualquer programa tomando alguns cuidados:

Na tela inicial de instalação existe a opção de instalar o python na variável de ambiente do Windows (PATH). Marque esta opção que permite que o comando python possa ser utilizado na linha de comando

A opção Install now (recomendada) instala todos os recursos no sistema inclusive o pip⁶³ que iremos precisar.

⁶² Todos os serviços disponíveis no WampServer podem ser acessados diretamente do ícone na barra de acesso. Utilize os botões (direito ou esquerdo) de seu mouse para ter as opções de configuração.

Uma dica é deixar o servidor como serviço na inicialização do computador. (Nota do Autor)

⁶³ Ferramenta de instalação das dependências do Python. (Nota do Autor)

Figura 96 - Verificação da versão do python

```
C:\ Prompt de Comando
Microsoft Windows [versão 10.0.18362.657]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.
C:\Users\robso>python --version
Python 3.7.2
```

Fonte: Do Autor

Abra o prompt de comando do Windows e digite o comando (python – version). Se a instalação estiver correta aparecerá a versão instalada em seu OS. O Analysis Project utiliza a versão 3.7.2.

7.2.2. Configurações do ambiente de desenvolvimento

Após a instalação dos aplicativos que vamos precisar devemos configurar o nosso ambiente de desenvolvimento. Recomendamos àqueles que desejam realizar um projeto com python, instalar um ambiente virtual (Virtualenv) e a partir dele definir os procedimentos e bibliotecas que irá utilizar. Agindo desta maneira conseguimos criar um espaço de trabalho limpo podendo instalar versões diferentes do python, django, ou qualquer aplicação que desejar⁶⁴.

7.2.2.1. Instalação do Virtualenv

O primeiro passo é a instalação do aplicativo Virtualenv

⁶⁴ O Analysis Project poderia ter mantido a versão do python 2.7 que foi utilizada na sua programação inicial. Não adotamos esta medida devido a incompatibilidade do Django que atualmente suporta

Figura 97 - Comando para instalação da Virtualenv

```

Prompt de Comando
Microsoft Windows [versão 10.0.18362.657]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\robso>pip install virtualenv
Requirement already satisfied: virtualenv in c:\python37-32\lib\site-packages (20.0.5)
Requirement already satisfied: distlib<1,>=0.3.0 in c:\python37-32\lib\site-packages (from virtualenv) (0.3.0)
Requirement already satisfied: importlib-metadata<2,>=0.12; python_version < "3.8" in c:\python37-32\lib\site-packages (from virtualenv) (1.5.0)
Requirement already satisfied: filelock<4,>=3.0.0 in c:\python37-32\lib\site-packages (from virtualenv) (3.0.12)
Requirement already satisfied: six<2,>=1.9.0 in c:\python37-32\lib\site-packages (from virtualenv) (1.12.0)
Requirement already satisfied: appdirs<2,>=1.4.3 in c:\python37-32\lib\site-packages (from virtualenv) (1.4.3)
Requirement already satisfied: zipp>=0.5 in c:\python37-32\lib\site-packages (from importlib-metadata<2,>=0.12; python_version < "3.8"->virtualenv) (3.0.0)

C:\Users\robso>

```

Fonte: Do Autor

Utilizando o comando (pip install Virtualenv) no prompt de comando instalamos o Virtualenv em nosso sistema.

Figura 98 - Comando para a instalação do virtualenvwrapper-win

```

Prompt de Comando
C:\Users\robso>pip install virtualenvwrapper-win
Requirement already satisfied: virtualenvwrapper-win in c:\python37-32\lib\site-packages (1.2.6)
Requirement already satisfied: virtualenv in c:\python37-32\lib\site-packages (from virtualenvwrapper-win) (20.0.5)
Requirement already satisfied: importlib-metadata<2,>=0.12; python_version < "3.8" in c:\python37-32\lib\site-packages (from virtualenv->virtualenvwrapper-win) (1.5.0)
Requirement already satisfied: appdirs<2,>=1.4.3 in c:\python37-32\lib\site-packages (from virtualenv->virtualenvwrapper-win) (1.4.3)
Requirement already satisfied: six<2,>=1.9.0 in c:\python37-32\lib\site-packages (from virtualenv->virtualenvwrapper-win) (1.12.0)
Requirement already satisfied: filelock<4,>=3.0.0 in c:\python37-32\lib\site-packages (from virtualenv->virtualenvwrapper-win) (3.0.12)
Requirement already satisfied: distlib<1,>=0.3.0 in c:\python37-32\lib\site-packages (from virtualenv->virtualenvwrapper-win) (0.3.0)
Requirement already satisfied: zipp>=0.5 in c:\python37-32\lib\site-packages (from importlib-metadata<2,>=0.12; python_version < "3.8"->virtualenv->virtualenvwrapper-win) (3.0.0)

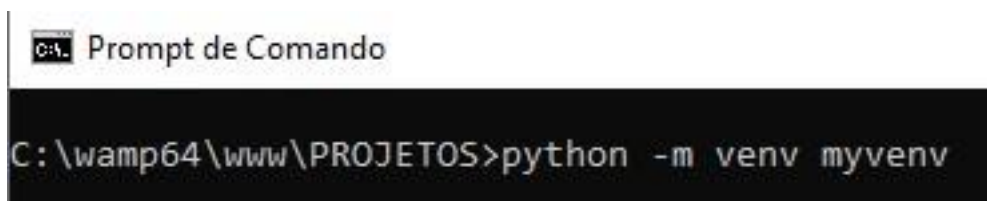
C:\Users\robso>

```

Fonte: Do Autor

Utilizando o comando (pip install virtualenvwrapper-win) instalamos o virtualenvwrapperwin, um conjunto de ferramentas para manipular ambientes virtuais.

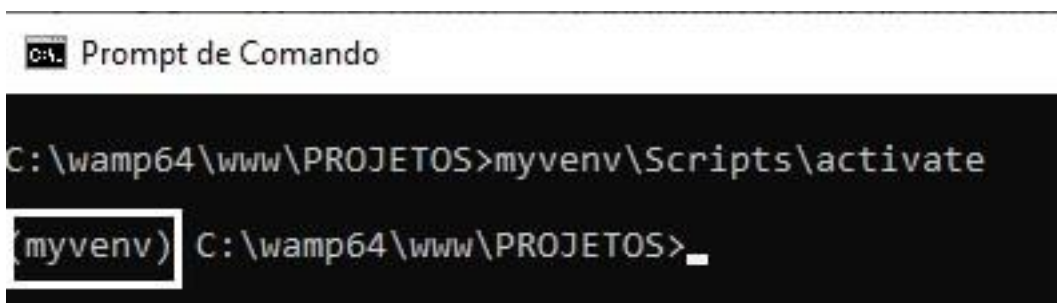
Fisicamente o meu ambiente web está localizado no endereço c:\wamp64\www. Dentro desta pasta criei uma outra pasta chamada PROJETOS. Desejo instalar meu ambiente virtual dentro desta pasta.

Figura 99 - Comando para a criação do ambiente virtualA screenshot of a Windows Command Prompt window. The title bar reads "C:\ Prompt de Comando". The command prompt shows the current directory as "C:\wamp64\www\PROJETOS" and the command "python -m venv myenv" being entered.

```
C:\wamp64\www\PROJETOS>python -m venv myenv
```

Fonte: Do Autor

Utilizando o comando (`python -m venv myenv`) crio um diretório de nome `myenv` onde estaciono todas as versões que desejo dos aplicativos. A expressão “`myenv`” pode ser substituída por aquela que o programador achar mais conveniente.

Figura 100 - Ativando o ambiente virtualA screenshot of a Windows Command Prompt window. The title bar reads "C:\ Prompt de Comando". The command prompt shows the current directory as "C:\wamp64\www\PROJETOS" and the command "myenv\Scripts\activate" being entered. The prompt changes to "(myenv) C:\wamp64\www\PROJETOS>".

```
C:\wamp64\www\PROJETOS>myenv\Scripts\activate
(myenv) C:\wamp64\www\PROJETOS>
```

Fonte: Do Autor

Utilizando o comando (`myenv\Scripts\activate`) ativamos nosso ambiente virtual. Observe que antes do caminho do diretório aparece o nome de nossa Virtualenv entre parênteses (`myenv`).

7.2.2.2. Instalação do Django

O painel de administração de nossa aplicação é desenvolvido em Django.

Figura 101 - Comando de instalação do Django



```
C:\> Prompt de Comando  
(myvenv) C:\wamp64\www\PROJETOS>pip install django
```

Fonte: Do Autor

Utilizando o comando (`pip install django`) instalamos a última versão em nosso ambiente virtual.

7.3. Framework Django e suas aplicações

Django é um framework escrito em Python. Utiliza o padrão MTV (Model, Template, View). Criado em sua origem como sistema de gerenciamento jornalístico por () para o jornal () na cidade de Lawrence, no Kansas. Em 2005 foi publicado com código aberto sob a licença BSD. O Nome Django se deve ao fato de um dos seus programadores gostar do guitarrista de jazz Django Reinhardt.

Principais características do framework Django:

- 1 Mapeamento Objeto-Relacional (ORM): A modelagem de dados é definida através de classes Python sendo possível gerar as tabelas que utilizaremos em nossa aplicação sem a necessidade da utilização de comandos SQL[NRP57].
- 2 Django-admin: Uma das maiores funcionalidades do Django, pelo menos para o nosso projeto. A Interface administrativa é criada automaticamente e assim temos um gerenciador automático das tabelas que criamos.
- 3 Formulários: É possível a geração dos formulários através das nossas tabelas
- 4 URLs Elegantes: Com o Django conseguimos direcionar nossas páginas utilizando nomes mais simples o que facilita ao usuário e aos motores de busca.
- 5 Sistema de Templates: Com o sistema de templates do Django podemos separar os códigos html[NRP58], css[NRP59], js[NRP61], e py[NRP60].

- 6 Internacionalização: Suporte para aplicações multi-idioma que permite que o programador defina tags para tradução.
- 7 Sistema de Cache O Django possui um sistema de cache que se integra ao memcached[NRP62] ou em outros frameworks de cache.

A instalação do Django é simples e deve ser realizada dentro da nossa máquina virtual como descrito anteriormente.

Com o Framework instalado precisamos de criar a aplicação. O local físico onde ela estará estacionada é `c:\wamp64\www\PROJETOS\` o mesmo diretório da nossa Virtualenv.

Figura 102 - Comando para criar uma aplicação em Django

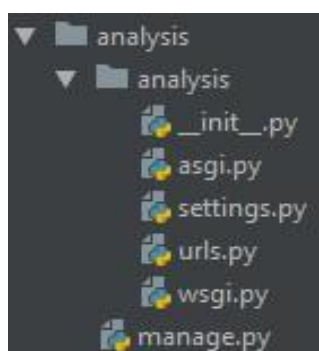


```
C:\> Prompt de Comando  
  
(myvenv) C:\wamp64\www\PROJETOS>django-admin startproject analysis
```

Fonte: Do Autor

Ao executar o comando acima, construímos uma aplicação em Django com o nome “analysis”. O django cria um diretório chamado analysis onde ficará depositado os arquivos do nosso projeto.

Figura 103 - Estrutura de um projeto Django



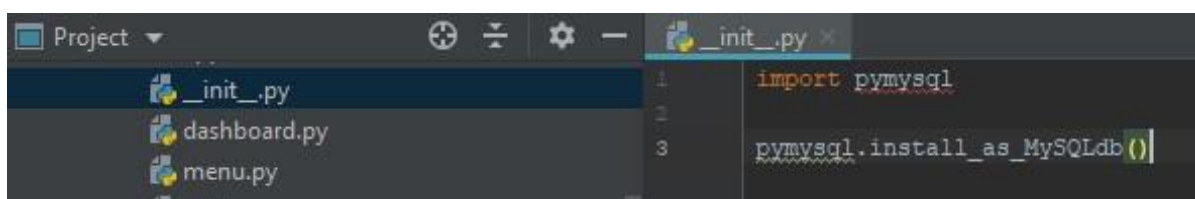
Fonte: Do Autor

No diretório principal temos o arquivo `manage.py`. É o arquivo gerenciador do nosso projeto. Ele gera as tabelas do banco de dados, realiza as atualizações necessárias⁶⁵.

O Django criou um diretório principal chamado `analysis` e dentro dele, outro de mesmo nome. No subdiretório `analysis` encontramos alguns arquivos. Descreveremos agora suas funções:

Arquivo `_init_.py` - Até a versão do Python 2.7, para que os módulos de um diretório fossem reconhecidos como um pacote python necessitávamos deste arquivo. Era um arquivo vazio. A partir da versão 3.3 não é mais necessário utilizar este arquivo para esta finalidade. Dentro do diretório ele é sempre o primeiro a ser executado sendo útil para a definição de variáveis globais ou funções.

Figura 104 - Arquivo `_init_.py`



Fonte: Do Autor

No caso do `Analysis Project` utilizo para agilizar a conexão com o banco de dados `MySQL`.

Arquivo `settings.py`. - Este é o arquivo das configurações gerais no nosso projeto. Objetivando o correto funcionamento do `Analysis Project` precisamos realizar algumas configurações. Abaixo relacionamos as mais importantes:

- Indicar nossa base de dados
- Configurar corretamente a localização física dos nossos diretórios `static`, `media` e `templates`
- Configurar o `TinyMCE` que será muito valioso para nossa aplicação.
- Configurar o `grapelli` que é um `Skin` para que a área de administração do projeto fique com aparência mais agradável.
- Configurar as nossas `APPS` que são as aplicações que desenvolvemos.

⁶⁵ Para maiores informações sobre os comandos aceitos pelo gerenciador de aplicação digite: `python manage.py --help` (Nota do Autor)

Figura 105 - Arquivo settings.py

```

26
27 # APLICACOES GRAPPELLI (NECESSITA ESTAR ANTES DE 'django.contrib.admin',)
28
29 'django.contrib.contenttypes',...# POSICIONAMENTO NECESSARIO PARA O GRAPPELLI
30 'grappelli.dashboard',
31 '#grappelli.menu'...# CÓDIGO ULTRAPASSADO
32 'grappelli',
33
34 # APLICACOES DJANGO
35 'django.contrib.admin',
36 'django.contrib.auth',
37 '#django.contrib.contenttypes',
38 'django.contrib.sessions',
39 'django.contrib.messages',
40 'django.contrib.staticfiles',
41
42 # APLICACOES DE TERCEIROS
43 'laravel',...# editor do PROJETO
44
45 # APLICACOES ANALYSIS PROJECT

```

Fonte: Do Autor

Arquivo views.py - onde construímos as funções que nos direcionam para nossos templates.

Figura 106 - Arquivo views.py

```

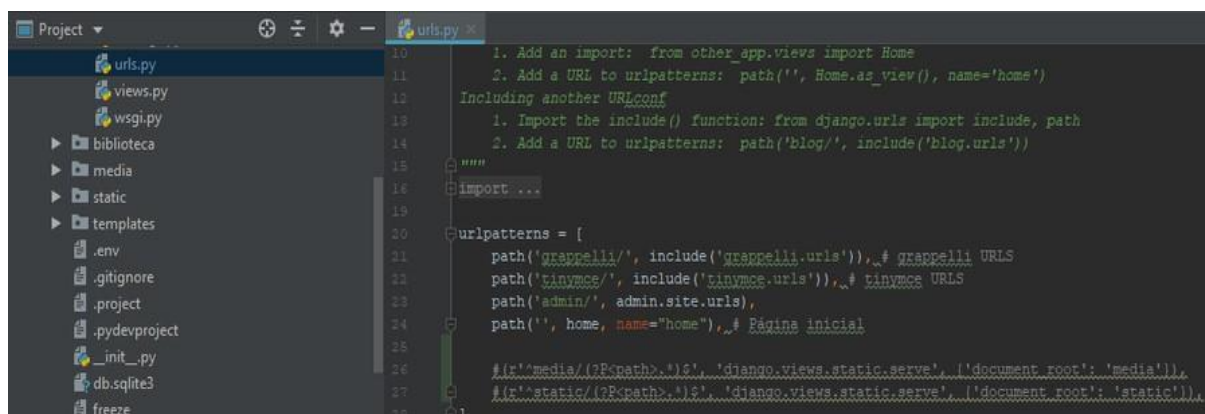
1 # -*- coding: utf-8 -*-
2 # VIEWS DO PROJETO ANALYSIS PROJECT
3
4 from django.shortcuts import render
5
6
7 def home(request):...# ENVIA PARA A PÁGINA INICIAL
8     return render(request, 'index.html')
9

```

Fonte: Do Autor

urls.py é responsável pela definição das nossas URLs (conceito de Url elegante) e para onde iremos direcioná-las.

Figura 107 - Arquivo urls.py



```

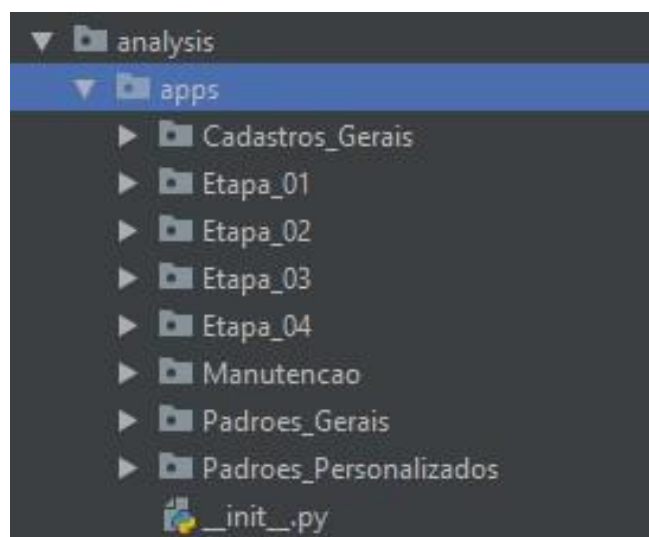
10 1. Add an import: from other_app.views import Home
11 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13 1. Import the include() function: from django.urls import include, path
14 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 import ...
17
18 urlpatterns = [
19     path('grappelli/', include('grappelli.urls')), _# grappelli URLs
20     path('tinymce/', include('tinymce.urls')), _# tinymce URLs
21     path('admin/', admin.site.urls),
22     path('', home, name="home"), _# Página inicial
23
24     #('media/(?P<path>.*)$', 'django.views.static.serve', {'document_root': 'media'}),
25     #('static/(?P<path>.*)$', 'django.views.static.serve', {'document_root': 'static'})
26 ]

```

Fonte: Do Autor

Conhecendo melhor a arquitetura de um projeto Django, temos como criar nossas aplicações. As aplicações (APPS) são os programas pessoais que desenvolvemos para que realizem as tarefas de que necessitamos. Uma característica interessante é que quando criamos uma APPS podemos reutilizá-las em outros projetos. Antes de criarmos as nossas aplicações, criamos um diretório chamado (apps) para melhor organizar o nosso projeto.

Figura 108 - Diretório das aplicações pessoais

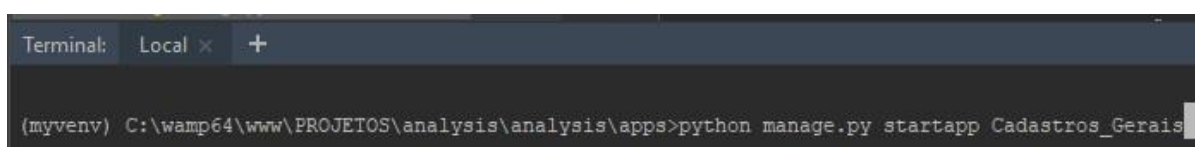


Fonte: Do Autor

Criamos no diretório apps as seguintes aplicações:

- Cadastros_Gerais
- Etapa_01
- Etapa_02
- Etapa_03
- Etapa_04
- Manutencao
- Padroes_Gerais
- Padroes_Personalizados

Figura 109 - Comando para a criação de apps.

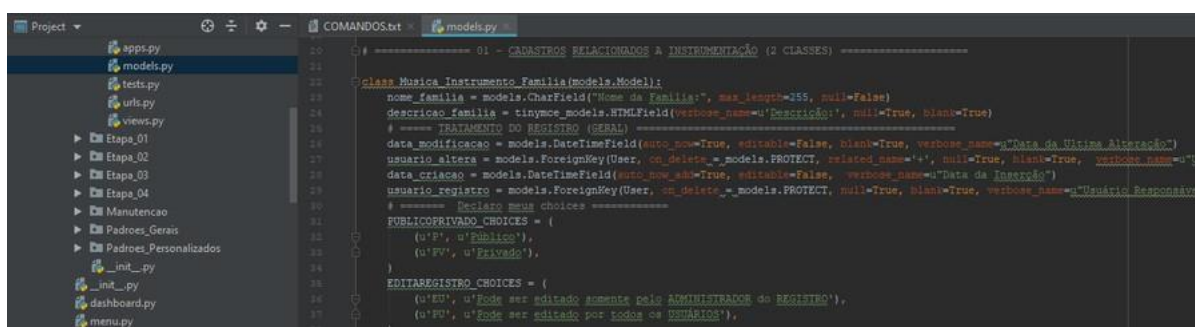


```
Terminal: Local x +
(myvenv) C:\wamp64\www\PROJETOS\analysis\analysis\apps>python manage.py startapp Cadastros_Gerais
```

Fonte: Do Autor

Dentro dos diretórios de nossas APPS a programação é feita basicamente em dois arquivos: O arquivo models.py e onde definimos nossas tabelas, seus campos e relacionamentos. É basicamente nossos modelos, a estrutura de programação do projeto

Figura 110 - Exemplo de arquivo models.py



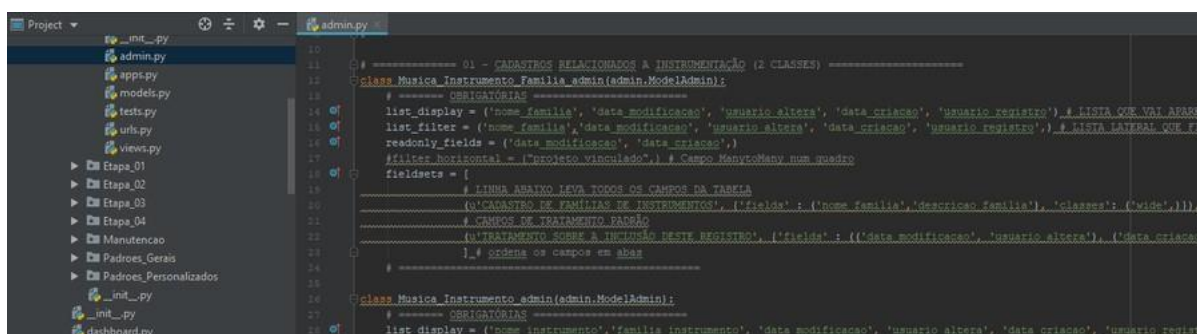
```
Project v COMANDOS.txt models.py
10 # ----- 01 - CADASTROS RELACIONADOS A INSTRUMENTAÇÃO (2 CLASSES) -----
11
12 class Musica_Instrumento_Familia(models.Model):
13     nome_familia = models.CharField("Nome da Família", max_length=255, null=False)
14     descricao_familia = tinymce_models.HTMLField(verbose_name="Descrição", null=True, blank=True)
15     # ----- TRATAMENTO DO REGISTRO (GERAL) -----
16     data_modificacao = models.DateTimeField(auto_now=True, editable=False, blank=True, verbose_name="Data da Última Alteração")
17     usuario_altera = models.ForeignKey(User, on_delete=models.PROTECT, related_name="+", null=True, blank=True, verbose_name="Usuário Responsável")
18     data_criacao = models.DateTimeField(auto_now_add=True, editable=False, verbose_name="Data da Inserção")
19     usuario_registro = models.ForeignKey(User, on_delete=models.PROTECT, null=True, blank=True, verbose_name="Usuário Responsável")
20     # ----- Declaro meus choices -----
21     PUBLICOPRIVADO_CHOICES = (
22         ('P', 'Público'),
23         ('PV', 'Privado'),
24     )
25     EDITAREGISTRO_CHOICES = (
26         ('EU', 'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
27         ('TV', 'Pode ser editado por todos os USUÁRIOS'),
28     )
```

Fonte: Do Autor

O arquivo admin.py é um arquivo onde declaramos quais as nossas aplicações que foram definidas no arquivo models.py serão utilizadas na administração.

Escrevemos os códigos informando ao Django-admin como queremos que ele trate cada uma das aplicações. Abaixo um exemplo utilizado no Analysis Project

Figura 111 - Exemplo de arquivo admin.py



```

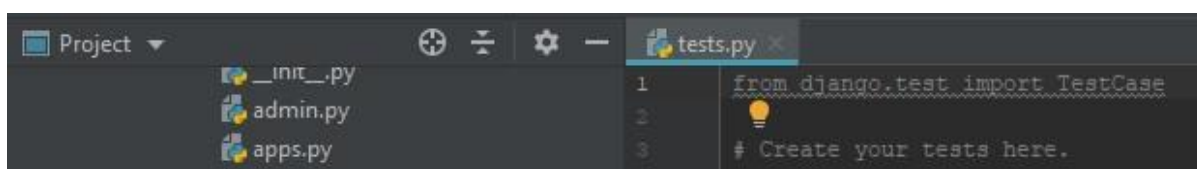
10 # ===== 01 - CADASTROS RELACIONADOS A INSTRUMENTAÇÃO (2 CLASSES) =====
11
12 class MusicaInstrumentoFamiliaAdmin(admin.ModelAdmin):
13     # OBRIGATORIAS
14     list_display = ('nome_familia', 'data_modificacao', 'usuario_altera', 'data_criacao', 'usuario_registro') # LISTA COM VOLTAR APARE
15     list_filter = ('nome_familia', 'data_modificacao', 'usuario_altera', 'data_criacao', 'usuario_registro') # LISTA LATERAL QUE FIL
16     readonly_fields = ('data_modificacao', 'data_criacao',)
17     filters_horizontal = ('data_modificacao',) # Campo MenorMany com quadro
18     fieldsets = [
19         # LITTA ABAIXO LEVA TODOS OS CAMPOS DA FAMILIA
20         (CADASTRO DE FAMILIAS DE INSTRUMENTOS', ('fields': ('nome_familia', 'descricao_familia', 'classes': ('wide',))),
21         # CAMPOS DE TRATAMENTO FACIL
22         (TRATAMENTO SOBRE A INCLUSÃO NESTE REGISTRO', ('fields': (('data_modificacao', 'usuario_altera'), ('data_criacao',
23         ] # ordena os campos em abas
24
25
26 class MusicaInstrumentoAdmin(admin.ModelAdmin):
27     # OBRIGATORIAS
28     list_display = ('nome_instrumento', 'familia_instrumento', 'data_modificacao', 'usuario_altera', 'data_criacao', 'usuario_regist

```

Fonte: Do Autor

Outro arquivo criado com as APPS é o teste.py que serve para realizarmos testes na nossa aplicação. Particularmente nunca o utilizamos.

Figura 112 - Exemplo de arquivo test.py



```

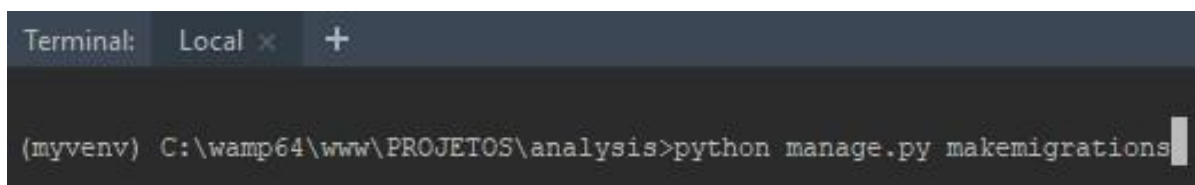
1 from django.test import TestCase
2
3 # Create your tests here.

```

Fonte: Do Autor

O início de qualquer projeto de programação é a modelagem de dados. Basicamente é a lógica necessária para relacionar os nossos registros que são guardados em tabelas num banco. Existem diversas ferramentas que auxiliam no processo de modelagem, contudo, as utilizamos. Escrevemos nossa aplicação diretamente nos arquivos models.py⁶⁶.

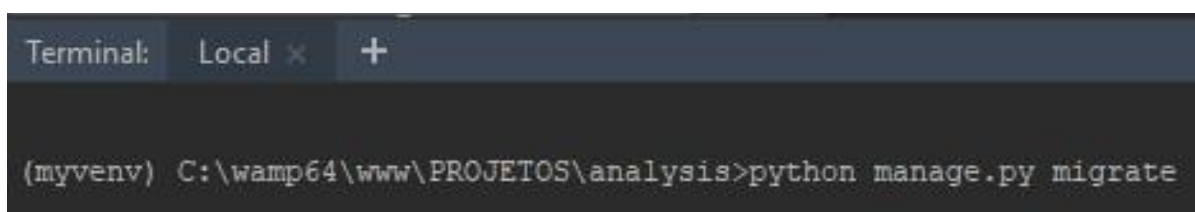
⁶⁶ Disponibilizei alguns arquivos e programação no APÊNDICE A. Infelizmente não seria possível disponibilizar todo o código por questão de espaço e ainda a incerteza sobre o tipo de licença de distribuição que será adotada para o Analysis Project.

Figura 113 - Comando makemigrations

```
Terminal: Local x +  
(myvenv) C:\wamp64\www\PROJETOS\analysis>python manage.py makemigrations
```

Fonte: Do Autor

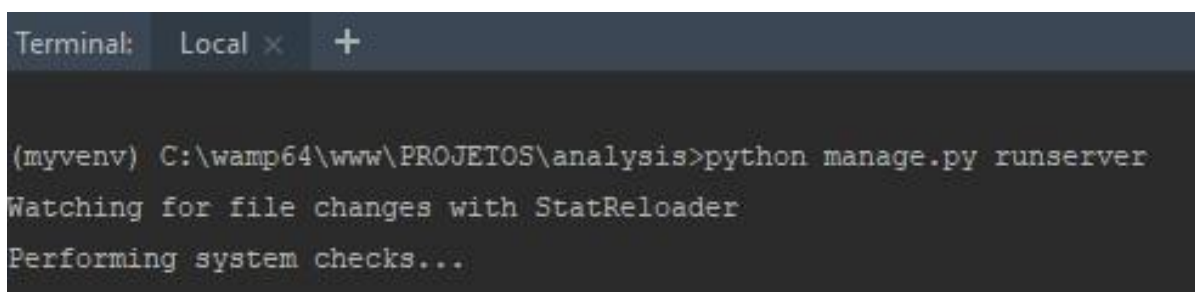
Depois de terminada toda a modelagem de dados, é necessário rodar nossa aplicação. Utilizamos o comando makemigrations para organizar os campos das nossas tabelas.

Figura 114 - Comando Migrate

```
Terminal: Local x +  
(myvenv) C:\wamp64\www\PROJETOS\analysis>python manage.py migrate
```

Do Autor

Com o comando migrate atualizamos o nosso banco de dados.

Figura 115 - Comando runserver

```
Terminal: Local x +  
(myvenv) C:\wamp64\www\PROJETOS\analysis>python manage.py runserver  
Watching for file changes with StatReloader  
Performing system checks...
```

Do Autor

Utilizamos o comando runserver para rodar o servidor nativo do django. Estamos prontos para começar a testar nosso aplicativo e inserir os dados em nosso banco.

7.4. Análise de desempenho

Possuímos uma diversidade de componentes na “web” (navegadores, servidores. . .) o que torna o serviço de monitoração uma tarefa complexa.

De modo a solucionar estas questões desenvolveu-se um conjunto de técnicas de monitoramento como os benchmarks (máquinas que enviam requisições a servidores e coletam dados de desempenho). No presente estudo trabalharemos com o httpperf e ele estará focado na análise através de ferramentas de benchmarks.

Um servidor (web) é composto de 4 partes:

- Plataforma de Hardware
- Sistema Operacional
- Software Servidor
- Conteúdo

As duas métricas mais importantes na monitoração de sistemas “web” são:

- Tempo de resposta (medida de maior interesse para o usuário do sistema)
- Taxa de Processamento (maior interesse para os administradores do sistema)

O desempenho de um sistema (web) depende muito das características de sua carga de trabalho. Estas características são determinadas pelo tempo de chegada, de conclusão da requisição, da CPU e tamanho do objeto solicitado. Precisamos então caracterizar a carga de trabalho. Tratamos de algumas características que devem ser consideradas:

- Tamanho do arquivo
- Distribuição da popularidade
- Autossimilaridade do tráfego (web)
- Localidade de referência
- Padrões de requisições do usuário, entre outras.

O motivo da observação acima é que o trabalho realizado com os benchmarks são em ambientes controlados e empregam (geralmente) uma carga de trabalho padrão que incluem páginas (HTML) geradas nas modalidades estática e dinâmica.

O Objetivo principal do trabalho é monitorar o sistema onde será implementada a aplicação Analysis Project. O sistema estudado é um sistema real, assim devemos considerar alguns fatores que podem atrapalhar ou “mascarar os resultados obtidos:

- As requisições podem demorar mais tempo que o necessário devido a alguma anomalia no sistema.
- Como o sistema é de produção, o acesso simultâneo como (motores de busca, usuários logados, requisições em espera, etc. . .) podem distorcer a informação recebida pelo httpperf.
- O sistema é compartilhado com outros usuários (hospedagem tradicional no servidor da HOMEHOST) assim não sabemos como ele irá se comportar mediante ao acesso de outros usuários a outras aplicações que não se relacionam diretamente com o Analysis Project.

7.4.1. Ferramentas utilizadas para análise de desempenho

Para realizarmos a análise de desempenho do nosso projeto, necessitamos da utilização de ferramentas que passamos a descrever de forma sucinta a sua funcionalidade:

7.4.1.1. httpperf

“ Httpperf - uma ferramenta para medir o desempenho do servidor web. Ele fornece uma facilidade flexível para gerar várias cargas de trabalho HTTP e para medir o desempenho do servidor. O foco da httpperf não está na implementação de um benchmark particular, mas no fornecimento de uma ferramenta robusta e de alto desempenho que facilita a construção de benchmarks de micro e macro nível. As três características distintivas do httpperf são a sua robustez, que inclui a capacidade de gerar e sustentar a sobrecarga do servidor, o suporte para os protocolos HTTP / 1.1 e SSL e a sua extensibilidade para a nova carga de trabalho geradores e medições de desempenho ”.

Fonte:<http://shiftright.com/mirrors/www.hpl.hp.com/research/linux/httpperf/index.html>

7.4.1.2. Escopo Padrão de Httpperf

É importante notar que, por padrão, o httpperf apenas testa a carga útil. Ele não carrega recursos (imagens, Java Script ou css) por padrão. É um excelente programa para execução de testes em servidores (web). Abre uma conexão na porta 80 da máquina alvo oferecendo múltiplas conexões, com diversas opções, tais como sessões, conexões seguras, cookies. Para realizar um benchmark para descobrir o real desempenho do seu servidor (web), devemos executar o httpperf de forma sincronizada, a partir de algumas máquinas distintas. Abaixo um exemplo de utilização básica:

```
httpperf - servidor analysisporject.org --port 80 --num-conns 10 --rate 1
```

Figura 116 - Conexões no servidor do domínio analysisproject.org

```
robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org --port 80 --num-conns 10 --rate 1
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --rate=1 --send-buffer=4096 --recv-buffer=16384
--num-conns=10 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 1

Total: connections 10 requests 10 replies 10 test-duration 9.001 s

Connection rate: 1.1 conn/s (900.1 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 1.0 avg 34.1 max 325.7 median 1.5 stddev 102.5
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 1.1 req/s (900.1 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 1.0 avg 1.0 max 1.0 stddev 0.0 (1 samples)
Reply time [ms]: response 33.9 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=10 3xx=0 4xx=0 5xx=0

CPU time [s]: user 1.36 system 7.52 (user 15.1% system 83.6% total 98.7%)
Net I/O: 12.6 KB/s (0.1*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
```

Fonte: Do Autor

Neste exemplo realizamos uma sequência de 10 conexões em meu domínio (analysisproject.org) com uma taxa de uma conexão por segundo (rate 1) Algumas considerações sobre os dados obtidos:

“Taxa de conexão” - Quando não passa o parâmetro “--rate”, as conexões são enviadas o mais rápido possível.

“Tempo de conexão [ms]” - É o processamento de várias métricas relacionadas ao teste em relação à sua carga básica.

“Tamanho da resposta [ms]” - isso é útil quando se testam mudanças que são orientadas para reduzir a carga básica de sua aplicação.

7.4.2. Etapas para a realização da análise do servidor

Precisamos determinar os possíveis problemas que encontraremos para averiguar nossas necessidades de implementação.

7.4.2.1. O que desejamos testar?

Possuímos um banco de dados com cerca de 50,000 obras, sendo que cada uma possui 3 **megabyte** (formato). O Analysis Project foi desenvolvido para ser utilizados por laboratórios de musicologia, assim acreditamos que nunca possuiríamos acesso simultâneo de mais de 100 usuários.

A nossa carga média de usuários seria de cerca de 10 usuários realizando buscas em 2000 obras que gerariam também de forma aproximada 20,000 a 30,000 requisições no total, assim cada usuário geraria um total de 2000 requisições.

1) EXPERIMENTO 01: O servidor poderia realizar satisfatoriamente as requisições que a aplicação Analysis Project necessitaria?

DESEJÁVEL: Tempo de pesquisa inferior a 10 segundos.

NÚMERO DE USUÁRIOS CONECTADOS - 10

Como realizar esta busca utilizando o httpperf?

Utilizaremos o comando abaixo:

\$ httpperf - servidor analysisproject.org -num-conns 2000 -num-calls=10 onde o [-num-conns 2000] representa o número de requisições por usuário e [-num-calls=10] representaria o número de chamadas.

Figura 117 - Primeiro teste no servidor

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org --num-conns 2000 --num-calls=10
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --send-buffer=4096 --recv-buffer=16384 --num-co
nns=2000 --num-calls=10
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 1

Total: connections 2000 requests 20000 replies 20000 test-duration 9.694 s

Connection rate: 206.3 conn/s (4.8 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 3.8 avg 4.8 max 20.1 median 4.5 stddev 1.9
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 10.000

Request rate: 2063.0 req/s (0.5 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 2067.6 avg 2067.6 max 2067.6 stddev 0.0 (1 samples)
Reply time [ms]: response 0.4 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=20000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 1.24 system 7.68 (user 12.8% system 79.2% total 92.0%)
Net I/O: 23447.0 KB/s (192.1*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Analisando rapidamente os resultados conclui que para o nosso servidor responder as 20,000 requisições ele gastou 9,694 segundos. Cada usuário seria atendido em 0,96 segundos, um tempo de espera dez vezes inferior ao que esperávamos. O servidor onde estacionamos a aplicação para o Analysis Project atende perfeitamente as necessidades do projeto neste quesito.

2) EXPERIMENTO 02: Qual seria o limite máximo de *usuários* que este sistema poderia suportar considerando uma carga de trabalho mediana / Tempo de resposta?

DESEJÁVEL: >100.

Não precisaríamos de realizar outra modalidade de experimento para responder a esta questão – com a análise dos dados obtidos na primeira pesquisa temos:

CARGA MÉDIA DE TRABALHO ESTIMADA: 2000 requisições

TEMPO DE RESPOSTA DESEJADO: 10 segundos

TEMPO DE RESPOSTA OBTIDO: 0,96 segundos para esta carga requisição

Se possuíssemos uma carga de 200,000 requisições no total, cada usuário precisaria esperar 9,694 segundos para ser atendido, assim o sistema pode atender simultaneamente mais de 100 usuários que eles seriam atendidos em um tempo inferior a 10 segundos.

3) EXPERIMENTO 03: Qual seria o limite máximo de *pesquisas* que este sistema poderia suportar considerando uma sobrecarga de trabalho com 100 usuários?

DESEJÁVEL: > 2000.

Ainda utilizando a mesma pesquisa concluiu que os usuários poderiam realizar mais de 2000 requisições simultâneas que seriam atendidas em tempo inferior a 10 segundos

7.4.2.2. Considerações sobre a primeira coleta de dados:

Utilizando somente a primeira coleta de dados conseguimos responder satisfatoriamente nossas questões. O servidor onde está estacionada a aplicação Analysis Project é perfeitamente compatível com as demandas por ela exigida.

7.4.2.3. Novas considerações e estudos mais aprofundados do servidor (web)

Como rapidamente concluímos nosso trabalho gostaria agora de passar a realizar outros testes (coleta de dados) para podermos conhecer um pouco mais sobre a desempenho do nosso servidor.

Para isso construímos duas cenas distintas:

A primeira cena é composta por um ambiente controlado onde submeteríamos a servidor a uma série de cargas que já sabemos ser perfeitamente capaz de suportar. Em média este servidor deveria levar cerca de 30 segundos para resolver estas “cargas”. O objetivo deste teste é verificar se nosso servidor está sofrendo a atuação de aplicações externas que poderiam prejudicar seu desempenho e observar igualmente a utilização da CPU pelo Usuário e pelo Sistema.

Na segunda cena vamos submeter o mesmo número de requisições da primeira, contudo, não especificaremos qual a carga de trabalho que ele deve executar por segundo (rate). O objetivo deste segundo teste e verificar a desempenho do servidor relacionado ao número de requisições que ele consegue executar por segundo.

A próxima etapa é definir os parâmetros de coleta de dados. Resolvemos realizar testes com 5 amostras variando simultaneamente o número de requisições por segundo [–rate] e o número total de requisições [–num-conns] na primeira cena,

não definindo os padrões de [-rate] e mantendo a configuração de [-num-conns] na segunda cena.

Os valores escolhidos foram determinados de forma que as requisições fossem processadas em 30 segundos na primeira cena (valor esperado) sendo que a ausência do parâmetro [-rate] na segunda cena nos informaria a capacidade máxima que nosso servidor possui de resolver requisições por segundo. Observe a tabela abaixo:

Tabela 6 - Parâmetros para a coleta de dados

Coleta de dados	[-rate]	[-num-conns]
Primeira coleta	300	9000
Segunda coleta	600	18000
Terceira coleta	900	27000
Quarta coleta	1000	30000
Quinta coleta	1200	36000

Fonte: Do Autor

7.4.3. Coleta de dados das cenas propostas:

Resolvemos colocar disponível no trabalho o resultado de todas as amostras que colhemos. Além dos dados que escolhemos para trabalho, ela nos traz a possibilidade de visualização de outros dados que podem inclusive gerar conclusões futuras de relevância.

1) Cena 01

Coleta 01 [-rate 300] [-num-conns 9000]

Figura 118 - Primeira coleta de dados cena 01

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_01 --num-conns
9000 --rate 300
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --rate=300 --send-buffer=4096 --recv-buffer=163
84 --num-conns=9000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 3

Total: connections 9000 requests 9000 replies 9000 test-duration 29.998 s

Connection rate: 300.0 conn/s (3.3 ms/conn, <=4 concurrent connections)
Connection time [ms]: min 0.5 avg 0.8 max 12.0 median 0.5 stddev 0.7
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 300.0 req/s (3.3 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 300.0 avg 300.0 max 300.0 stddev 0.0 (5 samples)
Reply time [ms]: response 0.7 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=9000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 4.36 system 23.15 (user 14.5% system 77.2% total 91.7%)
Net I/O: 3409.8 KB/s (27.9*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Coleta 02 [--rate 600] [--num-conns 18000]

Figura 119 - Segunda coleta de dados cena 01

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_01 --num-conns
18000 --rate 600
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --rate=600 --send-buffer=4096 --recv-buffer=163
84 --num-conns=18000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 13

Total: connections 18000 requests 18000 replies 18000 test-duration 30.002 s

Connection rate: 600.0 conn/s (1.7 ms/conn, <=48 concurrent connections)
Connection time [ms]: min 0.5 avg 1.7 max 98.4 median 0.5 stddev 5.6
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 600.0 req/s (1.7 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 599.8 avg 600.0 max 600.0 stddev 0.1 (6 samples)
Reply time [ms]: response 1.5 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=18000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 4.05 system 22.84 (user 13.5% system 76.1% total 89.6%)
Net I/O: 6818.7 KB/s (55.9*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Coleta 03 [--rate 900] [--num-conns 27000]

Figura 120 - Terceira coleta de dados cena 01

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_01 --num-conns
27000 --rate 900
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --rate=900 --send-buffer=4096 --recv-buffer=163
84 --num-conns=27000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 13

Total: connections 27000 requests 27000 replies 27000 test-duration 29.998 s

Connection rate: 900.1 conn/s (1.1 ms/conn, <=26 concurrent connections)
Connection time [ms]: min 0.5 avg 1.6 max 63.2 median 0.5 stddev 3.3
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 900.1 req/s (1.1 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 899.7 avg 900.0 max 900.5 stddev 0.3 (5 samples)
Reply time [ms]: response 1.5 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=27000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 4.24 system 25.12 (user 14.1% system 83.7% total 97.9%)
Net I/O: 10229.3 KB/s (83.8*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Coleta 04 [--rate 1000] [--num-conns 30000]

Figura 121 - Quarta coleta de dados cena 01

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_01 --num-conns
30000 --rate 1000
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --rate=1000 --send-buffer=4096 --recv-buffer=16
384 --num-conns=30000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 9

Total: connections 30000 requests 30000 replies 30000 test-duration 30.005 s

Connection rate: 999.8 conn/s (1.0 ms/conn, <=32 concurrent connections)
Connection time [ms]: min 0.5 avg 1.4 max 62.7 median 0.5 stddev 3.3
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 999.8 req/s (1.0 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 997.3 avg 999.5 max 1000.3 stddev 1.1 (6 samples)
Reply time [ms]: response 1.3 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=30000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 3.78 system 24.26 (user 12.6% system 80.9% total 93.5%)
Net I/O: 11363.4 KB/s (93.1*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Coleta 05 [--rate 1200] [--num-conns 36000]

Figura 122 - Quinta coleta de dados cena 01

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_01 --num-conns
36000 --rate 1200
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --rate=1200 --send-buffer=4096 --recv-buffer=16
384 --num-conns=36000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 16

Total: connections 36000 requests 36000 replies 36000 test-duration 29.998 s

Connection rate: 1200.1 conn/s (0.8 ms/conn, <=77 concurrent connections)
Connection time [ms]: min 0.5 avg 2.8 max 79.9 median 0.5 stddev 6.1
Connection time [ms]: connect 0.1
Connection length [replies/conn]: 1.000

Request rate: 1200.1 req/s (0.8 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 1195.1 avg 1200.1 max 1205.1 stddev 3.6 (5 samples)
Reply time [ms]: response 2.5 transfer 0.2
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=36000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 3.73 system 24.92 (user 12.4% system 83.1% total 95.5%)
Net I/O: 13639.3 KB/s (111.7*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

2) Cena 02

Coleta 01 [--num-conns 9000]

Figura 123 - Primeira coleta de dados cena 02

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_02/index.html -
--num-conns 9000
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --send-buffer=4096 --recv-buffer=16384 --num-co
nns=9000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 1

Total: connections 9000 requests 9000 replies 9000 test-duration 6.461 s

Connection rate: 1393.0 conn/s (0.7 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.6 avg 0.7 max 17.5 median 0.5 stddev 0.6
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 1393.0 req/s (0.7 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 1377.5 avg 1377.5 max 1377.5 stddev 0.0 (1 samples)
Reply time [ms]: response 0.5 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=9000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 0.67 system 5.12 (user 10.4% system 79.2% total 89.6%)
Net I/O: 15831.6 KB/s (129.7*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Coleta 02 [--num-conns 18000]

Figura 124 - Segunda coleta de dados cena 02

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_02/index.html -
-num-conns 18000
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --send-buffer=4096 --recv-buffer=16384 --num-co
nns=18000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 1

Total: connections 18000 requests 18000 replies 18000 test-duration 16.166 s

Connection rate: 1113.5 conn/s (0.9 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.6 avg 0.9 max 16.3 median 0.5 stddev 0.8
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 1113.5 req/s (0.9 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 1106.6 avg 1119.8 max 1130.7 stddev 12.2 (3 samples)
Reply time [ms]: response 0.6 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=18000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 1.12 system 9.30 (user 6.9% system 57.5% total 64.5%)
Net I/O: 12654.7 KB/s (103.7*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Coleta 03 [-num-conns 27000]

Figura 125 - Terceira coleta de dados cena 02

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_02/index.html -
-num-conns 27000
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --send-buffer=4096 --recv-buffer=16384 --num-co
nns=27000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 1

Total: connections 27000 requests 27000 replies 27000 test-duration 20.111 s

Connection rate: 1342.5 conn/s (0.7 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.5 avg 0.7 max 17.4 median 0.5 stddev 0.6
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 1342.5 req/s (0.7 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 1257.0 avg 1341.2 max 1395.5 stddev 59.2 (4 samples)
Reply time [ms]: response 0.5 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=27000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 2.27 system 15.20 (user 11.3% system 75.6% total 86.9%)
Net I/O: 15258.3 KB/s (125.0*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Coleta 04 [-num-conns 30000]

Figura 126 - Quarta coleta de dados cena 02

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_02/index.html -
-num-conns 30000
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --send-buffer=4096 --recv-buffer=16384 --num-co
nns=30000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 1

Total: connections 30000 requests 30000 replies 30000 test-duration 22.227 s

Connection rate: 1349.7 conn/s (0.7 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.5 avg 0.7 max 16.6 median 0.5 stddev 0.6
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 1349.7 req/s (0.7 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 1328.2 avg 1355.5 max 1381.5 stddev 22.9 (4 samples)
Reply time [ms]: response 0.5 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=30000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 2.29 system 17.16 (user 10.3% system 77.2% total 87.5%)
Net I/O: 15339.7 KB/s (125.7*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

Coleta 05 [-num-conns 36000]

Figura 127 - Quinta coleta de dados cena 02

```

robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_02/index.html -
-num-conns 36000
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --send-buffer=4096 --recv-buffer=16384 --num-co
nns=36000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 1

Total: connections 36000 requests 36000 replies 36000 test-duration 26.286 s

Connection rate: 1369.6 conn/s (0.7 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.5 avg 0.7 max 13.9 median 0.5 stddev 0.6
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 1369.6 req/s (0.7 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 1332.6 avg 1366.8 max 1414.3 stddev 35.3 (5 samples)
Reply time [ms]: response 0.5 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=36000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 2.81 system 20.68 (user 10.7% system 78.7% total 89.4%)
Net I/O: 15565.5 KB/s (127.5*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

```

Fonte: Do Autor

7.4.4. Requisições necessárias para resposta em cerca de 30 segundos

Figura 128 - Requisições para uma resposta de 30 segundos

```
robsonmaestro@RobsonMaestro:~$ httpperf - servidor analysisporject.org/icmc_desempenho/cena_02/index.html -
-num-conns 41000
httpperf --client=0/1 --server=localhost --port=80 --uri=/ --send-buffer=4096 --recv-buffer=16384 --num-co
nns=41000 --num-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 1

Total: connections 41000 requests 41000 replies 41000 test-duration 30.782 s

Connection rate: 1332.0 conn/s (0.8 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.6 avg 0.8 max 17.5 median 0.5 stddev 0.6
Connection time [ms]: connect 0.2
Connection length [replies/conn]: 1.000

Request rate: 1332.0 req/s (0.8 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 1232.9 avg 1330.9 max 1391.7 stddev 56.3 (6 samples)
Reply time [ms]: response 0.5 transfer 0.1
Reply size [B]: header 255.0 content 11321.0 footer 0.0 (total 11576.0)
Reply status: 1xx=0 2xx=41000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 3.33 system 23.40 (user 10.8% system 76.0% total 86.8%)
Net I/O: 15138.1 KB/s (124.0*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
```

Fonte: Do Autor

Apenas por uma questão de curiosidade realizei mais alguns testes para verificar qual o número aproximado de requisições que o servidor conseguiria realizar em 30 segundos chegando ao resultado de 41,000 requisições.

7.4.5. Construção das tabelas para verificação dos dados

Os dados coletados foram adicionados em uma planilha do libre office de forma manual não havendo necessidade de nenhum outro recurso. As variáveis descritas abaixo foram selecionadas de acordo com os nossos interesses:

- Número de requisições totais enviadas ao servidor
- Número de requisições submetidas para processamento por segundo [-rate]
- Tempo de resposta ao volume total de requisições
- Porcentagem de utilização de tempo da CPU pelo usuário
- Porcentagem de utilização de tempo da CPU pelo sistema

Figura 129 - Dados resultantes das coletas nas cenas 01 e 02

1	Cenas	Num_Reg	Rate	Tempo_Procesamento	CPU_Usuario	CPU_Sistema
2	Cena 01	9000	300	29,998	14,50	77,20
3	Cena 01	18000	600	30,002	13,50	76,10
4	Cena 01	27000	900	29,998	14,10	83,70
5	Cena 01	30000	1000	30,005	12,60	80,90
6	Cena 01	36000	1200	29,998	12,40	83,10
7	Cena 02	9000	1393	6,461	10,4	79,2
8	Cena 02	18000	1113,5	16,166	6,9	57,5
9	Cena 02	27000	1347	20,111	11,3	75,6
10	Cena 02	30000	1349,7	22,227	10,3	77,2
11	Cena 02	36000	1369,3	26,286	10,7	78,7

Fonte: Do Autor

Todos os dados levantados na pesquisa são relevantes para as decisões a serem tomadas quanto a hospedagem do Analysis Project. O objetivo é verificar se dentro deste servidor a aplicação rodaria com certa tranquilidade.

7.4.5.1. Considerações sobre os dados coletados.

Cena 01 – O tempo de processamento dos dados ficou em torno de 30 segundos (como o esperado) demonstrando que, dentro do horário em que foi realizada a pesquisa, o servidor não recebe influência externa que prejudique seu desempenho.

O tempo de utilização da CPU pelo usuário permanece constante variando um percentual aproximado de 2% com uma média de 13,42.

O tempo de utilização da CPU pelo sistema permanece constante variando um percentual aproximado de 6% com uma média de 80,2.

Cena 02 – Como não definimos o valor do [–rate] o tempo de processamento das requisições varia bastante, em torno de 20 segundos

O tempo de utilização da CPU pelo usuário permanece constante variando um percentual aproximado de 4% com uma média de 9,92.

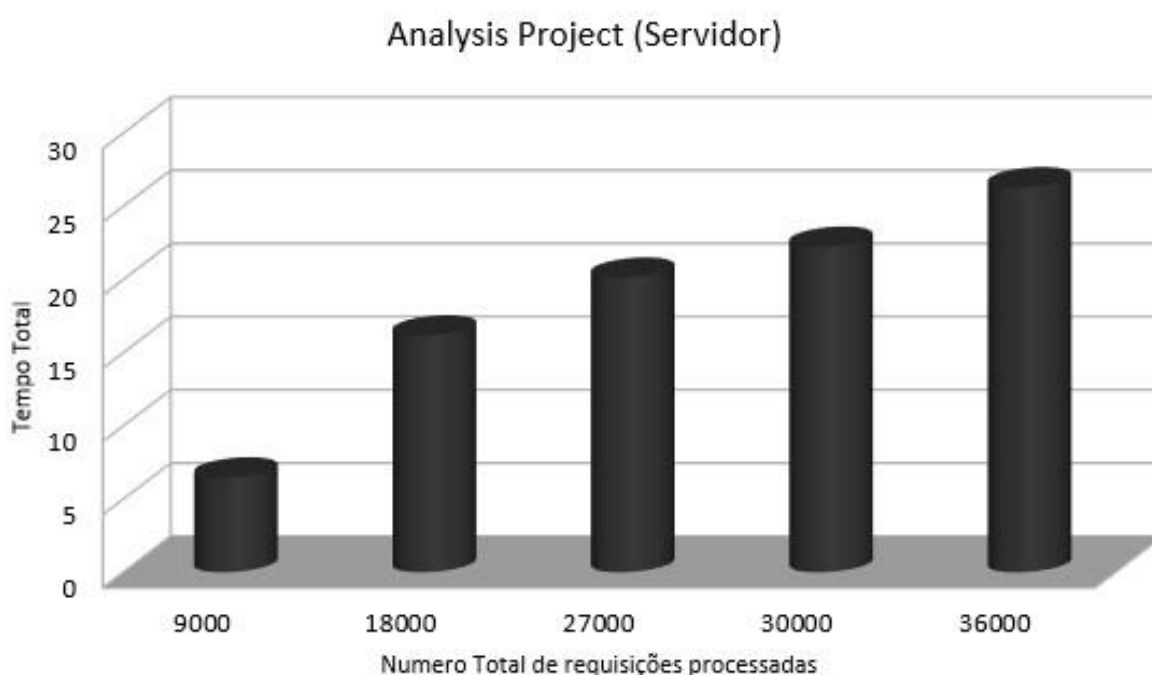
O tempo de utilização da CPU pelo sistema varia bastante um percentual aproximado de 20% com uma média de 73,64.

7.4.5.2. Geração dos gráficos para análise comparativa (Cena 01 X Cena 02)

Com o objetivo de realizar comparações entre as duas cenas estudadas, geramos alguns gráficos para facilitar nossas análises

Número total de requisições X Tempo de processamento (cena 02)

Figura 130 - Tempo Total X Número de requisições



Fonte: Do Autor

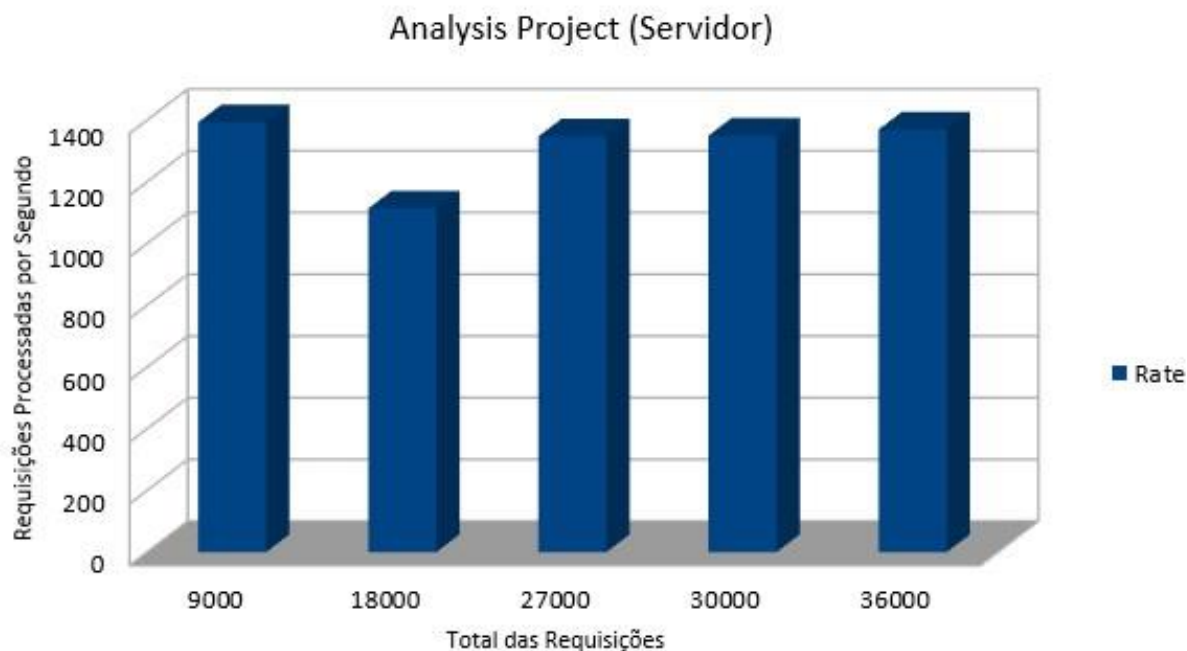
No gráfico acima podemos perceber que o tempo para o processamento das requisições aumenta à medida que aumentamos o número de requisições para o servidor. Algumas considerações:

Conseguimos processar 9000 requisições em cerca de 5 segundos, contudo, ao dobrarmos o número de requisições o tempo gasto para processá-las (que deveria em hipótese dobrar) triplica (em torno de 15 segundos). É um comportamento não esperado, assim as taxas de processamento no servidor variam demonstrando que fatores externos podem ocasionar perda de desempenho.

1) Número total de requisições X Requisições por segundo (cena 02)

Vamos analisar então o número de requisições que o servidor consegue atender por segundo.

Figura 131 - Requisições por segundo X total das requisições

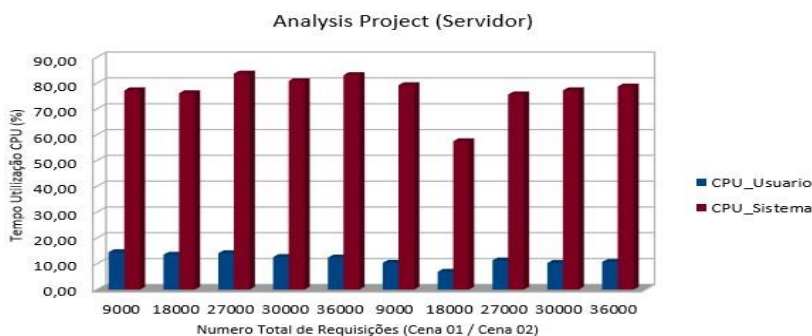


Fonte: Do Autor

Observamos no gráfico acima que a quantidade de requisições que o servidor consegue processar em um segundo é bastante regular. Ao enviarmos a segunda semente com 18,000 requisições o servidor apresentou uma taxa mais baixa de processamento. As causas para este outline podem ser variadas (acesso ao servidor por outro usuário, queda de comunicação etc.). Este segundo exemplo deve ser tratado no nosso caso como uma exceção. Podemos através desta visualização entender as irregularidades no tempo de processamento da semente 02 no estudo do gráfico anterior.

2) Número total de requisições X Tempo de Utilização da CPU pelo usuário e sistema (cena01 e cena 02)

Figura 132 - Tempo de utilização X número de requisições



Fonte: Do Autor

Observamos que quando não utilizamos a função `rate` (Cena 02) as taxas de utilização de tempo da CPU obtiveram uma melhora de desempenho tanto por parte do usuário quanto por parte do sistema, contudo, pouco significativa para nossa aplicação.

A única variação mais ostensiva encontrada nos resultados foi a segunda semente da Cena 02 que consideramos atípica (outline).

7.4.6. Considerações Finais

Primeiramente é necessário destacar que o servidor em que estacionamos a aplicação Analysis Project é *perfeitamente compatível com as suas necessidades*. A utilização da ferramenta `httperf` nos proporcionou resultados bastante rápidos e conclusivos em relação às indagações iniciais de nossa pesquisa. Conseguimos avançar no campo do conhecimento relativo as possibilidades de nosso servidor ao utilizarmos a ferramenta `httperf` de forma diferenciada.

7.5. Visualização Computacional

A visualização computacional é um processo de extrema importância para o sucesso de uma aplicação. Os dados colhidos devem ser apresentados de forma simples e clara para que o analista consiga tirar as conclusões necessárias ao seu projeto.

As fontes de coleta de dados mais comuns são os sensores, medições ou coletas, simulações ou computações. Os dados coletados podem aparecer basicamente de duas maneiras:

- Crus (aqueles que não sofreram nenhum tratamento).⁶⁷
- Processados (Aqueles que passaram por algum ajuste como suavização, normalização, remoção de ruído, interpolação, escala, entre outros, para poderem ser manipulados).

⁶⁷ Os dados colhidos pelo Analysis Project estão neste formato. Não existe a necessidade de nenhum tipo de processamento.

No Analysis Project temos sempre com um conjunto de dados que consiste em “n” instâncias (registros) ($r_1; r_2; r_3; \dots; r_n$) cada qual descrito por “m” (uma ou mais) atributos, ou variáveis ($v_1; v_2; v_3; \dots; v_m$)

Matriz de dados “nm”: cada linha descreve uma instância, cada coluna descreve um atributo

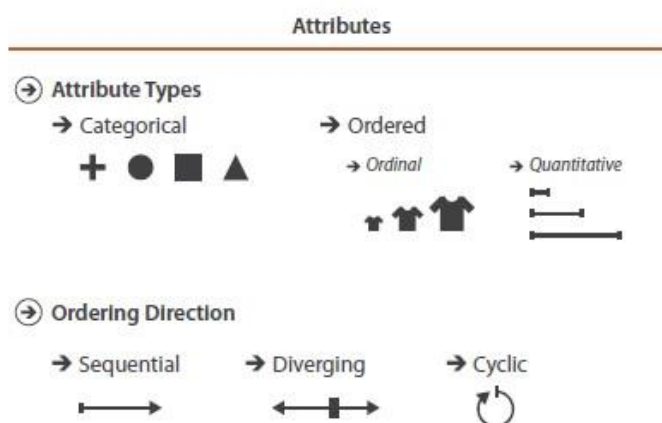
Cada atributo (variável) pode ser um valor único (número de compasso, célula rítmica, ou cadeia de caracteres), ou uma estrutura mais complexa (no caso de polifonia e linhas harmônicas).

7.5.1. Sobre os Dados coletados.

O Analysis Project trabalha com uma multiplicidade de situações e seria necessário aprender sobre os dados que nos fornece para podermos melhor manipulá-los.

Os atributos e as variáveis representam uma informação que categorizamos como ordenável ou categórico

Figura 133 - Tipos de Atributos



Fonte: T. Munzner, Visualization Analysis & Design (Fig. 2.7)

Observe que no primeiro exemplo temos atributos por categoria (símbolos matemáticos) e no segundo exemplo uma ordenação do mesmo elemento (camisa) de forma crescente (da menor para a maior). Concluimos que cada atributo ou variável é uma informação que classificamos de duas maneiras:

1 Ordinal (numérica)

- a. binários: valores 0 e 1
- b. discretos: valores inteiros
- c. contínuos: valores reais

2 Nominal (categórica, ou não-numérica)

- a. categóricos: valor de uma lista finita de possibilidades (vermelho, azul e verde)
- b. ranqueados: valor categórico com uma ordem (pequeno, médio e grande)
- c. arbitrários: faixa infinita de valores sem ordem (endereço)

Os dados podem também ser classificados considerando:

- Relação de ordem: se podem ser ordenados (Variáveis ranqueadas nominais e todas as ordinais).
- Métrica de distância: se é possível calcular a distância entre instâncias (não é comum em variáveis nominais).
- Existência de zero absoluto: menor valor fixo – Diferenciar variáveis ordinais (ex. Peso e saldo bancário) – Somente possui zero absoluto se as 4 operações matemáticas puderem ser aplicadas.

7.5.2. Apresentação de dados no Analysis Project

o Analysis Project é um projeto concebido para ambiente (web). Entre a grande variedade de opções para visualização de dados escolhemos o D3. O resultado superou nossas expectativas e resolvemos adotar como modelo a visualização no formato Tree ou dendograma para o nosso “software”.

7.5.2.1. Base de dados

.A base de dados do Analysis Project é construída em sua maioria por informações textuais, não possuímos muitos dados numéricos e a utilização de técnicas de redução de dimensionalidade como o PCA não é justificável.

Necessitamos dos dados da maneira que foram coletados e percebemos que uma visualização no modelo Tree ou um dendograma seria muito adequado para nossas necessidades.

Para a visualização no estilo Tree ou “arvore” os dados devem ser dispostos de forma a serem reconhecidos como eixos (principal e secundários). No caso específico Analysis Project os dados se encontram num banco de dados Mysql estacionado em domínio próprio facilitando assim sua manipulação. Foi gerado um arquivo com a extensão (.csv) para que a visualização fosse possível. Abaixo uma pequena amostra da estrutura arquivo gerado.

Tabela 8 – Demonstração do arquivo (.csv) e eixos

id,value
LADAINHA_FAUSTINO_XAVIER_DO_PRADO,
LADAINHA_FAUSTINO_XAVIER_DO_PRADO.Pattern_01,
LADAINHA_FAUSTINO_XAVIER_DO_PRADO.Pattern_01.Soprano,
LADAINHA_FAUSTINO_XAVIER_DO_PRADO.Pattern_01.Soprano.Adoramus_te_Christe,
LADAINHA_FAUSTINO_XAVIER_DO_PRADO.Pattern_01.Soprano.Adoramus_te_Christe.Tenor,
LADAINHA_FAUSTINO_XAVIER_DO_PRADO.Pattern_01.Soprano.Adoramus_te_Christe.Tenor. Compasso_06,
id,value
LADAINHA_FAUSTINO_XAVIER_DO_PRADO.Pattern_01.Soprano.Adoramus_te_Christe.Tenor. Compasso_06. augmented_4th _lower,

Fonte: Do Autor

8. RESULTADOS E DISCUSSÃO

Todas as pesquisas apresentadas foram realizadas em nossa base de dados⁶⁹. Apresentamos os resultados que obtivemos nos testes dos padrões cujos algoritmos demonstraram estar corretos.

8.1. Resultados

O Analysis Project possui padrões pré-cadastrados (Figuras de Retórica e Padrões do Gjerdingen) e oferece igualmente a possibilidade de o usuário inserir um padrão próprio e realizar escolhas referente as variações rítmicas, melódicas e estruturais.

Colocamos aqui oito exemplos sendo o último uma pequena melodia escrita pelo autor. Acreditamos ser suficientes para ilustrar a discussão e balizar nossas conclusões.

8.1.1. Pesquisas com figuras de retórica

Figuras que afetam a melodia, o ritmo e a harmonia de maneira geral são “programáveis”, ou seja, é possível desenvolver um algoritmo eficiente que as represente⁷⁰.

8.1.1.1. Passus Duriusculus

Passus Duriusculus, também chamado de “quarto cromático” é um fragmento melódico com a extensão de uma quarta justa com todos (ou quase todos) os intervalos cromáticos.

⁶⁹ A relação de músicas cadastradas encontra-se no APÊNDICE B (Nota do Autor)

⁷⁰ Figuras estruturais (ex: Exordio, narratio, confutatio, etc. . .) dependem da interpretação direta do ouvinte o que inviabiliza a construção de um algoritmo que as represente dentro do universo da música simbólica.(Nota do Autor)

Figura 135 - Prelúdio 09 - Cravo bem temperado compasso 21



Fonte: Cravo Bem Temperado - J. S. Bach - pag.36

A figura acima é um exemplo de Passus Duriusculus encontrado pelo Analysis Project no prelúdio número nove do cravo bem temperado de J. S. Bach.

8.1.1.2. Suspiratio

Interrupção da melodia para ilustração do discurso.

Figura 136 - Sonata III em Fá menor - 01 movimento - Telemann



Fonte: Sonata III em Fá menor - 01 movimento – Telemann

O Suspiratio é como “um suspiro” geralmente representado por uma sequência de pausas estruturais.

8.1.1.3. Aposiopesis

Aposiopese é uma figura em que uma sentença é deliberadamente inacabada. É uma interrupção coletiva representada por uma pausa em todas as vozes.

Figura 137 - Concerto Grosso em Ré maior Op.6 no. 5 - Handel

Fonte: Concerto Grosso Opus 06 n5 – Handel

Observe na figura acima que todas as vozes se interrompem. É um exemplo clássico da figura de retórica Aposiopesis.

8.1.2. Pesquisas com padrões e esquemas de Gjerdingen

Realizamos algumas pesquisas segundo os esquemas e padrões propostos por Robert Gjerdingen. Ressaltamos que em alguns casos como o padrão Comma não conseguimos nenhum resultado em nossa base de dados. Os resultados aqui expostos foram únicos com exceção do Dó Ré Mi que conseguimos três positivos.

Outra curiosidade é que encontramos estes padrões somente em obras que se encontrava em um diretório chamado trecento composto basicamente de madrigais renascentistas.

8.1.2.1. Romanesca com baixo progressivo.

Pesquisa realizada com o Analysis Project utilizando o modelo baixo progressivo.

Figura 138 - Romanesca com baixo progressivo

352

T. - tur, et con - glo - ri - fi - ca - tur:

Bar. con - glo - ri - fi - ca - tur qui

B. tur, et con - glo - ri - fi - ca - tur:

Fonte: Do Autor

A obra acima é o Credo Scabroso de Antônio Zacharias de Teramo. Observe que encontramos a sequência exata da Romanesca com baixo progressivo (linha do baixo realizando a progressão 1 - 7 - 6 - 5 - 4 - 3 e a linha do Tenor realizando a progressão 3 - 2 - 1 - 7 - 6 - 5).

8.1.2.2. Prinner

Pesquisa realizada com o Analysis Project utilizando o modelo Prinner.

Figura 139 - Resultado da pesquisa com o Prinner

43

Vo. se. Qual

ra. bio ti

bio ti

Fonte: Do Autor

A obra acima é um madrigal intitulado Quando L'aire Comença do compositor Magister Piero. Observe que encontramos a sequência com variações temporais interessantes (Mi / Sol no primeiro compasso e Ré / Fá no segundo).

8.1.2.3. Dó Ré - Mi

Realizamos em nosso corpo de obras uma pesquisa utilizando o padrão proposto por Gjerdingen chamado Dó - ré - mi.

Figura 140 - Resultado da pesquisa Do - Ré - Mi



Do Autor

A obra acima é o Sanctus do Lorenzo Fernandes e conseguimos encontrar o Prinner de forma literal.

8.1.3. Pesquisa com padrão determinado pelo usuário

Escolhi uma sequência de notas simples (escala de dó em descendência) com meu objetivo de busca.

Figura 141 - Padrão de pesquisa aleatório escolhido pelo usuário

Padrão do Usuário

Robson Dias

Fonte: Do Autor

Os parâmetros escolhidos para a busca foram:

- Qualquer transposição
- Não levar em consideração o ritmo escrito (olhar somente a frequência na ordem proposta)
- Descartar as sequencias em que a sucessão de notas fosse maior que 2 compassos.

Figura 142 - Padrão encontrado em Bach



Fonte: Do Autor

A obra acima é o bwv69.6 de J. S. Bach. Observe que a sequência de notas foi encontrada de forma literal desconsiderando o ritmo original, como foi solicitado pelo usuário.

Figura 143 - Padrão encontrado em Schumann

The image shows a musical score for a piece titled "Nicht schnell" in piano (*p*). The score is in 2/4 time and the key signature has two sharps (F# and C#). The vocal line is in the upper staff, and the piano accompaniment is in the lower staves. The lyrics are "Aus mei - nen Trä - nen sprie - Ben viel". In the bass line of the piano accompaniment, a specific melodic pattern is highlighted with red circles around the notes: G2, F#2, E2, D2, C#2, B1, A1, G1. This pattern is repeated across the first two measures of the piano part.

Fonte: Do Autor

Igualmente foi encontrado o padrão na obra II. Aus meinen Tränen sprießen, Dichterliebe, Op. 48 de Robert Schumann. Neste segundo exemplo houve uma transposição como solicitado pelo usuário.

Os dois exemplos têm a sequência exata de notas satisfazendo positivamente os critérios de busca, contudo, não consigo estabelecer qualquer relação musical entre as obras.

8.1.4. Exemplo interessante

No capítulo 03 criamos uma pesquisa para ilustrar os processos metodológicos desenvolvidos no Analysis Project. Igualmente criamos uma melodia para ilustração de alguns pontos que julgamos de extrema relevância para nossa discussão.

Figura 144 - Um exemplo de construção melódica

Um Exemplo

Robson Dias

♩ = 140

Passus Duriusculus Catabase

Passus Duriusculus Catabase

Passus Duriusculus Catabase

Passus Duriusculus Catabase

10

Passus Duriusculus Catabase

Passus Duriusculus / Catabase

Fonte: Do Autor

Neste exemplo construímos uma melodia em sol maior utilizando como base duas figuras retóricas: o Passus Duriusculus e a Anabase. De maneira geral o Passus Duriusculus é utilizado de forma descendente (Anabase) para situações musicais de tristeza e agouro. Na linha do baixo era chamado de “baixo lamentável”. A linha melódica acima é um frevo.

8.2. Discussão

Um trabalho multidisciplinar é um grande desafio. O Analysis Project tem sua origem na musicologia e se realiza programação de computadores. Como apresentar o trabalho final? Focar a escrita nos códigos e procedimentos desenvolvidos nesta trajetória é transformar o trabalho em um terreno árido e inacessível para o musicólogo. Igualmente nos foi informado que nenhuma tese em computação descreve sua contribuição em termos de códigos e procedimentos implementados, o que nos deixa aliviados.

A ideia geradora do trabalho é a construção de um modelo analítico musical que utilizasse música simbólica e buscasse por padrões pré-determinados em um grande volume de obras. Para verificar a eficácia (ou não) deste modelo precisávamos de ferramentas capazes de reconhecer os padrões nas obras estudadas e uma metodologia capaz de direcionar as etapas da pesquisa. Desenvolvemos algoritmos para identificar os padrões que desejávamos encontrar e

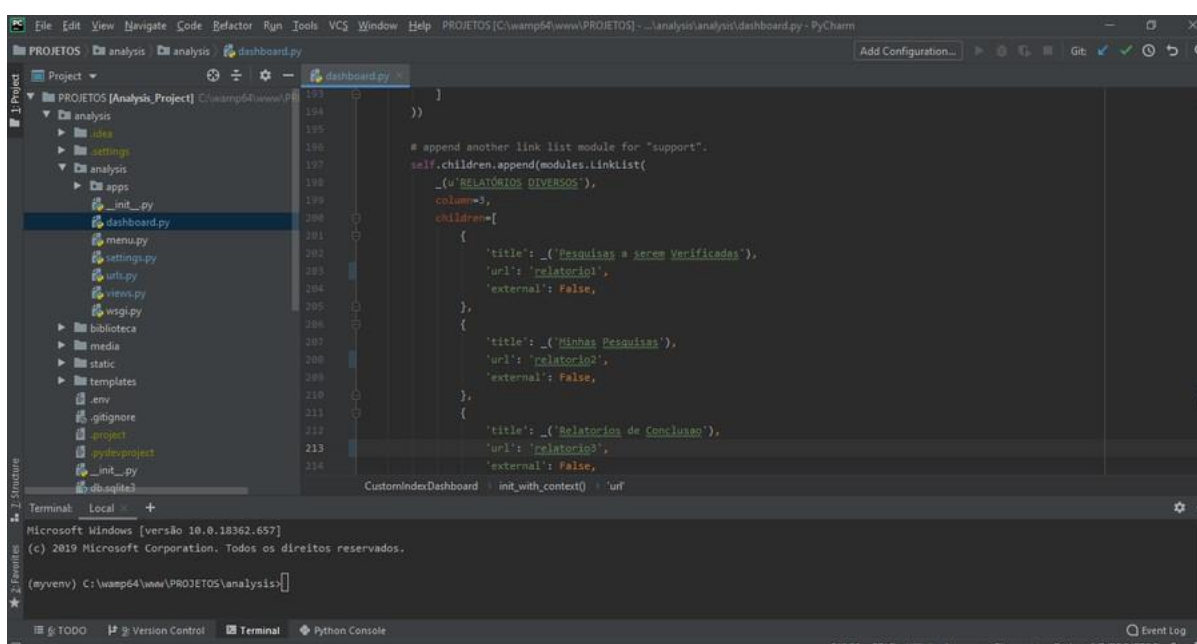
um método capaz de organizar os resultados obtidos. A ferramenta que desenvolvemos pretende preencher estas duas lacunas e oferecer ao pesquisador um suporte para a realização de seus trabalhos, ou pelo menos, utilizamos como suporte para o desenvolvimento desta tese.

Utilizando o Analysis Project conseguimos simular todas as fases de uma pesquisa: selecionamos as obras a serem analisadas, aplicamos um mecanismo de busca dos padrões que desejávamos, analisamos os resultados obtidos e geramos as conclusões. A análise de música simbólica utilizando padrões em grandes corpus é completamente possível, desde que se utilize os procedimentos e ferramentas adequadas.

Em diversos momentos do texto buscamos informar ao leitor sobre questões e decisões de implementação. São elementos para que ele possa avaliar, adequar e corrigir possíveis falhas ou inadequações adaptando as propostas a sua realidade. É justo, visto este trabalho ser somente uma opinião forjada em tentativas e erros e acertos. No momento esta é a nossa melhor versão.

No desenvolvimento de sistemas existem dois componentes: o Frontend (parte da plataforma que interage diretamente com o usuário) e o backend (os códigos desenvolvidos que sustenta a funcionalidade do sistema). Durante todo o processo trabalhamos exclusivamente no backend.

Figura 145 - Área de desenvolvimento



Fonte: Do Autor

A figura 145 descreve a minha área de desenvolvimento:

- Janela da esquerda visualizamos os diretórios com todos os arquivos do projeto.
- Janela da direita o arquivo que estamos editando.
- Janela na parte inferior temos um terminal onde rodamos toda a aplicação.

Todo o trabalho de verificação dos algoritmos desenvolvidos, (lógica, programação e eficácia) foi realizado em “linhas de comando” no terminal. O frontend foi acessado somente nos momentos de escrita deste trabalho onde demonstramos as funcionalidades do Analysis Project (Capítulo 06).

Criamos uma ferramenta que procura padrões de similaridades entre obras musicais. Os resultados proporcionaram outros questionamentos:

- As similaridades que encontramos definem as características da(s) obra(s) estudada(s)?
- Os resultados que obtivemos são sempre quantitativos. Quanto maior o número de similaridades mais parecidas são as obras?
- Encontrar determinado padrão em grande quantidade (passos Duriusculus por exemplo) em determinada obra significa que ela tem caráter melancólico e triste?

Os questionamentos anteriores possuem um grave erro de perspectiva. O Analysis Project, (ou qualquer outro aplicativo similar) trabalha com música simbólica. ***Música simbólica não é música. Música simbólica é a representação gráfica do som.*** O que estamos analisando é uma representação gráfica. Intencionalidade ou caráter de uma obra musical somente pode ser avaliado depois de exaustiva impressão sonora. Como exemplo escrevemos uma pequena melodia utilizando como motivo a figura Passus Duriusculus (Capítulo 7 item 7.1.4) que não tem caráter “melancólico e triste”. Na verdade, é um frevo. Ao classificá-la como Passus Duriusculus o Analysis Project cometeu um erro? Não. Ele achou o que foi programado para encontrar. Surge aqui uma discussão sobre as limitações do processamento computacional, inerentes ao modelo de representação do conteúdo de interesse. Realizamos uma análise baseada na identificação de padrões sintáticos que não capturam a semântica do texto. Observe no exemplo a palavra bonito:

Bonito é o jardim do meu vizinho (Bonito = Belo)

Bonito hein? Fazendo arte novamente. (Bonito = Feio)

Um descritor acharia as duas ocorrências ficando a cargo do analista verificar sentido analisando o contexto. Concluir sobre gêneros músicas e estilos utilizando como material de pesquisa somente música simbólica é terreno movediço e a probabilidade de resultados apócrifos é enorme. O trabalho *Music in Galant Style* do pesquisador Robert Gjerdingen(2007) adota um sistema de padrões e esquemas em um contexto mais amplo. Por esta medida, falar sobre situações pode ser um pouco mais confortável.

Ao iniciarmos a construção do Analysis Project partimos da premissa que padrões repetidos nos elucidaria sobre as características dominantes do nosso objeto de estudo. Os resultados nos demonstraram o contrário. Seriam somente um apontamento sobre um contexto, mas não necessariamente a característica que o define. No nosso exemplo interessante (Capítulo 8, item 8.1.4) existe uma grande quantidade de movimentos melódicos descendentes cromáticos, contudo, a característica dominante da peça é a velocidade da execução e os padrões rítmicos.

Para desenvolvermos um algoritmo que represente um certo padrão ou esquema musical, temos de conhecer profundamente seus elementos constitutivos. Se compreendemos o que vem a ser uma Romanesca, por exemplo, podemos traduzir em passos lógicos o que faríamos para encontrá-la em um contexto. A isto chamamos de lógica de programação⁷¹. Adotamos este procedimento em todo o desenvolvimento do Analysis Project. Os próximos passos da investigação musicológica é compreender melhor a biotipologia das estruturas musicais.

⁷¹ No capítulo 4 item 4.2 descrevi os passos lógicos para o desenvolvimento de um algoritmo representativo do esquema Romanesca.(Nota do Autor)

9. CONCLUSÃO

Buscar padrões em um elevado número de obras utilizando música simbólica era o modelo de análise musicológica que estimulou esta pesquisa. Partíamos do pressuposto de que estilos musicais poderiam ser definidos através do reconhecimento de padrões recorrentes. Como validar esta hipótese? Construimos uma ferramenta para dar suporte ao trabalho, o Analysis Project. Detectamos fragilidades que precisavam ser minoradas, assim desenvolveram-se funcionalidades, que passamos a descrever:

- ORGANIZAÇÃO DO ESPAÇO DE PESQUISA. Em seu módulo de grupos é possível a gerência completa de um laboratório de musicologia, criando subgrupos e associando todo o material necessário ao desenvolvimento das pesquisas.
- METODOLOGIA – processo lógico de realização das pesquisas em fases, organizando os resultados obtidos.
- REALIZAÇÃO DAS PESQUISAS - Realização de forma interna das buscas de padrões com a gravação dos resultados em banco de dados para futuras consultas.
- CONCLUSÃO AUTOMÁTICA DE PROJETOS - Com acesso a todos os detalhes e procedimentos realizados .

O trabalho de construção do Analysis Project passou por duas etapas:

- Estudo de tecnologias para desenvolvimento de soluções computacionais que nos demandou muito tempo por nossas deficiências de conhecimento na área.
- Discussão musicológica dos resultados obtidos.

Tópicos de importância para futuras discussões:

- Trabalhamos com música simbólica que, como dito anteriormente é uma representação gráfica dos sons. Utilizar este contexto para determinar características de um estilo pode ser temeroso.
- O caráter ou gênero musical não pode ser definido somente pelo quantitativo de padrões similares. Outras variáveis devem ser consideradas.

- Precisamos compreender melhor o nosso objeto de estudo para conseguir desenvolver algoritmos mais eficientes.
- Mecanismos de buscas são eficientes quando o pesquisador sabe o que procura e o que fazer com os resultados que encontra. Qualquer pesquisa genérica no âmbito da musicologia tende a resultados rasos.

Referências

AHO, A.; LAM, M.; SETHI, R.; ULLMAN, J. *Compilers: Principles, Techniques and Tools*. 2a ed. Addison-Wesley, 2007, 1000p.

ARISTOTELES. *Retorica*. Ed. bilíngüe em grego e espanhol por Antônio Tovar. Madrid:Centro de Estudios Constitucionales, 1990.

BARTEL, Dietrich. *Música poética: musical-rhetorical figures in German Baroque music*. Lincoln, Nebraska: University of Nebraska Press, 1997.

BENT, Ian. *Music Analysis in the Nineteenth Century: Fugue, form, and style*. New York: Cambridge University Press, 1994.

BENT, Ian & DRABKIN, William. Hong Kong, MacMillan Press, 1990. 184p. (The New Grove Handbook in Music)

BERVIAN, Pedro A.; CERVO, Amado L.; SILVA, Roberto da. *Metodologia Científica*. 6 ed. São Paulo: Pearson Prentice Hall, 2007.

BURMEISTER, Joachim. *Música Poética (Rostock, 1601)*. Laaber: Laaber Verlag, 2004.
CANO, Rubén Lopez. *Música y retórica en el barroco*. Cidade do México, D.F: Universidad Nacional Autónoma de México, 2000.

CARDOSO, João M. P.; DINIZ, Pedro C. *Compilation Techniques for Reconfigurable Architectures*. 1a ed. Springer, 2008, 234 p.

CARDOSO, João M. P.; DINIZ, Pedro; COUTINHO, José Gabriel; PETROV, Zlatko. *Compilation and Synthesis for Embedded Reconfigurable Systems: An Aspect-Oriented Approach*. 1a ed. Springer, 2013, 203p.

CARDOSO, João M. P.; HUEBNER, Michael. *Reconfigurable Computing: From FPGAs to Hardware/SoftwareCodesign*. 1a ed. Springer, 2011, 311p.

COOPER, K.; TORCZON, L. *Engineering a Compiler*. 2a ed. Morgan-Kaufman Publishers, 2010, 824p.

DUCKLES, V., PASLER, J., Stanley, G., CHRISTENSEN, T., HAGGH, B., BALCHIN, R., LIBIN, L., SEEBASS, T., PAGE, J., GOEHR, L., BUJIC, B., CLARKE, E., MCCLARY, S., GRIBENSKI, J., GIANTURCO, C., POTTER, P., FALLOWS, D., VELIMIROVIĆ, M., TOMLINSON, G., BÉHAGUE, G., KANAZAWA, M., & PLATT, P. (2014, January 31). *Musicology*. Grove Music Online. Acesso em 4 Jul. 2020, from <https://www.oxfordmusiconline.com/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-0000046710>.

FEW, Stephen. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*, Analytics Press, 2009.

GJERDINGEN, Robert. *Music in the Galant Style*. New York: Oxford University Press, Inc., 2007.

GOOD, M. (2001). *MusicXML for Notation and Analysis*. In Hewlett, W. B. and SelfridgeField, E. (eds.), *The Virtual Score: Representation, Retrieval, Restoration*. Cambridge (MA) and London (UK): MIT Press

GRACHTEN, M. A., Josep, L. and Mántaras R. L. (2002). A comparison of different approaches to melodic similarity. *Proceedings of the 2nd International Conference in Music and Artificial Intelligence (ICMAI) 2002*.

HAUCK, Scott; DEHON, Andre. *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*. Morgan Kaufmann/Elsevier, 2008, 908p.

HOPCROFT, John E.; MOTWANJ, Rajeev; ULMAN, Jeffrey D. *Introduction to Automata Theory, Languages, and Computation*. 3a ed. Prentice Hall, 2006, 750p.

JAIN, R., *The art of Computer Systems Performance Analysis*, John Wiley & Sons, Inc., 1996.

KANT, K., Introduction to Computer System Performance Evaluation, McGraw-Hill Computer Science Series, 1992. Shneiderman, B. Designing the User Interface, AddisonWesley, 1987.

KAUFMANN, M. Visual Thinking for Design, Colin Ware, 2013.

KEPPER, J., Schreiter, S. and Veit, J. (2014). ‚Freischütz‘ analog oder digital – Editionsformen im Spannungsfeld von Wissenschaft und Praxis. Editio

LUTZ, Mark. Programming Python - Powerful Object-Oriented Programming . 4 ed. Sebastopol: O'Reilly Media, 2010.

MACHADO NETO, Diósnio. O músico sob controle: o processo de licenciamento na primeira metade do século XVIII. Claves, São Paulo, n.7, p.33-52, 2009.

MENASCE, D.A.; ALMEIDA, V.A.F., Capacity Planning for web Services: Metrics, Models, and Methods, Prentice Hall, 2001. FORTIER, P. MICHEL, HOWARD, Computer Systems Performance Evaluation and Prediction, First Ediditon (Paperback), Digital Press 2003.

MENASCE, D.A.; DOWDY, L.W.; ALMEIDA, V.A.F., Performance by Design: Capacity Planning by Example (Paperback), 2004, Prentice Hall, ISBN 0-13-090673-5. MIURA, T. and Shioya, I. (2003). Similarity among melodies for music information retrieval. Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM) 2003.

MUCHNICK, S. Advanced Compiler Design and Implementation. 1a ed. SI: MorganKaufmann Publishers, 1997, 856p.

MÜLLENSIEFEN, D. and Frieler, K. (2004). Optimizing Measures Of Melodic Similarity For The Exploration Of A Large Folk Song Database. Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR) 2004

QUINTILIANO, Institutio Oratória, 1º edição.: 1921. Harvard, Loeb classical library.1996. (126) 4t.

RATNER, Leonard. Classic music: expression, form and style. London: MacMillan, 1980.

SAMMARTINO, Federico. Ceros y unos en la musicología. Software y análisis musical. En prensa, Resonancias, Pontificia Universidad Católica de Chile. Agosto, 2015.

SANTANA, M.J., SANTANA, R.H.C., FRANCÊS, C.R.L., Avaliação e Análise de Desempenho de Sistemas Computacionais: Técnicas e Ferramentas, Publicação Interna ICMC - USP, 1997.

SCHENKER, Heinrich. Five graphic music analyses. 2 ed. New York: Dover, 2012.

SCHOENBERG, Arnold. Fundamentos da composição musical. São Paulo: EDUSP, 1993.

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. Operating System Concepts. 9a ed. Wiley, 2012, 944p.

SPENCE, Robert. Information Visualization, 1. ed., Robert Spence, 2000 ACM Press.

TELEA, Alexander C. Data Visualization: Principles and Practice, Second Edition, 2015. A. K Peters.

TREITLER, Leo. Music and the Historical Imagination. Massachusetts: Harvard University Press, 1989.

TUFTE, Edward R. The Visual Display of Quantitative Information,2nd ed. 2001, Graphics Pr.

TYPKE, R., Wiering, F. and Veltkamp, R. C. (2005). A survey of music information retrieval systems. Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)

VEIT, J. (2015). Music notation beyond paper. On developing digital humanities tools for music editing. Forschungsforum Paderborn.

VIGLIANTI, R. (2007). MusicXML: An XML Based Approach to Musicological Analysis. Digital Humanities 2007: Conference Abstracts, pp. 235–37.

VIRO, V. (2011). Peachnote: Music Score Search and Analysis Platform. Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)

WARD, Matthew O., GRINSTEIN G., KEIM D. Interactive Data Visualization: Foundations, Techniques, and Applications, Second Edition, 2nd Edition, 2015, CRC Press. <http://shiftright.com/mirrors/www.hpl.hp.com/research/linux/httpperf/index.html>. Acesso em 10/12/2017 [https://en.wikipedia.org/wiki/Sar_\(Unix\)](https://en.wikipedia.org/wiki/Sar_(Unix)). Acesso em 10/12/2017

APÊNDICE

APÊNDICE A - Arquitetura de programação do Analysis Project

Colocaremos aqui como ilustração um exemplo dos arquivos models.py e admin.py. Utilizamos estes arquivos para declarar a nossa base de dados.

Cadastros_Gerais

models.py

```
# -*- coding: utf-8 -*-
# MODELS DO PROJETO ANALYSIS APLICAÇÃO CADASTROS_GERAIS
#IMPORTAÇÃO DOS APLICATIVOS NECESSARIOS DE TERCEIROS
from django.db import models
from django.contrib.auth.models import User
from tinymce import models as tinymce_models # NECESSARIO PARA A UTILI-

# =====
===
# ===== IMPORTAÇÃO DAS MINHAS APPS
=====

# from analysis.apps.Manutencao.models import * # IMPORTEI MINHAS
TABELAS DA APP
    # Cadastros_gerais.models import * # IMPORTEI MINHAS TABELAS DA
    APP
# Etapa_01.models import *# IMPORTEI MINHAS TABELAS DA APP
# Etapa_02.models import *# IMPORTEI MINHAS TABELAS DA APP
# Etapa_03.models import *# IMPORTEI MINHAS TABELAS DA APP
# Etapa_04.models import *# IMPORTEI MINHAS TABELAS DA APP
# =====
===
# === CRIAÇÃO DOS MODELS PARA CADASTROS_GERAIS =====
# = 01 - CADASTROS RELACIONADOS A INSTRUMENTAÇÃO (2
    CLASSES)
=
class Musica_Instrumento_Familia(models.Model):
nome_familia = models.CharField("Nome da Família:", max_length=255,
null=False)
```

```

descricao_familia = tinymce_models.HTMLField(verbose_name=u'Descrição:',
null=True, blank=True)
# == TRATAMENTO DO REGISTRO (GERAL) =====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
blank=True, verbose_name=u"Data da Última Alteração")
usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT, re-
lated_name='+', null=True, blank=True, verbose_name=u"Usuário que Alterou")
#Pego id do meu user

data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
ver-bose_name=u"Data da Inserção")
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, blank=True, verbose_name=u"Usuário Responsável") #Pego id do
meu user

# ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u'P', u'Público'),
(u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
(u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
(u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u"Este Registro é Público ou Privado?", default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u"Este Registro pode ser editado?", default = 'EU')
#=====
class Meta:
verbose_name, verbose_name_plural = u"Famílias de Instrumentos" ,
u"Famílias de Instrumentos"
ordering = ('nome_familia',)
def __str__(self):
return self.nome_familia
class Musica_Instrumento(models.Model):
nome_instrumento = models.CharField(u"Nome do Instrumento:",
max_length=255, null=False)
historia_instrumento = tinymce_models.HTMLField(verbose_name=u'História /
Curiosidades sobre o instrumento:', null=True, blank=True)

```

```

familia_instrumento = models.ForeignKey(Musica_Instrumento_Familia, on_delete
    = models.PROTECT, verbose_name=u"Selecione a Família deste Instrumento:")
#===== TRATAMENTO DO REGISTRO (GERAL) =====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
    verbose_name=u"Data da Última Alteração")
usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT,
    related_name='+', null=True, verbose_name=u"Usuário que Alterou") #Pego
id do meu user
data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
    verbose_name=u"Data da Inserção")
usuario_registro = models.ForeignKey(User,on_delete = models.PROTECT,
    null=True, verbose_name=u"Usuário Responsável") #Pego id do meu user
    # ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
    (u'P', u'Público'),
    (u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
    (u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
    (u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
    verbose_name=u"Este Registro é Público ou Privado?", default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
    verbose_name=u"Este Registro pode ser editado?", default = 'EU')
#=====
class Meta:
    verbose_name, verbose_name_plural = u"Instrumentos" , u"Instrumentos"
    ordering = ('nome_instrumento',)
    def __str__(self):
    return self.nome_instrumento
#=====
#== 02 - CADASTROS RELACIONADOS A OBRAS MUSICAIS (3 classes) ==
class Musica_Autor(models.Model):
    nome_autor = models.CharField(u"Nome do Autor:", max_length=255, null=False)
    nascimento_autor = models.CharField(u"Data de Nascimento:", max_length=30,
    null=True, blank=True)

```

```

falecimento_autor = models.CharField(u“Data de Falecimento:”,
max_length=30, null=True, blank=True)
nacionalidade_autor = models.CharField(u“Nacionalidade do Autor:”,
max_length=255, null=True, blank=True)
biografia_autor = tinymce_models.HTMLField(verbose_name=u‘Biografia do
Autor:’, null=True, blank=True)
fontebiografia_autor = models.CharField(u“Fonte da Biografia do Autor:”,
max_length=255, null=True, blank=True)
#==== TRATAMENTO DO REGISTRO (GERAL) =====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
verbose_name=u“Data da Ultima Alteração”)
usuario_altera = models.ForeignKey(User,on_delete = models.PROTECT,
related_name='+', null=True, verbose_name=u“Usuário que Alterou”) #Pego
id do meu user

data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
verbose_name=u“Data da Inserção”)
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, verbose_name=u“Usuário Responsável”) #Pego id do meu user
# ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u‘P’, u‘Público’),
(u‘PV’, u‘Privado’),
)
EDITAREGISTRO_CHOICES = (
(u‘EU’, u‘Pode ser editado somente pelo ADMINISTRADOR do REGISTRO’),
(u‘PU’, u‘Pode ser editado por todos os USUÁRIOS’),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u“Este Registro é Público ou Privado?”, default = ‘P’)
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u“Este Registro pode ser editado?”, default = ‘EU’)
#=====
class Meta:
verbose_name, verbose_name_plural = u“Autores” , u“Cadastro de Autores”
ordering = (‘nome_autor’,)
def __str__(self): return
self.nome_autor
class Musica_Obra(models.Model):

```

```

nome_obra = models.CharField(u"Nome da Obra:", max_length=255, null=False)
autor_obra = models.ForeignKey(Musica_Autor, on_delete = models.PROTECT,
verbose_name=u"Autor da Obra:", null=False)
instrumentacao_obra = models.ManyToManyField(Musica_Instrumento, ver-
bose_name=u"Instrumentação da Obra:", blank=True)
data_obra = models.CharField(max_length=255, null=True, blank=True, ver-
bose_name=u"Data em que a obra foi Composta:")
musicxml_obra = models.FileField(upload_to = u"MusicXml/", default="",
null=True, blank=True, verbose_name=u"Upload da Obra no formato
MusicXML:")
#==== TRATAMENTO DO REGISTRO (GERAL) =====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
verbose_name=u"Data da Última Alteração")
usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT,
related_name='+', null=True, verbose_name=u"Usuário que Alterou") #Pego
id do meu user
data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
verbose_name=u"Data da Inserção")
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, verbose_name=u"Usuário Responsável") #Pego id do meu user
# ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u'P', u'Público'),
(u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
(u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
(u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u"Este Registro é Público ou Privado?", default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u"Este Registro pode ser editado?", default = 'EU')
#=====
class Meta:
verbose_name, verbose_name_plural = u"Obras Musicais" , u"Obras
Musicais"
ordering = ('nome_obra',)

```

```

def __str__(self):
return self.nome_obra

#=====
#== 03 - CADASTROS RELACIONADOS A PATTERNS (6 classes) ==
class Pat_Tipodomaterial(models.Model):
nome_tipodomaterial = models.CharField("Tipo de Material Utilizado na
Pesquisa:", max_length=255, null=False)
descricao_tipodomaterial = tinymce_models.HTMLField(verbose_name=u'Descrição
da Tipo de Material Utilizado na Pesquisa:', null=True, blank=True)
#==== TRATAMENTO DO REGISTRO (GERAL) =====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
verbose_name=u"Data da Última Alteração")
usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT,
related_name='+', null=True, verbose_name=u"Usuário que Alterou") #Pego
id do meu user
data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
verbose_name=u"Data da Inserção")
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, verbose_name=u"Usuário Responsável")
#Pego id do meu user

        # ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u'P', u'Público'),
(u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
(u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
(u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u"Este Registro é Público ou Privado?", default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u"Este Registro pode ser editado?", default = 'EU')
#=====
class Meta:
verbose_name, verbose_name_plural = u"Materiais Utilizados nas Pesqui-
sas" , u"Materiais Utilizados nas Pesquisas"
ordering = ('nome_tipodomaterial',)

```

```

def __str__(self):
return self.nome_tipodomaterial
class Pat_Refinamentos(models.Model):
nome_refinamentos = models.CharField("Modalidade de Refinamento Utili-
zado na Pesquisa:", max_length=255, null=False)
descricao_refinamentos = tinymce_models.HTMLField(verbose_name=u'Descrição
da Tipo de Refinamentos Utilizados na Pesquisa:', null=True, blank=True)
#===== TRATAMENTO DO REGISTRO (GERAL) =====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
verbose_name=u'Data da Ultima Alteração')
usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT,
related_name='+', null=True, verbose_name=u'Usuário que Alterou') #Pego
id do meu user
data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
verbose_name=u'Data da Inserção')
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, verbose_name=u'Usuário Responsável') #Pego id do meu user
# ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u'P', u'Público'),
(u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
(u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
(u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u'Este Registro é Público ou Privado?', default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u'Este Registro pode ser editado?', default = 'EU')
#=====
class Meta:
verbose_name, verbose_name_plural = u'Refinamentos" , u'Refinamentos"
ordering = ('nome_refinamentos',)
def __str__(self):
return self.nome_refinamentos
class Pat_Transformacoes_Melodicas(models.Model):
nome_transformacoes_melodicas = models.CharField("Transformacoes Melo-
dicas Utilizadas na Pesquisa:", max_length=255, null=False)

```

```

descricao_transformacoes_melodicas = tinymce_models.HTMLField(verbose_name=u'De
da Tipo de Transformações Melódicas:', null=True, blank=True)
#===== TRATAMENTO DO REGISTRO (GERAL) =====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
verbose_name=u"Data da Última Alteração")
usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT,
related_name='+', null=True, verbose_name=u"Usuário que Alterou") #Pego
id do meu user
data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
verbose_name=u"Data da Inserção")
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, verbose_name=u"Usuário Responsável") #Pego id do meu user
# ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u'P', u'Público'),
(u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
(u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
(u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u"Este Registro é Público ou Privado?", default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u"Este Registro pode ser editado?", default = 'EU')
#=====
class Meta:
verbose_name, verbose_name_plural = u"Transformacoes Melodicas" ,
u"Transformacoes Melodicas"
ordering = ('nome_transformacoes_melodicas',)
def __str__(self):
return self.nome_transformacoes_melodicas class
Pat_Transformacoes_Ritmicas(models.Model):
nome_transformacoes_ritmicas = models.CharField("Transformacoes
Ritmicas Utilizadas na Pesquisa:", max_length=255, null=False)
descricao_transformacoes_ritmicas = tinymce_models.HTMLField(verbose_name=u'Desc
da Tipo de Transformações Ritmicas:', null=True, blank=True)
#===== TRATAMENTO DO REGISTRO (GERAL) =====

```

```

data_modificacao = models.DateTimeField(auto_now=True, editable=False,
verbose_name=u"Data da Ultima Alteração")
usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT,
related_name='+', null=True, verbose_name=u"Usuário que Alterou") #Pego
id do meu user
data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
verbose_name=u"Data da Inserção")
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, verbose_name=u"Usuário Responsável") #Pego id do meu user
# ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u'P', u'Público'),
(u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
(u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
(u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u"Este Registro é Público ou Privado?", default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u"Este Registro pode ser editado?", default = 'EU')
#=====
class Meta:
verbose_name, verbose_name_plural = u"Transformacoes Ritmicas" ,
u"Transformacoes Ritmicas"
ordering = ('nome_transformacoes_ritmicas',)
def __str__(self):
return self.nome_transformacoes_ritmicas
class Pat_Transformacoes_Transposicoes(models.Model):
nome_transformacoes_transposicoes = models.CharField("Transposições
Utilizadas na Pesquisa:", max_length=255, null=False)
descricao_transformacoes_transposicoes = tinymce_models.HTMLField(verbose_name=
da Tipo de Transposições Utilizado na Pesquisa:', null=True, blank=True)
#===== TRATAMENTO DO REGISTRO (GERAL) =====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
verbose_name=u"Data da Ultima Alteração")

```

```

usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT,
related_name='+', null=True, verbose_name=u"Usuário que Alterou") #Pego
id do meu user
data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
verbose_name=u"Data da Inserção")
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, verbose_name=u"Usuário Responsável") #Pego id do meu user
# ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u'P', u'Público'),
(u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
(u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
(u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u"Este Registro é Público ou Privado?", default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u"Este Registro pode ser editado?", default = 'EU')
#=====
class Meta:
verbose_name, verbose_name_plural = u"Transposições" , u"Transposições"
ordering = ('nome_transformacoes_transposicoes',)
def __str__(self):
return self.nome_transformacoes_transposicoes
class Pat_Transformacoes_Tipografia(models.Model):
nome_transformacoes_tipografia = models.CharField("Transformações da
Tipografia Utilizadas na Pesquisa:", max_length=255, null=False)
descricao_transformacoes_tipografia = tinymce_models.HTMLField(verbose_name=u'De
da Tipo Transformações Tipográficas Utilizadas na Pesquisa:', null=True, blank=True)
# ===== TRATAMENTO DO REGISTRO (GERAL)
=====
data_modificacao = models.DateTimeField(auto_now=True, editable=False,
verbose_name=u"Data da Última Alteração")
usuario_altera = models.ForeignKey(User, on_delete = models.PROTECT,
related_name='+', null=True, verbose_name=u"Usuário que Alterou") #Pego
id do meu user

```

```

data_criacao = models.DateTimeField(auto_now_add=True, editable=False,
verbose_name=u“Data da Inserção”)
usuario_registro = models.ForeignKey(User, on_delete = models.PROTECT,
null=True, verbose_name=u“Usuário Responsável”) #Pego id do meu user
# ===== Declaro meus choices =====
PUBLICOPRIVADO_CHOICES = (
(u'P', u'Público'),
(u'PV', u'Privado'),
)
EDITAREGISTRO_CHOICES = (
(u'EU', u'Pode ser editado somente pelo ADMINISTRADOR do REGISTRO'),
(u'PU', u'Pode ser editado por todos os USUÁRIOS'),
)
registro_publico_privado = models.CharField(max_length=2, choices=PUBLICOPRIVADO
verbose_name=u“Este Registro é Público ou Privado?”, default = 'P')
edita_registro = models.CharField(max_length=2, choices=EDITAREGISTRO_CHOICES,
verbose_name=u“Este Registro pode ser editado?”, default = 'EU')
#=====
class Meta:
verbose_name, verbose_name_plural = u“Transformações da Tipografia” ,
u“Transformações da Tipografia”
ordering = ('nome_transformacoes_tipografia',)
def __str__(self):
return self.nome_transformacoes_tipografia
# =====

```

admin.py

```

# -*- coding: utf-8 -*-
# ADMIN DO PROJETO ANALYSIS_PROJECT APLICAÇÃO
CADASTROS GERAIS from django.contrib import admin

# ===== IMPORT PERSONALIZADO
=====

#from django.db import models # ADICIONAR O MANYTOMANY COMO
CHECKBOX #from django.forms import CheckboxSelectMultiple #
ADICIONAR O MANYTOMANY COMO CHECKBOX
from .models import * # IMPORTAR MINHAS APPS
# =====

```

```

# === 01 - CADASTROS RELACIONADOS A INSTRUMENTAÇÃO (2
CLASSES) == class
Musica_Instrumento_Familia_admin(admin.ModelAdmin):
# ===== OBRIGATÓRIAS =====
list_display = ('nome_familia', 'data_modificacao', 'usuario_altera',
'data_criacao', 'usuario_registro') # LISTA QUE VAI APARECER NO ADMIN
list_filter = ('nome_familia', 'data_modificacao', 'usuario_altera',
'data_criacao', 'usuario_registro',) # LISTA LATERAL QUE FILTRA POR
CAMPO
readonly_fields = ('data_modificacao', 'data_criacao',)
#filter_horizontal = ("projeto_vinculado",) # Campo ManyToMany num
quadro fieldsets = [
# LINHA ABAIXO LEVA TODOS OS CAMPOS DA TABELA
(u'CADASTRO DE FAMÍLIAS DE INSTRUMENTOS', {'fields': ('nome_familia', 'descricao_fami
'classes': ('wide',)})),
# CAMPOS DE TRATAMENTO PADRÃO
(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fields':
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'),
('registro_publico_privado
'edita_registro')), 'classes': ('wide',)})),
    ] # ordena os campos em abas
# =====
class Musica_Instrumento_admin(admin.ModelAdmin):
# ===== OBRIGATÓRIAS =====
list_display = ('nome_instrumento', 'familia_instrumento', 'data_modificacao',
'usuario_altera', 'data_criacao', 'usuario_registro') # LISTA QUE VAI
APARECER NO ADMIN
list_filter = ('nome_instrumento', 'familia_instrumento', 'data_modificacao',
'usuario_altera', 'data_criacao', 'usuario_registro',) # LISTA LATERAL QUE
FILTRA POR CAMPO
readonly_fields = ('data_modificacao', 'data_criacao',)
#filter_horizontal = ("projeto_vinculado",) # Campo ManyToMany num quadro
fieldsets = [
(u'CADASTRO DE INSTRUMENTOS', {'fields': ('nome_instrumento',
'familia_instrumento', 'historia_instrumento'), 'classes': ['collapse']})),
# CAMPOS DE TRATAMENTO PADRÃO
(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fields':
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'), ('regis-
tro_publico_privado', 'edita_registro')), 'classes': ('wide',)})),

```

```

] # ordena os campos em abas
# =====
# =====
# ===02 - CADASTROS RELACIONADOS A OBRAS MUSICAIS (3 classes) ===
class Musica_Autor_admin(admin.ModelAdmin):
    # ===== OBRIGATÓRIAS =====
    list_display = ('nome_autor', 'nascimento_autor', 'falecimento_autor', 'nacionalidade_autor', 'data_modificacao', 'usuario_altera', 'data_criacao', 'usuario_registro') # LISTA QUE VAI APARECER NO ADMIN
    list_filter = ('nome_autor', 'nascimento_autor', 'falecimento_autor', 'nacionalidade_autor', 'data_modificacao', 'usuario_altera', 'data_criacao', 'usuario_registro',) # LISTA LATERAL QUE FILTRA POR CAMPO
    readonly_fields = ('data_modificacao', 'data_criacao',)
    fieldsets = [
        (u'CADASTRO DE AUTORES', {'fields' : ('nome_autor', 'nascimento_autor', 'falecimento_autor', 'nacionalidade_autor', 'biografia_autor', 'fontebiografia_autor'), 'clas-ses': ['collapse']}),
        # CAMPOS DE TRATAMENTO PADRÃO
        (u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fields' : (('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'), ('registro_publico_privado', 'edita_registro')), 'classes': ('wide',)}),
    ] # ordena os campos em abas
    # =====

class Musica_Obra_admin(admin.ModelAdmin):
    # ===== OBRIGATÓRIAS =====
    list_display = ('nome_obra', 'autor_obra', 'data_obra', 'data_modificacao', 'usuario_altera', 'data_criacao', 'usuario_registro') # LISTA QUE VAI APARECER NO ADMIN
    list_filter = ('nome_obra', 'autor_obra', 'data_obra', 'data_modificacao', 'usuario_altera', 'data_criacao', 'usuario_registro',) # LISTA LATERAL QUE FILTRA POR CAMPO
    readonly_fields = ('data_modificacao', 'data_criacao',)
    filter_horizontal = ("instrumentacao_obra",) # Campo ManyToMany num quadro
    #filter_horizontal = ("projeto_vinculado",) # Campo ManyToMany num quadro
    fieldsets = [

```

```

(u'CADASTRO DE OBRAS', {'fields': ('nome_obra', 'autor_obra',
'data_obra', 'musicxml_obra', 'instrumentacao_obra'), 'classes': ['collapse']}),
# CAMPOS DE TRATAMENTO PADRÃO
(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fields': :
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'), ('regis-
tro_publico_privado', 'edita_registro')), 'classes': ('wide',)}),
    ] # ordena os campos em abas
# =====
# =====
# 03 - CADASTROS RELACIONADOS A PATTERNS (6 classes) =====
class Pat_Tipodomaterial_admin(admin.ModelAdmin):
    # ===== OBRIGATÓRIAS =====
    list_display = ('nome_tipodomaterial', 'data_modificacao', 'usuario_altera',
'data_criacao', 'usuario_registro') # LISTA QUE VAI APARECER NO ADMIN
    list_filter = ('nome_tipodomaterial', 'data_modificacao', 'usuario_altera', 'data_criacao',
'usuario_registro',) # LISTA LATERAL QUE FILTRA POR CAMPO
    readonly_fields = ('data_modificacao', 'data_criacao',)
    fieldsets = [
(u'CADASTRO DE PATTERNS - TIPO DO MATERIAL UTILIZADO NA
PESQUISA', {'fields': ('nome_tipodomaterial', 'descricao_tipodomaterial'),
'classes': ['collapse']}),
# CAMPOS DE TRATAMENTO PADRÃO
(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fields': :
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'), ('re-
gistro_publico_privado', 'edita_registro')), 'classes': ('wide',)}), ] # ordena os
campos em abas
#=====
class Pat_Refinamentos_admin(admin.ModelAdmin):
#===== OBRIGATÓRIAS =====
list_display = ('nome_refinamentos', 'data_modificacao', 'usuario_altera', 'data_criacao',
'usuario_registro') # LISTA QUE VAI APARECER NO ADMIN
list_filter = ('nome_refinamentos', 'data_modificacao', 'usuario_altera', 'data_criacao',
'usuario_registro',) # LISTA LATERAL QUE FILTRA POR CAMPO
readonly_fields = ('data_modificacao', 'data_criacao',)
    # filter_horizontal = ("projeto_vinculado",) # Campo ManytoMany
    num quadro fieldsets = [
(u'CADASTRO DE PATTERNS - TIPO DE REFINAMENTOS UTILIZADOS
NA PESQUISA', {'fields': ('nome_refinamentos', 'descricao_refinamentos'), 'classes':

```

```

['collapse'])),
# CAMPOS DE TRATAMENTO PADRÃO

(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fi-elds' :
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'), ('regis-
tro_publico_privado', 'edita_registro')), 'classes': ('wide',)}),
    ] # ordena os campos em abas

#=====
class Pat_Transformacoes_Melodicas_admin(admin.ModelAdmin):
#===== OBRIGATÓRIAS =====
    list_display = ('nome_transformacoes_melodicas', 'data_modificacao', 'usua-
                    rio_altera', 'data_criacao', 'usuario_registro')

#LISTA QUE VAI APARECER NO ADMIN
list_filter = ('nome_transformacoes_melodicas', 'data_modificacao', 'usua-rio_altera',
'data_criacao', 'usuario_registro',) # LISTA LATERAL QUE FILTRA POR
CAMPO
readonly_fields = ('data_modificacao', 'data_criacao', )
#filter_horizontal = ("projeto_vinculado",) # Campo ManyToMany num quadro fieldsets
= [
(u'CADASTRO DE PATTERNS - TRANSFORMAÇÕES MELODICAS', {'fields' :
('nome_transformacoes_melodicas',          'descricao_transformacoes_melodicas'),
'classes': ['collapse'])),
# CAMPOS DE TRATAMENTO PADRÃO

(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fi-elds' :
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'), ('regis-
tro_publico_privado', 'edita_registro')), 'classes': ('wide',)}),
    ] # ordena os campos em abas

#=====
class Pat_Transformacoes_Ritmicas_admin(admin.ModelAdmin):
#===== OBRIGATÓRIAS =====
list_display = ('nome_transformacoes_ritmicas', 'data_modificacao', 'usua-rio_altera',
'data_criacao', 'usuario_registro') # LISTA QUE VAI APARECER NO ADMIN
list_filter = ('nome_transformacoes_ritmicas', 'data_modificacao', 'usuario_altera',
'data_criacao', 'usuario_registro',) # LISTA LATERAL QUE FILTRA POR CAMPO
readonly_fields = ('data_modificacao', 'data_criacao', )
#filter_horizontal = ("projeto_vinculado",) # Campo ManyToMany num quadro fieldsets
= [

```

```
(u'CADASTRO DE PATTERNS - TRANSFORMAÇÕES RÍTMICAS', {'fields' :
('nome_transformacoes_ritmicas', 'descricao_transformacoes_ritmicas'), 'classes':
['collapse']}),
```

```
# CAMPOS DE TRATAMENTO PADRÃO
```

```
(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fields' :
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'), ('regis-
tro_publico_privado', 'edita_registro')), 'classes': ('wide',)}),
```

```
# ordena os campos em abas
```

```
#=====
```

```
class Pat_Transformacoes_Transposicoes_admin(admin.ModelAdmin):
```

```
#===== OBRIGATÓRIAS =====
```

```
list_display = ('nome_transformacoes_transposicoes', 'data_modificacao',
'usu-ario_altera', 'data_criacao', 'usuario_registro') # LISTA QUE VAI
```

```
APARECER NO ADMIN list_filter = ('nome_transformacoes_transposicoes',
'data_modificacao', 'usu-ario_altera', 'data_criacao', 'usuario_registro',) #
```

```
LISTA LATERAL QUE FILTRA POR
```

```
CAMPO
```

```
readonly_fields = ('data_modificacao', 'data_criacao',)
```

```
    # filter_horizontal = ("projeto_vinculado",) # Campo ManyToMany
```

```
    num_quadro fieldsets = [
```

```
(u'CADASTRO DE PATTERNS - TRANSPOSIÇÕES', {'fields' :
```

```
('nome_transformacoes_transposicoes',
'descricao_transformacoes_transposicoes'), 'classes': ['collapse']}),
```

```
# CAMPOS DE TRATAMENTO PADRÃO
```

```
(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fields' :
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'),
('regis-tro_publico_privado', 'edita_registro')), 'classes': ('wide',)}),
```

```
    ] # ordena os campos em abas
```

```
#=====
```

```
class Pat_Transformacoes_Tipografia_admin(admin.ModelAdmin):
```

```
#===== OBRIGATÓRIAS =====
```

```
list_display = ('nome_transformacoes_tipografia', 'data_modificacao', 'usua-
rio_altera', 'data_criacao', 'usuario_registro') # LISTA QUE VAI APARECER
```

```
NO ADMIN list_filter = ('nome_transformacoes_tipografia',
```

```
'data_modificacao', 'usua-rio_altera', 'data_criacao', 'usuario_registro',) #
```

```
LISTA LATERAL QUE FILTRA POR
```

```
CAMPO
```

```
readonly_fields = ('data_modificacao', 'data_criacao',)
```

```

#filter_horizontal = ("projeto_vinculado",) # Campo ManyToMany num quadro
fieldsets = [
(u'CADASTRO DE PATTERNS - TIPOGRAFIA', {'fields' : ('nome_transformacoes_tipog
'descricao_transformacoes_tipografia'), 'classes': ['collapse']})),
# CAMPOS DE TRATAMENTO PADRÃO
(u'TRATAMENTO SOBRE A INCLUSÃO DESTE REGISTRO', {'fields' :
(('data_modificacao', 'usuario_altera'), ('data_criacao', 'usuario_registro'),
('registro_publico_privado', 'edita_registro')), 'classes': ('wide',)})),
] # ordena os campos em abas

=====
#===== REGISTRO DOS MODELS =====
# 01 - CADASTROS RELACIONADOS A INSTRUMENTAÇÃO (3 CLASSES)
=====
admin.site.register (Musica_Instrumento_Familia,
Musica_Instrumento_Familia_admin) admin.site.register (Musica_Instrumento,
Musica_Instrumento_admin)
# 02 - CADASTROS RELACIONADOS A OBRAS MUSICAIS (2 classes)
=====
admin.site.register (Musica_Autor, Musica_Autor_admin)
admin.site.register (Musica_Obra, Musica_Obra_admin)
# 03 - CADASTROS RELACIONADOS A PATTERNS (6 classes)
=====
admin.site.register (Pat_Tipodomaterial, Pat_Tipodomaterial_admin) admin.site.register
(Pat_Refinamentos, Pat_Refinamentos_admin) admin.site.register
(Pat_Transformacoes_Melodicas, Pat_Transformacoes_Melodicas_admin)
admin.site.register (Pat_Transformacoes_Ritmicas, Pat_Transformacoes_Ritmicas_admin)
admin.site.register (Pat_Transformacoes_Transposicoes,
Pat_Transformacoes_Transposicoes_admin) admin.site.register (Pat_Transformacoes_Tipografia,
Pat_Transformacoes_Tipografia_admin)

```

APÊNDICE B - Opções de permissões para grupos do Analyis Project

- admin | entrada de log | Can add log entry
- admin | entrada de log | Can change log entry
- admin | entrada de log | Can delete log entry
- admin | entrada de log | Can view log entry
- auth | grupo | Can add group auth |
- grupo | Can change group auth |
- grupo | Can delete group auth |
- grupo | Can view group
- auth | permissão | Can add permission
- auth | permissão | Can change permission
- auth | permissão | Can delete permission
- auth | permissão | Can view permission
- auth | usuário | Can add user
- auth | usuário | Can change user
- auth | usuário | Can delete user
- auth | usuário | Can view userCadastros_Gerais | Autores | Can add Autores
- Cadastros_Gerais | Autores | Can change Autores
- Cadastros_Gerais | Autores | Can delete Autores
- Cadastros_Gerais | Autores | Can view Autores
- Cadastros_Gerais | Instrumentos | Can add Instrumentos
- Cadastros_Gerais | Instrumentos | Can change Instrumento
- Cadastros_Gerais | Instrumentos | Can delete Instrumentos
- Cadastros_Gerais | Instrumentos | Can view Instrumentos
- Cadastros_Gerais | Familias de Instrumentos | Can add Familias de Instrumentos

- Cadastros_Gerais | Familias de Instrumentos | Can change Familias de Instrumentos
- Cadastros_Gerais | Familias de Instrumentos | Can delete Familias de Instrumentos
- Cadastros_Gerais | Familias de Instrumentos | Can view Familias de Instrumentos
- Cadastros_Gerais | Obras Musicais | Can add Obras Musicais
- Cadastros_Gerais | Obras Musicais | Can change Obras Musicais
- Cadastros_Gerais | Obras Musicais | Can delete Obras Musicais
- Cadastros_Gerais | Obras Musicais | Can view Obras Musicais
- Cadastros_Gerais | Refinamentos | Can add Refinamentos
- Cadastros_Gerais | Refinamentos | Can change Refinamentos
- Cadastros_Gerais | Refinamentos | Can delete Refinamentos
- Cadastros_Gerais | Refinamentos | Can view Refinamentos
- Cadastros_Gerais | Materiais Utilizados nas Pesquisas | Can add Materiais Utilizados nas Pesquisas
- Cadastros_Gerais | Materiais Utilizados nas Pesquisas | Can change Materiais Utilizados nas Pesquisas
- Cadastros_Gerais | Materiais Utilizados nas Pesquisas | Can delete Materiais Utilizados nas Pesquisas
- Cadastros_Gerais | Materiais Utilizados nas Pesquisas | Can view Materiais Utilizados nas Pesquisas
- Cadastros_Gerais | Transformacoes Melodicas | Can add Transformacoes Melodicas
- Cadastros_Gerais | Transformacoes Melodicas | Can change Transformacoes Melodicas
- Cadastros_Gerais | Transformacoes Melodicas | Can delete Transformacoes Melodicas

- Cadastros_Gerais | Transformacoes Melodicas | Can view Transformacoes Me-lodicas
- Cadastros_Gerais | Transformacoes Ritmicas | Can add Transformacoes Ritmi-cas
- Cadastros_Gerais | Transformacoes Ritmicas | Can change Transformacoes Ritmicas
- Cadastros_Gerais | Transformacoes Ritmicas | Can delete Transformacoes Ritmicas
- Cadastros_Gerais | Transformacoes Ritmicas | Can view Transformacoes Ritmi-cas
- Cadastros_Gerais | Transformações da Tipografia | Can add Transformações da Tipografia
- Cadastros_Gerais | Transformações da Tipografia | Can change Transformações da Tipografia
- Cadastros_Gerais | Transformações da Tipografia | Can delete Transformações da Tipografia
- Cadastros_Gerais | Transformações da Tipografia | Can view Transformações da Tipografia
- Cadastros_Gerais | Transposições | Can add Transposições
- Cadastros_Gerais | Transposições | Can change Transposições
- Cadastros_Gerais | Transposições | Can delete Transposições
- Cadastros_Gerais | Transposições | Can view Transposições
- contenttypes | tipo de conteúdo | Can add content type
- contenttypes | tipo de conteúdo | Can change content type
- contenttypes | tipo de conteúdo | Can delete content type
- contenttypes | tipo de conteúdo | Can view content type
- Etapa_01 | 01 - Projetos | Can add 01 - Projetos
- Etapa_01 | 01 - Projetos | Can change 01 - Projetos
- Etapa_01 | 01 - Projetos | Can delete 01 - Projetos

- Etapa_01 | 01 - Projetos | Can view 01 - Projetos
- Etapa_01 | 02 - INSTÂNCIAS de Projetos | Can add 02 - INSTÂNCIAS de Projetos
- Etapa_01 | 02 - INSTÂNCIAS de Projetos | Can change 02 - INSTÂNCIAS de Projetos
- Etapa_01 | 02 - INSTÂNCIAS de Projetos | Can delete 02 - INSTÂNCIAS de Projetos
- Etapa_01 | 02 - INSTÂNCIAS de Projetos | Can view 02 - INSTÂNCIAS de Projetos
- Etapa_02 | 01 - Modalidade de PESQUISA a ser realizada | Can add 01 - Modalidade de PESQUISA a ser realizada
- Etapa_02 | 01 - Modalidade de PESQUISA a ser realizada | Can change 01 - Modalidade de PESQUISA a ser realizada
- Etapa_02 | 01 - Modalidade de PESQUISA a ser realizada | Can delete 01 - Modalidade de PESQUISA a ser realizada
- Etapa_02 | 01 - Modalidade de PESQUISA a ser realizada | Can view 01 - Modalidade de PESQUISA a ser realizada
- Etapa_02 | 02 - CATEGORIAS de PESQUISA | Can add 02 - CATEGORIAS de PESQUISA
- Etapa_02 | 02 - CATEGORIAS de PESQUISA | Can change 02 - CATEGORIAS de PESQUISA
- Etapa_02 | 02 - CATEGORIAS de PESQUISA | Can delete 02 - CATEGORIAS de PESQUISA
- Etapa_02 | 02 - CATEGORIAS de PESQUISA | Can view 02 - CATEGORIAS de PESQUISA
- Etapa_03 | 01 - IMPORTAÇÃO dos Resultados da Pesquisa / VERIFICAÇÃO | Can add 01 - IMPORTAÇÃO dos Resultados da Pesquisa / VERIFICAÇÃO
- Etapa_03 | 01 - IMPORTAÇÃO dos Resultados da Pesquisa / VERIFICAÇÃO | Can change 01 - IMPORTAÇÃO dos Resultados da Pesquisa / VERIFICAÇÃO

- Etapa_03 | 01 - IMPORTAÇÃO dos Resultados da Pesquisa / VERIFICAÇÃO | Can delete 01 - IMPORTAÇÃO dos Resultados da Pesquisa / VERIFICAÇÃO
- Etapa_03 | 01 - IMPORTAÇÃO dos Resultados da Pesquisa / VERIFICAÇÃO | Can view 01 - IMPORTAÇÃO dos Resultados da Pesquisa / VERIFICAÇÃO
- Etapa_03 | 02 - VERIFICAÇÃO dos Resultados da Pesquisa | Can add 02 - VERIFICAÇÃO dos Resultados da Pesquisa
- Etapa_03 | 02 - VERIFICAÇÃO dos Resultados da Pesquisa | Can change 02 - VERIFICAÇÃO dos Resultados da Pesquisa
- Etapa_03 | 02 - VERIFICAÇÃO dos Resultados da Pesquisa | Can delete 02 - VERIFICAÇÃO dos Resultados da Pesquisa
- Etapa_03 | 02 - VERIFICAÇÃO dos Resultados da Pesquisa | Can view 02 - VERIFICAÇÃO dos Resultados da Pesquisa
- Etapa_04 | 01 - Conclusao Automatica de Projetos | Can add 01 - Conclusao Automatica de Projetos
- Etapa_04 | 01 - Conclusao Automatica de Projetos | Can change 01 - Conclusao Automatica de Projetos
- Etapa_04 | 01 - Conclusao Automatica de Projetos | Can delete 01 - Conclusao Automatica de Projetos
- Etapa_04 | 01 - Conclusao Automatica de Projetos | Can view 01 - Conclusao Automatica de Projetos
- Padroes_Gerais | 02 - Classe de Figura Retórico_Musical | Can add Classe de Figura Retórico_Musical
- Padroes_Gerais | 02 - Classe de Figura Retórico_Musical | Can change Classe de Figura Retórico_Musical
- Padroes_Gerais | 02 - Classe de Figura Retórico_Musical | Can delete Classe de Figura Retórico_Musical
- Padroes_Gerais | 02 - Classe de Figura Retórico_Musical | Can view Classe de Figura Retórico_Musical
- Padroes_Gerais | 01 - Figura Retórico Musical | Can add Figura Retórico Musical
Padroes_Gerais | 01 - Figura Retórico Musical | Can change Figura Retórico Musical

- Padroes_Gerais | 01 - Figura Retórico Musical | Can delete Figura Retórico Musical Padroes_Gerais | 01 - Figura Retórico Musical | Can view Figura Retórico Musical
- Padroes_Gerais | 01 - Padrões e Schematas | Can add 01 - Padrões e Schematas Padroes_Gerais | 01 - Padrões e Schematas | Can change 01 - Padrões e Schematas
- Padroes_Gerais | 01 - Padrões e Schematas | Can delete 01 - Padrões e Schematas Padroes_Gerais | 01 - Padrões e Schematas | Can view 01 - Padrões e Schematas
- Padroes_Gerais | 03 - Subclasse Nivel 01 de Figura Retórico_Musical | Can add Subclasse Nivel 01 de Figura Retórico_Musical
- Padroes_Gerais | 03 - Subclasse Nivel 01 de Figura Retórico_Musical | Can change Subclasse Nivel 01 de Figura Retórico_Musical
- Padroes_Gerais | 03 - Subclasse Nivel 01 de Figura Retórico_Musical | Can delete Subclasse Nivel 01 de Figura Retórico_Musical
- Padroes_Gerais | 03 - Subclasse Nivel 01 de Figura Retórico_Musical | Can view Subclasse Nivel 01 de Figura Retórico_Musical
- Padroes_Gerais | 04 - Subclasse Nivel 02 de Figura Retórico_Musical | Can add Subclasse Nivel 02 de Figura Retórico_Musical
- Padroes_Gerais | 04 - Subclasse Nivel 02 de Figura Retórico_Musical | Can change Subclasse Nivel 02 de Figura Retórico_Musical
- Padroes_Gerais | 04 - Subclasse Nivel 02 de Figura Retórico_Musical | Can delete Subclasse Nivel 02 de Figura Retórico_Musical
- Padroes_Gerais | 04 - Subclasse Nivel 02 de Figura Retórico_Musical | Can view Subclasse Nivel 02 de Figura Retórico_Musical
- Padroes_Gerais | 05 - Subclasse Nivel 03 de Figura Retórico_Musical | Can add Subclasse Nivel 03 de Figura Retórico_Musical
- Padroes_Gerais | 05 - Subclasse Nivel 03 de Figura Retórico_Musical | Can change Subclasse Nivel 03 de Figura Retórico_Musical
- Padroes_Gerais | 05 - Subclasse Nivel 03 de Figura Retórico_Musical | Can delete Subclasse Nivel 03 de Figura Retórico_Musical

- Padroes_Gerais | 05 - Subclasse Nivel 03 de Figura Retórico_Musical | Can view Subclasse Nivel 03 de Figura Retórico_Musical
- sessions | sessão | Can add session
- sessions | sessão | Can change session
- sessions | sessão | Can delete session
- sessions | sessão | Can view session

APÊNDICE C - Lista de Figuras de Retórica cadastradas no Analysis Project em ordem alfabética

A

- Abruptio
- Accentus
- Acciacatura
- Amplificatio
- Anabasis
- Anadiplosis
- Anadiplosis Noema
- Analepsis
- Anaphora
- Anaploce
- Anticipatio
- Anticipatio Notae
- Antimetabole
- Antistaechon
- Antistrophe
- Antithesis
- Antitheton
- Apocope
- Aposiopesis
- Apostrophe
- Apotomia
- Articulis

- Ascensus
- Assimilatio
- Asyndeton
- Atanacsis
- Auxesis

B

- Bombi
- Bombilans
- Bombus

C

- Cadentia Duriuscula
- Catabasis
- Catachresis
- Celeritas
- Cercar della nota
- Circulatio
- Circulo
- Climax
- Coloratura
- Commissura
- Complexio
- Complexus
- Congeries
- Consonantiae Impropiae

- Contrapositio
- Contrapositum
- Corta

D

- Deminutio
- Descensus
- Diabasis
- Diminutio
- Distributio
- Distributio Recaptulativa
- Dubitatio

E

- Ecphonesis
- Ellipsis
- Emphasis
- Endiadis
- Enumeratio
- Epanadiplosis
- Epanalepsis
- Epanodos
- Epiphonema
- Epiphora
- Episinalepha
- Epistrophe
- Epizeuxis

- Ethoponia
- Exclamatio
- Extensio

F

- Faux Bourdon
- Fuga
- Fuga Hypotiposis
- Fuga in alio sensu
- Fuga Realis

G

- Gradatio
- Groppo

H

- Harmonia Gemina
- Heterolepsis
- Homioptoton
- Homioiosis
- Homioiteleuton
- Hypallage
- Hyperbaton
- Hyperbole
- Hypobole
- Hypotyposis

I

- Imitatio
- Imperfecta
- Inchoatio
- Inchoatio Imperfecta
- Incrementum
- Interjectio
- Interrogatio

K

- Kyklosis

L

- Licentia
- Ligatura
- Longinqua Distancia

M

- Manubrium
- Meiosis
- Melalepsis
- Messanza
- Metabasis
- Metalepsis

- Mimesis
- Mimesis Noema
- Misticanza
- Mora
- Multiplicatio
- Mutatio Toni

N

- Noema

P

- Palilogia
- Paragoge
- Parembolè
- Parenthesis
- Paronomasia
- Parrhesia
- Passaggio
- Passus Duriusculus
- Pathopoeia
- Pausa
- Percursio
- Pleonasmus
- Polypoton
- Polysyndeton
- Praesumptio
- Prolepsis

- Prolongatio
- Prosopopoeia

Q

- Quaesitio Notae
- Quarta Superflua
- Quasi Transitus

- Quinta Superflua

R

- Reditus
- Reduplicatio
- Regressio
- Repercussio
- Repetitio
- Resumptio
- Retardatio
- Reticentia
- Ribatuta

S

- Salti Composti
- Salto Semplice
- Saltus Duriusculus
- Schematoides
- Sectio

- Sexta Superflua
- Simul Procedentia
- Stenasmus
- Subsumptio
- Subsumptio Postpositiva M.
- Superjectio
- Supplementum
- Suspensio
- Suspiratio
- Symblema
- Symploce
- Synaeresis
- Synathoroismus
- Syncopatio
- Syncopatio Carachresica
- Syncopatio Invertida
- Syncope
- Synecdoche
- Synecdoche-Metonymie
- Syneresis
- Synhaeresis
- synonymia

- Tenuta
- Tertia Deficiens
- Tirata
- Tmesis
- Totales Fugae
- Traductio
- Transgressio
- Transitus
- Transitus Inversus
- Transitus Notae
- Transumptio
- Tremolo
- Trillo

V

- Variatio

APÊNDICE D - Lista de autores relacionados a Retórica Musical cadastrados no Analysis Project (Referencias primarias e secundárias)

- Ahle, Johann G.
- Arbusow, Leonid
- Auerbach, Eric
- Bach, C. P. E.
- Bacon, Francis
- Barner, Wilfried
- Bartel, Dietrich
- Benary, Peter
- Benitez, Vincent
- Bernhard, Christoph
- Bianconi, Lorenzo
- Blume, Friedrich
- Burmeister, Joachim
- Butler, Charles
- Caldenbach, Christoph
- Calvisius, Seth
- Cano, Rubén López
- Cícero
- Descartes, René
- Dressler, Gallus
- Ernesti, Johann Chr.

- Forkel, Johann
- Glareanus, Henricus
- Gottsched, Johann Christoph
- Heinichen, Johann D.
- Herbst, Johann Andreas
- Hoffmann-Axthelm, D.
- Janovka, Tomáš
- Johnston, Gregory
- Kehrein, Joseph
- Kennedy, George A.
- Kircher, Athanasius.
- Kirkendale, Ursula
- Krause, Christian
- Kretzschmar, Hermann
- Kristeller, Paul O.
- Krones, Hartmut
- Kuhnau, Johann
- Lamy, Bernhard.
- Lausberg, Heinrich
- Le Huray, Peter
- Lenneberg, Hans
- Lester, Joel
- Lippius, Johannes
- Listenius, Nikolaus

- Lossius, Lucas
- Marpurg, Friedrich
- Massenkeil, Günther
- Mattheson, Johann
- Meier, Bernhard
- Melancton, Philipp
- Morely, Thomas
- Murphy, James
- Müller-Blattau, Joseph
- Neumann, Frederick
- Nucius, Johannes
- Palisca, Claude
- Peacham, Henry
- Pilszynski, Judith
- Praetorius, Michael
- Printz, Wolfgang C.
- Puttenham, George
- Quantz, Johann
- Quintilian
- Rhaw, Georg
- Ruhnke, Martin
- Scheibe, Johann
- Schering, Arnold

- Schmitz, Arnold
- Serauky, Walter
- Sevier, Z.
- Sonnino, Lee A.
- Speer, Daniel
- Spiess, Meinrad
- Steblin, Rita
- Susenbrotus, Johannes
- Sühngen, Oskar
- Thuringus, Joachim
- Tinctorus, Johannes
- Toussaint, Georg
- Unger, Hans-Heinrich
- Vickers, Brian
- Vogt, Mauritius
- Vossius, Gerardus
- Waite, William
- Walther, Elias
- Walther, Johann G.
- Weissenborn, M. Christoph
- Werckmeister, Andreas
- Wilhelm, I.
- Williams, Peter
- Zarlino, Giuseppe

Anexos

ANEXO A - Recursos do Music21

Alguns recursos existentes no Music21 para a manipulação de arquivos MusicXML. São 29 recursos que podem ser adaptados à necessidade do usuário.

```
from music21 import *
print(musicxml.lilypondTestSuite.__doc__)
The Lilypond MusicXML Test Suite comes from https://github.com/cuthbertLab/musicxmlTestSuite
and is a fork of http://lilypond.org/doc/v2.18/input/regression/musicxml/collated-files
The test suite is licensed under the MIT license (https://opensource.org/licenses/mit-license.php) and copyrighted by the Lilypond project.
```

```
import os
import re
import pathlib
allF = musicxml.lilypondTestSuite.allFiles()
untested = [f.name for f in sorted(allF)]
scores = {}
musicxmlOut = {}
def s(i):
    if isinstance(i, int):
        longFp = allF[i]
    else:
        for longFp in allF:
            shortFp = longFp.name
            if i in shortFp:
                break
        else:
            raise Exception("Cannot find: " + str(i))
        shortFp = longFp.name
        if shortFp in untested:
            untested.remove(shortFp)
        sc = converter.parse(longFp, forceSource=True)
        scores[i] = sc
    fp = pathlib.Path(sc.filePath).name
```

```

print(fp + '\n')
desc = sc.metadata.description
desc = re.sub(r'\s+', ' ', desc)

print(desc)
fpOut = sc.write('musicxml')
with open(fpOut, 'r') as musicxmlOutFile:
allOut = musicxmlOutFile.read()
musicxmlOut[i] = allOut
return sc.show()

```

Lista de recursos de manipulação dos arquivos MusicXML:

- 01 . . . Pitches
- 02 . . . Rests
- 03 . . . Rhythm
- 04 . . . Time signatures
- 05 . . . Clefs
- 06 . . . Key signatures
- 07 . . . Staff attributes
- 08 . . . Chorded notes
- 09 . . . Note settings, heads, etc.
- 10 . . . Triplets, Tuplets
- 11 . . . Grace notes
- 12 . . . Dynamics and other single symbols
- 13 . . . Notations and Articulations
- 14 . . . Spanners
- 15 . . . Multiple parts (staves)
- 16 . . . Multiple voices per staff
- 17 . . . One part on multiple staves
- 18 . . . Repeats
- 19 . . . Barlines, Measures
- 20 . . . Header information
- 21 . . . Page layout
- 22 . . . Lyrics
- 23 . . . Guitar notation
- 24 . . . Transposing instruments
- 25 . . . Percussion
- 26 . . . Figured bass

- 27 . . . Other instrumental notation
- 28 . . . Compressed MusicXML files
- 29 . . . Compatibility with broken MusicXML

ANEXO B- Listas as obras castradas no Analysis Project

Johann Sebastian Bach

Bwv 1.6 (musicxml): bach/bwv1.6.xml
Bwv 10.7 (musicxml): bach/bwv10.7.xml
Bwv 101.7 (musicxml): bach/bwv101.7.xml
Bwv 102.7 (musicxml): bach/bwv102.7.xml
Bwv 103.6 (musicxml): bach/bwv103.6.xml
Bwv 104.6 (musicxml): bach/bwv104.6.xml
Bwv 108.6 (musicxml): bach/bwv108.6.xml
Bwv 11.6 (musicxml): bach/bwv11.6.xml
Bwv 110.7 (musicxml): bach/bwv110.7.xml
Bwv 111.6 (musicxml): bach/bwv111.6.xml
Bwv 112.5 -Sc (musicxml): bach/bwv112.5-sc.xml
Bwv 112.5 (musicxml): bach/bwv112.5.xml
Bwv 113.8 (musicxml): bach/bwv113.8.xml
Bwv 114.7 (musicxml): bach/bwv114.7.xml
Bwv 115.6 (musicxml): bach/bwv115.6.xml
Bwv 116.6 (musicxml): bach/bwv116.6.xml
Bwv 117.4 (musicxml): bach/bwv117.4.xml
Bwv 119.9 (musicxml): bach/bwv119.9.xml
Bwv 12.7 (musicxml): bach/bwv12.7.xml
Bwv 120.6 (musicxml): bach/bwv120.6.xml
Bwv 120.8 -A (musicxml): bach/bwv120.8-a.xml
Bwv 121.6 (musicxml): bach/bwv121.6.xml
Bwv 122.6 (musicxml): bach/bwv122.6.xml
Bwv 123.6 (musicxml): bach/bwv123.6.xml
Bwv 124.6 (musicxml): bach/bwv124.6.xml
Bwv 125.6 (musicxml): bach/bwv125.6.xml
Bwv 126.6 (musicxml): bach/bwv126.6.xml
Bwv 127.5 (musicxml): bach/bwv127.5.xml
Bwv 128.5 (musicxml): bach/bwv128.5.xml
Bwv 13.6 (musicxml): bach/bwv13.6.xml
Bwv 130.6 (musicxml): bach/bwv130.6.xml
Bwv 133.6 (musicxml): bach/bwv133.6.xml
Bwv 135.6 (musicxml): bach/bwv135.6.xml
Bwv 136.6 (musicxml): bach/bwv136.6.xml

Bwv 137.5 (musicxml): bach/bwv137.5.xml
Bwv 139.6 (musicxml): bach/bwv139.6.xml
Bwv 14.5 (musicxml): bach/bwv14.5.xml
Bwv 140.7 (musicxml): bach/bwv140.7.xml
Bwv 144.3 (musicxml): bach/bwv144.3.xml
Bwv 144.6 (musicxml): bach/bwv144.6.xml
Bwv 145 -A (musicxml): bach/bwv145-a.xml
Bwv 145.5 (musicxml): bach/bwv145.5.xml
Bwv 146.8 (musicxml): bach/bwv146.8.xml
Bwv 148.6 (musicxml): bach/bwv148.6.xml
Bwv 149.7 (musicxml): bach/bwv149.7.xml
Bwv 151.5 (musicxml): bach/bwv151.5.xml
Bwv 153.1 (musicxml): bach/bwv153.1.xml
Bwv 153.5 (musicxml): bach/bwv153.5.xml
Bwv 153.9 (musicxml): bach/bwv153.9.xml
Bwv 154.3 (musicxml): bach/bwv154.3.xml
Bwv 154.8 (musicxml): bach/bwv154.8.xml
Bwv 155.5 (musicxml): bach/bwv155.5.xml
Bwv 156.6 (musicxml): bach/bwv156.6.xml
Bwv 157.5 (musicxml): bach/bwv157.5.xml
Bwv 158.4 (musicxml): bach/bwv158.4.xml
Bwv 159.5 (musicxml): bach/bwv159.5.xml
Bwv 16.6 (musicxml): bach/bwv16.6.xml
Bwv 161.6 (musicxml): bach/bwv161.6.xml
Bwv 162.6 -Lpz (musicxml): bach/bwv162.6-lpz.xml
Bwv 164.6 (musicxml): bach/bwv164.6.xml
Bwv 165.6 (musicxml): bach/bwv165.6.xml
Bwv 166.6 (musicxml): bach/bwv166.6.xml
Bwv 168.6 (musicxml): bach/bwv168.6.xml
Bwv 169.7 (musicxml): bach/bwv169.7.xml
Bwv 17.7 (musicxml): bach/bwv17.7.xml
Bwv 171.6 (musicxml): bach/bwv171.6.xml
Bwv 172.6 (musicxml): bach/bwv172.6.xml
Bwv 174.5 (musicxml): bach/bwv174.5.xml
Bwv 175.7 (musicxml): bach/bwv175.7.xml
Bwv 176.6 (musicxml): bach/bwv176.6.xml
Bwv 177.4 (musicxml): bach/bwv177.4.xml
Bwv 177.5 (musicxml): bach/bwv177.5.xml
Bwv 178.7 (musicxml): bach/bwv178.7.xml

Bwv 179.6 (musicxml): bach/bwv179.6.xml
Bwv 18.5 -Lz (musicxml): bach/bwv18.5-lz.xml
Bwv 18.5 -W (musicxml): bach/bwv18.5-w.xml

Bwv 180.7 (musicxml): bach/bwv180.7.xml
Bwv 183.5 (musicxml): bach/bwv183.5.xml
Bwv 184.5 (musicxml): bach/bwv184.5.xml
Bwv 185.6 (musicxml): bach/bwv185.6.xml
Bwv 187.7 (musicxml): bach/bwv187.7.xml
Bwv 188.6 (musicxml): bach/bwv188.6.xml
Bwv 19.7 (musicxml): bach/bwv19.7.xml
Bwv 190.7 -Inst (musicxml): bach/bwv190.7-inst.xml
Bwv 190.7 (musicxml): bach/bwv190.7.xml
Bwv 194.12 (musicxml): bach/bwv194.12.xml
Bwv 194.6 (musicxml): bach/bwv194.6.xml
Bwv 195.6 (musicxml): bach/bwv195.6.xml
Bwv 197.10 (musicxml): bach/bwv197.10.xml
Bwv 197.5 (musicxml): bach/bwv197.5.xml
Bwv 197.7 -A (musicxml): bach/bwv197.7-a.xml
Bwv 2.6 (musicxml): bach/bwv2.6.xml
Bwv 20.11 (musicxml): bach/bwv20.11.xml
Bwv 20.7 (musicxml): bach/bwv20.7.xml
Bwv 226.2 (musicxml): bach/bwv226.2.xml
Bwv 227.1 (musicxml): bach/bwv227.1.xml
Bwv 227.11 (musicxml): bach/bwv227.11.xml
Bwv 227.3 (musicxml): bach/bwv227.3.xml
Bwv 227.7 (musicxml): bach/bwv227.7.xml
Bwv 229.2 (musicxml): bach/bwv229.2.xml
Bwv 24.6 (musicxml): bach/bwv24.6.xml
Bwv 244.10 (musicxml): bach/bwv244.10.xml
Bwv 244.15 (musicxml): bach/bwv244.15.xml
Bwv 244.17 (musicxml): bach/bwv244.17.xml
Bwv 244.25 (musicxml): bach/bwv244.25.xml
Bwv 244.29 -A (musicxml): bach/bwv244.29-a.xml
Bwv 244.3 (musicxml): bach/bwv244.3.xml
Bwv 244.32 (musicxml): bach/bwv244.32.xml
Bwv 244.37 (musicxml): bach/bwv244.37.xml
Bwv 244.40 (musicxml): bach/bwv244.40.xml
Bwv 244.44 (musicxml): bach/bwv244.44.xml

Bwv 244.46 (musicxml): bach/bwv244.46.mxl
Bwv 244.54 (musicxml): bach/bwv244.54.mxl
Bwv 244.62 (musicxml): bach/bwv244.62.mxl
Bwv 245.11 (musicxml): bach/bwv245.11.mxl
Bwv 245.14 (musicxml): bach/bwv245.14.mxl
Bwv 245.15 (musicxml): bach/bwv245.15.mxl
Bwv 245.17 (musicxml): bach/bwv245.17.mxl
Bwv 245.22 (musicxml): bach/bwv245.22.mxl
Bwv 245.26 (musicxml): bach/bwv245.26.mxl
Bwv 245.28 (musicxml): bach/bwv245.28.mxl
Bwv 245.3 (musicxml): bach/bwv245.3.mxl
Bwv 245.37 (musicxml): bach/bwv245.37.mxl
Bwv 245.40 (musicxml): bach/bwv245.40.mxl
Bwv 245.5 (musicxml): bach/bwv245.5.mxl
Bwv 248.12 - 2 (musicxml): bach/bwv248.12-2.mxl
Bwv 248.17 (musicxml): bach/bwv248.17.mxl
Bwv 248.23 - 2 (musicxml): bach/bwv248.23-2.mxl
Bwv 248.23 -S (musicxml): bach/bwv248.23-s.mxl
Bwv 248.28 (musicxml): bach/bwv248.28.mxl
Bwv 248.33 - 3 (musicxml): bach/bwv248.33-3.mxl
Bwv 248.35 - 3 (musicxml): bach/bwv248.35-3.mxl
Bwv 248.35 - 3 C (musicxml): bach/bwv248.35-3c.mxl
Bwv 248.42 - 4 (musicxml): bach/bwv248.42-4.mxl
Bwv 248.42 -S (musicxml): bach/bwv248.42-s.mxl
Bwv 248.46 - 5 (musicxml): bach/bwv248.46-5.mxl
Bwv 248.5 (musicxml): bach/bwv248.5.mxl
Bwv 248.53 - 5 (musicxml): bach/bwv248.53-5.mxl
Bwv 248.59 - 6 (musicxml): bach/bwv248.59-6.mxl
Bwv 248.64 - 6 (musicxml): bach/bwv248.64-6.mxl
Bwv 248.64 -S (musicxml): bach/bwv248.64-s.mxl
Bwv 248.9 - 1 (musicxml): bach/bwv248.9-1.mxl
Bwv 248.9 -S (musicxml): bach/bwv248.9-s.mxl
Bwv 25.6 (musicxml): bach/bwv25.6.mxl
Bwv 250 (musicxml): bach/bwv250.mxl
Bwv 251 (musicxml): bach/bwv251.mxl
Bwv 252 (musicxml): bach/bwv252.mxl
Bwv 253 (musicxml): bach/bwv253.mxl
Bwv 254 (musicxml): bach/bwv254.mxl
Bwv 255 (musicxml): bach/bwv255.mxl

Bwv 256 (musicxml): bach/bwv256.mxl
Bwv 257 (musicxml): bach/bwv257.mxl
Bwv 258 (musicxml): bach/bwv258.mxl
Bwv 259 (musicxml): bach/bwv259.mxl
Bwv 26.6 (musicxml): bach/bwv26.6.mxl
Bwv 260 (musicxml): bach/bwv260.mxl
Bwv 261 (musicxml): bach/bwv261.mxl
Bwv 262 (musicxml): bach/bwv262.mxl
Bwv 263 (musicxml): bach/bwv263.mxl
Bwv 264 (musicxml): bach/bwv264.mxl
Bwv 265 (musicxml): bach/bwv265.mxl
Bwv 266 (musicxml): bach/bwv266.mxl
Bwv 267 (musicxml): bach/bwv267.mxl
Bwv 268 (musicxml): bach/bwv268.mxl
Bwv 269 (musicxml): bach/bwv269.mxl
Bwv 27.6 (musicxml): bach/bwv27.6.mxl
Bwv 270 (musicxml): bach/bwv270.mxl
Bwv 271 (musicxml): bach/bwv271.mxl
Bwv 272 (musicxml): bach/bwv272.mxl
Bwv 273 (musicxml): bach/bwv273.mxl
Bwv 274 (musicxml): bach/bwv274.mxl
Bwv 275 (musicxml): bach/bwv275.mxl
Bwv 276 (musicxml): bach/bwv276.mxl
Bwv 277

Bwv 277 (humdrum): bach/bwv277.krn

- Bwv 277 (musicxml): bach/bwv277.mxl

Bwv 278 (musicxml): bach/bwv278.mxl
Bwv 279 (musicxml): bach/bwv279.mxl
Bwv 28.6 (musicxml): bach/bwv28.6.mxl
Bwv 280 (musicxml): bach/bwv280.mxl Bwv 281

- Bwv 281 (humdrum): bach/bwv281.krn
- Bwv 281 (musicxml): bach/bwv281.mxl

Bwv 282 (musicxml): bach/bwv282.mxl
Bwv 283 (musicxml): bach/bwv283.mxl
Bwv 284 (musicxml): bach/bwv284.mxl

Bwv 285 (musicxml): bach/bwv285.mxl
Bwv 286 (musicxml): bach/bwv286.mxl
Bwv 287 (musicxml): bach/bwv287.mxl
Bwv 288 (musicxml): bach/bwv288.mxl
Bwv 289 (musicxml): bach/bwv289.mxl
Bwv 29.8 (musicxml): bach/bwv29.8.mxl
Bwv 290 (musicxml): bach/bwv290.mxl
Bwv 291 (musicxml): bach/bwv291.mxl
Bwv 292 (musicxml): bach/bwv292.mxl
Bwv 293 (musicxml): bach/bwv293.mxl
Bwv 294 (musicxml): bach/bwv294.mxl
Bwv 295 (musicxml): bach/bwv295.mxl
Bwv 296 (musicxml): bach/bwv296.mxl
Bwv 297 (musicxml): bach/bwv297.mxl
Bwv 298 (musicxml): bach/bwv298.mxl
Bwv 299 (musicxml): bach/bwv299.mxl
Bwv 3.6 (musicxml): bach/bwv3.6.mxl
Bwv 30.6 (musicxml): bach/bwv30.6.mxl
Bwv 300 (musicxml): bach/bwv300.mxl
Bwv 301 (musicxml): bach/bwv301.mxl
Bwv 302 (musicxml): bach/bwv302.mxl
Bwv 303 (musicxml): bach/bwv303.mxl
Bwv 304 (musicxml): bach/bwv304.mxl
Bwv 305 (musicxml): bach/bwv305.mxl
Bwv 306 (musicxml): bach/bwv306.mxl
Bwv 307 (musicxml): bach/bwv307.mxl
Bwv 308 (musicxml): bach/bwv308.mxl
Bwv 309 (musicxml): bach/bwv309.mxl
Bwv 31.9 (musicxml): bach/bwv31.9.mxl
Bwv 310 (musicxml): bach/bwv310.mxl
Bwv 311 (musicxml): bach/bwv311.mxl
Bwv 312 (musicxml): bach/bwv312.mxl
Bwv 313 (musicxml): bach/bwv313.mxl
Bwv 314 (musicxml): bach/bwv314.mxl
Bwv 315 (musicxml): bach/bwv315.mxl
Bwv 316 (musicxml): bach/bwv316.mxl
Bwv 317 (musicxml): bach/bwv317.mxl
Bwv 318 (musicxml): bach/bwv318.mxl
Bwv 319 (musicxml): bach/bwv319.mxl

Bwv 32.6 (musicxml): bach/bwv32.6.mxl
Bwv 320 (musicxml): bach/bwv320.mxl
Bwv 321 (musicxml): bach/bwv321.mxl
Bwv 322 (musicxml): bach/bwv322.mxl
Bwv 323 (musicxml): bach/bwv323.mxl
Bwv 324 (musicxml): bach/bwv324.mxl
Bwv 325 (musicxml): bach/bwv325.mxl
Bwv 326 (musicxml): bach/bwv326.mxl
Bwv 327 (musicxml): bach/bwv327.mxl
Bwv 328 (musicxml): bach/bwv328.mxl
Bwv 329 (musicxml): bach/bwv329.mxl
Bwv 33.6 (musicxml): bach/bwv33.6.mxl
Bwv 330 (musicxml): bach/bwv330.mxl
Bwv 331 (musicxml): bach/bwv331.mxl
Bwv 332 (musicxml): bach/bwv332.mxl
Bwv 333 (musicxml): bach/bwv333.mxl
Bwv 334 (musicxml): bach/bwv334.mxl
Bwv 335 (musicxml): bach/bwv335.mxl
Bwv 336 (musicxml): bach/bwv336.mxl
Bwv 337 (musicxml): bach/bwv337.mxl
Bwv 338 (musicxml): bach/bwv338.mxl
Bwv 339 (musicxml): bach/bwv339.mxl
Bwv 340 (musicxml): bach/bwv340.mxl
Bwv 341 (musicxml): bach/bwv341.mxl
Bwv 342 (musicxml): bach/bwv342.mxl
Bwv 343 (musicxml): bach/bwv343.mxl
Bwv 344 (musicxml): bach/bwv344.mxl
Bwv 345 (musicxml): bach/bwv345.mxl
Bwv 346 (musicxml): bach/bwv346.mxl
Bwv 347 (musicxml): bach/bwv347.mxl
Bwv 348 (musicxml): bach/bwv348.mxl
Bwv 349 (musicxml): bach/bwv349.mxl
Bwv 350 (musicxml): bach/bwv350.mxl
Bwv 351 (musicxml): bach/bwv351.mxl
Bwv 352 (musicxml): bach/bwv352.mxl
Bwv 353 (musicxml): bach/bwv353.mxl
Bwv 354 (musicxml): bach/bwv354.mxl
Bwv 355 (musicxml): bach/bwv355.mxl
Bwv 356 (musicxml): bach/bwv356.mxl

Bwv 357 (musicxml): bach/bwv357.mxl
Bwv 358 (musicxml): bach/bwv358.mxl
Bwv 359 (musicxml): bach/bwv359.mxl
Bwv 36.4 - 2 (musicxml): bach/bwv36.4-2.mxl
Bwv 36.8 - 2 (musicxml): bach/bwv36.8-2.mxl
Bwv 360 (musicxml): bach/bwv360.mxl
Bwv 361 (musicxml): bach/bwv361.mxl
Bwv 362 (musicxml): bach/bwv362.mxl

Bwv 363 (musicxml): bach/bwv363.mxl
Bwv 364 (musicxml): bach/bwv364.mxl
Bwv 365 (musicxml): bach/bwv365.mxl
Bwv 366

- Bwv 366 (humdrum): bach/bwv366.krn
- Bwv 366 (musicxml): bach/bwv366.mxl

Bwv 367 (musicxml): bach/bwv367.mxl
Bwv 368 (musicxml): bach/bwv368.mxl
Bwv 369 (musicxml): bach/bwv369.mxl
Bwv 37.6 (musicxml): bach/bwv37.6.mxl
Bwv 370 (musicxml): bach/bwv370.mxl
Bwv 371 (musicxml): bach/bwv371.mxl
Bwv 372 (musicxml): bach/bwv372.mxl
Bwv 373 (musicxml): bach/bwv373.mxl
Bwv 374 (musicxml): bach/bwv374.mxl
Bwv 375 (musicxml): bach/bwv375.mxl
Bwv 376 (musicxml): bach/bwv376.mxl
Bwv 377 (musicxml): bach/bwv377.mxl
Bwv 378 (musicxml): bach/bwv378.mxl
Bwv 379 (musicxml): bach/bwv379.mxl
Bwv 38.6 (musicxml): bach/bwv38.6.mxl
Bwv 380 (musicxml): bach/bwv380.mxl
Bwv 381 (musicxml): bach/bwv381.mxl
Bwv 382 (musicxml): bach/bwv382.mxl
Bwv 383 (musicxml): bach/bwv383.mxl
Bwv 384 (musicxml): bach/bwv384.mxl
Bwv 385 (musicxml): bach/bwv385.mxl
Bwv 386 (musicxml): bach/bwv386.mxl
Bwv 387 (musicxml): bach/bwv387.mxl

Bwv 388 (musicxml): bach/bwv388.xml
Bwv 389 (musicxml): bach/bwv389.xml
Bwv 39.7 (musicxml): bach/bwv39.7.xml
Bwv 390 (musicxml): bach/bwv390.xml
Bwv 391 (musicxml): bach/bwv391.xml
Bwv 392 (musicxml): bach/bwv392.xml
Bwv 393 (musicxml): bach/bwv393.xml
Bwv 394 (musicxml): bach/bwv394.xml
Bwv 395 (musicxml): bach/bwv395.xml
Bwv 396 (musicxml): bach/bwv396.xml
Bwv 397 (musicxml): bach/bwv397.xml
Bwv 398 (musicxml): bach/bwv398.xml
Bwv 399 (musicxml): bach/bwv399.xml
Bwv 4.8 (musicxml): bach/bwv4.8.xml
Bwv 40.3 (musicxml): bach/bwv40.3.xml
Bwv 40.6 (musicxml): bach/bwv40.6.xml
Bwv 40.8 (musicxml): bach/bwv40.8.xml
Bwv 400 (musicxml): bach/bwv400.xml
Bwv 401 (musicxml): bach/bwv401.xml
Bwv 402 (musicxml): bach/bwv402.xml
Bwv 403 (musicxml): bach/bwv403.xml
Bwv 404 (musicxml): bach/bwv404.xml
Bwv 405 (musicxml): bach/bwv405.xml
Bwv 406 (musicxml): bach/bwv406.xml
Bwv 407 (musicxml): bach/bwv407.xml
Bwv 408 (musicxml): bach/bwv408.xml
Bwv 409 (musicxml): bach/bwv409.xml
Bwv 41.6 (musicxml): bach/bwv41.6.xml
Bwv 410 (musicxml): bach/bwv410.xml
Bwv 411 (musicxml): bach/bwv411.xml
Bwv 412 (musicxml): bach/bwv412.xml
Bwv 413 (musicxml): bach/bwv413.xml
Bwv 414 (musicxml): bach/bwv414.xml
Bwv 415 (musicxml): bach/bwv415.xml
Bwv 416 (musicxml): bach/bwv416.xml
Bwv 417 (musicxml): bach/bwv417.xml
Bwv 418 (musicxml): bach/bwv418.xml
Bwv 419 (musicxml): bach/bwv419.xml
Bwv 42.7 (musicxml): bach/bwv42.7.xml

Bwv 420 (musicxml): bach/bwv420.mxl
Bwv 421 (musicxml): bach/bwv421.mxl
Bwv 422 (musicxml): bach/bwv422.mxl
Bwv 423 (musicxml): bach/bwv423.mxl
Bwv 424 (musicxml): bach/bwv424.mxl
Bwv 425 (musicxml): bach/bwv425.mxl
Bwv 426 (musicxml): bach/bwv426.mxl
Bwv 427 (musicxml): bach/bwv427.mxl
Bwv 428 (musicxml): bach/bwv428.mxl
Bwv 429 (musicxml): bach/bwv429.mxl
Bwv 43.11 (musicxml): bach/bwv43.11.mxl
Bwv 430 (musicxml): bach/bwv430.mxl
Bwv 431 (musicxml): bach/bwv431.mxl
Bwv 432 (musicxml): bach/bwv432.mxl
Bwv 433 (musicxml): bach/bwv433.mxl
Bwv 434 (musicxml): bach/bwv434.mxl
Bwv 435 (musicxml): bach/bwv435.mxl
Bwv 436 (musicxml): bach/bwv436.mxl
Bwv 437 (musicxml): bach/bwv437.mxl
Bwv 438 (musicxml): bach/bwv438.mxl
Bwv 44.7 (musicxml): bach/bwv44.7.mxl
Bwv 45.7 (musicxml): bach/bwv45.7.mxl
Bwv 46.6 (musicxml): bach/bwv46.6.mxl
Bwv 47.5 (musicxml): bach/bwv47.5.mxl
Bwv 48.3 (musicxml): bach/bwv48.3.mxl
Bwv 48.7 (musicxml): bach/bwv48.7.mxl
Bwv 5.7 (musicxml): bach/bwv5.7.mxl
Bwv 52.6 (musicxml): bach/bwv52.6.mxl
Bwv 55.5 (musicxml): bach/bwv55.5.mxl
Bwv 56.5 (musicxml): bach/bwv56.5.mxl
Bwv 57.8 (musicxml): bach/bwv57.8.mxl
Bwv 59.3 (musicxml): bach/bwv59.3.mxl
Bwv 6.6 (musicxml): bach/bwv6.6.mxl
Bwv 60.5 (musicxml): bach/bwv60.5.mxl
Bwv 62.6 (musicxml): bach/bwv62.6.mxl
Bwv 64.2 (musicxml): bach/bwv64.2.mxl
Bwv 64.4 (musicxml): bach/bwv64.4.mxl
Bwv 64.8 (musicxml): bach/bwv64.8.mxl
Bwv 65.2 (musicxml): bach/bwv65.2.mxl

Bwv 65.7 (musicxml): bach/bwv65.7.xml
Bwv 66.6 (musicxml): bach/bwv66.6.xml
Bwv 67.4 (musicxml): bach/bwv67.4.xml
Bwv 67.7 (musicxml): bach/bwv67.7.xml
Bwv 69.6 -A (musicxml): bach/bwv69.6-a.xml
Bwv 69.6 (musicxml): bach/bwv69.6.xml
Bwv 7.7 (musicxml): bach/bwv7.7.xml
Bwv 70.11 (musicxml): bach/bwv70.11.xml
Bwv 70.7 (musicxml): bach/bwv70.7.xml
Bwv 72.6 (musicxml): bach/bwv72.6.xml
Bwv 73.5 (musicxml): bach/bwv73.5.xml
Bwv 74.8 (musicxml): bach/bwv74.8.xml
Bwv 77.6 (musicxml): bach/bwv77.6.xml
Bwv 78.7 (musicxml): bach/bwv78.7.xml
Bwv 79.3 (musicxml): bach/bwv79.3.xml
Bwv 79.6 (musicxml): bach/bwv79.6.xml
Bwv 8.6 (musicxml): bach/bwv8.6.xml
Bwv 80.8 (musicxml): bach/bwv80.8.xml
Bwv 81.7 (musicxml): bach/bwv81.7.xml
Bwv 83.5 (musicxml): bach/bwv83.5.xml
Bwv 84.5 (musicxml): bach/bwv84.5.xml
Bwv 846 (musicxml): bach/bwv846.xml
Bwv 85.6 (musicxml): bach/bwv85.6.xml
Bwv 86.6 (musicxml): bach/bwv86.6.xml
Bwv 87.7 (musicxml): bach/bwv87.7.xml
Bwv 88.7 (musicxml): bach/bwv88.7.xml
Bwv 89.6 (musicxml): bach/bwv89.6.xml
Bwv 9.7 (musicxml): bach/bwv9.7.xml
Bwv 90.5 (musicxml): bach/bwv90.5.xml
Bwv 91.6 (musicxml): bach/bwv91.6.xml
Bwv 92.9 (musicxml): bach/bwv92.9.xml
Bwv 93.7 (musicxml): bach/bwv93.7.xml
Bwv 94.8 (musicxml): bach/bwv94.8.xml
Bwv 95.7 (musicxml): bach/bwv95.7.xml
Bwv 96.6 (musicxml): bach/bwv96.6.xml
Bwv 97.9 (musicxml): bach/bwv97.9.xml
Bwv 99.6 (musicxml): bach/bwv99.6.xml

Amy Beach

Child (musicxml): beach/prayer_of_a_tired_child.musicxml

Ludwig van Beethoven

Opus 132 (musicxml): beethoven/opus132.xml

Opus 133 (musicxml): beethoven/opus133.xml

Opus 18 No 1

- Movement 1 (humdrum): beethoven/opus18no1/movement1.krn
- Movement 1 (musicxml): beethoven/opus18no1/movement1.xml
- Movement 2 (humdrum): beethoven/opus18no1/movement2.krn
- Movement 2 (musicxml): beethoven/opus18no1/movement2.xml
- Movement 3 (humdrum): beethoven/opus18no1/movement3.krn
- Movement 3 (musicxml): beethoven/opus18no1/movement3.xml
- Movement 4 (humdrum): beethoven/opus18no1/movement4.krn
- Movement 4 (musicxml): beethoven/opus18no1/movement4.xml

Opus 18 No 3 (musicxml): beethoven/opus18no3.xml

Opus 18 No 4 (musicxml): beethoven/opus18no4.xml

Opus 18 No 5 (musicxml): beethoven/opus18no5.xml Opus 59

No 1

- Movement 1 (musicxml): beethoven/opus59no1/movement1.xml
- Movement 2 (musicxml): beethoven/opus59no1/movement2.xml
- Movement 3 (musicxml): beethoven/opus59no1/movement3.xml
- Movement 4 (musicxml):

beethoven/opus59no1/movement4.xml Opus 59 No 2

- Movement 1 (musicxml): beethoven/opus59no2/movement1.xml
- Movement 2 (musicxml): beethoven/opus59no2/movement2.xml
- Movement 3 (musicxml): beethoven/opus59no2/movement3.xml

- Movement 4 (musicxml):
beethoven/opus59no2/movement4.xml Opus 59 No 3
- Movement 1 (musicxml): beethoven/opus59no3/movement1.xml
- Movement 2 (musicxml): beethoven/opus59no3/movement2.xml
- Movement 3 (musicxml): beethoven/opus59no3/movement3.xml
- Movement 4 (musicxml): beethoven/opus59no3/movement4.xml

Opus 74 (musicxml): beethoven/opus74.xml

Johannes Ciconia

Quod Jactatur (musicxml): ciconia/quod_jactatur.xml

Arcangelo Corelli

1 Grave (musicxml): corelli/opus3no1/1grave.xml

C.P.EBach

H 186 (musicxml): cpebach/h186.xml

George Frideric Handel

Lascia Chio Pianga (musicxml): handel/rinaldo/Lascia_chio_pianga.xml

Concerto Grosso em Ré maior Op.6 no. 5 (musicxml): handel/grosso0605.xml

Joseph Haydn

Opus 1 No 1

- Movement 1 (musicxml): haydn/opus1no1/movement1.xml
- Movement 2 (musicxml): haydn/opus1no1/movement2.xml
- Movement 3 (musicxml): haydn/opus1no1/movement3.xml
- Movement 4 (musicxml): haydn/opus1no1/movement4.xml
- Movement 5 (musicxml):

haydn/opus1no1/movement5.xml Opus 74 No 1

- Movement 1 (musicxml): haydn/opus74no1/movement1.xml
- Movement 2 (musicxml): haydn/opus74no1/movement2.xml
- Movement 3 (musicxml): haydn/opus74no1/movement3.xml
- Movement 4 (musicxml): haydn/opus74no1/movement4.xml

Scott Joplin

Maple Leaf Rag (musicxml): joplin/maple_leaf_rag.xml

DLuca

Gloria (musicxml): luca/gloria.xml

Claudio Monteverdi

Madrigal .3.1

- Madrigal .3.1 (musicxml): monteverdi/madrigal.3.1.xml
- Madrigal .3.1 (romantext):
monteverdi/madrigal.3.1.rntxt Madrigal .3.10
- Madrigal .3.10 (musicxml): monteverdi/madrigal.3.10.xml
- Madrigal .3.10 (romantext):
monteverdi/madrigal.3.10.rntxt Madrigal .3.11
- Madrigal .3.11 (musicxml): monteverdi/madrigal.3.11.xml
- Madrigal .3.11 (romantext):
monteverdi/madrigal.3.11.rntxt Madrigal .3.12
- Madrigal .3.12 (musicxml): monteverdi/madrigal.3.12.xml
- Madrigal .3.12 (romantext):
monteverdi/madrigal.3.12.rntxt Madrigal .3.13
- Madrigal .3.13 (musicxml): monteverdi/madrigal.3.13.xml

- Madrigal .3.13 (romantext):
monteverdi/madrigal.3.13.rntxt Madrigal .3.14
- Madrigal .3.14 (musicxml): monteverdi/madrigal.3.14.xml
- Madrigal .3.14 (romantext):
monteverdi/madrigal.3.14.rntxt Madrigal .3.15
- Madrigal .3.15 (musicxml): monteverdi/madrigal.3.15.xml
- Madrigal .3.15 (romantext):
monteverdi/madrigal.3.15.rntxt Madrigal .3.16
- Madrigal .3.16 (musicxml): monteverdi/madrigal.3.16.xml
- Madrigal .3.16 (romantext): monteverdi/madrigal.3.16.rntxt

Madrigal .3.17

- Madrigal .3.17 (musicxml): monteverdi/madrigal.3.17.xml
- Madrigal .3.17 (romantext):
monteverdi/madrigal.3.17.rntxt Madrigal .3.18
- Madrigal .3.18 (musicxml): monteverdi/madrigal.3.18.xml
- Madrigal .3.18 (romantext):
monteverdi/madrigal.3.18.rntxt Madrigal .3.19
- Madrigal .3.19 (musicxml): monteverdi/madrigal.3.19.xml
- Madrigal .3.19 (romantext): monteverdi/madrigal.3.19.rntxt

Madrigal .3.1 Old (musicxml): monteverdi/madrigal.3.1_old.xml

Madrigal .3.2

- Madrigal .3.2 (musicxml): monteverdi/madrigal.3.2.xml
- Madrigal .3.2 (romantext):
monteverdi/madrigal.3.2.rntxt Madrigal .3.20
- Madrigal .3.20 (musicxml): monteverdi/madrigal.3.20.xml

- Madrigal .3.20 (romantext):
monteverdi/madrigal.3.20.rntxt Madrigal .3.3
- Madrigal .3.3 (musicxml): monteverdi/madrigal.3.3.mxl
- Madrigal .3.3 (romantext):
monteverdi/madrigal.3.3.rntxt Madrigal .3.4
- Madrigal .3.4 (musicxml): monteverdi/madrigal.3.4.mxl
- Madrigal .3.4 (romantext):
monteverdi/madrigal.3.4.rntxt Madrigal .3.5
- Madrigal .3.5 (musicxml): monteverdi/madrigal.3.5.mxl
- Madrigal .3.5 (romantext): monteverdi/madrigal.3.5.rntxt

Madrigal .3.6

- Madrigal .3.6 (musicxml): monteverdi/madrigal.3.6.mxl
- Madrigal .3.6 (romantext):
monteverdi/madrigal.3.6.rntxt Madrigal .3.7
- Madrigal .3.7 (musicxml): monteverdi/madrigal.3.7.mxl
- Madrigal .3.7 (romantext):
monteverdi/madrigal.3.7.rntxt Madrigal .3.8
- Madrigal .3.8 (musicxml): monteverdi/madrigal.3.8.mxl
- Madrigal .3.8 (romantext):
monteverdi/madrigal.3.8.rntxt Madrigal .3.9
- Madrigal .3.9 (musicxml): monteverdi/madrigal.3.9.mxl
- Madrigal .3.9 (romantext):
monteverdi/madrigal.3.9.rntxt Madrigal .4.1
- Madrigal .4.1 (musicxml): monteverdi/madrigal.4.1.mxl

- Madrigal .4.1 (romantext):
monteverdi/madrigal.4.1.rntxt Madrigal .4.10
- Madrigal .4.10 (musicxml): monteverdi/madrigal.4.10.xml
- Madrigal .4.10 (romantext):
monteverdi/madrigal.4.10.rntxt Madrigal .4.11
- Madrigal .4.11 (musicxml): monteverdi/madrigal.4.11.xml
- Madrigal .4.11 (romantext):
monteverdi/madrigal.4.11.rntxt Madrigal .4.12
- Madrigal .4.12 (musicxml): monteverdi/madrigal.4.12.xml
- Madrigal .4.12 (romantext):
monteverdi/madrigal.4.12.rntxt Madrigal .4.13
- Madrigal .4.13 (musicxml): monteverdi/madrigal.4.13.xml
- Madrigal .4.13 (romantext):
monteverdi/madrigal.4.13.rntxt Madrigal .4.14
- Madrigal .4.14 (musicxml): monteverdi/madrigal.4.14.xml
- Madrigal .4.14 (romantext):
monteverdi/madrigal.4.14.rntxt Madrigal .4.15
- Madrigal .4.15 (musicxml): monteverdi/madrigal.4.15.xml
- Madrigal .4.15 (romantext):
monteverdi/madrigal.4.15.rntxt Madrigal .4.16
- Madrigal .4.16 (musicxml): monteverdi/madrigal.4.16.xml
- Madrigal .4.16 (romantext):
monteverdi/madrigal.4.16.rntxt Madrigal .4.17
- Madrigal .4.17 (musicxml): monteverdi/madrigal.4.17.xml

- Madrigal .4.17 (romantext):
monteverdi/madrigal.4.17.rntxt Madrigal .4.18
- Madrigal .4.18 (musicxml): monteverdi/madrigal.4.18.mxl
- Madrigal .4.18 (romantext):
monteverdi/madrigal.4.18.rntxt Madrigal .4.19
- Madrigal .4.19 (musicxml): monteverdi/madrigal.4.19.mxl
- Madrigal .4.19 (romantext):
monteverdi/madrigal.4.19.rntxt Madrigal .4.2
- Madrigal .4.2 (musicxml): monteverdi/madrigal.4.2.mxl
- Madrigal .4.2 (romantext):
monteverdi/madrigal.4.2.rntxt Madrigal .4.20
- Madrigal .4.20 (musicxml): monteverdi/madrigal.4.20.mxl
- Madrigal .4.20 (romantext):
monteverdi/madrigal.4.20.rntxt Madrigal .4.3
- Madrigal .4.3 (musicxml): monteverdi/madrigal.4.3.mxl
- Madrigal .4.3 (romantext):
monteverdi/madrigal.4.3.rntxt Madrigal .4.4
- Madrigal .4.4 (musicxml): monteverdi/madrigal.4.4.mxl
- Madrigal .4.4 (romantext):
monteverdi/madrigal.4.4.rntxt Madrigal .4.5
- Madrigal .4.5 (musicxml): monteverdi/madrigal.4.5.mxl
- Madrigal .4.5 (romantext):
monteverdi/madrigal.4.5.rntxt Madrigal .4.6
- Madrigal .4.6 (musicxml): monteverdi/madrigal.4.6.mxl

- Madrigal .4.6 (romantext):
monteverdi/madrigal.4.6.rntxt Madrigal .4.7
- Madrigal .4.7 (musicxml): monteverdi/madrigal.4.7.xml
- Madrigal .4.7 (romantext):
monteverdi/madrigal.4.7.rntxt Madrigal .4.8
- Madrigal .4.8 (musicxml): monteverdi/madrigal.4.8.xml
- Madrigal .4.8 (romantext):
monteverdi/madrigal.4.8.rntxt Madrigal .4.9
- Madrigal .4.9 (musicxml): monteverdi/madrigal.4.9.xml
- Madrigal .4.9 (romantext):
monteverdi/madrigal.4.9.rntxt Madrigal .5.1
- Madrigal .5.1 (musicxml): monteverdi/madrigal.5.1.xml
- Madrigal .5.1 (romantext): monteverdi/madrigal.5.1.rntxt

Madrigal .5.2

- Madrigal .5.2 (musicxml): monteverdi/madrigal.5.2.xml
- Madrigal .5.2 (romantext):
monteverdi/madrigal.5.2.rntxt Madrigal .5.3
- Madrigal .5.3 (musicxml): monteverdi/madrigal.5.3.xml
- Madrigal .5.3 (romantext):
monteverdi/madrigal.5.3.rntxt Madrigal .5.4
- Madrigal .5.4 (musicxml): monteverdi/madrigal.5.4.xml
- Madrigal .5.4 (romantext):
monteverdi/madrigal.5.4.rntxt Madrigal .5.5
- Madrigal .5.5 (musicxml): monteverdi/madrigal.5.5.xml

- Madrigal .5.5 (romantext):
monteverdi/madrigal.5.5.rntxt Madrigal .5.6
- Madrigal .5.6 (musicxml): monteverdi/madrigal.5.6.xml
- Madrigal .5.6 (romantext):
monteverdi/madrigal.5.6.rntxt Madrigal .5.7
- Madrigal .5.7 (musicxml): monteverdi/madrigal.5.7.xml
- Madrigal .5.7 (romantext):
monteverdi/madrigal.5.7.rntxt Madrigal .5.8
- Madrigal .5.8 (musicxml): monteverdi/madrigal.5.8.xml
- Madrigal .5.8 (romantext): monteverdi/madrigal.5.8.rntxt

Wolfgang Amadeus Mozart K 155

- Movement 1 (musicxml): mozart/k155/movement1.xml
- Movement 2 (musicxml): mozart/k155/movement2.xml
- Movement 3 (musicxml):
mozart/k155/movement3.xml K 156
- Movement 1 (musicxml): mozart/k156/movement1.xml
- Movement 2 (musicxml): mozart/k156/movement2.xml
- Movement 3 (musicxml): mozart/k156/movement3.xml
- Movement 4 (musicxml):
mozart/k156/movement4.xml K 458
- Movement 1 (musicxml): mozart/k458/movement1.xml
- Movement 2 (musicxml): mozart/k458/movement2.xml
- Movement 3 (musicxml): mozart/k458/movement3.xml
- Movement 4 (musicxml): mozart/k458/movement4.xml

Movement 1 Exposition (musicxml): mozart/k545/movement1_exposition.xml K 80

- Movement 1 (musicxml): mozart/k80/movement1.xml
- Movement 2 (musicxml): mozart/k80/movement2.xml
- Movement 3 (musicxml): mozart/k80/movement3.xml
- Movement 4 (musicxml): mozart/k80/movement4.xml

Arnold Schoenberg

Opus 19

- Movement 2 (musicxml): schoenberg/opus19/movement2.xml
- Movement 6 (musicxml): schoenberg/opus19/movement6.xml

Franz Schubert

Lindenbaum (musicxml): schubert/Lindenbaum.xml

Robert Schumann

Dichterliebe No 2 (musicxml): schumann/dichterliebe_no2.xml

Opus 41 No 1

- Movement 1 (musicxml): schumann/opus41no1/movement1.xml
- Movement 2 (musicxml): schumann/opus41no1/movement2.xml
- Movement 3 (musicxml): schumann/opus41no1/movement3.xml
- Movement 4 (musicxml): schumann/opus41no1/movement4.xml
- Movement 5 (musicxml): schumann/opus41no1/movement5.xml

Opus 48 No 2 (musicxml): schumann/opus48no2.xml

Clara Schumann

Movement 3 (musicxml): schumann_clara/opus17/movement3.xml

Polonaise Op 1 N 1 (musicxml): schumann_clara/polonaise_op1n1.xml

Polonaise Op 1 N 2 (musicxml): schumann_clara/polonaise_op1n2.xml

Polonaise Op 1 N 3 (musicxml): schumann_clara/polonaise_op1n3.xml

Polonaise Op 1 N 4 (musicxml): schumann_clara/polonaise_op1n4.mxl

Fourteenth-Century Italian Music

Fava Dicant Nunc Iudei (musicxml): trecento/Fava_Dicant_nunc_iudei.xml

Pmfc 01 - Lugentium Siccentur (musicxml): trecento/PMFC_01-Lugentium siccentur.xml

Pmfc 01 - Rex Quem Metrorum (musicxml): trecento/PMFC_01-Rex quem metrorum.xml

Pmfc 01 - Virtutibus Laudabilis (musicxml): trecento/PMFC_01-Virtutibus laudabilis.xml

Pmfc 01 - Vos Qui Admiramini Gratissima Virginis Species (musicxml): trecento/PMFC_01-Vos Qui Admiramini Gratissima virginis species.xml

Pmfc 04 - A Lle S Andra Lo Spirt (musicxml): trecento/PMFC_04-A lle s_andra lo spirt.xml

Pmfc 04 - Cara Mi Donna (musicxml): trecento/PMFC_04-Cara mi donna.xml

Pmfc 04 - Quanto Piu Caro (musicxml): trecento/PMFC_04-Quanto piu caro.xml

Pmfc 04 - Si Dolce Non Sono (musicxml): trecento/PMFC_04-Si dolce non sono.xml

Pmfc 06 - Jacopo- 01 - Aquila- Altera (musicxml): trecento/PMFC_06-Jacopo-01-Aquila-Altera.xml

Pmfc 06 - Jacopo- 02 - Con- Gran- Furor (musicxml): trecento/PMFC_06-Jacopo-02-Con-Gran-Furor.xml

Pmfc 06 - Jacopo- 03 A- Di Novo (musicxml): trecento/PMFC_06-Jacopo-03a-Di_Novo.xml

Pmfc 06 - Jacopo- 03 B- Di Novo (musicxml): trecento/PMFC_06-Jacopo-03b-Di_Novo.xml

Pmfc 06 10 - I Senti Gia Come L Arco (musicxml): trecento/PMFC_06_10-I senti gia come l_arco.xml

Pmfc 06 8 - In Verde Prato (musicxml): trecento/PMFC_06_8-In Verde Prato.xml

Pmfc 06 Giovanni- 01 - Angnel Son Bianco (musicxml): trecento/PMFC_06_Giovanni-01-Angnel_son_biancho.xml

Pmfc 06 Giovanni- 02 - Apress Un Fiume A (musicxml): trecento/PMFC_06_Giovanni-02-Apress_un_fiume_a.xml

Pmfc 06 Giovanni- 03 - Chon Brachi Assai (musicxml): trecento/PMFC_06_Giovanni-03-Chon_brachi_assai.xml

Pmfc 06 Giovanni- 04 De Come Dolce Mente (musicxml): trecento/PMFC_06_Giovanni-04_De_Come_Dolce_Mente.xml

Pmfc 06 Giovanni- 05 Donna Gia Fu Leggiadra (musicxml):
trecento/PMFC_06_Giovanni-05_Donna_Gia_Fu_Leggiadra.xml

Pmfc 06 Giovanni- 06 Fra Mille Corvi (musicxml): trecento/PMFC_06_Giovanni-
06_Fra_Mille_Corvi.xml

Pmfc 06 Giovanni- 07 In Sulla Ripa (musicxml): trecento/PMFC_06_Giovanni-
07_In_Sulla_Ripa.xml

Pmfc 06 Giovanni- 09 Nascoso El Viso (musicxml):
trecento/PMFC_06_Giovanni-09_Nascoso_El_Viso.xml

Pmfc 06 Piero 1 - All Onbra D Un Perlato (musicxml):
trecento/PMFC_06_Piero_1-All_onbra_d_un_perlato.xml

Pmfc 06 Piero 2 - Chavalcando (musicxml): trecento/PMFC_06_Piero_2-Chavalcando.xml

Pmfc 06 Piero 6 A- Quando Laire Comenca (musicxml):
trecento/PMFC_06_Piero_6a-Quando_laire_comenca.xml

Pmfc 12 1 - Kyrie Rondello (musicxml): trecento/PMFC_12_1-Kyrie Rondello.xml

Pmfc 12 10 A- Credo Pmfc 12.10 A (musicxml): trecento/PMFC_12_10a-Credo
PMFC12.10a.xml

Pmfc 12 10 B- Credo Pmfc 12.10 B (musicxml): trecento/PMFC_12_10b-Credo
PMFC12.10b.xml

Pmfc 12 11 A- Credo Pmfc 12.11 A (musicxml): trecento/PMFC_12_11a-Credo
PMFC12.11a.xml

Pmfc 12 11 B- Credo Pmfc 12.11 B (musicxml): trecento/PMFC_12_11b-Credo
PMFC12.11b.xml

Pmfc 12 12 - Credo Bartholus (musicxml): trecento/PMFC_12_12-Credo Bartho-lus.xml

Pmfc 12 13 - Credo London F 83 - 85 (musicxml): trecento/PMFC_12_13-Credo
London f83-85.xml

Pmfc 12 14 - Credo Phillipoctus (musicxml): trecento/PMFC_12_14-Credo
Phillipoctus.xml

Pmfc 12 15 - Sanctus Lorenzo (musicxml): trecento/PMFC_12_15-Sanctus
Lorenzo.xml

Pmfc 12 16 - Sanctus Pmfc 12.16 (musicxml): trecento/PMFC_12_16-Sanctus
PMFC12.16.xml

Pmfc 12 17 - Sanctus Gratosus (musicxml): trecento/PMFC_12_17-Sanctus
Gratosus.xml

Pmfc 12 18 - Sanctus Mediolano (musicxml): trecento/PMFC_12_18-Sanctus
Mediolano.xml

Pmfc 12 19 - Sanctus Barbitonsoris (musicxml): trecento/PMFC_12_19-Sanctus
Barbitonsoris.xml

Pmfc 12 2 - Gloria (Gubbio) (musicxml): trecento/PMFC_12_2-Gloria (Gub-bio).xml

Pmfc 12 20 - Agnus Dei Gherardello (musicxml): trecento/PMFC_12_20-Agnus Dei Gherardello.xml

Pmfc 12 21 - Benedicamus Pmfc 12.21 (musicxml): trecento/PMFC_12_21-Benedicamus PMFC12.21.xml

Pmfc 12 22 - Benedicamus Pmfc 12.22 (musicxml): trecento/PMFC_12_22-Benedicamus PMFC12.22.xml

Pmfc 12 23 - Benedicamus Pmfc 12.23 (musicxml): trecento/PMFC_12_23-Benedicamus PMFC12.23.xml

Pmfc 12 24 - Benedicamus (musicxml): trecento/PMFC_12_24-Benedicamus.xml

Pmfc 12 3 - Gloria Gherardello (musicxml): trecento/PMFC_12_3-Gloria Gherardello.xml

Pmfc 12 4 - Gloria Rvat 1419 (musicxml): trecento/PMFC_12_4-Gloria Rvat 1419.xml

Pmfc 12 4 - Gloria Rvat 1419 Bak (musicxml): trecento/PMFC_12_4-Gloria Rvat 1419_bak.xml

Pmfc 12 5 - Gloria Pmfc 12.5 (musicxml): trecento/PMFC_12_5-Gloria PMFC 12.5.xml

Pmfc 12 6 - Gloria Gratosus (musicxml): trecento/PMFC_12_6-Gloria Gratio-sus.xml

Pmfc 12 7 - Gloria Egardus (musicxml): trecento/PMFC_12_7-Gloria Egar-dus.xml

Pmfc 12 8 - Gloria London 29987 (musicxml): trecento/PMFC_12_8-Gloria London 29987.xml

Pmfc 12 9 - Gloria Clementiae Pax (musicxml): trecento/PMFC_12_9-Gloria Clementiae Pax.xml

Pmfc 13 01 - Kyrie- Summe- Clementissime

- Pmfc 13 01 - Kyrie- Summe- Clementissime (musicxml): trecento/PMFC_13_01-Kyrie-Summe-Clementissime.mxl

- Pmfc 13 01 - Kyrie- Summe- Clementissime (musicxml): trecento/PMFC_13_01-Kyrie-Summe-Clementissime.xml

Pmfc 13 02 - Kyrie- Questa-Fanciulla (musicxml): trecento/PMFC_13_02-Kyrie-Questa-fanciulla.mxl

Pmfc 13 03 - Gloria Micinella (musicxml): trecento/PMFC_13_03-Gloria Mici-nella.xml

Pmfc 13 04 - Credo Cursor (musicxml): trecento/PMFC_13_04-Credo Cursor.xml

Pmfc 13 05 - Gloria Rosetta (musicxml): trecento/PMFC_13_05-Gloria Ro-setta.xml

Pmfc 13 06 - Gloria Fior Gentil (musicxml): trecento/PMFC_13_06-Gloria Fior Gentil.xml

Pmfc 13 07 - Gloria Laus Honor (musicxml): trecento/PMFC_13_07-Gloria Laus Honor.xml

- Pmfc 13 08 - Gloria (Ad Ongni Vento) (musicxml): trecento/PMFC_13_08-Gloria (Ad Ongni Vento).xml
- Pmfc 13 09 - Gloria Anglicana (musicxml): trecento/PMFC_13_09-Gloria Angli-cana.xml
- Pmfc 13 10 - Gloria (musicxml): trecento/PMFC_13_10-Gloria.xml
- Pmfc 13 12 - Gloria (musicxml): trecento/PMFC_13_12-Gloria.xml
- Pmfc 13 13 - Gloria (musicxml): trecento/PMFC_13_13-Gloria.xml
- Pmfc 13 14 - Gloria (musicxml): trecento/PMFC_13_14-Gloria.xml
- Pmfc 13 15 - Gloria (musicxml): trecento/PMFC_13_15-Gloria.xml
- Pmfc 13 16 - Gloria (musicxml): trecento/PMFC_13_16-Gloria.xml
- Pmfc 13 17 - Gloria Spiritus Et Alme (musicxml): trecento/PMFC_13_17-Gloria Spiritus et Alme.xml
- Pmfc 13 18 - Gloria Spiritus Et Alme (musicxml): trecento/PMFC_13_18-Gloria Spiritus et Alme.xml
- Pmfc 13 19 - Credo Scabroso (musicxml): trecento/PMFC_13_19-Credo Sca-broso.xml
- Pmfc 23 15 A- Kyrie Humano Generi A (musicxml): trecento/PMFC_23_15a-Kyrie Humano Generi a.xml
- Pmfc 23 15 B- Kyrie Humano Generi B (musicxml): trecento/PMFC_23_15b-Kyrie Humano Generi b.xml
- Pmfc 23 16 - Kyrie Apt 16 (musicxml): trecento/PMFC_23_16-Kyrie Apt 16.xml
- Pmfc 23 17 - Kyrie Principum Effectivum (musicxml): trecento/PMFC_23_17-Kyrie Principum Effectivum.xml
- Pmfc 23 18 - Kyrie Chipre (musicxml): trecento/PMFC_23_18-Kyrie Chipre.xml
- Pmfc 23 19 - Kyrie Perrinet (musicxml): trecento/PMFC_23_19-Kyrie Perrinet.xml
- Pmfc 23 20 - Kyrie O Sacra Virgo Beata (musicxml): trecento/PMFC_23_20-Kyrie O Sacra Virgo Beata.xml
- Pmfc 23 21 - Kyrie Guymont (musicxml): trecento/PMFC_23_21-Kyrie Guy-mont.xml
- Pmfc 23 22 - Kyrie Summe Clementissime (musicxml): trecento/PMFC_23_22-Kyrie Summe Clementissime.xml
- Pmfc 23 23 - Kyrie Rex Inmense Maiestasis (musicxml): trecento/PMFC_23_23-Kyrie Rex Inmense Maiestasis.xml
- Pmfc 23 24 - Kyrie Ave Desiderii (musicxml): trecento/PMFC_23_24-Kyrie Ave Desiderii.xml
- Pmfc 23 25 - Kyrie Barcelona 853 (musicxml): trecento/PMFC_23_25-Kyrie Barcelona 853.xml
- Pmfc 23 26 - Kyrie O Virgo Sacrata Maria (musicxml): trecento/PMFC_23_26-Kyrie O Virgo Sacrata Maria.xml
- Pmfc 24 1 - Gloria Ciconia 1 (musicxml): trecento/PMFC_24_1-Gloria Ciconia 1.xml
- Pmfc 24 10 - Credo Ciconia 10 (musicxml): trecento/PMFC_24_10-Credo Ciconia

10.xml

Pmfc 24 11 - Credo Ciconia 11 (musicxml): trecento/PMFC_24_11-Credo Ciconia

11.xml

Pmfc 24 15 - O Virum Omnimoda (musicxml): trecento/PMFC_24_15-O virum omnimoda.xml

Pmfc 24 16 - Albane Misse Celitus (musicxml): trecento/PMFC_24_16-Albane misse celitus.xml

Pmfc 24 17 - Doctorum Principem- Melodia Suavissima- Vir Mitis (musicxml): trecento/PMFC_24_17-Doctorum principem-Melodia suavissima-Vir mitis.xml

Pmfc 24 19 - Ut Te Per Omnes Celitus- Ingens Alumnus Padue (musicxml): trecento/PMFC_24_19-Ut te per omnes celitus- Ingens alumnus padue.xml

Pmfc 24 2 - Credo Ciconia 2 (musicxml): trecento/PMFC_24_2-Credo Ciconia 2.xml

Pmfc 24 20 - Padu . . . Serenans (musicxml): trecento/PMFC_24_20-Padu. . . serenans.xml

Pmfc 24 21 - O Proles Hispanie (musicxml): trecento/PMFC_24_21-O proles hispanie.xml

Pmfc 24 22 - O Beatum Incendium (musicxml): trecento/PMFC_24_22-O beatum incendium.xml

Pmfc 24 3 - Gloria Ciconia 3 (musicxml): trecento/PMFC_24_3-Gloria Ciconia 3.xml

Pmfc 24 4 - Credo Ciconia 4 (musicxml): trecento/PMFC_24_4-Credo Ciconia 4.xml

Pmfc 24 5 - Gloria Ciconia 5 Spiritus Et Alme (musicxml): trecento/PMFC_24_5-Gloria Ciconia 5 Spiritus et alme.xml

Pmfc 24 6 - Gloria Ciconia 6 Spiritus Et Alme (musicxml): trecento/PMFC_24_6-Gloria Ciconia 6 Spiritus et alme.xml

Pmfc 24 7 - Gloria Suscipe, Trinitas (musicxml): trecento/PMFC_24_7-Gloria Suscipe, trinitas.xml

Pmfc 24 8 - Gloria Ciconia 8 (musicxml): trecento/PMFC_24_8-Gloria Ciconia 8.xml

Pmfc 24 9 - Gloria Ciconia 9 (musicxml): trecento/PMFC_24_9-Gloria Ciconia 9.xml

Giuseppe Verdi

La Donna E Mobile (musicxml): verdi/laDonnaEMobile.mxl

Carl Maria von Weber

Concertino Clarinet (musicxml): weber/concertino_clarinet.mxl

ANEXO C - Lista de softwares compatíveis com o padrão MusicXML

- abc2xml
- abc4j
- Amadeus
- AnthemScore
- Archivarius 3000
- Arduino
- Audimus Notes
- AudioScore Ultimate
- audite PLUS
- Audiveris
- Audovia
- Auto-Tune EFX+
- Avid Scorch
- bach
- Band-in-a-Box
- Blackbinder
- Braille Music Compiler
- Braille Music Editor
- Braille Music Markup Language (BMML)
- BrailleMUSE
- BUZZle
- Cadencii
- Cakewalk

- Calligra Suite
- Canon
- Canorus
- capella
- capella playAlong
- capella reader
- capella start
- capella wave kit
- capella-scan
- Cavatina
- CelticPipes
- Chaconne
- CMME Editor
- Cornelius Composer
- Crescendo
- CrestMuse Toolkit
- CsoundAC
- Cubase
- D3 Pianogram
- daCapo
- Denemo
- Digital Performer
- Don's MusicXML Viewer
- Doo Bee Doo Composer
- Dorico
- Drumline Ensemble Composer

- EarMaster
- EasyABC
- Electric Pipes
- Encore
- Ensemble Composer
- Expresseur
- Fandango
- Finale
- Finale NotePad
- Flat
- FOMUS
- Forte
- Forte Sheet Music Reader
- Free Clef
- FreeDots
- Frescobaldi
- Frettable
- GLozart
- GoldenScroller
- GOODFEEL
- GStreamer
- GUIDO
- Guitar Pro
- Harmonia
- Harmony Assistant

- HarmonyWiz
- Haskell Library
- HTML5 Guitar Tab Player
- Humdrum Extras
- IBOS MusicXML Reader
- Impro-Visor
- INScore
- iReal Pro
- Java Music Specification Language (JMSL)
- jChing
- Jellynote
- JFugue
- Jniz
- KGuitar
- KlavarScript
- Komp
- Kooplet
- Kunkunshi Editor
- libMusicXML
- LilyPond
- Lime
- Logic Pro
- Ludwig
- MacroMusicUI
- Magenta
- MagicScore

- Manufaktura Controls
- Match My Sound
- MATLAB
- MaxScore
- McCarthy Music
- MDLscore
- Melody Assistant
- Melody Player
- Melody Scanner
- Melomics
- mercussion
- MidiAndMusicXmlPlayer
- mingus
- MorpheuS
- Mozart
- muDic
- muscript
- MuseBook Score
- MusEdit
- MuseScore
- Music Jotter
- Music Notation SDK
- Music Processing Suite
- Music Prodigy
- Music-to-XML

- music21
- MusicEase
- Musicista
- MusicJOT
- Musicrain
- MusicSQL
- MusicXML Analyzer
- MusicXML to MP3 Converter
- musicxml2mid
- musicxml2words
- MXMLiszt
- Myriad QuickLook Plug-In
- MyScript Music SDK
- Newzik
- Nightingale Notelist
- Notate
- NotateMe
- Notation Composer
- Notation Pad
- NoteAbility Pro
- Noteflight
- NoteWorthy Composer
- Notion
- Noutee
- Nuendo
- Obtiv Octava

- OpenMusic
- OpenSheetMusicDisplay
- Opus
- Opusmodus
- OrganMuse
- Ossia Viewer
- Overture
- Partitura
- PDFToMusic Pro
- PhonicScore
- PhotoScore Ultimate
- PHPMusicTools
- PianoLudic
- Pizzicato
- Plaine and Easie
- PlayScore
- PMX
- Power Tab
- Power Tracks Pro Audio
- Practice Bird
- Practice Player Live Midi
- PracticeFirst
- PriMus
- PrintMusic
- Progression

- ProxyMusic
- PSAM Control Library
- Ptolemaic
- Purely Musical
- PWGL
- pyScore
- QuickScore Elite Level II
- Rachmaniac Score
- ReadScoreLib
- REAPER
- Reflow
- Rosegarden
- Rousseau
- Samplitude
- ScanScore
- SCORE
- Score Creator
- Score Writer
- ScoreCloud
- Scored
- ScoreMaker
- Scorio
- Scroller
- SeeScore
- SeeScore SDK
- Sequoia

- SharpEye Music Reader
- Sheet Music Scanner
- Sibelius
- Sibelius Ultimate
- Simple Song Creator
- Singscope
- Sinsy
- SmartMusic
- SmartScore
- SmartScore NoteReader
- sol2snd
- Songs2See Editor
- Songs2See Game
- Sonja
- Soundslice
- Soundslice MusicXML Viewer
- Speech Analyzer
- StaffPad
- Stave'n'Tabs
- Stelupa
- StringyFi
- SuperScore
- Symphonix Evolution
- Symphony Pro
- Synfire

- TaBazar II
- TablEdit
- TEFpad
- TEFview
- teoría
- THoTH
- tonica fugata
- Touch Notation
- TuxGuitar
- VocalEasel
- Watch and Listen
- WhiteNote
- xml2abc
- Zong! Editor
- Zong! Player
- Zupfnoter
- mO

ANEXO D - MusicXML (Arquivos DTD)

Partwise

<!--

MusicXML Partwise DTD

Version 3.1

Copyright © 2004-2017 the Contributors to the MusicXML Specification, published by the W3C Music Notation Community Group under the W3C Community Final Specification Agreement (FSA):

<https://www.w3.org/community/about/agreements/final/>

A human-readable summary is available:

<https://www.w3.org/community/about/agreements/fsa-deed/>

->

<!--

The MusicXML format is designed to represent musical scores, specifically common western musical notation from the 17th century onwards. It is designed as an interchange format for notation, analysis, retrieval, and performance applications. Therefore it is intended to be sufficient, not optimal, for these applications.

The MusicXML format is based on the MuseData and Humdrum formats. Humdrum explicitly represents the two-dimensional nature of musical scores by a 2-D layout notation. Since the XML format is hierarchical, we cannot do this explicitly. Instead, there are two top-level formats:

partwise.dtd Represents scores by part/instrument
timewise.dtd Represents scores by time/measure

Thus partwise.dtd contains measures within each part, while timewise.dtd contains parts within each measure. XSLT stylesheets are provided to convert between the two formats.

The partwise and timewise score DTDs represent a single movement of music. Multiple movements or other musical collections are presented using opus.dtd. An opus document contains XLinks to individual scores.

Suggested use:

```
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD
MusicXML 3.1 Partwise//EN"
"http://www.musicxml.org/dtds/partwise.dtd">
```

This DTD is made up of a series of component DTD modules, all of which are included here.

→

```
<!-- Entities -->
```

```
<!--
```

The partwise and timewise entities are used with conditional sections to control the differences between the partwise and timewise DTDs. The values for these entities are what distinguish the partwise and timewise DTD files.

→

```
<!ENTITY % partwise "INCLUDE">
```

```
<!ENTITY % timewise "IGNORE">
```

```
<!-- Component DTD modules -->
```

```
<!--
```

The common DTD module contains the entities and elements that are shared among multiple component DTDs.

→

```
<!ENTITY % common PUBLIC "-//Recordare//ELEMENTS MusicXML
3.1 Common//EN" "common.mod">
```

```
%common;
```

```
<!--
```

The layout DTD module contains formatting information for pages, systems, staves, and measures.

→

```
<!ENTITY % layout PUBLIC "-//Recordare//ELEMENTS MusicXML
3.1 Layout//EN"
```

```
“layout.mod”>
```

```
%layout;
```

```
<!--
```

The identity DTD module contains identification and metadata elements.

```
—>
```

```
<!ENTITY % identity PUBLIC “-//Recordare//ELEMENTS  
MusicXML 3.1 Identity//EN” “identity.mod”>
```

```
%identity;
```

```
<!--
```

The attributes DTD module contains elements that usually change at the start of a measure, such as key signatures, time signatures, and clefs.

```
—>
```

```
<!ENTITY % attributes PUBLIC “-//Recordare//ELEMENTS  
MusicXML 3.1 Attributes//EN” “attributes.mod”>
```

```
%attributes;
```

```
<!--
```

The link DTD module contains XLink attributes.

```
—>
```

```
<!ENTITY % link PUBLIC  
“-//Recordare//ELEMENTS MusicXML 3.1 Link//EN”  
“link.mod”>
```

```
%link;
```

```
<!--
```

The note DTD module contains the bulk of the elements and attributes for a musical scores relating to individual notes and rests.

```
—>
```

```
<!ENTITY % note PUBLIC  
“-//Recordare//ELEMENTS MusicXML 3.1 Note//EN”  
“note.mod”>
```

```
%note;
```

```
<!--
```

The barline DTD module contains elements regarding barline style, repeats, and multiple endings.

→

```
<!ENTITY % barline PUBLIC
```

```
"-//Recordare//ELEMENTS MusicXML 3.1 Barline//EN"
```

```
"barline.mod">
```

```
%barline;
```

```
<!--
```

The direction DTD module contains elements for musical directions not tied to individual notes. This includes harmony and chord symbol elements.

→

```
<!ENTITY % direction PUBLIC "-//Recordare//ELEMENTS
```

```
MusicXML 3.1 Direction//EN" "direction.mod">
```

```
%direction;
```

```
<!--
```

The score DTD module contains the top-level elements for musical scores, including the root document elements.

→

```
<!ENTITY % score PUBLIC "-//Recordare//ELEMENTS
```

```
MusicXML 3.1 Score//EN" "score.mod">
```

```
%score;
```

Timewise

```
<!--
```

```
MusicXML Timewise DTD
```

```
Version 3.1
```

Copyright © 2004-2017 the Contributors to the MusicXML Specification, published by the W3C Music Notation Community Group under the W3C Community Final Specification Agreement (FSA):

<https://www.w3.org/community/about/agreements/final/>

A human-readable summary is available:

<https://www.w3.org/community/about/agreements/fsa-deed/>

→

<!--

The MusicXML format is designed to represent musical scores, specifically common western musical notation from the 17th century onwards. It is designed as an interchange format for notation, analysis, retrieval, and performance applications. Therefore it is intended to be sufficient, not optimal, for these applications.

The MusicXML format is based on the MuseData and Humdrum formats. Humdrum explicitly represents the two-dimensional nature of musical scores by a 2-D layout notation. Since the XML format is hierarchical, we cannot do this explicitly. Instead, there are two top-level formats:

partwise.dtd Represents scores by part/instrument
 timewise.dtd Represents scores by time/measure

Thus partwise.dtd contains measures within each part, while timewise.dtd contains parts within each measure. XSLT stylesheets are provided to convert between the two formats.

The partwise and timewise score DTDs represent a single movement of music. Multiple movements or other musical collections are presented using opus.dtd. An opus document contains XLinks to individual scores.

Suggested use:

```
<!DOCTYPE score-timewise PUBLIC "-//Recordare//DTD
MusicXML 3.1 Timewise//EN"
"http://www.musicxml.org/dtds/timewise.dtd">
```

This DTD is made up of a series of component DTD modules, all of which are included here.

→

<!-- Entities →

<!--

The partwise and timewise entities are used with conditional sections to control the differences between the partwise and timewise DTDs. The values for these entities are what distinguish the partwise and timewise

DTD files.

→

```
<!ENTITY % partwise "IGNORE">
<!ENTITY % timewise "INCLUDE">
<!-- Component DTD modules -->
<!--
```

The common DTD module contains the entities and elements that are shared among multiple component DTDs.

→

```
<!ENTITY % common PUBLIC "-//Recordare//ELEMENTS
MusicXML 3.1 Common//EN" "common.mod">
```

```
%common;
```

```
<!--
```

The layout DTD module contains formatting information for pages, systems, staves, and measures.

→

```
<!ENTITY % layout PUBLIC "-//Recordare//ELEMENTS
MusicXML 3.1 Layout//EN" "layout.mod">
```

```
%layout;
```

```
<!--
```

The identity DTD module contains identification and metadata elements.

→

```
<!ENTITY % identity PUBLIC "-//Recordare//ELEMENTS
MusicXML 3.1 Identity//EN" "identity.mod">
```

```
%identity;
```

```
<!--
```

The attributes DTD module contains elements that usually change at the start of a measure, such as key signatures, time signatures, and clefs.

→

```
<!ENTITY % attributes PUBLIC "-//Recordare//ELEMENTS
MusicXML 3.1 Attributes//EN" "attributes.mod">
```

```
%attributes;
```

```
<!--
```

The link DTD module contains XLink attributes.

→

```
<!ENTITY % link PUBLIC
 "-//Recordare//ELEMENTS MusicXML 3.1 Link//EN"
 "link.mod">
```

```
%link;
```

```
<!--
```

The note DTD module contains the bulk of the elements and attributes for a musical scores relating to individual notes and rests.

→

```
<!ENTITY % note PUBLIC
 "-//Recordare//ELEMENTS MusicXML 3.1 Note//EN"
 "note.mod">
```

```
%note;
```

```
<!--
```

The barline DTD module contains elements regarding barline style, repeats, and multiple endings.

→

```
<!ENTITY % barline PUBLIC "-//Recordare//ELEMENTS
MusicXML 3.1 Barline//EN" "barline.mod">
```

```
%barline;
```

```
<!--
```

The direction DTD module contains elements for musical directions not tied to individual notes. This includes harmony and chord symbol elements.

→

```
<!ENTITY % direction PUBLIC "-//Recordare//ELEMENTS
MusicXML 3.1 Direction//EN" "direction.mod">
```

%direction;

<!--

The score DTD module contains the top-level elements for musical scores, including the root document elements.

-->

```
<!ENTITY % score PUBLIC "-//Recordare//ELEMENTS
MusicXML 3.1 Score//EN" "score.mod">
```

%score;

Opus

<!--

MusicXML Opus DTD
Version 3.1

Copyright © 2004-2017 the Contributors to the MusicXML Specification, published by the W3C Music Notation Community Group under the W3C Community Final Specification Agreement (FSA):

<https://www.w3.org/community/about/agreements/final/>

A human-readable summary is available:

<https://www.w3.org/community/about/agreements/fsa-deed/>

-->

<!--

An opus collects MusicXML scores together into a larger entity. The individual scores could be movements in a symphony, scenes or acts in an opera, or songs in an album. The opus definition allows arbitrary nesting either via an opus (included in the document) or an opus-link (linked like scores). Future versions of the MusicXML format may expand this DTD to include reference data and other metadata related to musical scores.

Suggested use:

```
<!DOCTYPE opus PUBLIC "-//Recordare//DTD
MusicXML 3.1 Opus//EN"
"http://www.musicxml.org/dtds/opus.dtd">
```

→

```
<!-- Component DTD modules -->
```

```
<!--
```

The common DTD module contains the entities and elements that are shared among multiple DTD modules.

→

```
<!ENTITY % common PUBLIC "-//Recordare//ELEMENTS
MusicXML 3.1 Common//EN"
```

```
"common.mod">
```

```
%common;
```

```
<!--
```

The link DTD module contains XLink attributes.

→

```
<!ENTITY % link PUBLIC
```

```
"-//Recordare//ELEMENTS MusicXML 3.1 Link//EN"
```

```
"link.mod">
```

```
%link;
```

```
<!-- Elements -->
```

```
<!--
```

Opus is the document element. The document-attributes entity includes the version attribute and is defined in the common.mod file.

→

```
<!ELEMENT opus (title?, (opus | opus-link | score)*)>
```

```
<!ATTLIST opus
```

```
%document-attributes;
```

```
>
```

```
<!--
```

The score elements provide the links to the individual movements. The new-page attribute, added in Version 2.0, is used to indicate if the first page of the score is different than the last page of the previous score. If new-page is "yes", then a different page is used; if "no", then the same page is used. The default value is implementation-dependent.

```

->
<!ELEMENT score EMPTY>
<!ATTLIST score
%link-attributes;
new-page %yes-no; #IMPLIED
>
<!--

```

An opus-link provides a link to another opus document, allowing for multiple levels of opus collections via linking as well as nesting.

```

->
<!ELEMENT opus-link EMPTY>
<!ATTLIST opus-link %link-attributes;>
<!--

```

Future versions may include metadata elements. In this version, we just include the title of the opus.

```

->
<!ELEMENT title (#PCDATA)>

```

Container

```

<!--
MusicXML Container DTD
Version 3.1

```

Copyright © 2004-2017 the Contributors to the MusicXML Specification, published by the W3C Music Notation Community Group under the W3C Community Final Specification Agreement (FSA):

<https://www.w3.org/community/about/agreements/final/>

A human-readable summary is available:

<https://www.w3.org/community/about/agreements/fsa-deed/>

```

->
<!--

```

Starting with Version 2.0, the MusicXML format includes a standard zip compressed version. These zip files can contain multiple MusicXML files as well as other media files for images and sound. The container DTD describes the contents of the META-INF/container.xml file. The container describes the starting point for the MusicXML version of the file, as well as alternate renditions such as PDF and audio versions of the musical score.

The MusicXML zip file format is compatible with the zip format used by the java.util.zip package and Java JAR files. It is based on the Info-ZIP format described at: <http://www.info-zip.org/doc/appnote-19970311-iz.zip>

The JAR file format is specified at: <http://docs.oracle.com/javase/8/docs/technotes/guides/jar/jar.html> Note that, compatible with JAR files, file names should be encoded in UTF-8 format.

Files with the zip container are compressed the DEFLATE algorithm. The DEFLATE Compressed Data Format (RFC 1951) is specified at:

<http://www.ietf.org/rfc/rfc1951.txt>

The recommended media type for a compressed MusicXML file is:
application/vnd.recordare.musicxml

The recommended media type for an uncompressed MusicXML file is:
application/vnd.recordare.musicxml+xml

The first file in the zip container should be a file named mimetype. The contents of this file should be the MIME media type string

application/vnd.recordare.musicxml

encoded in US-ASCII. The mimetype file must not be compressed or encrypted, and there must not be an extra field in its zip header. The contents of the mimetype file must not contain any leading padding or white space and must not begin with a byte order mark. Older versions of MusicXML did not specify this mimetype file, so applications may see containers without a mimetype file.

The recommended file extension for compressed MusicXML files is .mxl. The recommended file extension for uncompressed MusicXML files is .musicxml. Older versions of MusicXML use

.xml as the extension for uncompressed MusicXML files, so it is recommended that applications be prepared to read files with the .xml extension as well.

It is recommended that applications that run on macOS and iOS use the following Uniform Type Identifier for MusicXML files:

```

<array>
  <dict>
    <key>UTTypeIdentifier</key>
    <string>com.recordare.musicxml.uncompressed</string>
    <key>UTTypeReferenceURL</key>
    <string>http://www.musicxml.org/</string>
    <key>UTTypeDescription</key>
    <string>MusicXML File</string>
    <key>UTTypeTagSpecification</key>
    <dict>
      <key>public.filename-extension</key>
      <array>
        <string>musicxml</string>
      </array>
      <key>public.mime-type</key>
      <array>
        <string>application/vnd.recordare.musicxml+xml</string>
      </array>
    </dict>
    <key>UTTypeConformsTo</key>
    <array>
      <string>public.xml</string>
    </array>
  </dict>
</dict>
<dict>
  <key>UTTypeIdentifier</key>
  <string>com.recordare.musicxml</string>
  <key>UTTypeReferenceURL</key>
  <string>http://www.musicxml.org/</string>
  <key>UTTypeDescription</key>
  <string>MusicXML File</string>
  <key>UTTypeTagSpecification</key>

```

```

<dict>
<key>public.filename-extension</key>
<array>
<string>mxl</string>
</array>
<key>public.mime-type</key>
<array>
<string>application/vnd.recordare.musicxml</string>
</array>
</dict>
<key>UTTypeConformsTo</key>
<array>
<string>public.zip-archive</string>
</array>
</dict>
</array>

```

Suggested use:

```

<!DOCTYPE container PUBLIC "-//Recordare//DTD
MusicXML 3.1 Container//EN"
"http://www.musicxml.org/dtds/container.dtd">

```

→

<!-- Elements -->

<!--

Container is the document element.

→

<!ELEMENT container (rootfiles)>

<!--

Rootfiles include the starting points for the different representations of a MusicXML score. The MusicXML root must be described in the first rootfile element. Additional rootfile elements can describe alternate versions such as PDF and audio files.

→

<!ELEMENT rootfiles (rootfile+)>

<!--

The rootfile element describes each top-level file in the MusicXML container. The full-path attribute provides the path relative to the root folder of the zip file. The media-type identifies

the type of different top-level root files. It is an error to have a non-MusicXML media-type value in the first rootfile in a rootfiles element. If no media-type value is present, a MusicXML file is assumed. A MusicXML file used as a rootfile may have score-partwise, score-timewise, or opus as its document element.

→

```
<!ELEMENT rootfile EMPTY>
```

```
<!ATTLIST rootfile
```

```
full-path CDATA #REQUIRED
```

```
media-type CDATA #IMPLIED
```

```
>
```