

Camila Barbosa Gouveia

**LoboSig: Sistema de gerenciamento de dados  
do Conservatório Estadual de Música Lobo de  
Mesquita**

Diamantina

2018



Camila Barbosa Gouveia

## **LoboSig: Sistema de gerenciamento de dados do Conservatório Estadual de Música Lobo de Mesquita**

Monografia submetida ao Curso de Graduação em Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM, como pré-requisito para obtenção do grau de Bacharel.

Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM

Faculdade de Ciências Exatas

Departamento de Computação

Curso de Graduação em Sistemas de Informação

Orientador: Alessandro Vivas Andrade

Diamantina

2018

---

Camila Barbosa Gouveia

LoboSig: Sistema de gerenciamento de dados do Conservatório Estadual de Música Lobo de Mesquita/ Camila Barbosa Gouveia. – Diamantina, 2018-  
181 p. : il. (algumas color.) ; 30 cm.

Orientador: Alessandro Vivas Andrade

Trabalho de Conclusão de Curso – Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM  
Faculdade de Ciências Exatas  
Departamento de Computação  
Curso de Graduação em Sistemas de Informação, 2018.

1. Gestão de Dados. 2. Sistema Web. I. Alessandro Vivas Andrade. II. Universidade Federal dos Vales do Jequitinhonha e Mucuri. III. Faculdade de Ciências Exatas - Sistemas de Informação. IV. Sistema de gerenciamento de dados do Conservatório Estadual de Música Lobo de Mesquita

CDU \*\*\*.\*\*\*.\*\*\*

---



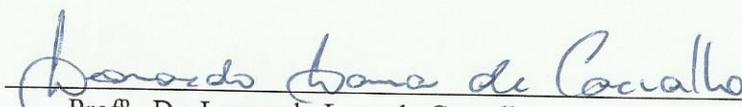
MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI  
DIAMANTINA – MINAS GERAIS

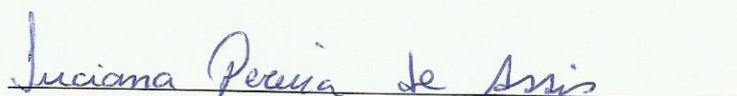
www.ufvjm.edu.br

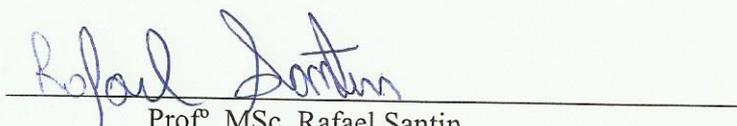


Monografia de projeto final de graduação sob o título “LoboSig: Sistema de gerenciamento de dados do Conservatório Estadual de Música Lobo de Mesquita”, defendida por Camila Barbosa Gouveia e aprovada em 27 de fevereiro de 2018, em Diamantina, Minas Gerais.

Banca Examinadora:

  
Prof.<sup>o</sup> Dr. Leonardo Lana de Carvalho

  
Prof.<sup>a</sup> Dra. Luciana Pereira de Assis

  
Prof.<sup>o</sup> MSc. Rafael Santin

*Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida, autor do meu destino, meu guia, socorro presente nos momentos mais difíceis; a minha mãe, guerreira, Eliana e a minha família.*

# Agradecimentos

A Universidade Federal dos Vales do Jequitinhonha e Mucuri pela oportunidade de fazer o curso, ao corpo docente do departamento, direção e administração.

Ao Professor Dr. Alessandro Vivas, pelo paciente trabalho, pela orientação, apoio e confiança.

Agradeço a minha mãe Eliana, heroína, que passou por momentos difíceis, mas mesmo assim me deu apoio, incentivo nas horas difíceis, de desânimo e cansaço.

Meus agradecimentos aos amigos, companheiros de trabalhos e irmãos na amizade, que fizeram parte da minha formação e que vão continuar presentes em minha vida, com certeza. Agradeço em especial ao amigo e colega Álvaro Luiz Marinho, que foi pessoa de suma importância para a realização e finalização desse trabalho.

Enfim, a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.



*“Ouça conselhos e aceite instruções,  
e acabará sendo sábio”.*  
*(Bíblia Sagrada, Provérbios 19, 20)*



# Resumo

O Conservatório Estadual de Música Lobo de Mesquita, na cidade de Diamantina-MG, é uma escola de música na qual o trabalho de gerenciamento de dados dos alunos é feito manualmente, ou com o auxílio de ferramentas computacionais inadequadas. Este trabalho apresenta as etapas e o resultado do desenvolvimento de um sistema realizado juntamente com a escola de música. O objetivo foi elaborar um sistema de gestão dos dados pessoais e escolares dos alunos que eliminasse processos manuais de trabalho, redundâncias e erros nas etapas dos serviços, bem como que diminuísse o tempo da execução de tarefas por parte dos funcionários. Utilizou-se das linguagens HTML e o PHP, adotando o padrão de desenvolvimento MVC. Utilizou-se também o Framework CakePHP e outras ferramentas como Wamp Server, GitHub e Sublime Text. Através da implantação do Sistema poderão ser observadas melhorias, tais como a confiabilidade dos dados, agilidade e padrão nos processos de trabalho, bem como a diminuição do ônus causado pelo retrabalho.

**Palavras-chaves:** Sistema de Gerenciamento. Dados. Conservatório.



# Abstract

The Lobo de Mesquita State Conservatory of Music, in the city of Diamantina-MG, is a music school in which the work of data management of students' is done manually, or with the aid of inadequate computational tools. This work presents the stages and the result of the development of a system performed together with the music school. The objective was to develop a system for the management of students' personal and school data, eliminating manual labor processes, redundancies and errors in the service stages, as well as reducing the time the staff performed. We used HTML and PHP, adopting the MVC development standard. Also used was the CakePHP Framework and other tools like Wamp Server, GitHub and Sublime Text. Improvements were observed through the implementation of the System, such as data reliability, agility and standard in work processes, as well as reduction of the burden caused by rework.

**Key-words:** Development System . Data. Conservatory.



# Lista de ilustrações

Figura 1 – Diagrama de funcionamento do MVC . . . . .	48
Figura 2 – Diagrama de Caso de Uso do Sistema LoboSig . . . . .	78
Figura 3 – Diagrama de Classes do Sistema LoboSig . . . . .	79
Figura 4 – Relacionamentos das Tabelas do Banco de Dados do Sistema LoboSig . . . . .	81
Figura 5 – Tela de login do Sistema LoboSig. . . . .	84
Figura 6 – Tela Painel do Administrador. . . . .	85
Figura 7 – Tela de Visualização de Alunos já cadastrados. . . . .	86
Figura 8 – Tela de cadastro de novo Aluno 1. . . . .	87
Figura 9 – Tela de cadastro de novo Aluno 2. . . . .	87
Figura 10 – Tela de cadastro de novo Aluno 3. . . . .	87
Figura 11 – Tela de edição de dados de Aluno. . . . .	89
Figura 12 – Tela de impressão do Comprovante de Matrícula. . . . .	91
Figura 13 – Tela de Visualização de Funcionários já cadastrados. . . . .	92
Figura 14 – Tela de cadastro de novo Funcionário 1. . . . .	93
Figura 15 – Tela de cadastro de novo Funcionário 2. . . . .	93
Figura 16 – Tela de cadastro de novo Funcionário 3. . . . .	93
Figura 17 – Tela de edição de dados dos Funcionários. . . . .	95
Figura 18 – Tela de visualização de Disciplinas já cadastradas. . . . .	96
Figura 19 – Tela de cadastro de novas Disciplinas. . . . .	97
Figura 20 – Tela de edição de Disciplinas. . . . .	98
Figura 21 – Tela de visualização de Vínculos. . . . .	99
Figura 22 – Tela vincular Professor/Disciplina. . . . .	100
Figura 23 – Tela de visualizar Turmas já cadastradas. . . . .	101
Figura 24 – Tela de cadastro de nova Turma. . . . .	102
Figura 25 – Tela de visualização de alunos já matriculados. . . . .	103
Figura 26 – Tela de realização de Matrícula. . . . .	104
Figura 27 – Tela de impressão do Comprovante de Lista de Presença. . . . .	105
Figura 28 – Tela Painel do Aluno. . . . .	106
Figura 29 – Tela de visualização de Horários das Aulas. . . . .	107
Figura 30 – Tela de visualização de Notas e Frequências. . . . .	108
Figura 31 – Tela Painel do Professor. . . . .	109
Figura 32 – Tela de seleção da Turma. . . . .	110
Figura 33 – Tela de visualização de Horário das Aulas. . . . .	111
Figura 34 – Tela de lançar de Notas e Frequências. . . . .	112



# Lista de tabelas

Tabela 1 – Plano curricular do curso de Educação Musical - Ciclo Inicial. . . . .	36
Tabela 2 – Plano curricular do curso de Educação Musical - Ciclo Intermediário. . .	36
Tabela 3 – Plano curricular do curso de Educação Musical - Ciclo Complementar. . .	36
Tabela 4 – Plano curricular do curso de Formação Profissional. . . . .	36
Tabela 5 – Cadastro de Alunos - Painel Administrador. . . . .	66
Tabela 6 – Edição de dados de Alunos - Painel Administrador. . . . .	67
Tabela 7 – Exclusão de dados de Alunos - Painel Administrador. . . . .	67
Tabela 8 – Impressão de Comprovante de Matrícula - Painel Administrador. . . . .	68
Tabela 9 – Cadastro de Funcionários - Painel Administrador. . . . .	68
Tabela 10 – Alteração de dados de Funcionários - Painel Administrador. . . . .	69
Tabela 11 – Exclusão de dados de Funcionários - Painel Administrador. . . . .	69
Tabela 12 – Cadastro de Disciplinas - Painel Administrador. . . . .	70
Tabela 13 – Alteração de dados de Disciplinas - Painel Administrador. . . . .	70
Tabela 14 – Exclusão de dados de Disciplina - Painel Administrador. . . . .	71
Tabela 15 – Vincular Professor a Disciplina - Painel Administrador. . . . .	71
Tabela 16 – Exclusão de Vínculo - Painel Administrador. . . . .	71
Tabela 17 – Cadastro de Turmas - Painel Administrador. . . . .	72
Tabela 18 – Exclusão de Turma - Painel Administrador. . . . .	72
Tabela 19 – Exclusão de Vínculo - Painel Administrador. . . . .	72
Tabela 20 – Matrícula de Aluno - Painel Administrador. . . . .	73
Tabela 21 – Desmatricular Aluno - Painel Administrador. . . . .	73
Tabela 22 – Imprimir Listagem de alunos - Painel Administrador. . . . .	73
Tabela 23 – Imprimir de Lista de presença - Painel Administrador. . . . .	74
Tabela 24 – Visualização de horários - Painel Aluno. . . . .	74
Tabela 25 – Visualização de Notas e Frequências - Painel Aluno. . . . .	74
Tabela 26 – Visualização de horários - Painel Professor. . . . .	75
Tabela 27 – Lançamento de Notas e Frequências - Painel Professor. . . . .	75
Tabela 28 – Seleção de Turmas - Painel Professor. . . . .	75
Tabela 29 – Descrição dos campos – Tela de Login no Sistema. . . . .	84
Tabela 30 – Descrição das ações – Tela de Login no Sistema. . . . .	84
Tabela 31 – Descrição das ações – Tela Painel do Administrador. . . . .	85
Tabela 32 – Descrição dos campos – Tela Visualização de Alunos já cadastrados. . .	86
Tabela 33 – Descrição das ações – Tela Visualização de Alunos já cadastrados. . . .	86
Tabela 34 – Descrição dos campos – Tela de cadastro de novo Aluno - Dados Pessoais. .	88
Tabela 35 – Descrição das ações – Tela de cadastro de novo Aluno - Dados Pessoais. .	89

Tabela 36 – Descrição dos campos – Tela de edição de dados de Aluno - Dados Pessoais. . . . .	90
Tabela 37 – Descrição das ações – Tela de edição de dados de Aluno - Dados Pessoais.	91
Tabela 38 – Descrição das ações – Tela de impressão do Comprovante de Matrícula.	91
Tabela 39 – Descrição dos campos – Tela de Visualização de Funcionários já cadastrados. . . . .	92
Tabela 40 – Descrição das ações – Tela de Visualização de Funcionários já cadastrados.	92
Tabela 41 – Descrição dos campos – Tela de cadastro de novo Funcionário - Dados Pessoais. . . . .	94
Tabela 42 – Descrição das ações – Tela de cadastro de novo Funcionário - Dados Pessoais. . . . .	94
Tabela 43 – Descrição dos campos – Tela de edição de dados dos Funcionários - Dados Pessoais. . . . .	96
Tabela 44 – Descrição das ações – Tela de edição de dados dos Funcionários - Dados Pessoais. . . . .	96
Tabela 45 – Descrição dos campos – Tela de visualização de Disciplinas já cadastradas.	97
Tabela 46 – Descrição das ações – Tela de visualização de Disciplinas já cadastradas.	97
Tabela 47 – Descrição dos campos – Tela de cadastro de novas Disciplinas. . . . .	98
Tabela 48 – Descrição das ações – Tela de cadastro de novas Disciplinas. . . . .	98
Tabela 49 – Descrição dos campos – Tela de edição de Disciplinas. . . . .	99
Tabela 50 – Descrição das ações – Tela de edição de Disciplinas. . . . .	99
Tabela 51 – Descrição dos campos – Tela de visualização de Vínculos. . . . .	100
Tabela 52 – Descrição das ações – Tela de visualização de Vínculos. . . . .	100
Tabela 53 – Descrição dos campos – Tela vincular Professor/Disciplina. . . . .	101
Tabela 54 – Descrição das ações – Tela vincular Professor/Disciplina. . . . .	101
Tabela 55 – Descrição dos campos – Tela de visualizar Turmas já cadastradas. . . . .	101
Tabela 56 – Descrição das ações – Tela de visualizar Turmas já cadastradas. . . . .	102
Tabela 57 – Descrição dos campos – Tela de cadastro de nova Turma. . . . .	103
Tabela 58 – Descrição das ações – Tela de cadastro de nova Turma. . . . .	103
Tabela 59 – Descrição dos campos – Tela de visualização de alunos já matriculados.	104
Tabela 60 – Descrição das ações – Tela de visualização de alunos já matriculados. . . . .	104
Tabela 61 – Descrição dos campos – Tela de realização de Matrícula. . . . .	105
Tabela 62 – Descrição das ações – Tela de realização de Matrícula. . . . .	105
Tabela 63 – Descrição dos campos – Tela de impressão de Lista de Presença. . . . .	105
Tabela 64 – Descrição das ações – Tela de impressão de Lista de Presença. . . . .	106
Tabela 65 – Descrição das ações – Tela Painel do Aluno. . . . .	106
Tabela 66 – Descrição dos campos – Tela de visualização de Horários das Aulas. . . . .	107
Tabela 67 – Descrição das ações – Tela de visualização de Horários das Aulas. . . . .	107
Tabela 68 – Descrição dos campos – Tela de visualização de Notas e Frequências. . . . .	108

Tabela 69 – Descrição das ações – Tela de visualização de Notas e Frequências. . . . .	109
Tabela 70 – Descrição das ações – Tela Painel do Professor. . . . .	109
Tabela 71 – Descrição dos campos – Tela de seleção da Turma. . . . .	110
Tabela 72 – Descrição das ações – Tela de seleção da Turma. . . . .	110
Tabela 73 – Descrição dos campos – Tela de visualização de Horário das Aulas. . . . .	111
Tabela 74 – Descrição das ações – Tela de visualização de Horário das Aulas. . . . .	111
Tabela 75 – Descrição dos campos – Tela de lançar de Notas e Frequências. . . . .	112
Tabela 76 – Descrição das ações – Tela de lançar de Notas e Frequências. . . . .	113



# Lista de abreviaturas e siglas

SI	Sistemas de Informação
UFVJM	Univerisidade Federal dos Vales do Jequitinhonha e Mucuri
IDE	Integrated Development Environment
MVC	Model-View-Controller
CGI	Computer Graphic Imagery
POO	Programação Orientada a Objetos
CRUD	Creat, Retrieve, Update e Delete
URL	Uniform, Resource, Locator
HTML	Hypertext Markup Language
RoR	Ruby On Rails
XML	Extensible Markup Language
AJAX	Asynchronous Javascript and XML
CSS	Cascading Style Sheets
ASP	Active Server Pages
HTTP	HyperText Transfer Protocol
GNU	Gnu's Not Unix
API	Application Programming Interface
OS/2	Operating System/2
POSIX	Portable Operating System Interface
RAM	Random Access Memory
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structure Query Language
TCP	Transmission Control Protocol

CSV	Comma-separated values
PDF	Portable Document Format
QBE	Query-by-example
ISO	International Organization for Standardization
LoboSig	Sistema de Gerenciamento de Dados do Conservatório Estadual de Música Lobo de Mesquita
CEM	Conservatório Estadual de Música
RF	Requisito Funcional
RNF	Requisito Não Funcional
UML	Unified Modeling Language

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>27</b>
<b>1.1</b>	<b>Justificativa</b>	<b>28</b>
<b>1.2</b>	<b>Objetivos</b>	<b>28</b>
1.2.1	Objetivo geral	28
1.2.2	Objetivos específicos	28
<b>1.3</b>	<b>Estrutura do Trabalho</b>	<b>30</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>31</b>
<b>3</b>	<b>DESCRIÇÃO DO PROBLEMA</b>	<b>35</b>
<b>4</b>	<b>FERRAMENTAS COMPUTACIONAIS DE SUPORTE</b>	<b>39</b>
<b>4.1</b>	<b>PHP</b>	<b>39</b>
4.1.1	Como funciona o PHP	40
<b>4.2</b>	<b>Programação Orientada a Objetos (POO)</b>	<b>41</b>
4.2.1	Diferença entre objetos de classes	42
4.2.1.1	Classes	42
4.2.1.2	Objetos	43
4.2.2	Encapsulamento	44
4.2.3	Herança	45
4.2.4	Sobrecarga	46
<b>4.3</b>	<b>MVC (Model-View-Controller)</b>	<b>47</b>
4.3.1	MVC e PHP	48
4.3.1.1	Exemplo prático de MVC	49
<b>4.4</b>	<b>Framework CakePHP</b>	<b>50</b>
4.4.1	Razão para usar um framework	50
4.4.2	Razões para usar CakePHP	51
<b>4.5</b>	<b>HTML</b>	<b>52</b>
4.5.1	Tags no HTML	52
4.5.2	Desenvolvimento em Camadas	53
<b>4.6</b>	<b>WAMPserver</b>	<b>54</b>
4.6.1	Apache	54
<b>4.7</b>	<b>Banco de Dados</b>	<b>55</b>
4.7.1	MySQL	56
4.7.2	phpMyAdmin	57
<b>4.8</b>	<b>Sublime Text</b>	<b>58</b>

<b>4.9</b>	<b>GitHub</b> . . . . .	<b>59</b>
<b>4.10</b>	<b>Windows 10</b> . . . . .	<b>60</b>
4.10.1	Prós . . . . .	61
4.10.2	Contras . . . . .	61
<b>5</b>	<b>ANÁLISE DE REQUISITOS</b> . . . . .	<b>63</b>
<b>5.1</b>	<b>Propósito do Sistema</b> . . . . .	<b>63</b>
5.1.1	Funcionalidades do sistema . . . . .	63
<b>5.2</b>	<b>Perspectiva do Sistema</b> . . . . .	<b>64</b>
5.2.1	Restrições Gerais . . . . .	64
<b>5.3</b>	<b>Requisitos de Software: Definições e Abreviações</b> . . . . .	<b>64</b>
5.3.1	Estrutura de um Requisito . . . . .	65
<b>5.4</b>	<b>Requitos Funcionais</b> . . . . .	<b>66</b>
5.4.1	Requisitos Funcionais - Painel Administrador . . . . .	66
5.4.1.1	Cadastro de Alunos . . . . .	66
5.4.1.2	Alteração do Cadastro de Alunos . . . . .	67
5.4.1.3	Exclusão do Cadastro de Aluno . . . . .	67
5.4.1.4	Impressão de Matrícula do Aluno . . . . .	68
5.4.1.5	Cadastro de Funcionário . . . . .	68
5.4.1.6	Alteração de Cadastro de Funcionário . . . . .	69
5.4.1.7	Exclusão do Cadastro de Funcionário . . . . .	69
5.4.1.8	Cadastro de Disciplina . . . . .	70
5.4.1.9	Alteração de Cadastro de Disciplina . . . . .	70
5.4.1.10	Exclusão do Cadastro de Disciplina . . . . .	71
5.4.1.11	Vincular Professor a uma Disciplina . . . . .	71
5.4.1.12	Exclusão do Vínculo Professor/Disciplina . . . . .	71
5.4.1.13	Cadastro de nova Turma . . . . .	72
5.4.1.14	Exclusão de Turma . . . . .	72
5.4.1.15	Consulta a Turmas . . . . .	72
5.4.1.16	Matricular Aluno . . . . .	73
5.4.1.17	Desmatricular Aluno . . . . .	73
5.4.1.18	Impressão Listagem de Alunos . . . . .	73
5.4.1.19	Impressão Lista de Presença . . . . .	74
5.4.2	Requisitos Funcionais - Painel Aluno . . . . .	74
5.4.2.1	Visualização de Horários . . . . .	74
5.4.2.2	Visualização de Notas e Frequência . . . . .	74
5.4.3	Requisitos Funcionais - Painel Professor . . . . .	75
5.4.3.1	Visualização de Horários . . . . .	75
5.4.3.2	Lançamento de Notas e Frequência no Sistema . . . . .	75
5.4.3.3	Seleção de Turma para lançamento de Notas e Frequência . . . . .	75

<b>5.5</b>	<b>Requisitos Não Funcionais</b>	<b>76</b>
5.5.1	Requisitos Organizacionais	76
5.5.1.1	RNF01: Software	76
5.5.1.2	RNF02: Linguagem de Programação	76
5.5.1.3	RNF03:Hardware	76
5.5.2	Outros Requisitos	76
<b>5.6</b>	<b>Atributos</b>	<b>76</b>
<b>5.7</b>	<b>Diagrama de Caso de Uso</b>	<b>78</b>
<b>5.8</b>	<b>Diagrama de Classes</b>	<b>79</b>
<b>5.9</b>	<b>Modelo Entidade Relacionamento</b>	<b>80</b>
<b>6</b>	<b>DESENVOLVIMENTO DO PROJETO</b>	<b>83</b>
6.1	Tela de Login no Sistema	84
6.2	Tela Painel do Administrador	85
6.3	Tela de visualização de Alunos já cadastrados	86
6.4	Tela de cadastro de novo Aluno	87
6.5	Tela de edição de dados dos Alunos	89
6.6	Tela de impressão de Comprovante de Matrícula	91
6.7	Tela de visualização de Funcionários já cadastrados	92
6.8	Tela de cadastro de novo Funcionário	93
6.9	Tela de edição de dados dos Funcionários	95
6.10	Tela de visualização de Disciplinas já cadastradas	96
6.11	Tela de cadastro de novas Disciplinas	97
6.12	Tela de edição de Disciplinas	98
6.13	Tela de visualização de Vínculos Professor/Disciplina	99
6.14	Tela vincular Professor/Disciplina	100
6.15	Tela de visualização de Turmas já cadastradas	101
6.16	Tela de cadastro de nova Turma	102
6.17	Tela de visualização de alunos já matriculados	103
6.18	Tela de realização de Matrícula	104
6.19	Tela de impressão de Lista de Presença	105
6.20	Tela Painel do Aluno	106
6.21	Tela de visualização de Horários das Aulas dos Alunos	107
6.22	Tela de visualização de Notas e Frequências do Aluno	108
6.23	Tela Painel do Professor	109
6.24	Tela de seleção da Turma	110
6.25	Tela de visualização de Horário das Aulas do Professor	111
6.26	Tela de lançar Notas e Frequências	112
<b>7</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>115</b>

<b>Conclusão</b> . . . . .	<b>117</b>
----------------------------	------------

<b>REFERÊNCIAS</b> . . . . .	<b>121</b>
------------------------------	------------

<b>ANEXOS</b>	<b>125</b>
---------------	------------

<b>ANEXO A – BANCO DE DADOS</b> . . . . .	<b>127</b>
---	------------

<b>ANEXO B – VIEW</b> . . . . .	<b>131</b>
---------------------------------	------------

<b>B.1 View Alunos</b> . . . . .	<b>131</b>
----------------------------------	------------

B.1.1 Horário . . . . .	131
-------------------------	-----

B.1.2 Index . . . . .	131
-----------------------	-----

B.1.3 Turma . . . . .	132
-----------------------	-----

<b>B.2 View Autenticações</b> . . . . .	<b>133</b>
---	------------

B.2.1 Login . . . . .	133
-----------------------	-----

<b>B.3 View Coordenação</b> . . . . .	<b>133</b>
---------------------------------------	------------

B.3.1 Aluno . . . . .	133
-----------------------	-----

B.3.2 Comprovante de Matrícula . . . . .	134
--	-----

B.3.3 Disciplina . . . . .	136
----------------------------	-----

B.3.4 Editar Aluno . . . . .	136
------------------------------	-----

B.3.5 Editar Disciplina . . . . .	139
-----------------------------------	-----

B.3.6 Editar Funcionário . . . . .	140
------------------------------------	-----

B.3.7 Funcionário . . . . .	142
-----------------------------	-----

B.3.8 Index . . . . .	143
-----------------------	-----

B.3.9 Lista de presença . . . . .	144
-----------------------------------	-----

B.3.10 Matrícula . . . . .	146
----------------------------	-----

B.3.11 Nova Disciplina . . . . .	147
----------------------------------	-----

B.3.12 Nova Turma . . . . .	149
-----------------------------	-----

B.3.13 Novo Aluno . . . . .	150
-----------------------------	-----

B.3.14 Novo Funcionário . . . . .	152
-----------------------------------	-----

B.3.15 Novo Vínculo . . . . .	155
-------------------------------	-----

B.3.16 Turma . . . . .	155
------------------------	-----

B.3.17 Vínculo . . . . .	156
--------------------------	-----

B.3.18 Visualizar Matrícula . . . . .	157
---------------------------------------	-----

<b>B.4 View Professores</b> . . . . .	<b>159</b>
---------------------------------------	------------

B.4.1 Horário . . . . .	159
-------------------------	-----

B.4.2 Index . . . . .	159
-----------------------	-----

B.4.3 Notas e Frequência . . . . .	160
------------------------------------	-----

B.4.4 Selecionar Turma . . . . .	162
----------------------------------	-----

	<b>ANEXO C – CONTROLLER</b> . . . . .	<b>163</b>
<b>C.1</b>	<b>Controller Alunos</b> . . . . .	<b>163</b>
<b>C.2</b>	<b>Controller App</b> . . . . .	<b>164</b>
<b>C.3</b>	<b>Controller Autenticações</b> . . . . .	<b>165</b>
<b>C.4</b>	<b>Controller Coordenação</b> . . . . .	<b>166</b>
<b>C.5</b>	<b>Controller Paginação</b> . . . . .	<b>175</b>
<b>C.6</b>	<b>Controller Professor</b> . . . . .	<b>177</b>
	<b>ANEXO D – MODEL</b> . . . . .	<b>181</b>



# 1 Introdução

Com a diminuição de custos computacionais, por causa dos grandes avanços de hardware e software, e o crescimento da necessidade de automatização e modernização de processos de gestão de dados escolares, faz-se necessário para as instituições educacionais sistemas que gerenciem de maneira mais confiável as informações. Este trabalho trata sobre o desenvolvimento do Sistema de Gestão de dados escolares dos alunos do Conservatório Estadual de Música Lobo de Mesquita.

O Conservatório Estadual de Música Lobo de Mesquita é uma escola que destoa das escolas de ensino regular, não somente pelas disciplinas musicais e cursos ofertados, mas principalmente pelo modelo de ensino. Tal modelo é aplicado em ciclos e fases no ensino fundamental, onde o aluno pode obter um certificado de conclusão do Curso de Educação Musical. Ao finalizar o ensino fundamental o estudante pode escolher encerrar os estudos ou continuar por meio do ensino médio ou Curso Técnico em Educação Musical, onde há a possibilidade de se formar Tecnólogo em Educação Musical. Toda essa complexidade do negócio está associada ao acúmulo, ao longo de anos, de informações e processos de serviços atrasados e redundantes. Sendo assim, faz-se necessário um tratamento especial para esse cenário caótico com relação a administração dos dados.

O Sistema de gestão eliminará os processos manuais e irá diminuir o tempo de execução das tarefas por parte dos funcionários, pois será capaz de dividir tais tarefas com os professores que, com a implantação do software, poderão ajudar no lançamento de informações no Banco de Dados. Haverá a padronização dos processos e melhoria do fluxo da informação, o que aumentará a qualidade de trabalho, a produtividade e uma considerável redução no tempo de execução total e o tempo de resposta à supervisores do serviço.

A aplicação será dividida em três módulos: Aluno, Professor e Administrador. Foi desenvolvido em ambiente web, o que facilitará a administração escolar no que diz respeito à execução, acompanhamento e controle das atividades finais. Permitirá melhor divisão de trabalho e a atualização em tempo real do Banco de Dados, além de melhorar o atendimento à comunidade escolar e o acompanhamento das informações por parte dos alunos/pais de alunos, que poderão acessar os dados no Sistema através do módulo Aluno.

Com o novo Sistema, serão integradas todas as informações do Conservatório, o que evitará distorções no planejamento escolar e permitirá a identificação da demanda real de vagas, através do fácil controle de alunos que evadem ou não. Haverá uma padronização dos documentos escolares emitidos e a eficiência na gestão dos dados dos alunos e professores por parte da direção e supervisão.

O software será capaz de associar cada aluno a uma turma, às disciplinas que ele irá cursar durante o ano letivo, bem como aos respectivos professores de cada disciplina. Foi implementada a função de login para o Sistema, com verificação na autenticação de cada tipo de usuário, Aluno, Professor ou Administrador. No módulo Aluno os alunos terão a possibilidade de acessar o painel e verificar em quais disciplinas está matriculado, bem como visualizar as notas e frequências. Foi implementado também o módulo Professor, onde professores terão acesso ao sistema para lançar notas (conceitos) e frequências dos alunos, o que contribuirá com o trabalho dos funcionários da secretaria da escola, que desempenham tal tarefa atualmente. No módulo Administrador o funcionário que possui a permissão de acesso poderá visualizar os dados pessoais de alunos e funcionários, bem como cadastrar, editar ou remover dados do Banco de Dados.

A implantação de um Sistema de Gestão propiciará diversas melhorias perceptíveis ao Conservatório, entre elas está a maior confiabilidade dos dados, que serão atualizados em tempo real e sem perdas. Ao agilizar e padronizar todos os processos, a escola terá os dados necessários para planejar, diminuir custos causados pelo retrabalho e melhorar a comunicação entre funcionários. O tempo e o processo de tomada de decisão também ganharão nova dinâmica, devido à confiabilidade e rapidez na obtenção de dados.

## 1.1 Justificativa

O presente trabalho tem a proposta de desenvolver um Sistema de Gestão de dados escolares, considerando-se a necessidade da escola de música Conservatório Estadual de Música Lobo de Mesquita por um Sistema de Informação que realize o gerenciamento de tais dados e, ainda, que torne mais fáceis, rápidas e eficazes as atividades dos funcionários.

## 1.2 Objetivos

Nesta seção são apresentados os objetivos do presente trabalho.

### 1.2.1 Objetivo geral

Promover a melhoria dos processos de gestão e implantação de sistema para gerenciamento de dados escolares do Conservatório Estadual de Música Lobo de Mesquita.

### 1.2.2 Objetivos específicos

Um tema relevante para as organizações é o ajuste dos processos de serviço, que apoiam à busca incessante pela eficácia nas operações e por ótimas estratégias para a empresa. Na era do conhecimento e da informação faz-se necessário estar capacitado a

trabalhar com a tecnologia, sendo que tais processos devem ser apoiados por Sistemas de Gestão altamente favoráveis, integrados, aderentes.

Em um âmbito geral, as organizações educacionais, pelo fato de possuírem um grande número de informações e dados escolares dos alunos, demandam cada vez mais de processos automatizados, rápidos e eficientes para gerenciar e trabalhar com tais dados com maior qualidade e eficácia, com redução de tempo e retrabalho.

Nesse contexto de anseio das organizações educacionais por uma melhor organização das informações e redução no tempo de realização do trabalho, está enquadrado o Conservatório Estadual de Música Lobo de Mesquita, escola direcionada ao ensino de disciplinas voltadas à música, situado na cidade de Diamantina.

O Conservatório é um lugar onde todo o trabalho de gerenciamento de dados é feito manualmente ou com auxílio de ferramentas computacionais inadequadas. Ademais, há na escola um problema visível de redundância e duplicação de dados gerado pela falta de comunicação entre funcionários e também por falta de conhecimento, por parte de alguns desses funcionários, das tecnologias de informação que são utilizadas na escola. Esse fato provoca uma constante perda de dados que já haviam sido informatizados e, como consequência, toda essa situação torna o retrabalho uma constante na instituição de ensino.

Atualmente, os profissionais não devem mais rejeitar o fato de que é cada vez mais necessária a convivência com o computador e as tecnologias de informação, que podem auxiliar os usuários e aumentar a qualidade na elaboração dos trabalhos. Por isso, faz-se necessário que os funcionários do Conservatório tenham conhecimento das ferramentas computacionais existentes para não se verem pressionados e inseguros quando for introduzido o Sistema de Gestão e as mudanças tecnológicas nos processos de serviço da escola.

O funcionário não precisa se envolver profundamente com a área de informática a ponto de ser um "expert" e dominá-la. Contudo, hoje é exigida certa competência nesta área e, sendo assim, será necessário desempenho e esforço por parte de cada um no momento da implantação do Sistema, a fim de que realmente os processos de trabalho e armazenamento de dados sejam melhorados na Instituição.

Além da falta de afinidade com as ferramentas computacionais por parte de alguns funcionários, há um visível acúmulo de trabalho relacionado ao gerenciamento de dados e geração de relatórios. Tal acúmulo recai sobre a supervisão da escola. Dentro desse trabalho realizado por esta supervisão, está o lançamento no sistema das notas (conceitos) e frequências bimestrais de cada aluno. Isso acontece através de um software utilizado pela escola, o Microsoft Works, da empresa Microsoft e a tarefa atualmente é realizada por uma única pessoa.

## 1.3 Estrutura do Trabalho

O presente trabalho é composto por seis capítulos organizados da seguinte maneira:

- Capítulo 1: Introduz o trabalho e faz uma breve descrição sobre o tema proposto;
- Capítulo 2: Realiza uma revisão de trabalhos já existentes sobre o tema abordado, como livros, artigos, monografias, teses, mídias eletrônicas e outros materiais cientificamente confiáveis, afim de verificar o estado do problema a ser pesquisado, sob o aspecto teórico e de outros estudos e pesquisas já realizados;
- Capítulo 3: Apresenta a problemática que as escolas vivem no tocante ao armazenamento de dados e o papel dos Sistemas de Informação nesse âmbito, trazendo os conceitos necessários para o entendimento deste trabalho;
- Capítulo 4: Será apresentada a metodologia, as linguagens e ferramentas utilizadas na criação da aplicação e o resultado da aplicação em funcionamento.
- Capítulo 5: Apresenta a documentação da análise dos requisitos, com uma descrição geral do Sistema.
- Capítulo 6: Demonstra detalhadamente a descrição das funcionalidades do programa, com inclusão das imagens de cada tela do Sistema.

## 2 Referencial Teórico

A inovação tecnológica é algo que tem tomado uma grande dimensão na contemporaneidade e isso reflete em um maior fluxo de informações. Assim, o acesso a essa abundância de informações e conhecimentos, disponibilizados pelos Sistemas de Informação (SI), tem dado suporte as práticas do gerenciamento administrativo de diversos ambientes não somente no âmbito comercial, mas também no educacional.

Os SI podem ser definidos como "conjunto de partes, componentes, que interagem entre si, de forma ordenada, a fim de atingir um objetivo comum"(STAIR; REYNOLDS, 2005; LAUDON; LAUDON, 2011). São utilizados na automação de processos e consistem na administração dos fluxos de informações geradas em um determinado ambiente e sobre um determinado contexto, provendo a organização das informações. McGarry (1999 apud OLIVEIRA; FREITAS; TONO, 2008, p.2) define informação como sendo o termo que designa o conteúdo daquilo que permutamos com o mundo exterior ao ajustar-nos a ele, e que faz com que nosso ajustamento seja nele percebido.

Um Sistema de Informação é caracterizado pelo o processamento, armazenamento, recuperação dos dados conforme os critérios e fluxos estipulados pelos processos existentes na organização, propiciando aperfeiçoamento de desempenho, eficiência e racionalização de etapas na qual entende-se como processos. Segundo Hammer e Champy (1994 apud QUEIROZ; MUCELINI, 2010, p.23), como sendo um grupo de atividades realizadas numa sequência lógica com objetivo de produzir um bem ou serviço que tem valor para um grupo específico de clientes.

Surgiram sites que possibilitam o acesso a informações institucionais por clientes e parceiros, com o crescimento e evolução da World Wide Web. Surgiram também grandes SI, que utilizam navegadores como meio para que os clientes efetivem transações com banco de dados, como consultas e cadastro de novos dados pela Web. MacGarry (1999 apud JUNIOR; VIDAL, 2006, p.235) elucidou as aplicações Web como representantes da quarta geração dos sistemas de processamento de dados empresariais, sendo posterior as gerações dos sistemas de processamento em lote, sistemas de compartilhamento de tempo e das aplicações cliente-servidor. Um enfoque deve ser dado à definição de MacGarry (1999 apud JUNIOR; VIDAL, 2006, p.235), em que se deve considerar que muitos Sistemas de Informação baseados na Web são modelos cliente-servidor.

Em termos um pouco mais generalizados, de acordo com Tanenbaum (2003, p.19), a questão é o compartilhamento de recursos, e o objetivo é deixar todos os programas, equipamentos e especialmente dados e informações ao alcance de todas as pessoas na rede, independentemente da localização física do recurso e do usuário. Um exemplo visível e

bastante disseminado, é um grupo de funcionários de um escritório que compartilham uma impressora comum. Nenhuma das pessoas necessita, realmente, de uma impressora particular, e uma impressora de grande capacidade ligada na rede é, na maioria das vezes, muito mais econômica, mais rápida e de mais fácil manutenção que um aglomerado de impressoras pessoais.

Entretanto, para uma organização, compartilhar recursos físicos como pendrives, gravadores de CDs, impressoras, scanners, pode ser menos relevante que compartilhar informações. Muitas empresas pequenas e toda empresa de médio e grande porte possuem suma dependência de informações digitalizadas. A maioria delas tem cadastro de clientes, estoques, contas a receber, extratos financeiros, dados sobre impostos e muitas outras informações online. Se todos os computadores de uma rede bancária tivessem uma falha, provavelmente a organização não duraria mais que cinco minutos. Em um equipamento industrial moderno, como uma linha de montagem controlada por computadores, uma pane não duraria nem isso. Até mesmo uma pequena agência de viagens ou uma firma jurídica são totalmente dependentes de redes de computadores que permitem aos funcionários o acesso a dados e documentos relevantes de forma instantânea e em qualquer lugar.

A arquitetura cliente/servidor transformou-se em um dos princípios fundamentais das redes de computadores, amplamente utilizado por muitas aplicações de negócios. Esse modelo descreve a relação de programas em uma aplicação, onde o elemento servidor fornece uma função ou serviço a um ou mais clientes, que iniciam os pedidos de serviço. Normalmente os clientes e servidores se comunicam por uma rede de computadores em máquinas distintas, entretanto tanto o cliente quanto o servidor podem encontra-se no mesmo computador.

Não obstante, a utilização de padrões baseados no modelo cliente/servidor tem se tornado cada vez mais custosa. Modelo esse que como o próprio nome já diz, precisa de um servidor e um ou vários clientes para se rodar uma interface amigável e necessita utilizar computadores de alto rendimento. Sistemas Web são soluções desenvolvidas para estarem acessíveis na Intranet ou Extranet de uma instituição para serem acessados por clientes, funcionários e parceiros conforme a demanda. Assim, eles requerem um baixo custo em licenças, uma interface amplamente amigável e um ônus com equipamentos baixíssimo, pois para que haja eficiência no sistema só é necessário que no equipamento funcione um navegador.

Berners-Lee (1994 apud JUNIOR; VIDAL, 2006, p.236) destaca uma grande vantagem atrelada aos sistemas de informações baseados na Web:

"Acesso universal significa que você põe algo na Web e pode acessá-lo de qualquer lugar; não importa qual sistema de computador você esteja utilizando, ele é independente de onde você está, que plataforma você está rodando, ou qual sistema operacional você comprou [...]".

É considerável elucidar que não basta desenvolver um aplicativo e colocar para rodar em um servidor para que o sistema funcione. É preciso considerar e avaliar as questões de segurança do servidor onde será instalado esse sistema. Não é desejo de ninguém que dados confidenciais acabem de posse de pessoas inadequadas, o que não é impossível de acontecer caso os servidores possuam baixa segurança.

Existem, já, diversas organizações com soluções desenvolvidas para a Web e mesmo aquelas que ainda não entraram nesse mercado, tem cogitado tais possibilidades. Em uma instituição educacional, a aplicação da tecnologia da informação se evidencia, cada vez mais, como uma questão de muita relevância, de acordo com [Aguiar \(2004\)](#):

"A utilização da informática no âmbito educacional encontra-se, atualmente, em crescimento progressivo. A arte de ensinar e de aprender, nos dias de hoje, não se limita mais ao trabalho realizado dentro da sala de aula. Há um vasto leque de recursos que as tecnologias oferecem e que deve ser vivenciado pela comunidade escolar. Ter acesso a informações, comunicar-se e trocar experiências a grandes distâncias de forma rápida, pesquisar e buscar soluções cada vez mais atuais e eficientes para os problemas, conhecer o mundo em que vivemos sem a necessidade de deslocamento físico e, sobretudo, desenvolver novos níveis de convivência dentro e fora da escola são alguns dos recursos hoje disponíveis com o avanço da tecnologia da informação e da comunicação".

Quando se utiliza da tecnologia da informação no âmbito da administração de dados escolares, faz-se necessário preservar os bancos de dados bem atualizados e consolidados, visto que são conjuntos de informações que se associam para gerar um significado dentro de um contexto computacional. Mantê-los com dados atuais e consistentes é fator determinante para a transparência dos processos de uma organização. Ademais, outros problemas operacionais precisam ser observados para se alcançar um alto nível de organização e coesão. Problemas como a falta de recursos de hardware e software, estabelecimento e aplicação de uma estratégia de backup, ausência de um profissional dedicado ou que ao menos tenha conhecimento no uso do banco de dados, a falta de manutenção e de métricas de controle, minam os processos organizacionais e atingem praticamente todos os setores.

O acompanhamento e administração do banco de dados é imprescindível para que ocorra uma solução de problemas eficaz. Além de reconhecê-los, é necessário detectar qual a causa geradora desses problemas para poder saná-los ou prevê-los em um processo mais avançado de gestão dos SI. Uma vez que isso se transforma em um obstáculo, pode desencadear anos de transtornos para os administradores que não detêm metodologias de gerência e processos bem definidos.

Assim sendo, é indispensável que as organizações educacionais detenham tanto o recurso material quanto a capacidade dos profissionais para que se tenha uma boa execução no uso dos SI na gestão administrativa de informações. Os recursos tecnológicos são, também, fortes aliados do diretor na rotina escolar, visto que proporcionam a disponibilidade

de um grande número de informações transparentes, elaborar relatórios, supervisionar as notas de alunos e o rendimento desses em conjunto com os professores e permite que qualquer outra informação seja colocada em rede.

As escolas não utilizam todo o potencial que os SI oferecem e é nesse cenário que surge a relevância da cooperação entre professores e funcionários, para que a tecnologia não seja utilizada de maneira exclusiva, mas faça parte do coletivo. No cenário atual, em que se deve possuir conhecimentos sobre os recursos tecnológicos para não ficar atrasado e perdido em meio a evolução dos SI, as tecnologias nos ajudam a alcançar o que está consolidado e a organizar o que está desordenado, desconexo, desarranjado, caótico. Segundo [Lima \(1994 apud ROSINI, 2003\)](#), a tecnologia é:

"Muito mais que apenas equipamentos, máquinas e computadores. A organização funciona a partir da operação de dois sistemas que dependem um do outro de maneira variada. Existe um sistema técnico, formado pelas técnicas e ferramentas e utilizadas para realizar cada tarefa. Existe também um sistema social, com suas necessidades, expectativas, e sentimentos sobre o trabalho. Os dois sistemas são simultaneamente otimizados quando os requisitos da tecnologia e as necessidades das pessoas são atendidos conjuntamente. Assim, é possível distinguir entre tecnologia (conhecimento) e sistema técnico (combinação específica de máquinas e métodos empregados para obter um resultado desejado)".

É preciso que as instituições saibam utilizar as tecnologias para realizar uma boa gestão dos dados, não basta apenas adquirí-las. Uma gestão realizada de maneira consciente pode reduzir consideravelmente as redundâncias e custos no tempo de trabalho em uma organização. Assim sendo, certamente haverá ganhos em relação ao tempo de realização dos processos de trabalho.

### 3 Descrição do Problema

Conservatórios Estaduais de Música são escolas públicas regidas pelas mesmas leis que guiam as escolas de ensino regular, porém são voltadas para o ensino de música através de disciplinas que vão desde o instrumento escolhido pelo candidato a matérias relacionadas (História da Arte, História da Música e Apreciação Musical, Percepção Musical, Estruturação, Folclore Regional e Música Popular, Música de Câmara e Prática de ensino). Ao todo, no estado de Minas Gerais, estão em funcionamento 12 Conservatórios Estaduais.

O Conservatório Estadual de Música Lobo de Mesquita de Diamantina - MG foi inaugurado oficialmente em 1971, através da Lei 811 de 13 de dezembro de 1951, que ementa a criação de cinco conservatórios estaduais e música. A instituição tem cerca de 1500 alunos com idades diversas, que variam desde crianças (a partir de 6 anos) a adultos, possui em seu grupo docente 48 professores que são auxiliados por uma supervisora. Conta ainda com 14 funcionários de funções diversas. Oferece os seguintes cursos:

- A. Guitarra
- B. Flauta Doce
- C. Flauta Transversal
- D. Percussão
- E. Piano
- F. Violão
- G. Violino
- H. Saxofone
- I. Trompa
- J. Trombone
- K. Trompete
- L. Bateria
- M. Cavaquinho
- N. Contrabaixo Elétrico

O. Teclado

P. Violoncelo

O Conservatório oferece duas modalidades de curso com seus devidos e respectivos planos curriculares. O curso Educação Musical (conforme as Tabelas 1, 2, 3) subdivide-se em três ciclos: Inicial, Intermediário e Complementar e cada ciclo se subdivide em três fases: I, II e III; e o curso de Formação Profissional (conforme a Tabela 4).

<b>Fase I</b>	<b>Fase II</b>	<b>Fase III</b>
Musicalização	Musicalização/Percepção	Musicalização/Percepção
Canto Coral	Canto Coral	Canto Coral
Instrumento	Instrumento	Instrumento
		Atividade Artística Complementar

Tabela 1: Plano curricular do curso de Educação Musical - Ciclo Inicial.

<b>Fase I</b>	<b>Fase II</b>	<b>Fase III</b>
Musicalização/Percepção	Musicalização/Percepção	Percepção
Canto Coral	Canto Coral	Canto Coral
Instrumento	Instrumento	Instrumento
Atividade Artística Complementar		Prática de Conjunto

Tabela 2: Plano curricular do curso de Educação Musical - Ciclo Intermediário.

<b>Fase I</b>	<b>Fase II</b>	<b>Fase III</b>
Percepção	Percepção	Percepção
Canto Coral	Oficina Multimeios	Oficina Multimeios
Instrumento	Instrumento	Instrumento
		Atividade Artística Complementar

Tabela 3: Plano curricular do curso de Educação Musical - Ciclo Complementar.

<b>1º Ano</b>	<b>2º Ano</b>	<b>3º Ano</b>
Atividade Artística Complementar	Estrutura Musical	Estrutura Musical
Folclore	História da Música e Apreciação Musical	Instrumento
Instrumento	Instrumento	Prática de Conjunto
Noções de Educação Musical	Percepção	Prática de Ensino
Oficina Multimeios	Prática de Conjunto	Produção Cultural e Empreendedorismo
Percepção	Prática de Ensino	
Prática de Conjunto		

Tabela 4: Plano curricular do curso de Formação Profissional.

Cada aluno ingressante é classificado em determinado Ciclo/Fase de acordo com a idade, nível de conhecimento e aptidão musical, podendo ser reclassificado através de prova classificatória. O aluno também possui, em cada matéria do respectivo plano curricular em

que se encaixa, professores a quem é relacionado, os quais farão a avaliação do estudante no decorrer de bimestres, atribuindo-lhe notas/conceitos e frequência escolar.



## 4 Ferramentas Computacionais de Suporte

Para a execução do projeto de desenvolvimento do sistema para gerenciamento de alunos e turmas, utilizou-se a linguagem de programação PHP orientada a objetos, bem como o padrão de projeto arquitetural MVC (Model-View-Controller).

Para a implementação dessa especificação utilizou-se o CakePHP como framework e por se tratar de uma aplicação desenvolvida para funcionar como um sistema Web, foi utilizada a linguagem de marcação HTML e necessitou-se de um servidor de aplicação embutido. Foi escolhido o software WAMPServer, que instala automaticamente um ambiente de desenvolvimento e proporciona a criação de aplicações em PHP, MySQL e Apache. Como editor de texto utilizou-se o Sublime Text 3 e para controle versão optou-se por utilizar o sistema de controle GitHub. Todas as aplicações foram instaladas em uma máquina com sistema operacional Windows 10 Pro.

### 4.1 PHP

A Linguagem PHP começou em 1994, quando Rasmus Lerdorf resolveu criar um gerenciador de visitas para o seu site para monitorar o número de acessos. Sua primeira criação não era uma linguagem, e sim uma ferramenta. Chamada inicialmente de Personal Home Page, a linguagem era formada por um conjunto de scripts escritos em linguagem C e usava comandos simples inspirados da linguagem PERL. Com o tempo, mais pessoas passaram a utilizá-la e Rasmus adicionou vários recursos, como a interação com bancos de dados (CAKEPHP, 2016).

Em 1995, Rasmus liberou o acesso ao código-fonte da CGI (Computer Graphic Imagery), e com isso mais desenvolvedores puderam se juntar ao projeto. Uma segunda versão da ferramenta foi lançada em novembro de 1997 e já permitia a manipulação de formulários e pequenos livros de visita: o Personal Home Page/ Forms Interpreter, ou PHP/FI 2.0. Essa versão ficou bastante popular e começou a se espalhar. Podia se observar que cerca de 60.000 domínios, algo em torno de 1% da Internet na época, usavam PHP, suportado especialmente por Rasmus, conforme o Manual do PHP Group (2001).

No mesmo ano, dois discentes israelenses que usavam PHP em um projeto acadêmico de comércio eletrônico, Andi Gutmans e Zeev Suraski, decidiram contribuir com Rasmus para melhorar a linguagem. Baseados no PHP/FI 2.0, eles reescreveram o núcleo da ferramenta, e a transformou em uma Linguagem de Scripts, nos moldes em que o PHP é conhecido atualmente. Assim, esse fato deu início ao PHP 3, oficialmente viabilizado em junho de 1998. As características mais marcantes do PHP 3 eram a oportunidade de

conexão com múltiplos bancos de dados, suporte à orientação a objetos, a extensibilidade, protocolos novos, uma sintaxe mais estável e uma API nova, o que viabilizou a concepção de novos padrões e atraiu vários desenvolvedores para o PHP. A presença da linguagem já era marcante em aproximadamente 10% dos domínios da Internet e já despontava como uma das linguagens mais usadas. O significado da sigla mudou para PHP: Hypertext Preprocessor, mostrando a nova perspectiva de uma linguagem com finalidades mais abrangentes.

Zeev e Andi, no inverno de 1998, iniciaram os trabalhos de uma reestruturação do núcleo do PHP, fugindo quase que totalmente da proposta primária de Rasmus Lerdorf. Eles tinham a pretensão de aprimorar o desempenho e a modularidade do PHP em projetos mais complexos e o resultado foi renomeado para Zend Engine (Zeev e Andi). A quarta versão do PHP surge em 2000 para acabar com a maioria das falhas de segurança e erros encontrados pela comunidade, proporcionando aperfeiçoamentos como suporte a diversos servidores Web, seções, bem como a abstração de sua API, podendo ser utilizada, até mesmo, como linguagem para shell script.

Mesmo com toda essa dedicação, o PHP precisava ainda de mais aporte no quesito orientação a objetos. Tais melhorias vieram com o PHP 5, que foi disponibilizado oficialmente em julho de 2004, depois de um extenso período de quatro anos de desenvolvimento. Por mais de uma década a versão ficou globalmente conhecida, recebeu mais atribuições e se estabeleceu ano após ano como uma das linguagens orientadas a objetos que mais cresce. Hoje, segundo Dall'oglio (2007), conta com 32 versões e mais de mil correções de falhas de segurança. Calcula-se que o PHP é utilizado em mais de 80% de servidores Web, sendo, no momento, a linguagem para desenvolvimento Web mais usada.

PHP 7 é o primeiro grande lançamento PHP desde a versão 5.0, que foi lançada em 2004, há mais 12 anos. Os desenvolvedores que persistem a linguagem, resolveram passar o nome do PHP 6.0 para o 7.0, porque várias mudanças existentes na versão 5.6 já equivalem ao PHP 6, assim sendo, atualmente está disponível somente a versão 7.0 da linguagem.

### 4.1.1 Como funciona o PHP

PHP <sup>1</sup> é uma linguagem interpretada. Ele é interpretado em uma máquina servidora (server-side) onde o código fonte é executado por um programa chamado interpretador, gerando um código binário executável. Isto é, quando a máquina cliente envia uma requisição à máquina servidora, ela é responsável por tratá-la de acordo com a necessidade da aplicação: fazer consultas de dados e armazená-los no banco, processar os dados, gerar o HTML que será interpretado e exibido pelo navegador do cliente, entre outros.

---

<sup>1</sup> <<http://www.petsi.facom.ufu.br/system/files/Minicurso%20PHP.pdf>>

O PHP, em conformidade com o Manual do (PHP Group, 2001), tem como principal função o desenvolvimento de aplicações Web. E essa é a melhor aplicabilidade do PHP, já que é uma linguagem criada para isso, utilizada para implementar páginas automáticas e dinâmicas. Dinâmicas porque as páginas em PHP são geradas no momento em que é efetuada uma requisição ao servidor. Um servidor jamais poderia armazenar todas as combinações possíveis de páginas, o que existem armazenados são conteúdos que serão alocados nos devidos locais conforme pedido efetuado, sobre qualquer assunto, pelo cliente.

O funcionamento do PHP para automatizar uma página é simples. Toda saída de código PHP é HTML, logo é imprescindível ter conhecimentos de HTML para que se saiba desenvolver o desejado. Em um simples esquema, pode-se entender:

PHP -> Servidor Web -> Interpretador -> HTML -> Cliente

Um arquivo PHP, geralmente, tem a extensão .php e o código dos programas iniciam-se com `<?php` e terminam com `?>`, como pode ser visto no trecho de código abaixo<sup>2</sup>:

---

```
1 <?php
2 //codigo ;
3 //codigo ;
4 //codigo ;
5 ?>
```

---

## 4.2 Programação Orientada a Objetos (POO)

Nessa seção serão apresentados os conceitos primordiais que permeiam o uso do padrão de orientação<sup>3</sup> a objetos na programação, sempre utilizando a linguagem PHP como motivador. De acordo com Ivan Luiz Marques Ricarte (2001), objetos são instâncias de classes, que definem quais características e comportamentos um objeto possui e como ele pode manipulá-los. Em POO, um dos pontos mais marcantes em relação a outros paradigmas de programação, que também possibilitam a declaração de estruturas e operações, está no conceito de herança. Simultaneamente à definição de herança, deve ser destacada a relevância polimorfismo, que permite distinguir funcionalidades que um programa irá operar de forma dinâmica no momento da execução. O padrão de programação orientada a objetos se distingue dos demais pois permite o reuso de código, essa concepção está intimamente presente através dos conceitos de herança e polimorfismo.

---

<sup>2</sup> <<https://www.mendeley.com/viewer/?fileId=725c4ca1-1d49-8d97-6006-677eb1930045&documentId=bd7894b2-1b9b-3d23-a6fd-a642cc0af4fe>>

<sup>3</sup> <<https://social.technet.microsoft.com/wiki/pt-br/contents/articles/9644.conceitos-de-orientacao-a-objetos.aspx>>

A POO<sup>4</sup> trata os sistemas como uma projeção da realidade, uma analogia, onde os elementos do mundo real comportam-se como objetos que interagem entre si. As aplicações que são desenvolvidas sobre esse paradigma são ordenadas em módulos, onde cada módulo exibe atributos e operações que interagem sobre esses módulos. Os modelos são chamados de classes e as operações presentes nas classes são chamados de métodos.

### 4.2.1 Diferença entre objetos de classes

Para trabalhar diretamente com os elementos importantes da POO, é preciso um entendimento primordial das diferenças entre objetos e classes. Essa seção discorrerá sobre a base das classes, suas diferentes capacidades e alguns de seus usos dentro da linguagem PHP.

Uma analogia pode ser feita entre classe e um molde para fabricar bonecos de gesso, que representa claramente a forma, o tamanho e várias outras características dos bonecos que serão fabricados, mesmo que eles ainda não existam (SANTIAGO, 2016).

Por sua vez, o objeto seria um boneco de gesso propriamente dito, fabricado conforme o molde. Os atributos contidos no objeto são como o gesso, tinta e outros materiais que fazem parte do boneco de gesso. Sem o formato dado pelo molde, são só um amontoado de matérias primas. No entanto, quando tudo é misturado e enformado seguindo um padrão, surgem os bonecos.

"As classes servem de estrutura para os dados e ações, e usam essa informação para construir objetos"(Jason Legstorf, 2011). Um objeto ou mais podem ser constituídos da mesma classe, concomitantemente e independentemente. Em continuidade com a comparação entre classes e objetos no tocante aos bonecos de gesso, é análoga à maneira que pode-se constituir uma fábrica de bonecos, utilizando o mesmo molde: vários bonecos diversificados, muito semelhantes uns aos outros, mas com detalhes e ornamentos distintos.

#### 4.2.1.1 Classes

Classe é um padrão para o estabelecimento de objetos. Por meio da definição de uma classe, esboça-se quais características ou atributos o objeto conterá. A declaração de uma classe especifica também qual o comportamento de objetos da classe, além da especificação dos atributos. Isto é, quais aplicabilidades podem ser usadas em objetos da classe, descritas através de métodos. Um método é proporcional a um procedimento ou função, com a ressalva que ele trabalha apenas suas variáveis locais e os atributos que foram estabelecidos para a classe.

---

<sup>4</sup> <<<http://pt.slideshare.net/leonardomelosantos/conceitos-bsicos-de-programao-orientada-a-objetos>>>  
>

Uma vez definidas as classes que irão constituir uma aplicação, bem como qual deve ser sua estrutura interna e procedimentos, é permitido criar essas classes em PHP, ou qualquer linguagem orientada a objetos.

As funções de cada classe são definidas pelos métodos. Assim sendo, as ações peculiares que os objetos estão aptos a executar, bem como o que é possível fazer com os objetos da classe, tudo isso é definido na forma de métodos. "Cada método é caracterizado por uma assinatura, composta por um identificador para o método (o nome), o tipo para o valor de retorno e sua lista de argumentos, sendo cada argumento identificado por seu tipo e nome"(Ivan Luiz Marques Ricarte, 2001).

Por meio da técnica de sobrecarga, dois métodos de uma classe podem possuir nomes iguais, uma vez que seus atributos sejam diferenciados. Essa condição não acarreta incompatibilidade pois o compilador está capacitado a identificar qual método deve ser melhor indicado com base na interpretação dos tipos dos argumentos do método. Nessa situação, diz-se que acontece a ligação prematura para o método adequado.

Em PHP, variáveis pertencentes a uma classe são denominadas propriedades. É possível também verificar serem descritas usando outras expressões como atributos ou campos. São declaradas utilizando uma das palavras-chave *public*, *protected*, ou *private*, seguidas de uma declaração convencional de variável. Esta declaração pode compreender sua inicialização, mas esta inicialização tem que possuir um valor constante, ou seja, tem que ser permitido avaliá-la em tempo de compilação e não se deve necessitar de informações de tempo de execução.

#### 4.2.1.2 Objetos

Objetos são instâncias de classes. É por meio deles que, praticamente, todo o processamento acontece em aplicações desenvolvidas por meio das linguagens de programação orientadas a objetos. A utilização coerente de objetos, seguindo os conceitos relacionados à sua definição de acordo com o especificado no padrão de desenvolvimento orientado a objetos, é fundamento para a elaboração de sistemas complexos e efetivos. No campo da solução, um objeto é o que denota alguma entidade, concreta ou abstrata, do campo de interesse do problema que está em avaliação.

Objetos semelhantes são associados em classes e em POO tudo pode ser caracterizado como um objeto em potencial. Um objeto não possui muita distinção de uma variável convencional, na visão da programação orientada a objetos.

Quando se declara uma variável do tipo *int* em uma linguagem de programação, essa variável possui um espaço na memória para armazenar o seu valor e um conjunto de operações que podem ser utilizadas por meio de operadores estabelecidos e que podem ser aplicados a valores inteiros. De forma análoga, um objeto recebe um espaço de memória

quando é instanciado. Esse espaço serve para armazenar o valor (atributos definidos pela classe) e o conjunto de operações (métodos definidos pela classe) que podem ser utilizadas pelo objeto, de acordo com [Ivan Luiz Marques Ricarte \(2001\)](#).

Uma aplicação orientada a objetos é formada por uma coleção de objetos que se comunicam por meio de trocas de mensagens. Na realidade, essa permuta de mensagens significa o uso de métodos em objetos.

Os métodos em POO indicam que a estrutura de um objeto e a implementação de suas operações precisam ser, ao máximo, privados. De maneira geral, as características de um objeto não podem ser percebidas externamente. O mesmo deve ser aplicado aos métodos, conhecer somente a especificação é o bastante. Não há necessidade de conhecer detalhes da implementação das funcionalidades.

### 4.2.2 Encapsulamento

O Encapsulamento é uma técnica que consiste em deixar visível/acessível apenas algumas partes dos dados do objeto, ocultando os detalhes de implementação dos usuários da classe. De acordo com [José Carlos Macoratti \(2010\)](#), dessa forma, o objeto encapsula e protege seus dados de outros objetos, permitindo que seus dados sejam acessados por intermédio de seus próprios métodos. Cada classe configura sua estrutura própria, indicando quais e de qual modo seus métodos e dados devem ser encapsulados.

Para utilizar esse elemento, somente deve ser revelado o mínimo possível dos atributos e métodos. O uso de encapsulamento sugere que, na orientação a objetos, a representação do estado do objeto deve manter-se oculta. A manipulação de todo objeto deve ser feita exclusivamente pelos métodos públicos do mesmo, aos quais as características são reveladas.

Cada objeto deve ser manipulado exclusivamente através dos métodos públicos, dos quais apenas a assinatura deve ser revelada. "O conjunto de assinaturas dos métodos públicos da classe constitui sua interface operacional"([Ivan Luiz Marques Ricarte, 2001](#)).

Assim, particularidades das operações do objeto não são mostradas, possibilitando que o usuário do objeto manuseie em uma camada mais alta de abstração, sem se preocupar com os pormenores intrínsecos da classe. Essa simplicidade possibilita facilitar o desenvolvimento de programas com funcionalidades complexas, como interfaces gráficas ou aplicações distribuídas.

Um simples exemplo seria o de uma pessoa que vai usar um instrumento musical eletrônico, por exemplo um teclado musical. Ela não precisa ter conhecimento de toda complexidade utilizada na construção dos circuitos e partes internas do aparelho. Precisa apenas conhecer as funções existentes na interface que são os botões de acesso a funções e teclas.

Em PHP, a visibilidade de um atributo ou método pode ser determinada preestabelecendo a declaração através dos códigos *private*, *public* ou *protected*. Segundo o [PHP Group \(2001\)](#), membros instituídos como públicos podem ser manipulados por qualquer usuário. Membros protegidos só podem ser manipulados pela classe onde foram declarados e em classes herdeiras. Já os membros privados só podem ser manipulados na classe que os definem.

### 4.2.3 Herança

A definição de encapsulamento de atributos e métodos em variáveis não é específico da orientação a objetos, a programação estruturada segue essa mesma definição. O que diferencia a orientação a objetos e a deixa ímpar é o fato de utilizar a concepção de herança.

Herança é uma técnica que possibilita que atributos em comum a várias classes sejam declarados em uma classe base, ou superclasse. Com base nessa superclasse, existe a possibilidade de instanciar novas classes. Cada classe ou subclasse oriunda traz consigo as características (atributos e métodos) da classe base e soma a elas o que for estabelecido de especificidade. Segundo ([Ivan Luiz Marques Ricarte, 2001](#)), existem alguns tipos de relacionamentos com herança<sup>5</sup>, dentre eles estão:

- **Especificação:** a superclasse não estabelece nenhuma funcionalidade da subclasse, apenas indica o que ela poderá disponibilizar. Somente o conjunto de características dos métodos públicos, ou seja, a interface da superclasse, é herdada pela subclasse.
- **Extensão:** a subclasse estende a superclasse, acrescentando novas características (métodos e atributos). Não ocorrem modificações na superclasse, razão pela qual este modelo de relacionamento é usualmente chamado de herança estrita.
- **Combinação de extensão e especificação:** a subclasse recebe uma implementação modelo de ao menos um dos métodos e a interface da superclasse. Assim, a subclasse pode modificar métodos para descrever o desempenho no tocante ao que é disponibilizado pela superclasse. Pode ocorrer que a superclasse tenha declarado algum método, porém não tenha implementado. Assim, a subclasse poderá estender alguma implementação para esse método. Este modelo de relacionamento é chamado de herança polimórfica.

O tipo de relacionamento que mais ocorre na POO<sup>6</sup>, de acordo com [Ivan Luiz Marques Ricarte \(2001\)](#), é sem dúvida, o último. Algumas modelagens trazem um tipo de

<sup>5</sup> <[https://pt.wikibooks.org/wiki/Programa%C3%A7%C3%A3o\\_Orientada\\_a\\_Objeto:\\_Uma\\_Abordagem\\_com\\_Java/Introdu%C3%A7%C3%A3o/Heran%C3%A7a](https://pt.wikibooks.org/wiki/Programa%C3%A7%C3%A3o_Orientada_a_Objeto:_Uma_Abordagem_com_Java/Introdu%C3%A7%C3%A3o/Heran%C3%A7a)>

<sup>6</sup> <<http://www.dca.fee.unicamp.br/cursos/PooJava/heranca/formas.html>>

herança denominado como contração. Contração é uma diversificação de herança, onde a subclasse suprime métodos da superclasse com o propósito de gerar uma classe mais acessível. A eliminação dos métodos pode ocorrer pelo redirecionamento de métodos com corpo vazio. O problema com este mecanismo é que ele infringe a premissa da substituição, pelo qual uma subclasse pode ser utilizada em todos os itens onde a superclasse poderia ser empregada. Caso a contração configure um recurso pertinente em uma hierarquia de classes, certamente a hierarquia precisa de ser reanalisada para localização de inconsistências. De uma maneira geral, o mecanismo de contração precisa ser evitado.

O PHP utiliza o conceito de herança em seu modelo de objetos. Este fundamento influencia a maneira com que classes e objetos se associam uns com os outros. Por exemplo, a subclasse herda todos os métodos públicos e protegidos da classe base, ao se estender uma classe. A não ser que uma classe transcreva esses métodos, eles preservarão suas funções originais. Isto é válido para definir uma funcionalidade abstrata, e possibilitar a implementação de métodos adicionais em objetos similares sem que seja preciso reimplementar todas as funcionalidades compartilhadas.

#### 4.2.4 Sobrecarga

Em PHP, a sobrecarga possibilita meios para instanciar dinamicamente os métodos e atributos. Estas associações dinâmicas oferecem a uma classe muitas maneiras de agir e são processadas por métodos mágicos.

Os métodos de sobrecarga são chamados ao compartilhar características ou métodos que não foram evidenciados ou não são perceptíveis no escopo vigente. Esta seção utilizará as expressões propriedades inacessíveis e métodos inacessíveis para aludir a esta combinação de declaração e visibilidade.

Os métodos de sobrecarga precisam ser declarados como públicos. A interpretação do PHP de sobrecarga, segundo o Manual do [PHP Group \(2001\)](#), é diferente da maioria das linguagens orientadas a objeto. Geralmente, sobrecarga dispõe da capacidade de se possuir métodos com nomes iguais, mas com tipos de argumentos e quantias diferentes.

De acordo com [José Malcher Jr. \(2015\)](#), é possível incluir elementos do objeto em tempo de execução por meio do uso da sobrecarga em PHP. Isso ocorre com a implementação da classe utilizando os métodos sobrecarregados `__set`, `__get`, `__call` and, `__callStatic`. "O significado de sobrecarga no PHP é diferente de muitas outras linguagens de programação" ([PHP Group, 2001](#)).

Os métodos **setter** e **getter** são implementados pois os métodos `__set` e `__get` propiciam um modo apropriado para isso. Eles são utilizados com frequência para ler e escrever com segurança. Quando os parâmetros chamados são privados, tais métodos sobrecarregados são usados, seja por não estarem declarados na classe ou porque são

privados a partir do escopo atual.

O método `__set` é chamado com o nome do atributo e o valor como seus argumentos. Isso se dá ao definir o valor de uma característica que não é pública. Semelhantemente, o método `__get` é chamado com o nome da propriedade como seu argumento ao acessar uma propriedade privada.

No que diz respeito a sobrecarga de métodos, [José Malcher Jr. \(2015\)](#) disse que há duas formas para manipular chamadas a métodos privados de uma classe que são: `__callStatic` e `call`.

Ocorre a invocação do método `__call` quando há chamada de método de instância. O nome do método que está sendo chamado é o primeiro argumento para `__call`. Uma matriz numérica que possui os parâmetros transmitidos para o método é o segundo argumento. Tais argumentos são os mesmos utilizados para o método `__callStatic`, que manipula chamadas para métodos estáticos privados.

Existem as funções `empty`, `isset` e `unset`, que não funcionam em atributos sobrecarregados, mas somente em atributos definidos explicitamente. Sobrecarregando os métodos `__unset` e `__isset`, é possível que esta funcionalidade seja incorporada a uma classe. [José Malcher Jr. \(2015\)](#) explica que o método `__isset` será chamado quando `isset` for chamado em um atributo privado. Já o método `__unset` será chamado se `unset` for chamado em uma propriedade privada.

De acordo com [PHP Group \(2001\)](#), se o resultado de `__isset` for `false`, a função `empty` irá retornar `true`. Se `__isset` e `__get` estão aplicados, então o `empty` irá funcionar em atributos sobrecarregados. Mas somente se ambos os métodos estiverem sendo utilizados.

## 4.3 MVC (Model-View-Controller)

O sucesso no desenvolvimento de sistemas que utilizam o paradigma de orientação a objetos está ligado com os padrões utilizados para a construção do mesmo. Com o aumento do tamanho e complexidade das aplicações tornou-se fundamental a utilização de padrões que organizem e separam partes dos sistemas.

O padrão de projeto MVC (Model-View-Controller) estabelece um modelo que separa as funcionalidades envolvidas na manutenção e lógica de negócios da apresentação da aplicação para os usuários do sistema (figura 1). Em uma implementação desse padrão tem-se:

- A. **View - Apresentação ou visualização:** Atua como interface com o usuário, não implementa acesso aos dados e atua na entrada e exibição dos dados através das ferramentas de ferramentas como HTML, XHTML, CSS, XML, ASP, Applets, dentre

outras.

- B. **Model - Lógica da Aplicação:** Modela dos dados de acordo com a lógica dos processos do negócio, está atrelado com o modelo de armazenamento, manipulação e geração de dados. Em um paradigma de orientação a objetos, as classes que descrevem os objetos da aplicação são implementados aqui.
- C. **Controller - Controle:** Serve como uma parte intermediária entre a View e a Model, determina o fluxo da aplicação. O Controller é responsável por validar os dados recebidos da View e repassar a Model.

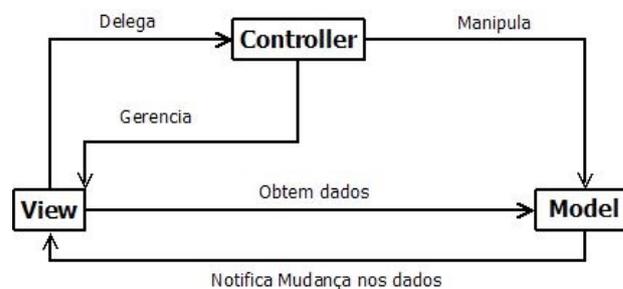


Figura 1: Diagrama de funcionamento do MVC

O padrão MVC, em resumo, divide e desacopla a interface da aplicação, permitindo que alterações feitas na View não altere o Model e vice-versa. Isso implica em independência entre as camadas, o que aumenta a facilidade de manutenção de código. É bastante usado desde aplicações pequenas até as de fato muito grandes, como por exemplo os frameworks Zend Framework, Laravel, **CakePHP** e muitos outros que usam amplamente o MVC.

### 4.3.1 MVC e PHP

Nesta seção pode-se ver um exemplo de código PHP usando MVC<sup>7</sup> sem usar qualquer mecanismo externo, o código exibido é bastante simples, mas mostra perfeitamente como funciona o padrão.

Há três arquivos separados que irão guardar todo o código em camadas. A primeira camada (Model) armazena toda a lógica, ou seja, conexão ao banco, selects, inserts, entre outros. A do meio, a segunda (Controller), é quem decide quando chamar as outras camadas (a primeira e a terceira) e o que elas irão fazer. A terceira (View) mostra os dados. Funciona como um mapa, analogamente.

<sup>7</sup> <<https://www.webdevbr.com.br/mvc-na-pratica-entendendo-o-padrao-mvc-na-pratica>>

#### 4.3.1.1 Exemplo prático de MVC

Esta subseção<sup>8</sup> mostra um exemplo prático, um *hello world* em MVC. A princípio, cria-se um arquivo chamado *index.php* com o seguinte:

---

```
1 <?php
2 //executa o controler
3 include "controller.php";
4 (new Controller) -> action();
5 ?>
```

---

O *controller* foi escrito dentro de parênteses, assim não é preciso instanciar em uma variável, uma vez que só será usado uma vez. Então executou-se um método chamado *action()*. A *action* faz parte do *controller* e normalmente é utilizada para definir uma página típica do método. Um *controller* denominado *Usuarios* poderia possuir as *actions* login, logout, editar, ver, apagar, entre outros, por exemplo. Posteriormente cria-se o arquivo *controller.php*:

---

```
1 <?php
2 include "model.php";
3 include "view.php";
4 class Controller {
5     public function action()
6     {
7         $message = (new Model()) -> getMessage();
8         return (new View($message)) -> render();
9     }
10 }
11 ?>
```

---

Em MVC é possível que o *controller* possua menor tamanho que as outras camadas, mas isso não é uma regra. Agora a descrição do Model:

---

```
1 <?
2 class Model
3 {
4     public function getMessage()
5     {
6         //Aqui criam-se as regras ,
7         //por exemplo, buscar esta mensagem
8         //no banco de dados.
9         return "Hello Word!";
10    }
11 }
12 ?>
```

---

<sup>8</sup> <<https://www.webdevbr.com.br/mvc-na-pratica-entendendo-o-padrao-mvc-na-pratica>>

E em seguida o View:

---

```
1 <?php
2 class View
3 {
4     private $message;
5     public function __construct($message){
6         $this -> message = $message;
7     }
8     public function render()
9     {
10         echo $this -> message;
11     }
12 }
13 ?>
```

---

Aqui utilizou-se a impressão de um texto estático apenas com a finalidade de exemplificar o padrão MVC. Mas é evidente que isso não é necessário na prática, é apenas uma ilustração das inúmeras possibilidades de aplicação do padrão. Além de separar as regras do negócio do HTML, pode-se melhorar muito mais a camada do Model com validações, tratamentos, alerta por email ou sms sempre que alguém exibir a mensagem, busca a banco de dados, trocar o tipo do banco de dados. Em suma, o Controller e a View ficariam íntegros. MVC é ideal para qualquer projeto de maior porte que o descrito nesse exemplo.

## 4.4 Framework CakePHP

Segundo a documentação, o [Cookbook \(2016\)](#), o CakePHP é um framework de desenvolvimento rápido para PHP, livre e de código aberto. O framework modifica o modo de desenvolvimento Web maçante, propiciando as ferramentas necessárias para programar o que se quer realmente, que é a lógica própria da aplicação. Ao contrário de ter o mesmo trabalho toda vez que é desenvolvido uma nova aplicação, basta pegar uma cópia do CakePHP e começar a modelar o novo projeto.

### 4.4.1 Razão para usar um framework

Pelo simples fato de não ter que ficar refazendo trabalho a todo tempo. Não é preciso ficar rescrevendo códigos, como as conhecidas *queries* de *insert*, *update*, *select* e todas as funções essenciais que os sistemas de CRUD possuem. O uso de um framework agiliza o desenvolvimento, sem que seja necessário ficar refazendo a escrita daquele código que se repete em quase todo o projeto. Assim, a preocupação do programador fica apenas na regra de negócio do sistema ([Allan Douglas, 2014](#)).

### 4.4.2 Razões para usar CakePHP

Baseado no framework Ruby On Rails (RoR), o CakePHP é excelente para quem já está habituado com a linguagem ou quer começar a programar através da aplicação. É muito simples de utilizar o framework e ele usa o termo *Convention over Configuration*, que quer dizer convenção ao invés de configuração. Com o uso CakePHP, somente se executa tarefas habituais, baseadas em algumas regras do framework. Ao contrário de outros frameworks, onde é necessário escrever grandes arquivos XML de configuração da aplicação.

O CakePHP detém um time ativo de desenvolvedores e uma grande comunidade, agregando grande prestígio ao projeto. "Usar o Cake significa que o núcleo da aplicação será bem testado e estará em permanente aprimoramento, além de permitir que o usuário fique longe da reinvenção da roda" (CAKEPHP, 2016).

De acordo com a documentação do CakePHP, o Cookbook (2016), através do framework o usuário pode usufruir dos vários recursos listados a seguir:

- Arquitetura MVC
- Comporta o PHP 5.2.6 ou superior
- Componentes de Segurança, Email, Sessão, Cookie e Tratamento de Requisições
- Comunidade amigável e dinâmica
- Controle de Acessos adaptável
- Criação de código
- Criação de protótipos através de Scaffolding
- Flexibilidade na licença
- Helpers para AJAX, JavaScript, formulários HTML e outros
- Interação com o banco de dados por meio de CRUD integrado
- Limpeza dos dados
- Localização
- Por meio de pouca ou nenhuma configuração do servidor Web, funciona a partir de qualquer diretório do website
- Sistema de Cache maleável
- Templates versáteis e rápidos (Sintaxe PHP, com helpers)

- URLs e rotas customizáveis, requisições realizadas com transparência
- Validações embutidas

## 4.5 HTML

Conforme [Diego Eis \(2011\)](#), HTML é uma das linguagens utilizadas para desenvolver Web sites, cuja sigla vem do inglês Hypertext Markup Language e quer dizer em português Linguagem de Marcação de Hipertexto.

Foi elaborada para ser de simples assimilação por usuários humanos e também pelas máquinas e é a linguagem pilar da Internet. Como exemplo, tem-se sistemas de busca ou outros sistemas que exploram a Internet colhendo informações.

Tim Berners-Lee é o nome do criador do HTML. Ele implementou a linguagem para difundir pesquisas e para a comunicação entre ele e um grupo de colegas. O HTML ficou bastante conhecido quando começou a ser usado para conceber a rede pública na época, o que se transformaria mais adiante na Internet dos dias atuais.

### 4.5.1 Tags no HTML

A linguagem HTML é fundamentada em marcação. Os fundamentos são evidenciados para exibir quais informações a página mostra, como por exemplo, um título. O título de um artigo, da manchete do site, são marcados com um elemento/tag chamada H1, como no exemplo:

---

```
1 <h1>Aqui vai o texto do titulo</h1>
```

---

O texto está entre duas marcações denominadas tags, que são abertas e depois fechadas. No exemplo acima, a tag foi aberta com `<h1>` e fechada com `</h1>`. O que está dentro é o conteúdo que será exibido para o usuário.

Com o uso das tags, passa-se ao navegador o que significa cada informação, o que é um parágrafo, o que é um título, um formulário, o que é um botão e assim por diante. Informa-se também o que é cada dado para os sistemas de busca. Por exemplo, o Google que, nesse caso, para mostrar os resultados de busca necessita distinguir o que é um parágrafo e o que é um título. Isso ocorre pelo uso das tags.

Todo HTML inicia do mesmo modo e pode-se usar o mesmo código toda vez que se inicia um novo HTML.

---

```
1 <!DOCTYPE html>
2
3 <html lang="pt-br">
4 <head>
```

```
5     <meta charset="utf-8">
6     <title>Titulo da pagina</title>
7 </head>
8 <body>
9     <h1>Aqui vai o texto do titulo</h1>
10    <p>Aqui vai o texto do paragrafo .
11    Geralmente paragrafos tem muitas palavras ,
12    letras menores que as do titulo</p>
13 </body>
14 </html>
```

---

DOCTYPE é como é denominada a primeira linha. O Doctype, segundo [Diego Eis \(2011\)](#), comunica aos navegadores, a leitores de tela, motores de busca e outras coisas, qual é o tipo de documento que eles irão carregar. Ainda de acordo com o autor, há outros códigos prováveis de carregar, como por exemplo o XML. Assim sendo, o Doctype é uma maneira de informar ao navegador como ele deve proceder ao ler o código.

Depois inicia-se com a Tag HTML. Logo, tudo o que estiver entre as tags `<html></html>` será escrito em HTML. Ao lado da palavra HTML há um atributo chamado *lang*, onde é informado o idioma em que o texto será escrito.

Depois da tag html vem a tag `<head>`, que indica o título do documento e a tabela de caracteres que o navegador tem que utilizar para compilar o texto.

É através da tag tag `<tittle>` que o título do documento será informado. Por isso ela é muito relevante. Essa tag é utilizada pelo Google, por exemplo, e outros sistemas de busca para mostrar em seus resultados o título da página, o que é bastante importante para esses motores.

Por fim, logo depois da tag de fechamento `</head>` vem a tag `<body>`. Dentro desta tag é escrito todo o código HTML do restante do site.

## 4.5.2 Desenvolvimento em Camadas

No que diz respeito ao desenvolvimento Web, um dos problemas mais relevantes é a fusão de vários códigos. Não é utilizada somente a linguagem HTML para desenvolver aplicações, mas ainda mais uma linguagem capaz de representar o visual de páginas, o CSS. Há também o JavaScript, que é responsável pelo comportamento da página, como por exemplo, o que irá ocorrer no momento que o usuário utilizar um botão.

Existem ainda as linguagens chamadas Linguagens Server-Side, ou seja, que são usadas na máquina servidora. Incluem o PHP, Python, Ruby, ASP, entre outras. Todo o funcionamento do sistema depende dessas linguagens, que dão a resposta para o usuário após fazer os cálculos necessários nos servidores.

Os códigos são separados em diversas camadas, a fim de que não se misturem. É como se o HTML fosse o esqueleto do site, através dele toda a estrutura de código é desenvolvida. É onde será indicado o que é um parágrafo, uma imagem, o que é um título e entre outros. O CSS é como a parte externa do corpo, é o que deixará o esqueleto mais vistoso. É através do CSS que o título ganha cores, são determinadas as largura das colunas, o tamanho do texto é ajustado. Dessa forma, o código HTML e o código CSS não se misturam.

## 4.6 WAMPserver

WAMPServer, de acordo com [Edivaldo Brito \(2013\)](#), é um software que propicia um ambiente de desenvolvimento Web em Sistemas Operacionais Windows. Através dele é possível administrar os bancos de dados com a ferramenta que faz parte da aplicação, o phpMyAdmin. Também pode-se conceber aplicações Web com PHP, banco de dados MySQL e Apache2.

O acrônimo WAMP são as primeiras letras de Windows, Apache, MySQL e PHP – Perl – Python. Através dele são instalados todos esses softwares e ainda outras aplicações como o phpMyAdmin e todas as opções para gerenciar e configurar, o que possibilita agilidade quando a necessidade é de desenvolver em ambiente Web no Windows.

Através do software é instalado tudo que é necessário para iniciar o desenvolvimento de aplicações Web e o uso é de entendimento extremamente fácil. Na barra de tarefas do sistema operacional fica disponibilizado, em um pequeno menu, todas as funcionalidades sempre prontas para serem utilizadas. O usuário pode sincronizar o servidor sem a necessidade de alterar os arquivos de configuração.

O WAMPServer dispensa o usuário da obrigação de ter que instalar e configurar cada uma das aplicações de um ambiente Web no Windows, deixando que ele se preocupe apenas em ter foco para criar, testar e usar as ferramentas.

É possível ter acesso às funcionalidades mais importantes do programa ao clicar no ícone WAMPServer. Também pode-se gerenciar os serviços do banco de dados MySQL e do Apache, além de dar acesso a todos os usuários ou somente acesso local, instalar e mudar releases do MySQL, PHP e Apache, gerenciar as configurações dos servidores, ficar online, off-line, acessar registros e arquivos.

### 4.6.1 Apache

Concebido em 1995 por Rob McCool, o servidor Apache ou HTTP Apache é o servidor Web livre existente que mais obteve êxito. Pode-se dizer que é um servidor Web muito popular e mais amplamente usado por usuários do Linux.

O Apache é encarregado por disponibilizar todos os recursos e páginas que podem ser acessadas pelo usuário, bem como qualquer servidor do tipo. Várias funcionalidades podem ser realizadas com o auxílio do Apache, como compras online, envio de e-mails, mensagens e várias outras. Com tudo isso, vale ressaltar que o Apache é distribuído sob a licença GNU, é gratuito e pode ser estudado e transformado por meio do código fonte por qualquer usuário (ALECRIM, 2006).

A Apache Software Foundation<sup>9</sup> é a encarregada pelo projeto, além de desenvolver e trabalhar com outras inovações de comunicação pela Web, processar dados e executar aplicativos distribuídos. Segundo Alecrim (2006), a expressão que dá nome ao servidor e a empresa responsável por ele faz menção à nação Apache, uma tribo de nativos índios americanos que de grande resistência e que detinha de táticas notáveis em combate. O sentido do nome caracteriza uma comparação à objeção da comunidade do projeto livre às investidas da ambição de algumas corporações privadas. O significado expõe também a solidez do servidor e a sua variedade de recursos e ferramentas que estão aptos a trabalhar com qualquer tipo de requisição feita na Web.

Sobre o nome, existem ainda dois outros motivos para o servidor ser denominado de Apache. Negada pela Fundação, a primeira é de que haveria uma relação do nome com o termo *patchy server*, que em inglês quer dizer algo como servidor remendado ou melhoria do servidor. Isso porque o programa foi criado através o código do servidor da já existente NCSA, onde foram incorporados vários patches. A segunda tem ligação com a apresentação de um sistema adjunto do Apache, o Tomcat. Na realidade, Tomcat é o nome de uma avião e Apache é o nome de um helicóptero de ataque, ambos americanos.

O servidor Apache tem compatibilidade com o protocolo HTTP. Suas aplicabilidades são preservadas por meio de uma estrutura de padrões, assim, através da API da aplicação, os usuários escrevem os próprios módulos. O software está disponível para Windows, OS/2, Novell Netware e outros sistemas do padrão POSIX, como o Unix e o Linux, onde é muito utilizado.

No tocante ao hardware para apoio ao servidor, tudo depende de como ele será utilizado. Os computadores ofertados pelo mercado são qualificados para colocar um servidor Apache em funcionamento, lembrando que tudo depende da aplicabilidade. Uma máquina com processamento básico e que não detém de muita de memória RAM já é capaz de rodar a aplicação sem maiores problemas em um ambiente empresarial pequeno.

## 4.7 Banco de Dados

A grande maioria dos softwares necessitam de armazenar os dados com os quais trabalham. Precisam persistir seus dados para inserções, exclusões e alterações dos mesmos,

<sup>9</sup> <<https://canaltech.com.br/o-que-e/internet/O-que-e-servidor-Apache/>>

futuras consultas, necessidades que são chamadas de CRUD. Esses dados devem se apresentar de maneira organizada para facilitar essas operações. Para isso, em sua grande maioria, os softwares utilizam-se dos chamados banco de dados. Esses, por sua vez, são administrados por softwares chamados de SGBD, que a tem como objetivo prover maneiras de modo transparente para que os usuários de um Banco de Dados possam realizar as operações de CRUD em suas aplicações (Ricardo Rezende, 2006).

A maioria dos SGBD existentes no mercado utiliza como modelo de estruturação e organização de seus dados o Modelo Relacional, que surgiu em 1970 como uma revolução dos bancos de dados (Rodrigo Almeida, 2008). É largamente utilizado até os tempos de hoje pelos SGBD. Nesse modelo, as informações são separadas e organizadas de acordo com suas correlações em tabelas que permitem relacionamento de registros (linhas de uma tabela). Assim, é possível realizar as operações de consultas, feitas através de atributos (colunas de uma tabela) chamados de chaves.

Uma das grandes vantagens do uso dos bancos de dados é a independência em relação as aplicações, ou seja, a existência banco de dados não está necessariamente atrelada a existência de uma aplicação. No contexto atual dos SGBD, existem tanto pagos quanto gratuitos.

Para este trabalho será utilizado o SGBD MySQL por ser gratuito, sem limitação de armazenamento e bem-conceituado entre os projetistas de aplicações Web. O MySQL será gerenciado pelo phpMyAdmin.

### 4.7.1 MySQL

O MySQL é um SGBD relacional, de código aberto, usado na maioria das aplicações gratuitas (PISA, 2012). Ele utiliza, assim como a maioria dos Sistemas Gerenciadores de bancos de dados, a linguagem SQL. Foi criado em 1995 pela a empresa pela empresa sueca MySQL AB, com o intuito de possibilitar ao usuário um banco de dados que apresentasse segurança, agilidade e ainda que necessitasse de requisitos de hardware com baixo custo. Hoje é desenvolvido pela Oracle Corporation, na qual apresenta uma versão gratuita, com algumas restrições de uso, e uma versão paga. É um SGBD multiplataforma, sendo compatível com os sistemas operacionais Windows e sistemas baseados em Linux.

O MySQL é usado na comunicação entre as aplicações e os dados, disponibilizando métodos de consulta, gravação e alteração dos dados que se apresentam sobre diversos tipos de campos. É um servidor que funciona por TCP, mas não por HTTP, em aplicações como o WAMPsServer. Na realidade é um servidor de banco de dados completamente a parte, não está conectado diretamente ao PHP ou Apache. Pode ser instalado até em um computador diferente de uma mesma rede ou de uma rede externa que seja acessível e fica em uma porta diferente do HTTP.

Não possui navegação por pastas como os servidores HTTP. Fica em uma porta diferente do Apache, que geralmente fica em portas como 80, 8000, 8080, 9000, ao passo que o MySQL, na maioria das vezes, fica na porta 3306.

Quando softwares ou linguagens dinâmicas conectam ao MySQL, na realidade ele somente manda comandos ao servidor MySQL via TCP, parecidos com os comandos de terminal. É possível instalar o banco em uma máquina e usar como cliente para acessar um servidor, podendo estar na mesma máquina ou não.

### 4.7.2 phpMyAdmin

phpMyAdmin é uma aplicação Web projetada em PHP para, pela Internet, gerenciar o MySQL. Por meio dele pode-se criar e deletar bases de dados, criar, deletar e alterar tabelas, inserir, remover e editar campos, executar códigos SQL e modificar campos chaves (HOSTNET, 2000). O phpMyAdmin é amplamente usado por desenvolvedores Web que precisam utilizar bases de dados. Geralmente, o phpMyAdmin é tratado como uma aplicação obrigatória em quase todas as hospedagens da Web, bem como em pacotes off-line, como o WAMPServer.

O phpmyadmin não é um servidor, mas sim um administrador escrito em HTML e PHP. Ou seja, ele é uma ferramenta e não é uma aplicação que deve estar acessível nem disponível para os usuários. Apenas o gerenciador do banco de dados e, por vezes, por quem persiste os scripts é que podem ter acesso à ferramenta. Não é um banco dados, não integra o MySQL, é apenas uma aplicação que simplifica a administração do banco de dados e existem outras opções para isso.

É um programa que roda em conjunto com o PHP e Apache e aplicações como PHP não se conectam a ele. Bem como uma página de Web comum que usa banco de dados, a única ligação do phpmyadmin com o MySQL é que ele se relaciona ao banco no momento de efetivar *queries*.

Os recursos fornecidos pelo programa incluem<sup>10</sup>:

1. Ativar consultas de monitor (processos);
2. Carregar arquivos de texto nas tabelas;
3. Comunicar-se em 54 línguas diferentes.
4. Criar consultas complexas usando QBE, criar e arrastar bancos de dados, criar e ler dumps nas tabelas, criar gráficos PDF no layout do banco de dados;
5. Criar, copiar, arrastar, renomear e alterar tabelas;

<sup>10</sup> <<https://pt.wikipedia.org/wiki/PhpMyAdmin>>

6. Deletar, editar e adicionar campos;
7. Executar qualquer declaração SQL, até mesmo queries em lote;
8. Exporta dados para vários formatos: CSV, SQL, XML, PDF, ISO/IEC 26300 - Abre documentos de texto e planilha, Word, Excel, LaTeX e outros;
9. Fazer manutenção nas tabelas;
10. Gerenciar chaves nos campos, vários servidores, dados MySQL, usuários MySQL e seus privilégios;
11. Interface da Web;
12. Importar dados de CSV e SQL;
13. Realizar buscas a nível mundial em um banco de dados ou em subparte dele;
14. Suporte a `mysql`, a extensão MySQL otimizada;
15. Suporte a tabelas InnoDB e teclas estrangeiras;
16. Transformar dados armazenados para qualquer formato usando funções pré-definidas;
17. Usar QBE, criar queries complexas que conectam automaticamente as tabelas requeridas;

## 4.8 Sublime Text

O usuário desenvolvedor de aplicações necessita de um editor de código fonte que seja acessível e eficiente para manter a produtividade diária. Um editor de texto que tem se tornado a alternativa preferida por muitos desenvolvedores é o Sublime Text, por ser leve, simples e com interface agradável (João Otávio, 2016). Com tantas ferramentas compactas e gratuitas, como por exemplo o Eclipse, o Sublime Text se evidencia exatamente por ser simples e descomplicado. Ademais, é rápido, customizável e compatível com várias linguagens de programação.

Um editor de texto intuitivo e muito fácil de utilizar, foi desenvolvido em Python e projetado para ser claro e ter inteligibilidade. Além do que, segundo João Otávio (2016), podem ser adicionados muitos recursos e funcionalidades para complementar o uso.

A versão de avaliação pode ser utilizada pelo tempo que se deseja, a avaliação gratuita, até o momento, não possui limitação. A instalação é realizada de forma padrão e não é preciso fazer nenhum tipo de configuração complementar, é necessário apenas seguir os passos da instalação normalmente (João Otávio, 2016).

Alguns dos recursos incluem<sup>11</sup>:

- Autocompletar e correspondência de parênteses;
- Edição multi-panel;
- Minimap: uma pré-visualização de todo o código-fonte;
- Personalização da coloração das sintaxes;
- Possibilidade de selecionar várias partes do código;
- Salvamento automático;
- Suporte a macros e extensões baseadas em Python;
- Teclas de atalho personalizáveis.

## 4.9 GitHub

O GitHub surgiu pois, além de gostar de desenvolver, muitos desenvolvedores procuram trocar informações sobre o assunto sobre o qual estão desenvolvendo, bem como compartilhar informações e ainda alguns códigos.

Antes da descrição de alguns dos recursos e funcionalidades do GitHub, é preciso entender o que é o Git. Nada mais é do que um sistema de controle de versão compartilhado. A princípio foi concebido por Linus Torvalds, utilizado no desenvolvimento do Kernel do Linux (Rafaela Pozzebom, 2015). Isso se deve ao fato da busca de Torvalds por algo que fosse relativamente ágil para tratar com uma série progressiva de procedimentos relacionados ao Linux. Segundo Rafaela Pozzebom (2015), sem sucesso em algo que pudesse ser útil, Linus desenvolveu o Git. De acordo com a Git Wiki, o nome refere-se a uma gíria para *cabeça dura, que acha sempre certo, briguento*.

Por meio do Git pode-se trabalhar continuamente em um mesmo diretório, gravar comentários e a documentação, e ainda fazer alterações de projeto. Assim sendo, o Git está qualificado a guardar tudo o que é projetado. Este registro do que foi feito é muito vantajoso para que o desenvolvedor tenha a possibilidade de voltar a uma versão salva do projeto caso ocorra algum problema. Ainda, por meio do Git tem-se a possibilidade de criar divisões para executar testes e até mesmo diferentes projetos. Poder trabalhar em equipe é mais um ponto forte do sistema.

---

<sup>11</sup> <<http://imasters.com.br/box/ferramenta/sublimetext-2/>>

## 4.10 Windows 10

A versão mais atual de sistemas operacionais (SO) da Microsoft é o Windows 10<sup>12</sup>. O Software, que é multiplataforma, pode ser baixado e instalado em computadores por meio de arquivos ISO ou pelo Windows Update. Pode ser instalado também em dispositivos móveis, como smartphones e tablets, por meio do Windows 10 Mobile.

A versão que é oferecida para computadores, Windows 10 ou Windows 10 Pro, junta o design inovador do Windows 8 e 8.1 com a interface clássica do Windows 7 (Edivaldo Brito, 2016). Propicia um ambiente eclético, preparado para adequar-se a telas de diferentes proporções e excelente para uso com mouse e teclado, como o clássico desktop.

As mudanças que podem ser destacadas, com relação às versões anteriores, são o menu Iniciar de volta, múltiplos ambientes de trabalho, navegador novo, renovação de aplicativos e junção de múltiplas plataformas da Microsoft. A empresa afirmou que oferecerá a nova versão tanto para quem faz uso do sistema original tanto para que utiliza a versão pirata. Mas esses últimos não terão acesso a direito de suporte.

O Windows 10 é fruto de um trabalho da Microsoft associado com usuários e uma sincera autocrítica das versões de prévias de avaliação. A consequência é um software com ótimas funcionalidades e que proporciona uma agradável experiência de uso.

Mesmo sem oferecer uma funcionalidade diferente, o navegador Edge sobressai por propiciar uma chance inovadora a um navegador da Microsoft, uma vez que o Explorer perdeu espaço e o apreço dos usuários com o passar dos anos (Edivaldo Brito, 2016).

Uma das competências importantes do Windows 10, são as funcionalidades que o SO trouxe de outros sistemas como a Central de Notificação, as múltiplas áreas de trabalho, bem como outras mudanças. Essa especificidade do sistema novo possibilita aos usuários o acesso a um maior número de funcionalidades, sem a necessidade de utilizar outros SO.

Porém, em comparação com outros sistemas, o Windows 10 ainda deixa a desejar no quesito personalização da interface. O Sistema até possibilita um certo grau de alteração do layout padrão, mas é ínfimo perante a concorrência. Ao usuário que aprecia uma apresentação diferente, somente resta o uso de aplicações de terceiros, que podem apresentar inconsistências e, até então, não são compatíveis com o novo sistema.

O Windows 10 obteve análises positivas, em grande parte. Desde a data de lançamento em julho de 2015, os críticos enalteceram a decisão da Microsoft de retirar alguns processos pertencentes ao Windows 8. Segundo Edivaldo Brito (2016), também foi elogiada a inclusão de um ambiente mais focado no uso em computadores desktop. Porém, a interface de touch do Windows 10 foi criticada por ter regredido quando comparada a interface

---

<sup>12</sup> <<http://www.techtudo.com.br/tudo-sobre/windows-10.html>>

touch do Windows 8. Os críticos elogiaram também os avanços do software integrado com o Windows 10 em comparação ao 8.1, o acréscimo da associação com o Xbox Live, bem como a substituição do Internet Explorer pelo Edge. Ademais, os críticos desaprovam o fato de o navegador ser incorporado ao Sistema e ainda estar em desenvolvimento.

#### 4.10.1 Prós

- Cortana pode modificar a experiência de uso do SO<sup>13</sup> no desktop.
- No quesito navegadores, Edge é a nova possibilidade da Microsoft.
- O Sistema traz uma ótima experiência de uso e tem boas funções.
- Possui ótimas aplicações trazidas de outros sistemas.

#### 4.10.2 Contras

- Com relação às opções de personalização da interface ainda deixa a desejar<sup>14</sup>.

---

<sup>13</sup> <<http://www.techtudo.com.br/tudo-sobre/windows-10.html>>

<sup>14</sup> <<http://www.techtudo.com.br/tudo-sobre/windows-10.html>>



# 5 Análise de Requisitos

Este capítulo apresenta uma descrição geral do Sistema, e logo em seguida descreve as funcionalidades especificando as entradas e saídas para todos os requisitos funcionais. Faz também uma descrição sucinta dos requisitos não funcionais contidos neste Sistema.

## 5.1 Propósito do Sistema

O programa LoboSig foi idealizado para ser um sistema que efetua cadastros e gerencia os dados de alunos, assim sendo, através dele é possível realizar o acompanhamento de notas e frequências dos alunos cadastrados, por exemplo. Somente os usuários administradores terão permissão para cadastrar, editar, excluir, consultar e imprimir dados. Apenas usuários previamente cadastrados no sistema poderão ter acesso, por meio de login, a suas funcionalidades, sendo que há níveis de permissões diferentes de acesso ao programa. Para isso, o software será dividido em três painéis diferentes:

1. **Administrador:** Poderá ter acesso a todas as funcionalidades do sistema, que serão melhores descritas na próxima subseção;
2. **Professor:** Poderá acessar o sistema para visualizar quais são as turmas em que lecionará, bem como os respectivos dias e horários das aulas. Também poderá lançar as notas e frequências dos alunos matriculados em suas turmas.
3. **Aluno:** Poderá visualizar em quais turmas está matriculado, os dias e horários, a sala onde fará aula e também poderá ver as notas e frequências.

### 5.1.1 Funcionalidades do sistema

- **Gerenciamento de Dados Cadastrais:** O usuário administrador poderá cadastrar, editar, excluir, consultar e imprimir os dados cadastrados do banco de dados.
- **Gerenciamento de Usuários:** O usuário administrador poderá cadastrar, editar e excluir usuários cadastrados no sistema.
- **Gerenciamento de Notas e Frequências:** O usuário professor poderá cadastrar e editar as notas e frequências dos alunos.
- **Visualização de Dados:** Os usuários devidamente cadastrados poderão ter acesso aos painéis do sistema para visualizar os dados, cada um com sua respectiva permissão, seja de administrador, professor ou aluno.

## 5.2 Perspectiva do Sistema

O Sistema LoboSig opera com o banco de dados detendo o controle ao acesso às informações, é instalado em uma máquina que disponibiliza o serviço Web através do software Apache.

Foram feitos testes com o sistema rodando nas plataforma Windows e Linux, nas quais ele funcionou corretamente. Não foram realizados testes com o LoboSig instalado em ambiente macOS.

Também foram feitos testes com o programa em diferentes tipos de navegadores de Internet, dentre eles o Google Chrome, Microsoft Edge, Opera Mini e Firefox, sendo que em todos eles o sistema apresentou-se o mesmo, com todas as funcionalidades de cadastro, edição, exclusão, impressão e visualização em perfeito funcionamento.

### 5.2.1 Restrições Gerais

O Sistema LoboSig não permitirá o acesso aos dados por pessoas não cadastradas no sistema.

## 5.3 Requisitos de Software: Definições e Abreviações

A elaboração de um projeto de software é a junção de várias etapas de desenvolvimento, não apenas técnico, bem como de metodologia e organização. Compreender o que é de suma importância, fundamentado nos problemas que o software terá que resolver, faz a diferença entre um sistema de sucesso e outro obsoleto.

Faz-se necessário estabelecer padrões a fim de que o levantamento dos requisitos de um projeto de software seja voltado para maximizar a utilização dos recursos. O propósito desse tipo de procedimento é buscar, localizar, classificar, organizar e registrar as variáveis que constituirão um sistema.

A execução dessa estratégia resulta na descrição de quais serão os requisitos funcionais e quais requisitos não serão. Compreende-se um requisito como um estado, habilidade ou capacidade onde a aplicação deve estar de acordo. Pode-se compreender melhor a diferença entre eles na descrição abaixo:

- **Requisito Funcional (RF):** De agora em diante entende-se RF como sendo um Requisito Funcional, que determina os elementos ou a finalidade de uma aplicação. Entende-se como finalidade o conjunto de entradas, saídas e desempenho comportamental que o software terá. O RF caracteriza, em termos de serviços e tarefas, o que o sistema pode fazer. Requisitos Funcionais podem ser a utilização de dados, operações, particularidades, especificidades, processamento de informações, dentre outras

aplicações específicas que indicam o que um programa poderá fazer ([WIKIPÉDIA, 2014](#)). De maneira abrangente, tal como estabelecido na engenharia de requisitos, os RF caracterizam resultados específicos da arquitetura de um sistema.

- **Requisito Não Funcional (RNF):** Os Requisitos Não Funcionais, de agora em diante entendidos como RNF, também são considerados requisitos de qualidade. São os requisitos que determinam limitações sobre a execução ou o projeto. Em contraste com os RF, os RNF estão ligados ao uso do sistema em no que diz respeito a disponibilidade, usabilidade, e especificam características gerais, tais como gastos, performance, infalibilidade e confiabilidade, bem como as tecnologias envolvidas caracterizam a arquitetura técnica de um sistema ([WIKIPÉDIA, 2014](#)).

O projeto de execução dos RF é detalhado na concepção do sistema. Já o projeto de execução de RNF é especificado na arquitetura da aplicação ([WIKIPÉDIA, 2014](#)). Muitas vezes, os RNF acabam gerando restrições aos RF.

Com o aumento gradativo do número de usuários de dispositivos móveis, RNF como portabilidade e flexibilidade fazem-se indispensáveis em aproximadamente todos os sistemas hospedados na Web. Esse raciocínio segue-se também no que tange a posição geográfica, interoperabilidade e alta disponibilidade, que também são integrantes do esquema de desenvolvimento de RNF ([VENTURA, 2016](#)).

### 5.3.1 Estrutura de um Requisito

Não existe um modelo definido sobre a essência de um requisito. Geralmente utiliza-se um modelo parecido, que possui peculiaridades nos campos. No presente trabalho contempla-se os campos mais relevantes, com a descrição de cada um à seguir:

- **Identificador:** Sufixo seguido de um identificador único. O sufixo geralmente utilizado são RF (Requisito Funcional) ou RNF (Requisito Não-Funcional) e o identificador numérico único, geralmente é composto de quatro dígitos.
- **Nome:** Nome curto do requisito, mas que possibilite entender bem o que ele faz apenas pelo nome.
- **Módulo:** Módulo ao qual o requisito pertence. Se for um sistema pequeno que não possua nenhum módulo, somente o próprio sistema, deve ser preenchido com N/A (não se aplica).
- **Descrição:** Descrição mais detalhada do requisito.
- **Entrada:** Quais serão os dados do requisito que serão cadastrados no Sistema.
- **Restrição:** Se os dados de entrada do requisito podem ser nulos ou não.

- **Observação:** Considerações a serem feitas a respeito dos dados pertencentes ao requisito.
- **Processo:** Evolução dos dados do requisito do sistema até o Banco de Dados.
- **Saída:** Resultado de uma operação com o requisito entregues ao usuário em forma de mensagem na tela ou relatórios do Sistema.

## 5.4 Requisitos Funcionais

### 5.4.1 Requisitos Funcionais - Painel Administrador

#### 5.4.1.1 Cadastro de Alunos

<b>Identificador:</b>	RF0001	
<b>Nome:</b>	Cadastro de dados de alunos no Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá cadastrar os alunos.	
<b>Entrada</b>	<b>Restrição</b>	<b>Observação</b>
Nome do Aluno	Não Nulo	Campo de preenchimento obrigatório
Número de matrícula	Não Nulo	Caracteres gerados aleatoriamente
Número da Identidade	Nulo	Campo não possui preenchimento obrigatório
Número do CPF	Nulo	Campo não possui preenchimento obrigatório
Data de Nascimento	Não Nulo	Campo de preenchimento obrigatório
Sexo	Não Nulo	Campo de preenchimento obrigatório
Naturalidade	Não Nulo	Campo de preenchimento obrigatório
Rua	Não Nulo	Campo de preenchimento obrigatório
Número	Não Nulo	Campo de preenchimento obrigatório
Bairro	Não Nulo	Campo de preenchimento obrigatório
Cidade	Não Nulo	Campo de preenchimento obrigatório
Estado	Não Nulo	Campo de preenchimento obrigatório
Nome da Mãe	Não Nulo	Campo de preenchimento obrigatório
Nome do Pai	Nulo	Campo não possui preenchimento obrigatório
Turno	Nulo	Campo não possui preenchimento obrigatório
Nome da escola	Nulo	Campo não possui preenchimento obrigatório
Ano Escolar	Nulo	Campo não possui preenchimento obrigatório
Observações	Nulo	Campo não possui preenchimento obrigatório
Número de Telefone	Nulo	Campo não possui preenchimento obrigatório
Número de Celular	Nulo	Campo não possui preenchimento obrigatório
E-mail	Nulo	Campo não possui preenchimento obrigatório
<b>Processo:</b>	O cadastro será incluído no Banco de Dados.	
<b>Saída:</b>	Confirmação de cadastro bem sucedido, caso contrário, alerta para preenchimento de algum campo que obrigatoriamente não possa ficar vazio.	

Tabela 5: Cadastro de Alunos - Painel Administrador.

## 5.4.1.2 Alteração do Cadastro de Alunos

<b>Identificador:</b>	RF0002	
<b>Nome:</b>	Alteração de cadastro de dados de alunos no Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Descrição:</b>	O usuário administrador entra com os novos dados no campo que deseja alterar e atualiza o Banco de Dados.	
<b>Entrada</b>	<b>Restrição</b>	<b>Observação</b>
Nome do Aluno	Não Nulo	Campo de preenchimento obrigatório
Número da Identidade	Nulo	Campo não possui preenchimento obrigatório
Número do CPF	Nulo	Campo não possui preenchimento obrigatório
Data de Nascimento	Não Nulo	Campo de preenchimento obrigatório
Sexo	Não Nulo	Campo de preenchimento obrigatório
Naturalidade	Não Nulo	Campo de preenchimento obrigatório
Rua	Não Nulo	Campo de preenchimento obrigatório
Número	Não Nulo	Campo de preenchimento obrigatório
Bairro	Não Nulo	Campo de preenchimento obrigatório
Cidade	Não Nulo	Campo de preenchimento obrigatório
Estado	Não Nulo	Campo de preenchimento obrigatório
Nome da Mãe	Não Nulo	Campo de preenchimento obrigatório
Nome do Pai	Nulo	Campo não possui preenchimento obrigatório
Turno	Nulo	Campo não possui preenchimento obrigatório
Nome da escola	Nulo	Campo não possui preenchimento obrigatório
Ano Escolar	Nulo	Campo não possui preenchimento obrigatório
Observações	Nulo	Campo não possui preenchimento obrigatório
Número de Telefone	Nulo	Campo não possui preenchimento obrigatório
Número de Celular	Nulo	Campo não possui preenchimento obrigatório
E-mail	Nulo	Campo não possui preenchimento obrigatório
<b>Processo:</b>	As alterações dos dados serão atualizadas no Banco de Dados.	
<b>Saída:</b>	Confirmação na tela de modificação bem sucedida, caso contrário, alerta para preenchimento de algum campo obrigatoriamente não possa ficar vazio.	

Tabela 6: Edição de dados de Alunos - Painel Administrador.

## 5.4.1.3 Exclusão do Cadastro de Aluno

<b>Identificador:</b>	RF0003
<b>Nome:</b>	Exclusão de cadastro de dados de alunos no Banco de Dados
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Clicar no botão de excluir cadastro do aluno desejado
<b>Descrição:</b>	O usuário administrador poderá excluir o cadastro de determinado aluno do Banco de Dados.
<b>Processo:</b>	Os dados serão excluídos do Banco de Dados.
<b>Saída:</b>	Confirmação na tela de exclusão realizada com sucesso.

Tabela 7: Exclusão de dados de Alunos - Painel Administrador.

## 5.4.1.4 Impressão de Matrícula do Aluno

<b>Identificador:</b>	RF0004
<b>Nome:</b>	Impressão de comprovantes de matrículas dos alunos cadastrados no sistema.
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Clicar no botão de imprimir comprovante de matrícula do aluno desejado
<b>Descrição:</b>	O usuário administrador poderá imprimir o comprovante de matrícula de determinado aluno cadastrado no Sistema.
<b>Processo:</b>	O Sistema gera uma página para impressão do comprovante de matrícula do aluno selecionado.
<b>Saída:</b>	Comprovante de matrícula.

Tabela 8: Impressão de Comprovante de Matrícula - Painel Administrador.

## 5.4.1.5 Cadastro de Funcionário

<b>Identificador:</b>	RF0005	
<b>Nome:</b>	Cadastro de dados de funcionários no Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá cadastrar os funcionários.	
<b>Entrada</b>	<b>Restrição</b>	<b>Observação</b>
Nome do Funcionário	Não Nulo	Campo de preenchimento obrigatório
Número de matrícula no Estado	Não Nulo	Campo de preenchimento obrigatório
Cargo	Não Nulo	Campo de preenchimento obrigatório
Número da Identidade	Não Nulo	Campo de preenchimento obrigatório
Número do CPF	Não Nulo	Campo de preenchimento obrigatório
Data de Nascimento	Não Nulo	Campo de preenchimento obrigatório
Rua	Não Nulo	Campo de preenchimento obrigatório
Número	Não Nulo	Campo de preenchimento obrigatório
Bairro	Não Nulo	Campo de preenchimento obrigatório
Cidade	Não Nulo	Campo de preenchimento obrigatório
Estado	Não Nulo	Campo de preenchimento obrigatório
Número de Telefone	Nulo	Campo não possui preenchimento obrigatório
Número de Celular	Nulo	Campo não possui preenchimento obrigatório
E-mail	Nulo	Campo não possui preenchimento obrigatório
<b>Processo:</b>	O cadastro será incluído no Banco de Dados.	
<b>Saída:</b>	Confirmação de cadastro bem sucedido, caso contrário, alerta para preenchimento de algum campo que obrigatoriamente não possa ficar vazio.	

Tabela 9: Cadastro de Funcionários - Painel Administrador.

## 5.4.1.6 Alteração de Cadastro de Funcionário

<b>Identificador:</b>	RF0006	
<b>Nome:</b>	Alteração do cadastro de dados de funcionários no Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Descrição:</b>	O usuário administrador entra com os novos dados no campo que deseja alterar e atualiza o Banco de Dados.	
<b>Entrada</b>	<b>Restrição</b>	<b>Observação</b>
Nome do Funcionário	Não Nulo	Campo de preenchimento obrigatório
Número de matrícula no Estado	Não Nulo	Campo de preenchimento obrigatório
Cargo	Não Nulo	Campo de preenchimento obrigatório
Número da Identidade	Não Nulo	Campo de preenchimento obrigatório
Número do CPF	Não Nulo	Campo de preenchimento obrigatório
Data de Nascimento	Não Nulo	Campo de preenchimento obrigatório
Rua	Não Nulo	Campo de preenchimento obrigatório
Número	Não Nulo	Campo de preenchimento obrigatório
Bairro	Não Nulo	Campo de preenchimento obrigatório
Cidade	Não Nulo	Campo de preenchimento obrigatório
Estado	Não Nulo	Campo de preenchimento obrigatório
Número de Telefone	Nulo	Campo não possui preenchimento obrigatório
Número de Celular	Nulo	Campo não possui preenchimento obrigatório
E-mail	Nulo	Campo não possui preenchimento obrigatório
<b>Processo:</b>	As alterações dos dados serão atualizadas no Banco de Dados.	
<b>Saída:</b>	Confirmação na tela de modificação bem sucedida, caso contrário, alerta para preenchimento de algum campo obrigatoriamente não possa ficar vazio.	

Tabela 10: Alteração de dados de Funcionários - Painel Administrador.

## 5.4.1.7 Exclusão do Cadastro de Funcionário

<b>Identificador:</b>	RF0007	
<b>Nome:</b>	Exclusão de cadastro de dados de funcionários no Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Entrada:</b>	Clicar no botão de excluir cadastro do funcionário desejado	
<b>Descrição:</b>	O usuário administrador poderá excluir o cadastro de determinado funcionário do Banco de Dados.	
<b>Processo:</b>	Os dados serão excluídos do Banco de Dados.	
<b>Saída:</b>	Confirmação na tela de exclusão realizada com sucesso.	

Tabela 11: Exclusão de dados de Funcionários - Painel Administrador.

## 5.4.1.8 Cadastro de Disciplina

<b>Identificador:</b>	RF0008	
<b>Nome:</b>	Cadastro de dados de disciplinas no Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá cadastrar os novas disciplinas.	
<b>Entrada</b>	<b>Restrição</b>	<b>Observação</b>
Nome da Disciplina	Não Nulo	Campo de preenchimento obrigatório
Nome do Curso	Não Nulo	Campo de preenchimento obrigatório
Ciclo	Não Nulo	Campo de preenchimento obrigatório
Fase	Não Nulo	Campo de preenchimento obrigatório
<b>Processo:</b>	O cadastro será incluído no Banco de Dados.	
<b>Saída:</b>	Confirmação de cadastro bem sucedido.	

Tabela 12: Cadastro de Disciplinas - Painel Administrador.

## 5.4.1.9 Alteração de Cadastro de Disciplina

<b>Identificador:</b>	RF0009	
<b>Nome:</b>	Alteração do cadastro de dados de disciplinas no Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Descrição:</b>	O usuário administrador entra com os novos dados no campo que deseja alterar e atualiza o Banco de Dados.	
<b>Entrada</b>	<b>Restrição</b>	<b>Observação</b>
Nome da Disciplina	Não Nulo	Campo de preenchimento obrigatório
Nome do Curso	Não Nulo	Campo de preenchimento obrigatório
Ciclo	Não Nulo	Campo de preenchimento obrigatório
Fase	Não Nulo	Campo de preenchimento obrigatório
<b>Processo:</b>	As alterações dos dados serão atualizadas no Banco de Dados.	
<b>Saída:</b>	Confirmação na tela de modificação bem sucedida.	

Tabela 13: Alteração de dados de Disciplinas - Painel Administrador.

## 5.4.1.10 Exclusão do Cadastro de Disciplina

<b>Identificador:</b>	RF0010
<b>Nome:</b>	Exclusão de cadastro de dados de disciplina no Banco de Dados
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Clicar no botão de excluir cadastro da disciplina desejada
<b>Descrição:</b>	O usuário administrador poderá excluir o cadastro de determinada disciplina do Banco de Dados.
<b>Processo:</b>	Os dados serão excluídos do Banco de Dados.
<b>Saída:</b>	Confirmação na tela de exclusão realizada com sucesso.

Tabela 14: Exclusão de dados de Disciplina - Painel Administrador.

## 5.4.1.11 Vincular Professor a uma Disciplina

<b>Identificador:</b>	RF0011
<b>Nome:</b>	Vincular Professor a uma Disciplina, ambos previamente cadastrados no Sistema.
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Nome do Professor, nome da disciplina.
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá vincular Professor a uma Disciplina.
<b>Processo:</b>	O vínculo realizado será incluído no banco de dados.
<b>Saída:</b>	Mensagem de confirmação de vínculo realizado com sucesso.

Tabela 15: Vincular Professor a Disciplina - Painel Administrador.

## 5.4.1.12 Exclusão do Vínculo Professor/Disciplina

<b>Identificador:</b>	RF0012
<b>Nome:</b>	Exclusão de Vínculo entre professor e disciplina do Banco de Dados
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Clicar no botão de excluir vínculo desejado
<b>Descrição:</b>	O usuário administrador poderá excluir o cadastro de determinado vínculo entre um professor e uma disciplina do Banco de Dados.
<b>Processo:</b>	Os dados serão excluídos do Banco de Dados.
<b>Saída:</b>	Confirmação na tela de exclusão realizada com sucesso.

Tabela 16: Exclusão de Vínculo - Painel Administrador.

## 5.4.1.13 Cadastro de nova Turma

<b>Identificador:</b>	RF0013	
<b>Nome:</b>	Cadastro de dados de turmas no Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá cadastrar as novas turmas.	
<b>Entrada</b>	<b>Restrição</b>	<b>Observação</b>
Nome da Turma	Não Nulo	Campo de preenchimento obrigatório
Nome do Professor	Não Nulo	Campo de preenchimento obrigatório
Identificação da Disciplina	Não Nulo	Campo de preenchimento obrigatório
Número da sala	Não Nulo	Campo de preenchimento obrigatório
Horário da Aula	Não Nulo	Campo de preenchimento obrigatório
Dia da Aula	Não Nulo	Campo de preenchimento obrigatório
<b>Processo:</b>	O cadastro será incluído no Banco de Dados.	
<b>Saída:</b>	Confirmação de cadastro bem sucedido.	

Tabela 17: Cadastro de Turmas - Painel Administrador.

## 5.4.1.14 Exclusão de Turma

<b>Identificador:</b>	RF0014	
<b>Nome:</b>	Exclusão de Turma do Banco de Dados	
<b>Módulo:</b>	Painel Administrador	
<b>Entrada:</b>	Clicar no botão de excluir turma desejada	
<b>Descrição:</b>	O usuário administrador poderá excluir o cadastro de determinada turma cadastrada no do Banco de Dados.	
<b>Processo:</b>	Os dados serão excluídos do Banco de Dados.	
<b>Saída:</b>	Confirmação na tela de exclusão realizada com sucesso.	

Tabela 18: Exclusão de Turma - Painel Administrador.

## 5.4.1.15 Consulta a Turmas

<b>Identificador:</b>	RF0015	
<b>Nome:</b>	Consulta a turmas cadastradas	
<b>Módulo:</b>	Painel Administrador	
<b>Entrada:</b>	Clicar no botão de Consulta da turma desejada	
<b>Descrição:</b>	O usuário administrador poderá consultar e visualizar os dados das turmas cadastradas no Sistema.	
<b>Processo:</b>	O Sistema redireciona o usuário para outra tela.	
<b>Saída:</b>	Visualização da tela referente à consulta efetuada e listagem dos alunos matriculados na referida Turma.	

Tabela 19: Exclusão de Vínculo - Painel Administrador.

## 5.4.1.16 Matricular Aluno

<b>Identificador:</b>	RF0016
<b>Nome:</b>	Matricular aluno
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Nome do Aluno
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá matricular os Alunos nas Turmas desejadas.
<b>Processo:</b>	A matrícula será incluída no banco de dados.
<b>Saída:</b>	Mensagem de confirmação de matrícula realizada com sucesso.

Tabela 20: Matrícula de Aluno - Painel Administrador.

## 5.4.1.17 Desmatricular Aluno

<b>Identificador:</b>	RF0017
<b>Nome:</b>	Desmatricular aluno
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Clicar no botão de desmatricular o Aluno desejado.
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá desmatricular determinado Aluno na Turma desejada.
<b>Processo:</b>	O Sistema desmatricula o aluno da turma
<b>Saída:</b>	Mensagem de confirmação de exclusão realizada com sucesso.

Tabela 21: Desmatricular Aluno - Painel Administrador.

## 5.4.1.18 Impressão Listagem de Alunos

<b>Identificador:</b>	RF0018
<b>Nome:</b>	Imprimir Listagem de Alunos
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Clicar no botão de Imprimir na Tela de Consulta de Turma.
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá imprimir a listagem de alunos da Turma desejada.
<b>Processo:</b>	O Sistema redireciona o usuário para a página de impressão do navegador de Internet utilizado.
<b>Saída:</b>	Listagem de Alunos por Turma, nome da Turma e do Professor pronta para impressão.

Tabela 22: Imprimir Listagem de alunos - Painel Administrador.

## 5.4.1.19 Impressão Lista de Presença

<b>Identificador:</b>	RF0019
<b>Nome:</b>	Imprimir Lista de Presença
<b>Módulo:</b>	Painel Administrador
<b>Entrada:</b>	Clicar no botão de Lista de Presença na Tela de Consulta de Turma.
<b>Descrição:</b>	Somente o usuário administrador do Sistema poderá imprimir a Lista de Presença referente a Turma desejada.
<b>Processo:</b>	O Sistema redireciona o usuário para outra tela.
<b>Saída:</b>	Visualização e impressão da tela referente à Lista de Presença dos alunos da referida Turma.

Tabela 23: Imprimir de Lista de presença - Painel Administrador.

## 5.4.2 Requisitos Funcionais - Painel Aluno

## 5.4.2.1 Visualização de Horários

<b>Identificador:</b>	RF0020
<b>Nome:</b>	Visualização de Horários
<b>Módulo:</b>	Painel Aluno
<b>Entrada:</b>	Clicar no botão Visualizar Horários na tela inicial do Painel Aluno.
<b>Descrição:</b>	O Aluno cadastrado no Sistema conseguirá visualizar os horários das aulas das Turmas nas quais está matriculado.
<b>Processo:</b>	Redirecionamento do usuário a outra tela.
<b>Saída:</b>	Visualização na tela de uma tabela com as Turmas e Horários das aulas do usuário.

Tabela 24: Visualização de horários - Painel Aluno.

## 5.4.2.2 Visualização de Notas e Frequência

<b>Identificador:</b>	RF0021
<b>Nome:</b>	Visualização de Notas e Frequências
<b>Módulo:</b>	Painel Aluno
<b>Entrada:</b>	Clicar no botão Visualizar Notas e Frequência na tela inicial do Painel Aluno.
<b>Descrição:</b>	O Aluno cadastrado no Sistema conseguirá visualizar as notas e frequências lançadas pelos Professores das Turmas onde ele está matriculado.
<b>Processo:</b>	Redirecionamento do usuário a outra tela.
<b>Saída:</b>	Visualização na tela uma tabela com as Turmas onde o Aluno está matriculado e suas respectivas Notas e Frequências.

Tabela 25: Visualização de Notas e Frequências - Painel Aluno.

### 5.4.3 Requisitos Funcionais - Painel Professor

#### 5.4.3.1 Visualização de Horários

<b>Identificador:</b>	RF0022
<b>Nome:</b>	Visualização de Horários
<b>Módulo:</b>	Painel Professor
<b>Entrada:</b>	Clicar no botão Visualizar Horários na tela inicial do Painel Professor.
<b>Descrição:</b>	O Professor cadastrado no Sistema conseguirá visualizar os horários das Turmas onde ele está vinculado.
<b>Processo:</b>	Redirecionamento do usuário a outra tela.
<b>Saída:</b>	Visualização na tela de uma tabela com as Turmas e Horários das aulas do usuário.

Tabela 26: Visualização de horários - Painel Professor.

#### 5.4.3.2 Lançamento de Notas e Frequência no Sistema

<b>Identificador:</b>	RF0023
<b>Nome:</b>	Lançamento de Notas e Frequências
<b>Módulo:</b>	Painel Professor
<b>Entrada:</b>	Clicar no botão Lançar Notas e Frequências na tela inicial do Painel Professor.
<b>Descrição:</b>	O Professor cadastrado no Sistema poderá lançar no Sistema notas e frequências dos alunos matriculados em sua Turma.
<b>Processo:</b>	Redirecionamento do usuário a outra tela.
<b>Saída:</b>	Visualização na tela de um formulário que possibilita o Professor escolher em qual Turma ele lançará as Notas e Frequências.

Tabela 27: Lançamento de Notas e Frequências - Painel Professor.

#### 5.4.3.3 Seleção de Turma para lançamento de Notas e Frequência

<b>Identificador:</b>	RF0024
<b>Nome:</b>	Seleção de Turma
<b>Módulo:</b>	Painel Professor
<b>Entrada:</b>	Seleção do nome da Turma desejada.
<b>Descrição:</b>	O Professor cadastrado no Sistema poderá cadastrar no Sistema notas e frequências dos alunos matriculados em sua Turma.
<b>Processo:</b>	Redirecionamento do usuário a outra tela.
<b>Saída:</b>	Mostra na tela uma tabela com a Turma escolhida, o nome dos alunos matriculados na turma e os campos para lançamento de notas e frequência dos referidos alunos.

Tabela 28: Seleção de Turmas - Painel Professor.

## 5.5 Requisitos Não Funcionais

### 5.5.1 Requisitos Organizacionais

#### 5.5.1.1 RNF01: Software

O SGBD utilizado será o MySQL 5.7.14. Este SGBD é gratuito e é destinado para uso não comercial.

#### 5.5.1.2 RNF02: Linguagem de Programação

O Sistema será feito PHP, HTML e CSS, que são, respectivamente, linguagens de programação, de marcação e de folha de estilos muito utilizadas no desenvolvimento de sites.

#### 5.5.1.3 RNF03:Hardware

A máquina que será a utilizada como servidora do banco de dados possui um processador Intel Core i3, 4Gb de memória RAM e 500Mb de HD.

### 5.5.2 Outros Requisitos

O Sistema foi testado em funcionamento em Sistemas Linux e Windows e apresentou o mesmo comportamento satisfatório em ambos. Será utilizado em máquinas que rodam Sistema Operacional Windows, através do servidor local que é o sistema Apache.

## 5.6 Atributos

### 1. Disponibilidade

O sistema deve estar sempre disponível, caso ocorra alguma interrupção, ele deve ser restaurado o mais rápido possível.

### 2. Segurança

Como o sistema funcionará em rede, ele deverá ser seguro para que pessoas não autorizadas não tenham acesso aos dados inseridos do Sistema. Para isso, somente usuários devidamente cadastrados, munidos de login e senha, conseguem acesso aos painéis do programa. Cada usuário possui um nível de permissão diferente, sendo que um usuário cadastrado como Aluno ou Professor não possui acesso ao Painel do Administrador e somente o Administrador tem acesso a todos os dados do Sistema.

O sistema de backup não existe se o programa for instalado em rede local. Caso funcione através da Web, por meio de um provedor de hospedagem, o mesmo oferece um serviço de backup dos dados.

### 3. Manutenção

A manutenção será feita por graduandos do Curso de Sistemas de Informação da UFVJM que queiram dar continuidade ao Projeto de Desenvolvimento do Sistema LoboSig em possíveis trabalhos futuros.

## 5.7 Diagrama de Caso de Uso

O propósito Diagrama de Caso de Uso é favorecer o diálogo entre o desenvolvedor e o usuário. Esse tipo de diagrama representa um contexto que demonstra as aplicabilidades do sistema pela ótica do cliente, que deve visualizar no diagrama as funções indispensáveis do sistema.

Segue o Diagrama de Caso de Uso do Sistema LoboSig:

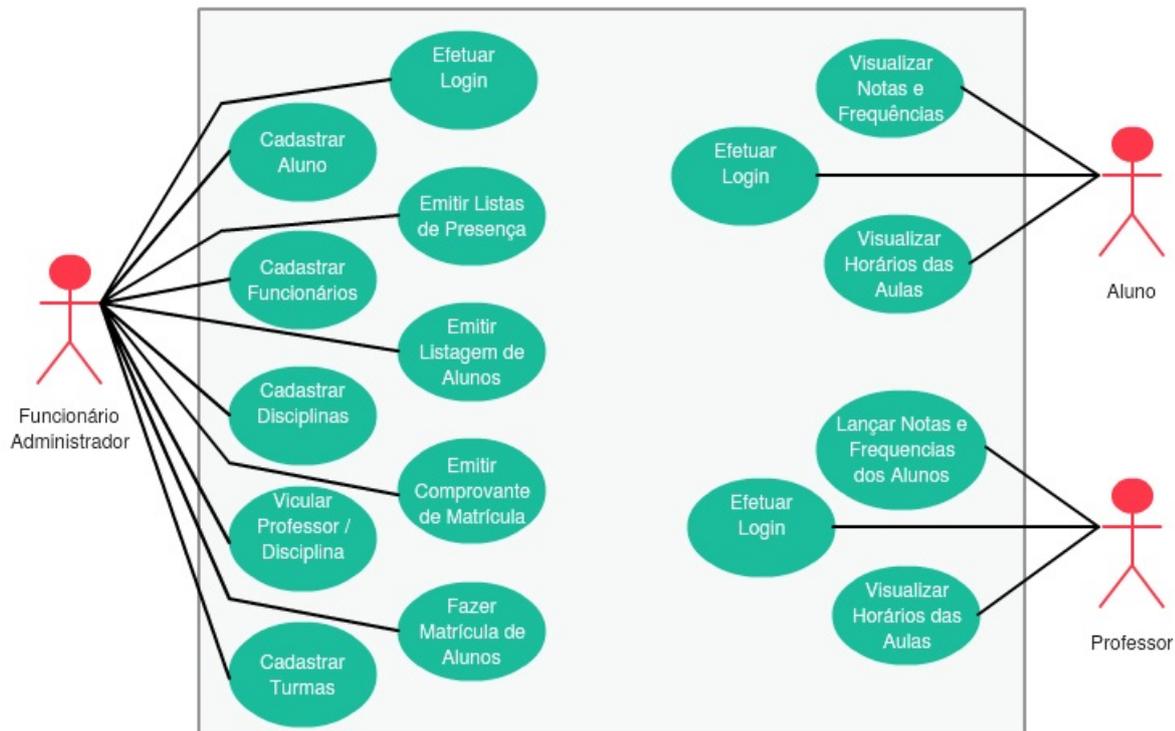


Figura 2: Diagrama de Caso de Uso do Sistema LoboSig

## 5.8 Diagrama de Classes

Diagrama de classes entende-se como uma representação da constituição e relacionamentos das classes que servem de modelo para objetos. De uma maneira mais simples, pode-se entender como um conjunto de objetos com características iguais, assim sendo, é possível identificar objetos e ordená-los de forma a encontrar suas próprias classes. Na Unified Modeling Language (UML), um retângulo representa uma classe com três divisões: O nome da classe, os atributos e os métodos.

Segue o Diagrama de Classes do Sistema LoboSig:

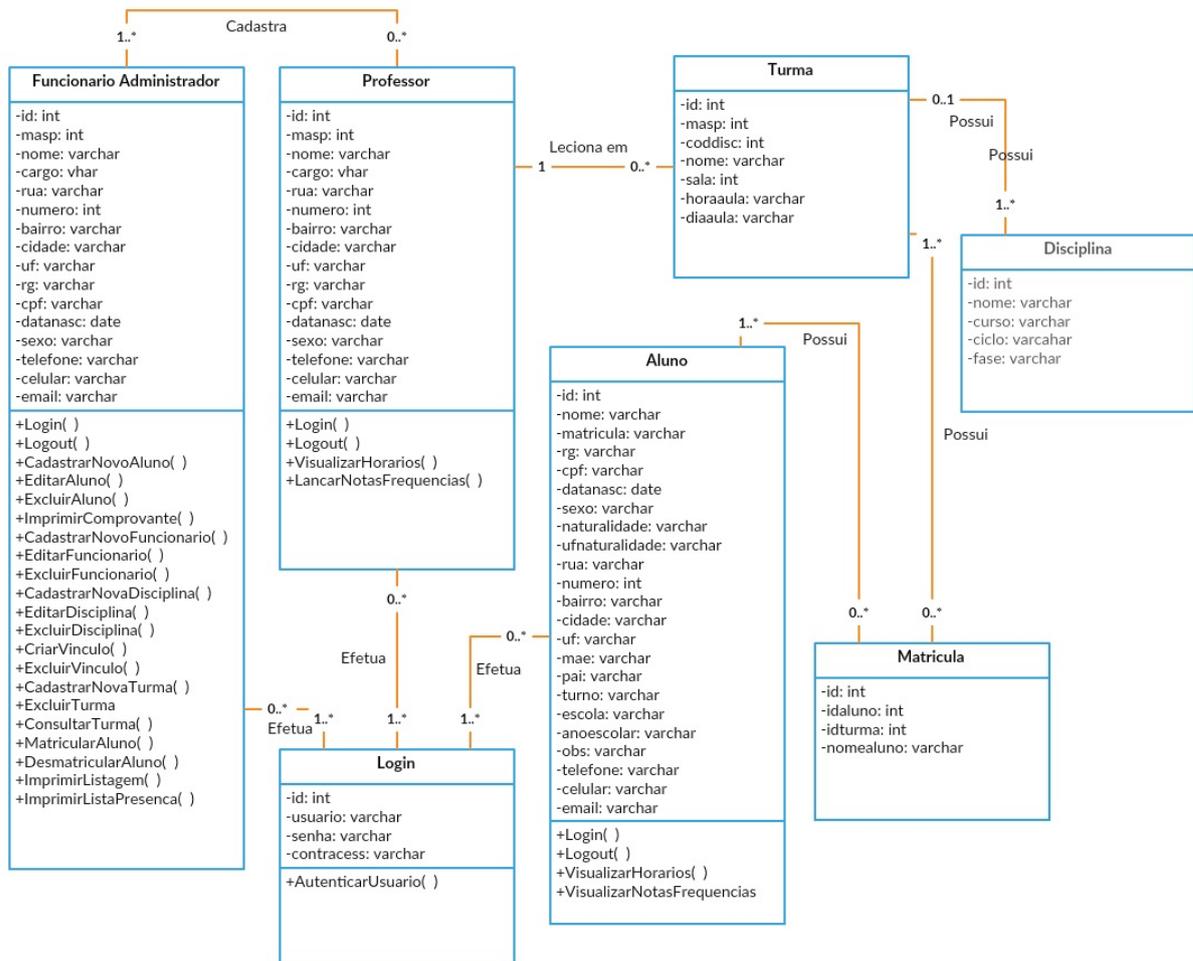


Figura 3: Diagrama de Classes do Sistema LoboSig

## 5.9 Modelo Entidade Relacionamento

No momento em que dá início ao processo e desenvolvimento de um novo software, ou até mesmo de uma função nova para uma aplicação que já existe, uma das primeiras etapas que deverá ser executada é a análise e levantamento dos requisitos que são indispensáveis para a elaboração do sistema. No processo de análise, percebe-se os objetos e elementos fundamentais envolvidos, as condutas e atribuições, bem como suas particularidades e como os objetos relacionam-se consigo mesmos (RODRIGUES, 2015).

Por meio dos dados recolhidos, é possível produzir um paradigma que será utilizado para direcionar a elaboração do sistema, propiciando dados a respeito das questões associadas ao âmbito do planejamento da aplicação.

A partir das informações obtidas, pode-se desenvolver um modelo conceitual que será utilizado para orientar o desenvolvimento, fornecendo informações sobre os aspectos relacionados ao domínio do projeto em questão.

Segundo (RODRIGUES, 2015), o Modelo Entidade Relacionamento é um paradigma da Engenharia de Software usado para representar as entidades que estão sendo utilizadas em uma área de negócio, com seus atributos e a maneira como é feito o relacionamentos entre esses atributos.

Genericamente, este modelo retrata de maneira subjetiva como será a organização do banco de dados do sistema. O banco de dados poderá possuir diferentes entidades, como por exemplo tabelas intermediárias e chaves, que só terão algum significado no âmbito de bases de dados relacionais.

Abaixo segue o Modelo Entidade Relacionamento do Sistema LoboSig:

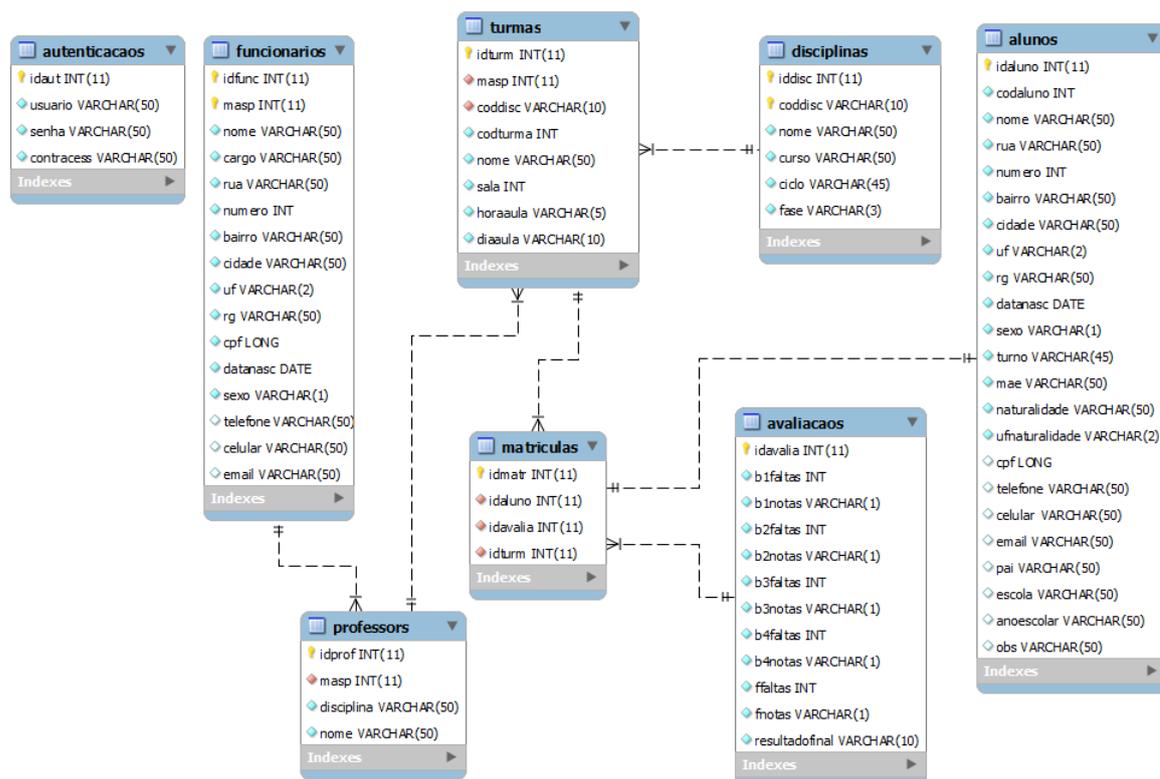


Figura 4: Relacionamentos das Tabelas do Banco de Dados do Sistema LoboSig



## 6 Desenvolvimento do Projeto

Para o Levantamento de Requisitos e coleta de dados, foram utilizadas técnicas de entrevista e questionário junto a diretoria e supervisão do CEM Lobo de Mesquita, visando a obtenção de informações para a criação do sistema. Além de contar com o levantamento de requisitos pronto, conta-se com a experiência *in situ* da analista na organização. O Sistema desenvolvido foi denominado LoboSig – Sistema de Gerenciamento de Dados do Conservatório Estadual de Música Lobo de Mesquita, por ter sido desenvolvido especialmente para atender às demandas do Conservatório de Diamantina. Feito com o propósito de auxiliar no gerenciamento de alunos e turmas, o funcionamento do Sistema seguirá a seguinte linha de raciocínio:

- A. Os alunos, funcionários e professores são cadastrados no sistema fornecendo seus dados pessoais e dados relacionados com a instituição de ensino.
- B. Para alunos, funcionários e professores cadastrados são fornecidas ao administrador funcionalidades de consultas dos dados dos mesmos.
- C. Além da funcionalidade de consulta, são fornecidas as funcionalidades de alteração e exclusão de alunos, funcionários e professores cadastrados, bem como disciplinas e turmas.
- D. As disciplinas oferecidas pelo conservatório são cadastradas fornecendo informações necessárias.
- E. Uma turma é cadastrada a partir da escolha e vinculação de uma disciplina e professor.
- F. Para as turmas cadastradas são cadastradas as aulas dessa turma (que vincula disciplina e professor).
- G. Os alunos cadastrados podem ser matriculados em turmas previamente cadastradas.
- H. Para alunos matriculados em turmas, fica disponível campos para lançamento, pelo professor, de notas e frequência do aluno, de forma bimestral.

Nesse capítulo será apresentado de forma detalhada o funcionamento do Sistema através de imagens. Em cada subseção será apresentada a imagem de uma respectiva tela do Sistema, uma tabela com a descrição dos campos presentes na tela e outra tabela com explicações das ações dos botões de links existentes.

## 6.1 Tela de Login no Sistema

Na Figura 5 pode ser visto o Painel de Login no Sistema, que dispõe de links para as demais painéis e suas funcionalidades.

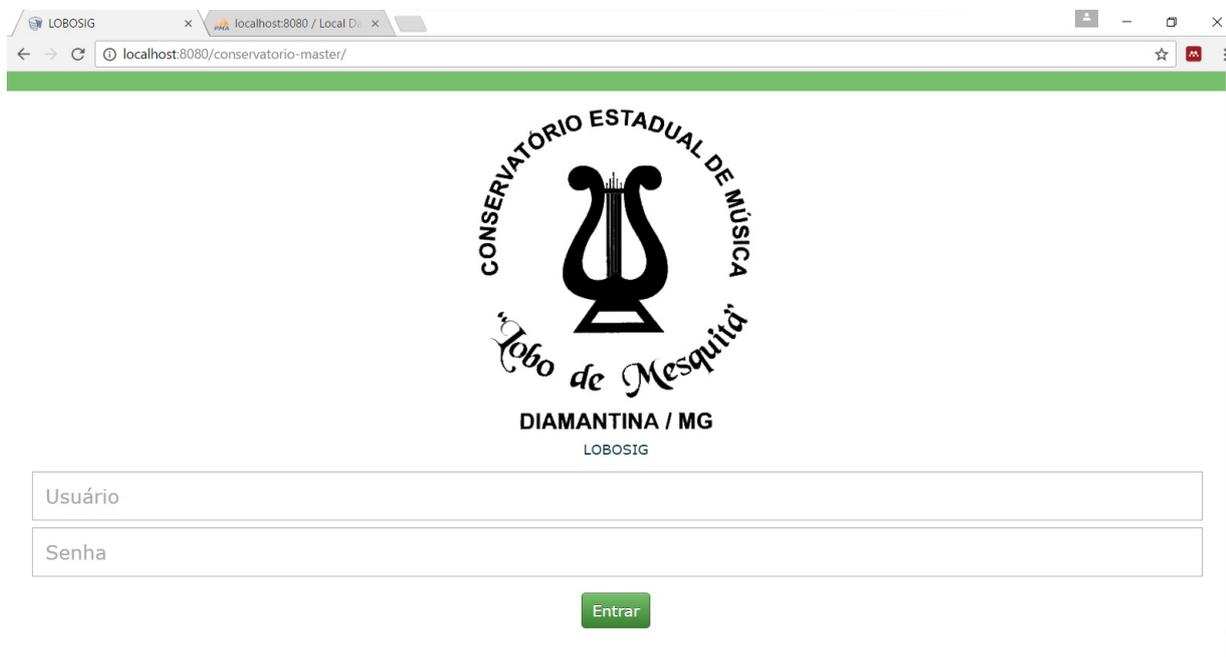


Figura 5: Tela de login do Sistema LoboSig.

Campo	Descrição	Tipos de Dados
Usuário	Campo para digitar nome do usuário que deseja efetuar o login	Caractere
Senha	Campo para digitar senha do usuário que deseja efetuar	Caractere

Tabela 29: Descrição dos campos – Tela de Login no Sistema.

Nome do Botão	Descrição
Entrar	Realiza o login do usuário no Sistema e redireciona para a tela do Painel Administrador, Aluno ou Professor, de acordo com o tipo de autenticação de usuário, sendo esse administrador, aluno ou professor

Tabela 30: Descrição das ações – Tela de Login no Sistema.

## 6.2 Tela Painel do Administrador

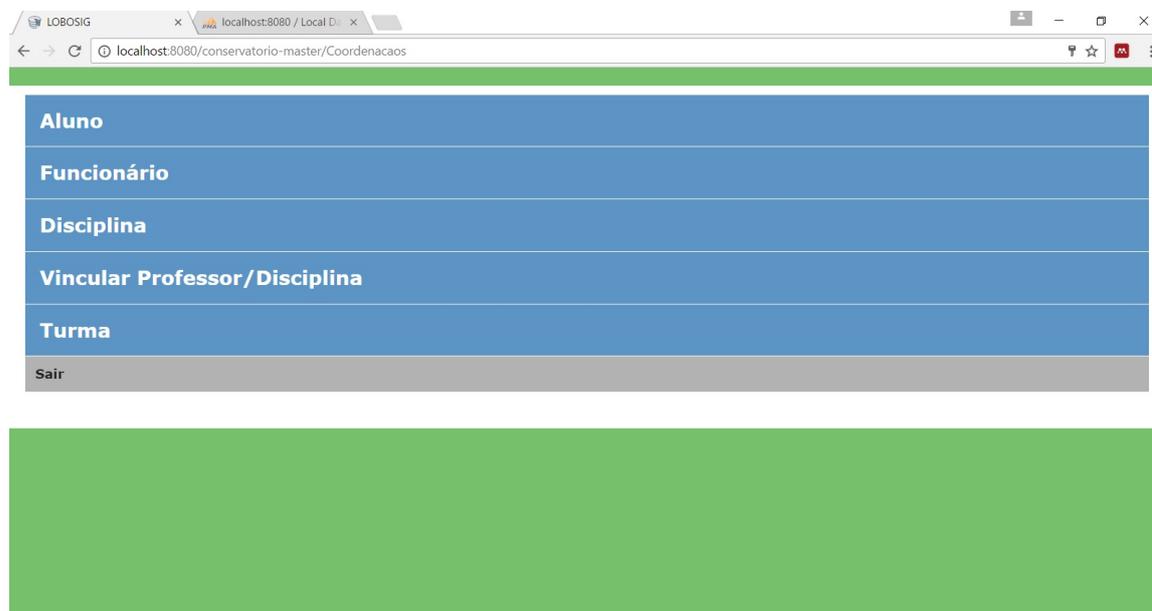


Figura 6: Tela Painel do Administrador.

Nome do Botão	Descrição
Aluno	Redireciona o usuário Administrador para a tela que exibe o nome dos alunos já cadastrados no Sistema
Funcionário	Redireciona o usuário Administrador para a tela que exibe o nome dos funcionários já cadastrados no Sistema
Disciplina	Redireciona o usuário Administrador para a tela que exibe o nome das disciplinas já cadastradas no Sistema
Vincular Professor/- Disciplina	Redireciona o usuário Administrador para a tela que exibe o vínculo entre determinado professor e disciplina já cadastrados no Sistema
Turma	Redireciona o usuário Administrador para a tela que exibe o nome das turmas já cadastradas no Sistema
Sair	Redireciona o usuário Administrador para a tela de login do Sistema

Tabela 31: Descrição das ações – Tela Painel do Administrador.

### 6.3 Tela de visualização de Alunos já cadastrados

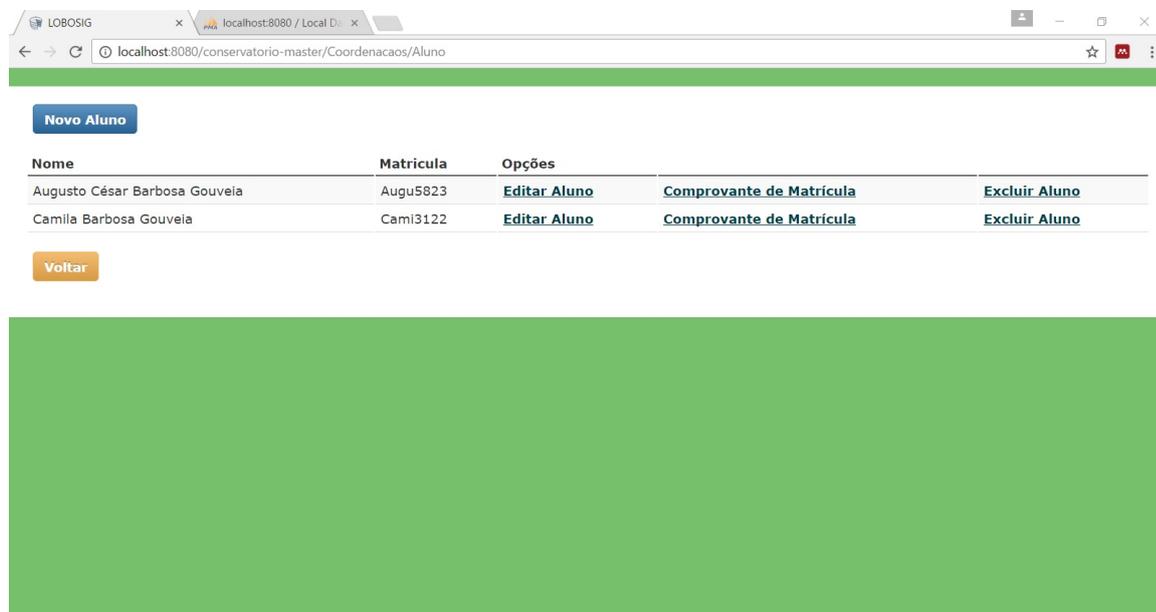


Figura 7: Tela de Visualização de Alunos já cadastrados.

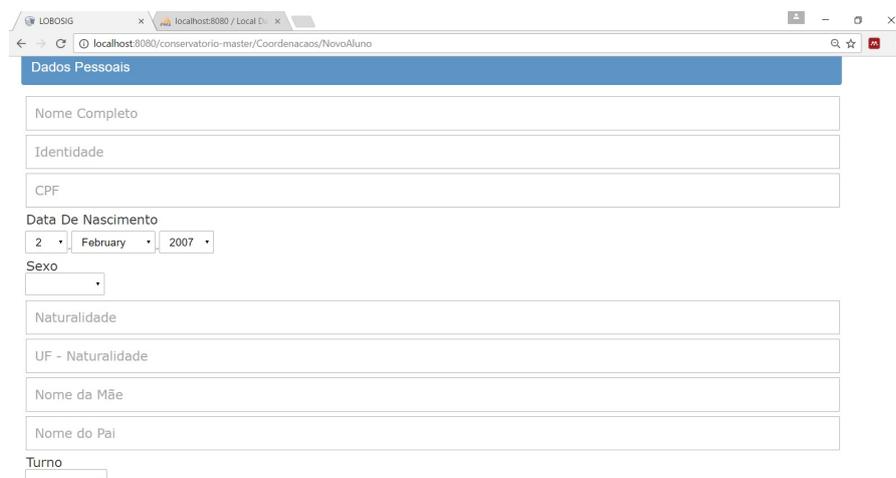
Campo	Descrição	Tipos de Dados
Nome	Nome do aluno cadastrado no Sistema	Caractere
Matrícula	Número de inscrição do aluno no Sistema	Caractere
Opções	Exibe botões de link da tela	

Tabela 32: Descrição dos campos – Tela Visualização de Alunos já cadastrados.

Nome do Botão	Descrição
Editar Aluno	Redireciona o usuário Administrador para a tela que exibe os campos para editar dados de Alunos já cadastrados
Comprovante de Matrícula	Redireciona o usuário Administrador para a tela que exibe o Comprovante de Matrícula do Aluno
Excluir Aluno	Exclui todos os dados do Aluno selecionado do banco de dados
Novo Aluno	Redireciona o usuário Administrador para a tela que exibe os campos para cadastrar dados de novos Alunos
Voltar	Redireciona o usuário Administrador para a tela inicial do Painel do Administrador

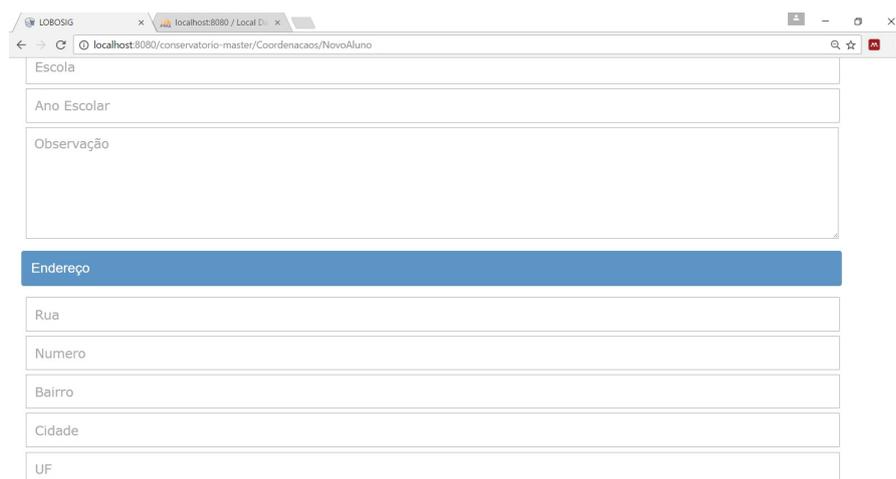
Tabela 33: Descrição das ações – Tela Visualização de Alunos já cadastrados.

## 6.4 Tela de cadastro de novo Aluno



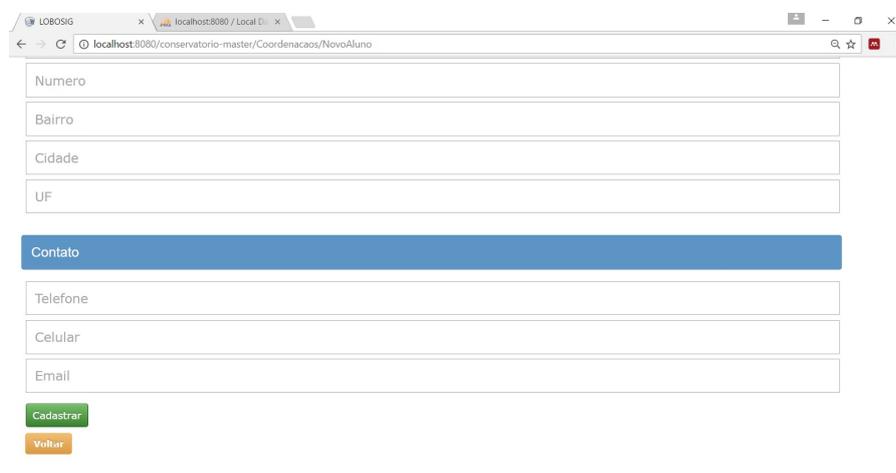
A screenshot of a web browser showing the 'Dados Pessoais' (Personal Data) registration form. The form is titled 'Dados Pessoais' in a blue header. It contains several input fields: 'Nome Completo', 'Identidade', 'CPF', 'Data De Nascimento' (with a date picker set to 2 February 2007), 'Sexo' (dropdown), 'Naturalidade', 'UF - Naturalidade', 'Nome da Mãe', 'Nome do Pai', and 'Turno' (dropdown).

Figura 8: Tela de cadastro de novo Aluno 1.



A screenshot of a web browser showing the 'Endereço' (Address) registration form. The form is titled 'Endereço' in a blue header. It contains several input fields: 'Escola', 'Ano Escolar', 'Observação', 'Rua', 'Numero', 'Bairro', 'Cidade', and 'UF'.

Figura 9: Tela de cadastro de novo Aluno 2.



A screenshot of a web browser showing the 'Contato' (Contact) registration form. The form is titled 'Contato' in a blue header. It contains several input fields: 'Numero', 'Bairro', 'Cidade', 'UF', 'Telefone', 'Celular', and 'Email'. At the bottom of the form, there are two buttons: 'Cadastrar' (green) and 'Voltar' (orange).

Figura 10: Tela de cadastro de novo Aluno 3.

<b>Campo</b>	<b>Descrição</b>	<b>Tipos de Dados</b>
Nome Completo	Nome completo do aluno a ser cadastrado no Sistema	Caractere
Identidade	Identidade do aluno a ser cadastrado no Sistema	Caractere
CPF	CPF do aluno a ser cadastrado no Sistema	Caractere
Data de Nascimento	Data de Nascimento do aluno a ser cadastrado no Sistema	Data
Sexo	Sexo do aluno a ser cadastrado no Sistema	Caractere
Naturalidade	Cidade de origem do aluno a ser cadastrado no Sistema	Caractere
UF - Naturalidade	Estado de origem do aluno a ser cadastrado no Sistema	Caractere
Nome da Mãe	Nome completo da mãe do aluno a ser cadastrado no Sistema	Caractere
Nome da Pai	Nome completo do pai do aluno a ser cadastrado no Sistema	Caractere
Turno	Turno no qual o aluno a ser cadastrado no Sistema irá estudar no CEM	Caractere
Escola	Nome da escola regular onde aluno a ser cadastrado no Sistema estuda	Caractere
Ano Escolar	Escolaridade do aluno a ser cadastrado no Sistema	Caractere
Observação	Observações que podem ser feitas a respeito do aluno a ser cadastrado no Sistema	Caractere
Rua	Nome da rua onde reside o aluno a ser cadastrado no Sistema	Caractere
Número	Número da casa onde reside o aluno a ser cadastrado no Sistema	Numérico
Bairro	Bairro onde reside o aluno a ser cadastrado no Sistema	Caractere
Cidade	Cidade onde reside o aluno a ser cadastrado no Sistema	Caractere
UF	Cidade onde reside o aluno a ser cadastrado no Sistema	Caractere
Telefone	Número de telefone do aluno (ou responsável) a ser cadastrado no Sistema	Caractere
Celular	Número de celular do aluno (ou responsável) a ser cadastrado no Sistema	Caractere
E-mail	Endereço de e-mail do aluno (ou responsável) a ser cadastrado no Sistema	Caractere

Tabela 34: Descrição dos campos – Tela de cadastro de novo Aluno - Dados Pessoais.

Nome do Botão	Descrição
Cadastrar	Realiza o cadastro do novo aluno e redireciona para a tela de visualizar alunos já cadastrados
Voltar	Redireciona o usuário Administrador para a tela de visualizar alunos já cadastrados

Tabela 35: Descrição das ações – Tela de cadastro de novo Aluno - Dados Pessoais.

## 6.5 Tela de edição de dados dos Alunos

A tela de edição de dados dos alunos possui os mesmos campos e botões de ação da tela de cadastro de novos alunos, porém os dados já aparecem preenchidos na tela quando o usuário clica no botão editar do aluno desejado.

Dados Pessoais

Nome  
Augusto César Barbosa Gouveia

RG  
MG0987655

CPF  
0987654321

Data de Nascimento  
10 January 1993

Sexo  
[ ]

Naturalidade  
Diamantina

UF - Naturalidade  
MG

Figura 11: Tela de edição de dados de Aluno.

<b>Campo</b>	<b>Descrição</b>	<b>Tipos de Dados</b>
Nome Completo	Nome completo do aluno a ser editado no Sistema	Caractere
Identidade	Identidade do aluno a ser editado no Sistema	Caractere
CPF	CPF do aluno a ser editado no Sistema	Caractere
Data de Nascimento	Data de Nascimento do aluno a ser editado no Sistema	Data
Sexo	Sexo do aluno a ser editado no Sistema	Caractere
Naturalidade	Cidade de origem do aluno a ser editado no Sistema	Caractere
UF - Naturalidade	Estado de origem do aluno a ser editado no Sistema	Caractere
Nome da Mãe	Nome completo da mãe do aluno a ser editado no Sistema	Caractere
Nome da Pai	Nome completo do pai do aluno a ser editado no Sistema	Caractere
Turno	Turno no qual o aluno a ser editado no Sistema irá estudar no CEM	Caractere
Escola	Nome da escola regular onde aluno a ser editado no Sistema estuda	Caractere
Ano Escolar	Escolaridade do aluno a ser editado no Sistema	Caractere
Observação	Observações que podem ser feitas a respeito do aluno a ser editado no Sistema	Caractere
Rua	Nome da rua onde reside o aluno a ser editado no Sistema	Caractere
Número	Número da casa onde reside o aluno a ser editado no Sistema	Numérico
Bairro	Bairro onde reside o aluno a ser editado no Sistema	Caractere
Cidade	Cidade onde reside o aluno a ser editado no Sistema	Caractere
UF	Cidade onde reside o aluno a ser editado no Sistema	Caractere
Telefone	Número de telefone do aluno (ou responsável) a ser editado no Sistema	Caractere
Celular	Número de celular do aluno (ou responsável) a ser cadastrado no Sistema	Caractere
E-mail	Endereço de e-mail do aluno (ou responsável) a ser editado no Sistema	Caractere

Tabela 36: Descrição dos campos – Tela de edição de dados de Aluno - Dados Pessoais.

Nome do Botão	Descrição
Atualizar	Atualiza os dados do aluno no banco de dados e redireciona para a tela de visualizar alunos já cadastrados
Voltar	Redireciona o usuário Administrador para a tela de visualizar alunos já cadastrados

Tabela 37: Descrição das ações – Tela de edição de dados de Aluno - Dados Pessoais.

## 6.6 Tela de impressão de Comprovante de Matrícula

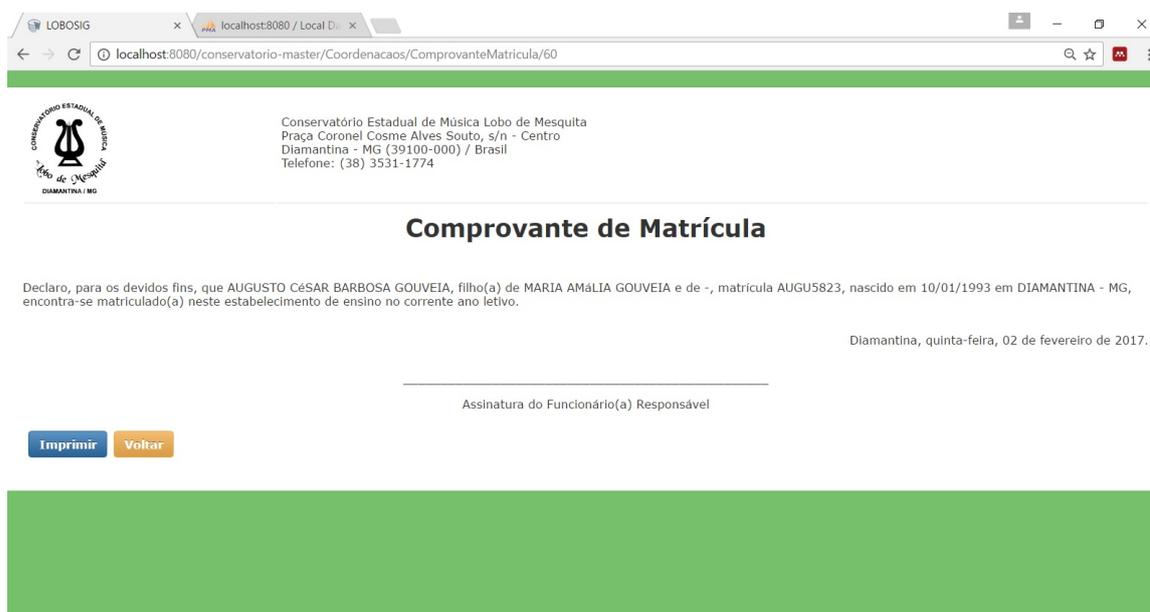


Figura 12: Tela de impressão do Comprovante de Matrícula.

Nome do Botão	Descrição
Imprimir	Redireciona o usuário Administrador para a tela de impressão do navegador utilizado
Voltar	Redireciona o usuário Administrador para a tela de visualizar alunos já cadastrados

Tabela 38: Descrição das ações – Tela de impressão do Comprovante de Matrícula.

## 6.7 Tela de visualização de Funcionários já cadastrados

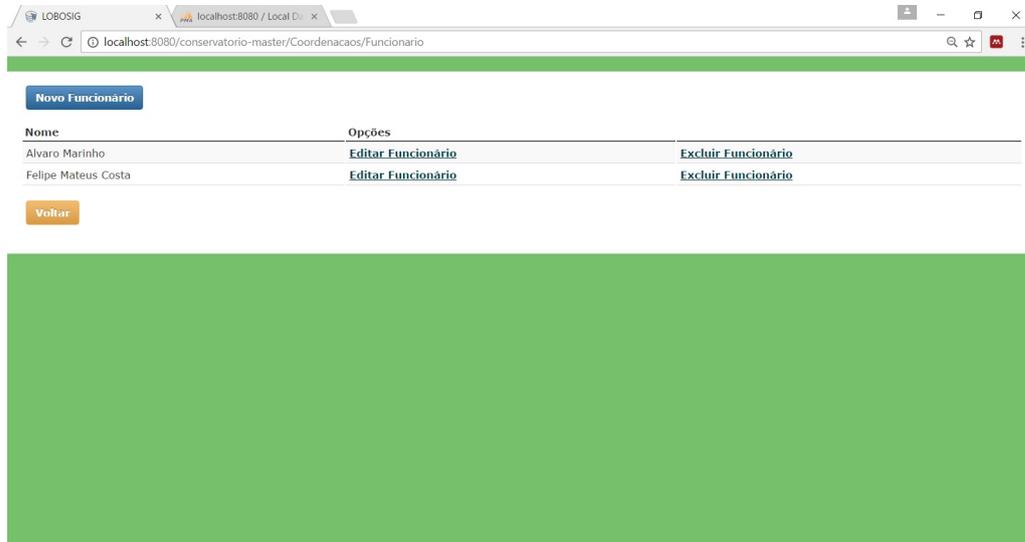


Figura 13: Tela de Visualização de Funcionários já cadastrados.

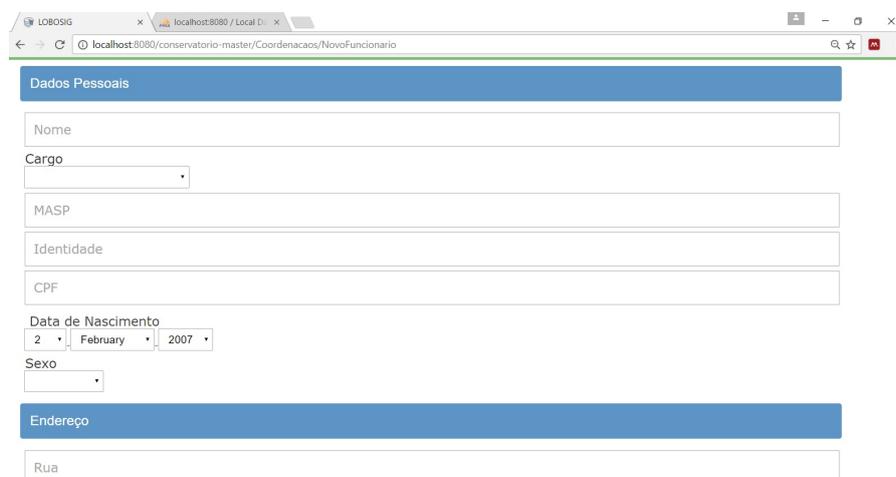
Campo	Descrição	Tipos de Dados
Nome	Nome do funcionário cadastrado no Sistema	Caractere
Opções	Exibe botões de link da tela	

Tabela 39: Descrição dos campos – Tela de Visualização de Funcionários já cadastrados.

Nome do Botão	Descrição
Editar Funcionário	Redireciona o usuário Administrador para a tela que exibe os campos para editar dados de Funcionários já cadastrados
Excluir Funcionário	Exclui todos os dados do Funcionário selecionado do banco de dados
Novo Funcionário	Redireciona o usuário Administrador para a tela que exibe os campos para cadastrar dados de novos Funcionários
Voltar	Redireciona o usuário Administrador para a tela inicial do Painel do Administrador

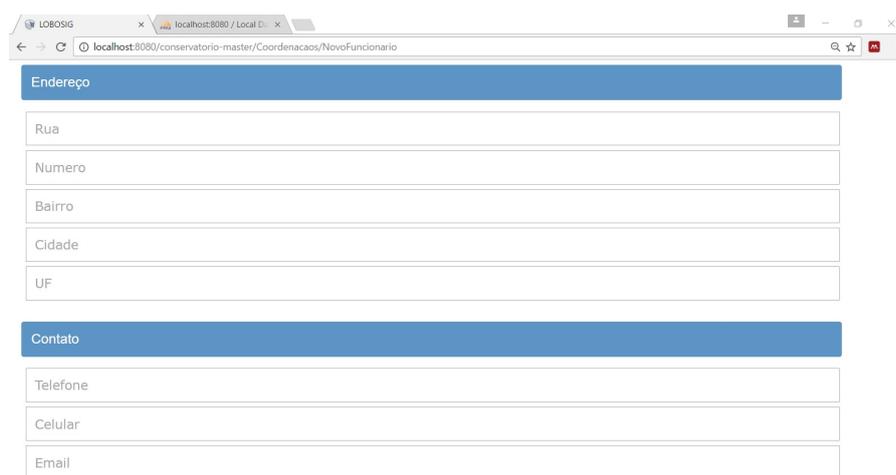
Tabela 40: Descrição das ações – Tela de Visualização de Funcionários já cadastrados.

## 6.8 Tela de cadastro de novo Funcionário



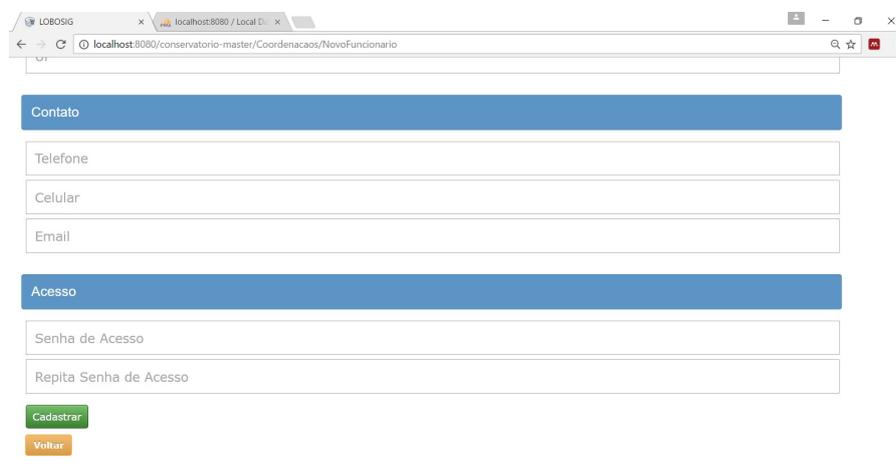
A screenshot of a web browser showing the 'Dados Pessoais' section of a registration form. The form includes the following fields: 'Nome' (text input), 'Cargo' (dropdown menu), 'MASP' (text input), 'Identidade' (text input), 'CPF' (text input), 'Data de Nascimento' (date picker with month 'February' and year '2007'), and 'Sexo' (dropdown menu). The section is titled 'Dados Pessoais' in a blue header bar.

Figura 14: Tela de cadastro de novo Funcionário 1.



A screenshot of a web browser showing the 'Endereço' and 'Contato' sections of a registration form. The 'Endereço' section includes fields for 'Rua', 'Numero', 'Bairro', 'Cidade', and 'UF'. The 'Contato' section includes fields for 'Telefone', 'Celular', and 'Email'. The sections are titled 'Endereço' and 'Contato' in blue header bars.

Figura 15: Tela de cadastro de novo Funcionário 2.



A screenshot of a web browser showing the 'Contato' and 'Acesso' sections of a registration form. The 'Contato' section includes fields for 'Telefone', 'Celular', and 'Email'. The 'Acesso' section includes fields for 'Senha de Acesso' and 'Repita Senha de Acesso'. At the bottom, there are two buttons: 'Cadastrar' (green) and 'Voltar' (orange). The sections are titled 'Contato' and 'Acesso' in blue header bars.

Figura 16: Tela de cadastro de novo Funcionário 3.

<b>Campo</b>	<b>Descrição</b>	<b>Tipos de Dados</b>
Nome Completo	Nome completo do funcionário a ser cadastrado no Sistema	Caractere
Cargo	Tipo de função que o funcionário desempenha na escola	Caractere
Masp	Número de Registro na Secretaria de Educação do funcionário a ser cadastrado no Sistema	Caractere
Identidade	Identidade do funcionário a ser cadastrado no Sistema	Caractere
CPF	CPF do funcionário a ser cadastrado no Sistema	Caractere
Data de Nascimento	Data de Nascimento do funcionário a ser cadastrado no Sistema	Data
Sexo	Sexo do funcionário a ser cadastrado no Sistema	Caractere
Rua	Nome da rua onde reside o funcionário a ser cadastrado no Sistema	Caractere
Número	Número da casa onde reside o funcionário a ser cadastrado no Sistema	Numérico
Bairro	Bairro onde reside o funcionário a ser cadastrado no Sistema	Caractere
Cidade	Cidade onde reside o funcionário a ser cadastrado no Sistema	Caractere
UF	Cidade onde reside o funcionário a ser cadastrado no Sistema	Caractere
Telefone	Número de telefone do funcionário a ser cadastrado no Sistema	Caractere
Celular	Número de celular do funcionário a ser cadastrado no Sistema	Caractere
E-mail	Endereço de e-mail do funcionário a ser cadastrado no Sistema	Caractere
Senha de Acesso	Usuário digita a senha escolhida para fazer login no Sistema	Caractere
Repete Senha de Acesso	Usuário redige a senha escolhida para fazer login no Sistema	Caractere

Tabela 41: Descrição dos campos – Tela de cadastro de novo Funcionário - Dados Pessoais.

<b>Nome do Botão</b>	<b>Descrição</b>
Cadastrar	Realiza o cadastro do novo funcionário e redireciona para a tela de visualizar funcionários já cadastrados
Voltar	Redireciona o usuário Administrador para a tela de visualizar funcionários já cadastrados

Tabela 42: Descrição das ações – Tela de cadastro de novo Funcionário - Dados Pessoais.

## 6.9 Tela de edição de dados dos Funcionários

A tela de edição de dados dos funcionários possui os mesmos campos e botões de ação da tela de cadastro de novos funcionários, porém os dados já aparecem preenchidos na tela quando o usuário clica no botão editar do funcionário desejado.

The screenshot shows a web browser window with the URL `localhost:8080/conservatorio-master/Coordenacao/EditarFuncionario/25`. The page displays a form for editing employee data. The form is divided into two main sections: 'Dados Pessoais' (Personal Data) and 'Endereço' (Address). The 'Dados Pessoais' section includes fields for 'Nome' (filled with 'Alvaro Marinho'), 'Cargo' (empty dropdown), 'MASP' (filled with '123456'), 'Data de Nascimento' (filled with '3 February 1990'), and 'Sexo' (empty dropdown). The 'Endereço' section includes fields for 'Rua' (filled with 'Rua Diamantina') and 'Numero' (empty). The browser's address bar and tabs are visible at the top.

Figura 17: Tela de edição de dados dos Funcionários.

Campo	Descrição	Tipos de Dados
Nome Completo	Nome completo do funcionário a ser editado no Sistema	Caractere
Cargo	Tipo de função que o funcionário desempenha na escola	Caractere
Masp	Número de Registro na Secretaria de Educação do funcionário a ser editado no Sistema	Caractere
Identidade	Identidade do funcionário a ser editado no Sistema	Caractere
CPF	CPF do funcionário a ser editado no Sistema	Caractere
Data de Nascimento	Data de Nascimento do funcionário a ser editado no Sistema	Data
Sexo	Sexo do funcionário a ser editado no Sistema	Caractere
Rua	Nome da rua onde reside o funcionário a ser editado no Sistema	Caractere
Número	Número da casa onde reside o funcionário a ser editado no Sistema	Numérico
Bairro	Bairro onde reside o funcionário a ser editado no Sistema	Caractere
Cidade	Cidade onde reside o funcionário a ser editado no Sistema	Caractere
UF	Cidade onde reside o funcionário a ser editado no Sistema	Caractere

Telefone	Número de telefone do funcionário a ser editado no Sistema	Caractere
Celular	Número de celular do funcionário a ser editado no Sistema	Caractere
E-mail	Endereço de e-mail do funcionário a ser editado no Sistema	Caractere
Senha de Acesso	Usuário digita a senha escolhida para fazer login no Sistema	Caractere
Repete Senha de Acesso	Usuário redige a senha escolhida para fazer login no Sistema	Caractere

Tabela 43: Descrição dos campos – Tela de edição de dados dos Funcionários - Dados Pessoais.

Nome do Botão	Descrição
Atualizar	Atualiza os dados do funcionário e redireciona para a tela de visualizar funcionários já cadastrados
Voltar	Redireciona o usuário Administrador para a tela de visualizar funcionários já cadastrados

Tabela 44: Descrição das ações – Tela de edição de dados dos Funcionários - Dados Pessoais.

## 6.10 Tela de visualização de Disciplinas já cadastradas

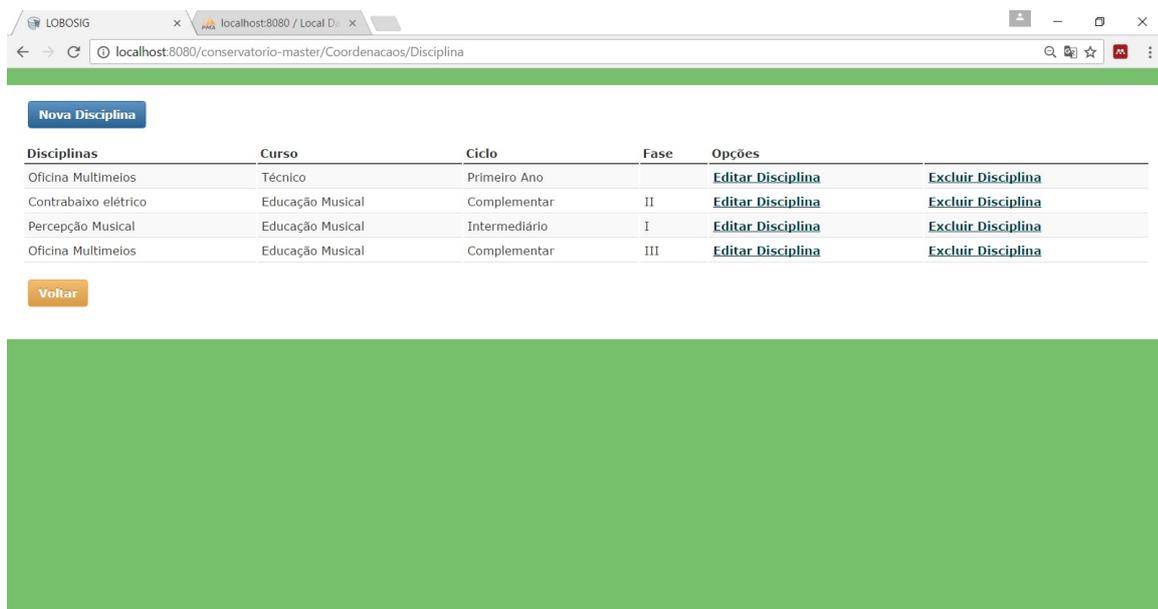


Figura 18: Tela de visualização de Disciplinas já cadastradas.

Campo	Descrição	Tipos de Dados
Disciplinas	Nome da Disciplina cadastrada no Sistema	Caractere
Curso	Curso ao qual a Disciplina cadastrada no Sistema pertence	Caractere
Ciclo	Ciclo ao qual a Disciplina cadastrada no Sistema pertence	Caractere
Fase	Fase a qual a Disciplina cadastrada no Sistema pertence	Caractere
Opções	Exibe botões de link da tela	

Tabela 45: Descrição dos campos – Tela de visualização de Disciplinas já cadastradas.

Nome do Botão	Descrição
Editar Disciplina	Redireciona o usuário Administrador para a tela que exibe os campos para editar dados de Disciplina já cadastradas
Excluir Disciplina	Exclui todos os dados da Disciplina selecionado do banco de dados
Nova Disciplina	Redireciona o usuário Administrador para a tela que exibe os campos para cadastrar dados de novas Disciplinas
Voltar	Redireciona o usuário Administrador para a tela inicial do Painel do Administrador

Tabela 46: Descrição das ações – Tela de visualização de Disciplinas já cadastradas.

## 6.11 Tela de cadastro de novas Disciplinas

The screenshot shows a web browser window with the following elements:

- Browser tabs: LOBOSIG, localhost:8080 / Local D...
- Address bar: localhost:8080/conservatorio-master/Coordenacaos/NovaDisciplina
- Page title: Nova Disciplina
- Form fields:
  - Disciplina (dropdown menu)
  - Curso (dropdown menu)
  - Ciclo - Ano (dropdown menu)
  - Fase (dropdown menu)
- Buttons:
  - Cadastrar (green button)
  - Voltar (orange button)

Figura 19: Tela de cadastro de novas Disciplinas.

Campo	Descrição	Tipos de Dados
Disciplina	Seleção da Disciplina a ser cadastrada no Sistema	Caractere
Curso	Seleção do curso ao qual a Disciplina será cadastrada no Sistema	Caractere
Ciclo - Ano	Seleção do ciclo ao qual a Disciplina será cadastrada no Sistema	Caractere
Fase	Seleção da fase a qual a Disciplina será cadastrada no Sistema	Caractere

Tabela 47: Descrição dos campos – Tela de cadastro de novas Disciplinas.

Nome do Botão	Descrição
Cadastrar	Realiza o cadastro da nova disciplina e redireciona para a tela de visualizar Disciplinas já cadastradas
Voltar	Redireciona o usuário Administrador para a tela de visualizar Disciplinas já cadastradas

Tabela 48: Descrição das ações – Tela de cadastro de novas Disciplinas.

## 6.12 Tela de edição de Disciplinas

A tela de edição de dados das disciplinas possui os mesmos campos e botões de ação da tela de cadastro de novas disciplinas.



Figura 20: Tela de edição de Disciplinas.

Campo	Descrição	Tipos de Dados
Disciplinas	Nome da Disciplina a ser editada no Sistema	Caractere
Curso	Curso ao qual pertence a Disciplina a ser editado no Sistema	Caractere
Ciclo	Ciclo ao qual pertence a Disciplina a ser editado no Sistema	Caractere
Fase	Fase a qual pertence a Disciplina a ser editada no Sistema	Caractere

Tabela 49: Descrição dos campos – Tela de edição de Disciplinas.

Nome do Botão	Descrição
Atualizar	Atualiza os dados da disciplina e redireciona para a tela de visualizar disciplina já cadastrados
Voltar	Redireciona o usuário Administrador para a tela de visualizar disciplinas já cadastrados

Tabela 50: Descrição das ações – Tela de edição de Disciplinas.

## 6.13 Tela de visualização de Vínculos Professor/Disciplina

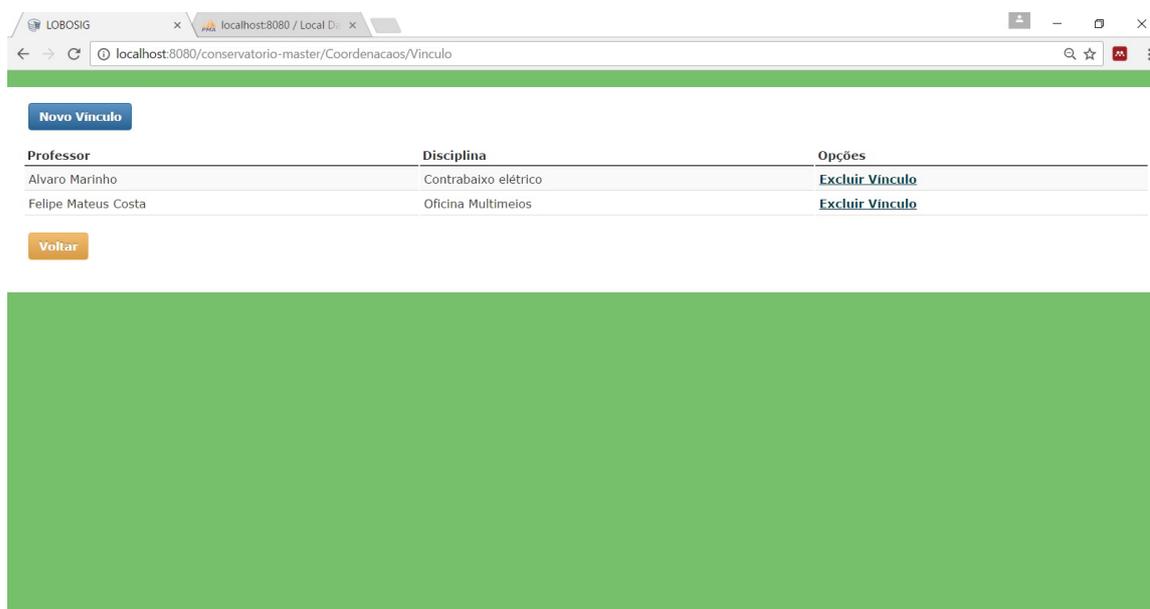


Figura 21: Tela de visualização de Vínculos.

Campo	Descrição	Tipos de Dados
Professor	Nome do professor vinculado a determinada disciplina	Caractere
Disciplina	Nome da disciplina vinculada a determinado professor	Caractere
Opções	Exibe botões de link da tela	

Tabela 51: Descrição dos campos – Tela de visualização de Vínculos.

Nome do Botão	Descrição
Excluir Vínculo	Exclui o vínculo existente entre um professor e uma disciplina
Novo Vínculo	Redireciona o usuário Administrador para a tela que exibe os campos para vincular professor e disciplina
Voltar	Redireciona o usuário Administrador para a tela inicial do Painel do Administrador

Tabela 52: Descrição das ações – Tela de visualização de Vínculos.

## 6.14 Tela vincular Professor/Disciplina

Figura 22: Tela vincular Professor/Disciplina.

Campo	Descrição	Tipos de Dados
Professor	Seleção do professor que será vinculado a uma disciplina	Caractere
Disciplina	Seleção da Disciplina que será vinculada a um professor	Caractere

Tabela 53: Descrição dos campos – Tela vincular Professor/Disciplina.

Nome do Botão	Descrição
Cadastrar	Realiza o novo vínculo entre professor e disciplina já cadastrados no Sistema
Voltar	Redireciona o usuário Administrador para a tela de visualizar Vínculos

Tabela 54: Descrição das ações – Tela vincular Professor/Disciplina.

## 6.15 Tela de visualização de Turmas já cadastradas

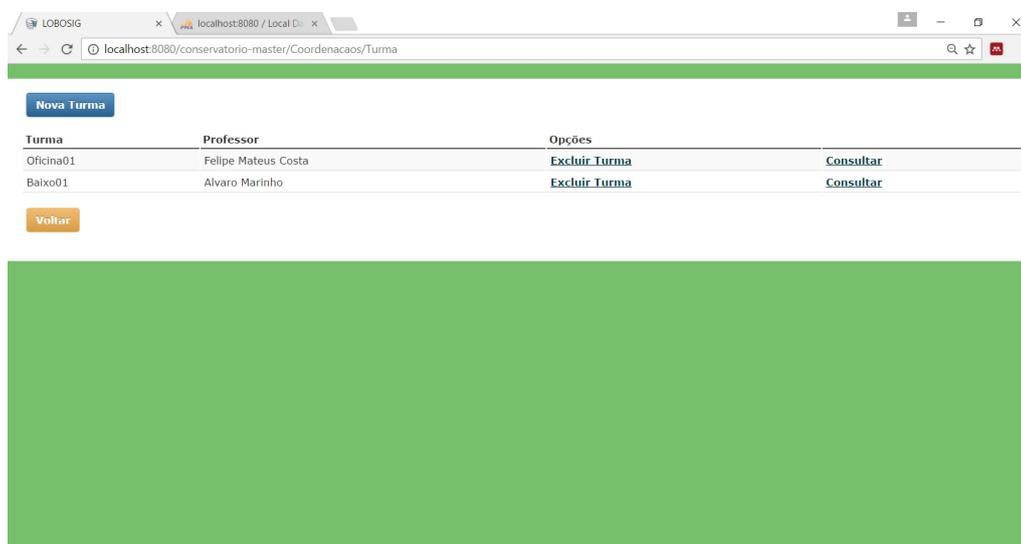


Figura 23: Tela de visualizar Turmas já cadastradas.

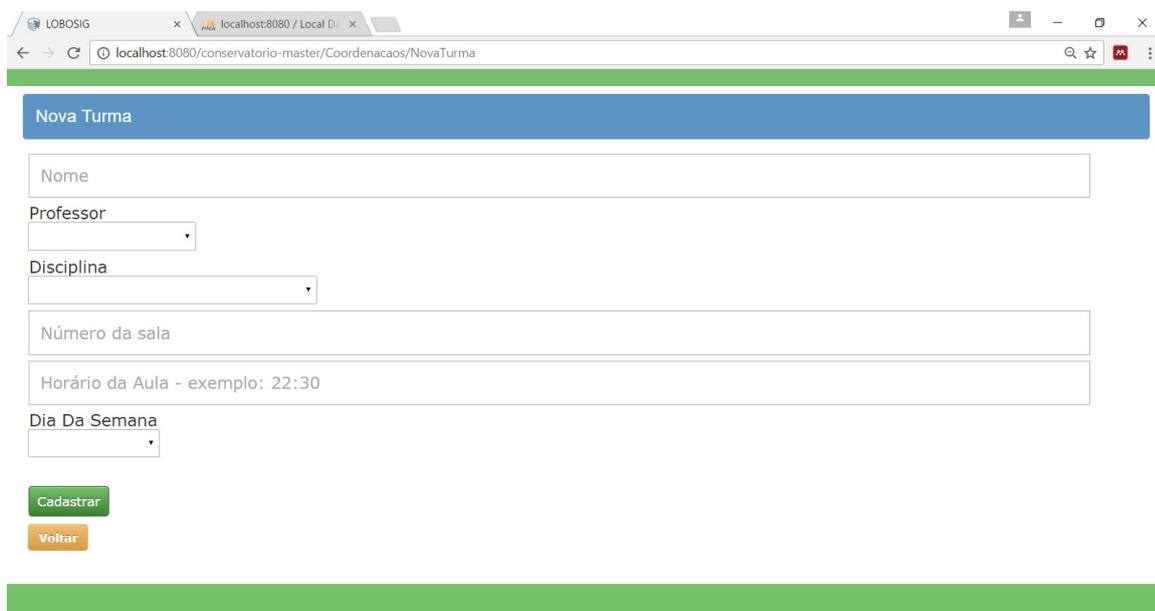
Campo	Descrição	Tipos de Dados
Turma	Nome da turma cadastrada no Sistema	Caractere
Professor	Nome no professor que possui o vínculo na turma	Caractere
Opções	Exibe botões de link da tela	

Tabela 55: Descrição dos campos – Tela de visualizar Turmas já cadastradas.

Nome do Botão	Descrição
Excluir Turma	Exclui todos os dados da Turma selecionada do banco de dados
Cosulta	Redireciona o usuário Administrador para a tela de exibição dos alunos matriculados em determinada turma
Nova Turma	Redireciona o usuário Administrador para a tela que exibe os campos para cadastrar dados de novas Turmas
Voltar	Redireciona o usuário Administrador para a tela inicial do Painel do Administrador

Tabela 56: Descrição das ações – Tela de visualizar Turmas já cadastradas.

## 6.16 Tela de cadastro de nova Turma



The screenshot shows a web browser window with the URL `localhost:8080/conservatorio-master/Coordenacoes/NovaTurma`. The page has a blue header bar with the text "Nova Turma". Below the header, there is a form with the following fields:

- Nome: A text input field.
- Professor: A dropdown menu.
- Disciplina: A dropdown menu.
- Número da sala: A text input field.
- Horário da Aula - exemplo: 22:30: A text input field.
- Dia Da Semana: A dropdown menu.

At the bottom of the form, there are two buttons: a green "Cadastrar" button and an orange "Voltar" button.

Figura 24: Tela de cadastro de nova Turma.

Campo	Descrição	Tipos de Dados
Nome	Nome da nova Turma a ser cadastrada no Sistema	Caractere
Professor	Seleção do professor que será vinculado a referida Turma	Caractere
Disciplina	Seleção da disciplina que será vinculada a referida Turma	Caractere
Número da Sala	Número da sala em que ocorrerá a aula com a referida Turma	Número
Horário da Aula	Horário em que a aula em que ocorrerá com a referida Turma	Caractere
Dia da Semana	Seleção do dia da semana em que ocorrerá a aula com a referida Turma	Caractere

Tabela 57: Descrição dos campos – Tela de cadastro de nova Turma.

Nome do Botão	Descrição
Cadastrar	Realiza o cadastro da nova Turma e redireciona para a tela de visualizar Turmas já cadastradas
Voltar	Redireciona o usuário Administrador para a tela de visualizar Turmas já cadastradas

Tabela 58: Descrição das ações – Tela de cadastro de nova Turma.

## 6.17 Tela de visualização de alunos já matriculados

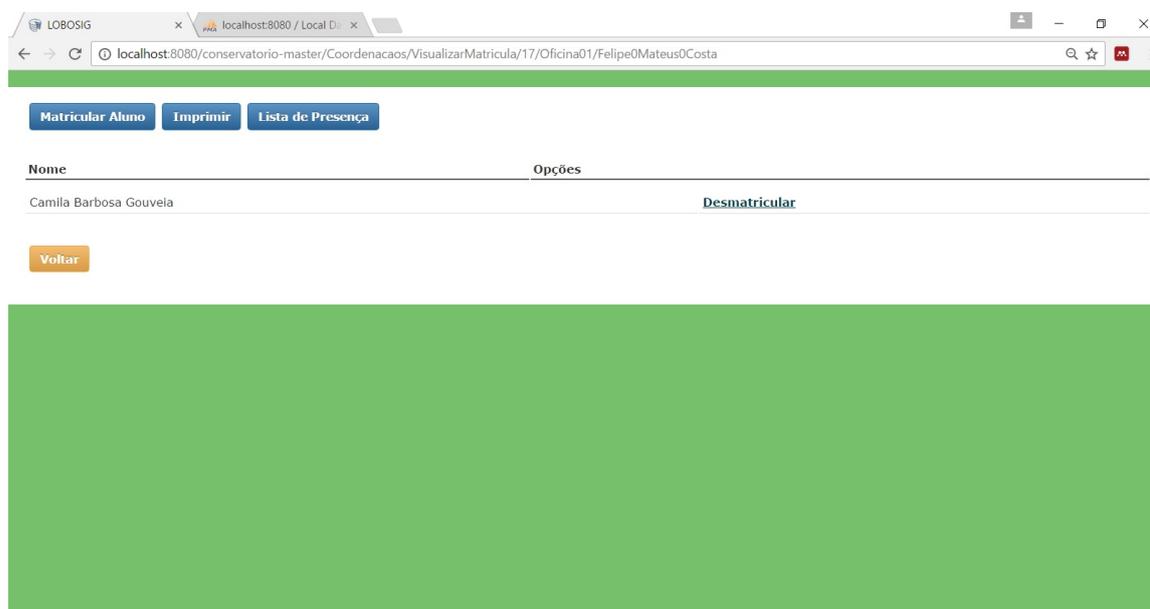


Figura 25: Tela de visualização de alunos já matriculados.

Campo	Descrição	Tipos de Dados
Nome	Nomes dos alunos matriculados na referida turma	Caractere
Opções	Exibe botões de link da tela	

Tabela 59: Descrição dos campos – Tela de visualização de alunos já matriculados.

Nome do Botão	Descrição
Desmatricular	Desmatricula o aluno da referida turma em que ele está matriculado
Matricular Aluno	Redireciona o usuário Administrador para a tela de matricular alunos na turma desejada
Imprimir	Redireciona o usuário Administrador para a página de impressão do navegador utilizado para imprimir a lista com os alunos matriculados
Lista de Presença	Redireciona o usuário Administrador para a tela que exibe a Lista de Presença
Voltar	Redireciona o usuário Administrador para a tela de visualização das turmas

Tabela 60: Descrição das ações – Tela de visualização de alunos já matriculados.

## 6.18 Tela de realização de Matrícula

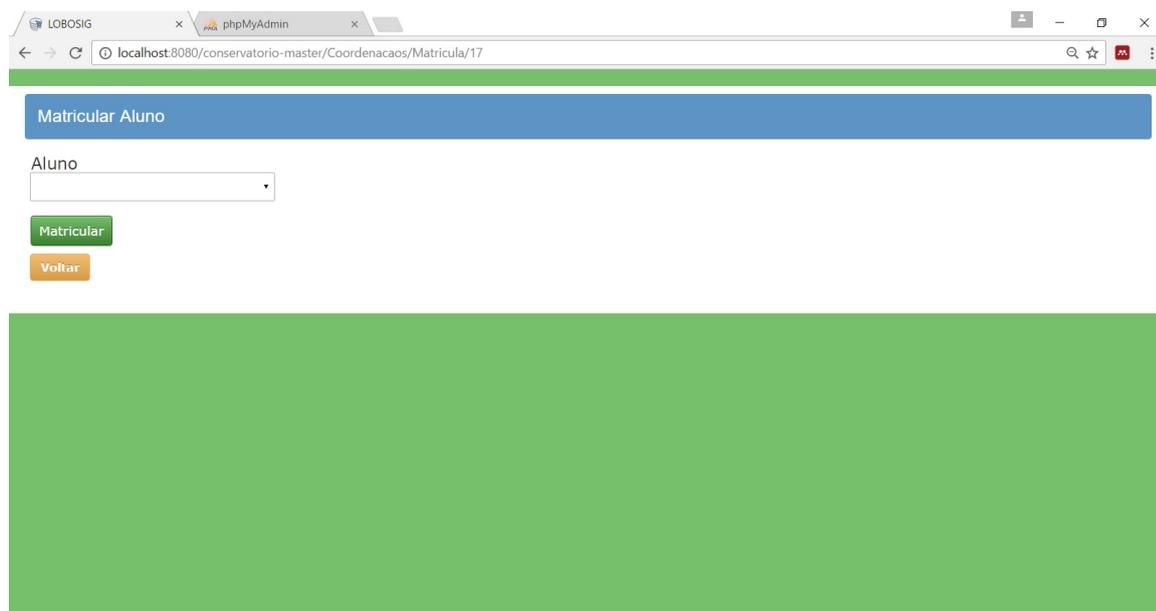


Figura 26: Tela de realização de Matrícula.

Campo	Descrição	Tipos de Dados
Aluno	Seleção do nome do aluno que se deseja matricular em uma referida turma	Caractere

Tabela 61: Descrição dos campos – Tela de realização de Matrícula.

Nome do Botão	Descrição
Matricular	Realiza a matrícula do aluno desejado em uma referida turma e redireciona o usuário Administrador para a tela de visualização de Turmas
Voltar	Redireciona o usuário Administrador para a tela de visualização de Turmas

Tabela 62: Descrição das ações – Tela de realização de Matrícula.

## 6.19 Tela de impressão de Lista de Presença

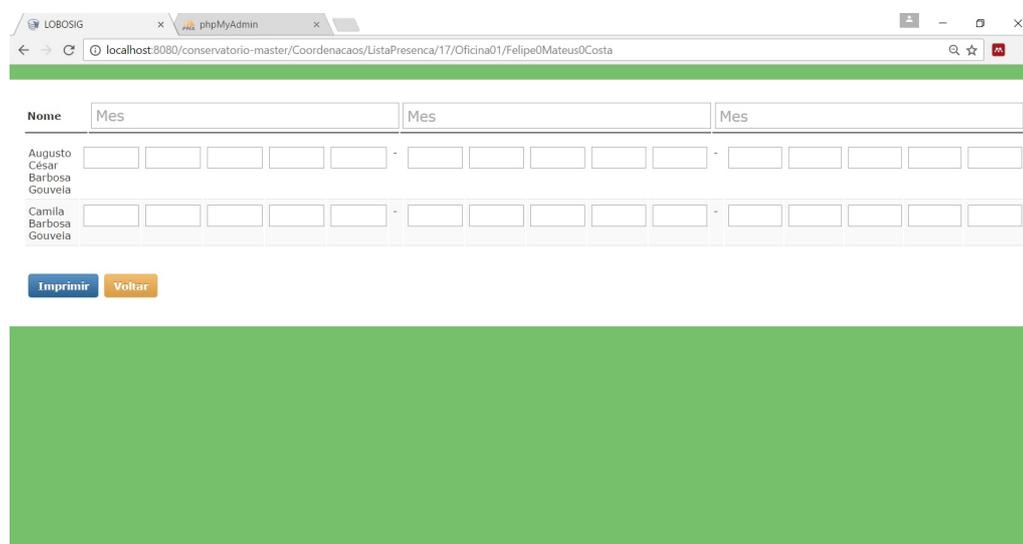


Figura 27: Tela de impressão do Comprovante de Lista de Presença.

Campo	Descrição	Tipos de Dados
Nome	Nome do aluno matriculado na referida turma	Caractere
Mês	Mês em que será lançada a frequência do aluno	Caractere

Tabela 63: Descrição dos campos – Tela de impressão de Lista de Presença.

Nome do Botão	Descrição
Imprimir	Redireciona o usuário Administrador para a tela de visualização das turmas já cadastradas no Sistema
Voltar	Redireciona o usuário Administrador para a tela de visualização das turmas já cadastradas no Sistema

Tabela 64: Descrição das ações – Tela de impressão de Lista de Presença.

## 6.20 Tela Painel do Aluno

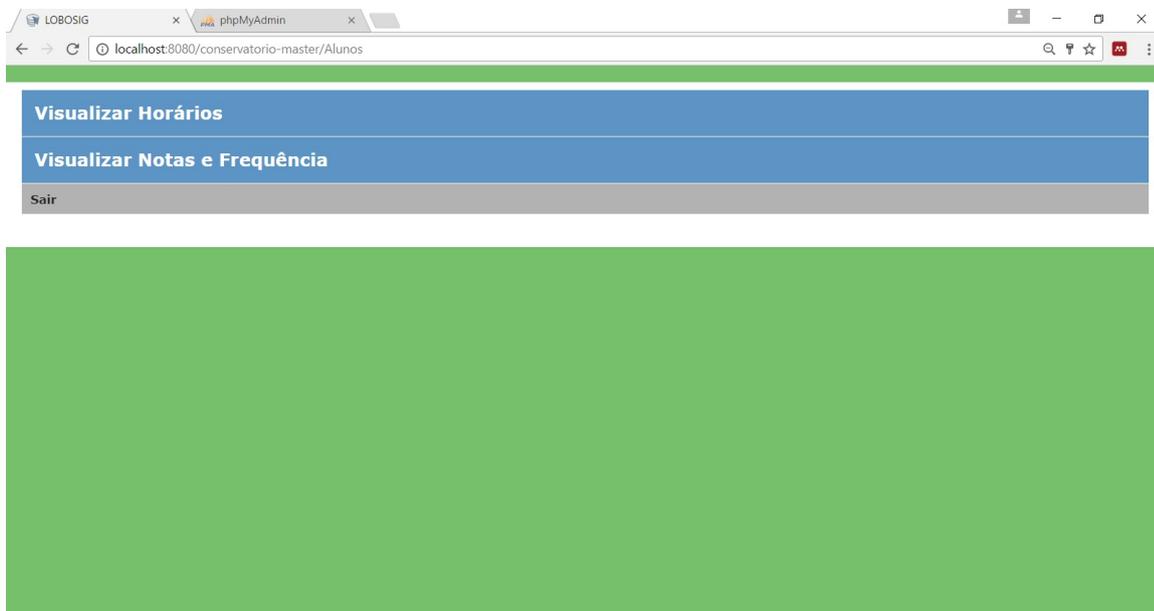


Figura 28: Tela Painel do Aluno.

Nome do Botão	Descrição
Visualizar Horários	Redireciona o usuário Aluno para a tela que exibe as turmas e horários das aulas em que está matriculado
Visualizar Notas e Frequências	Redireciona o usuário Aluno para a tela que exibe as notas e frequências das turmas em que está matriculado
Sair	Redireciona o usuário Aluno para a tela de login do Sistema

Tabela 65: Descrição das ações – Tela Painel do Aluno.

## 6.21 Tela de visualização de Horários das Aulas dos Alunos

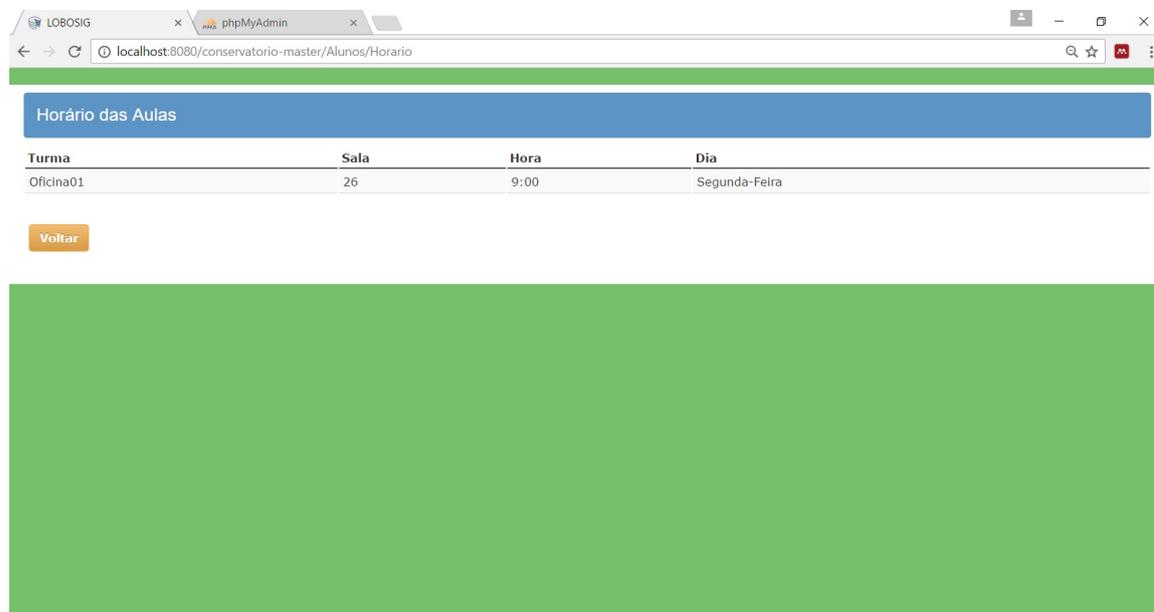


Figura 29: Tela de visualização de Horários das Aulas.

<b>Campo</b>	<b>Descrição</b>	<b>Tipos de Dados</b>
Turma	Nome da turma na qual o aluno está matriculado	Caractere
Sala	Número da sala onde ocorrerá a aula da turma	Numérico
Hora	Horário em que ocorrerá a aula da turma	Caractere
Dia	Dia em que ocorrerá a aula da turma	Caractere

Tabela 66: Descrição dos campos – Tela de visualização de Horários das Aulas.

<b>Nome do Botão</b>	<b>Descrição</b>
Voltar	Redireciona o usuário Aluno para a tela inicial do Painel do Aluno

Tabela 67: Descrição das ações – Tela de visualização de Horários das Aulas.

## 6.22 Tela de visualização de Notas e Frequências do Aluno



Figura 30: Tela de visualização de Notas e Frequências.

Campo	Descrição	Tipos de Dados
Turma	Nome da turma na qual o aluno está matriculado	Caractere
1º Bi - Faltas	Número de faltas do aluno no referido bimestre	Numérico
1º Bi - Notas	Conceito (nota) do aluno no referido bimestre	Caractere
2º Bi - Faltas	Número de faltas do aluno no referido bimestre	Numérico
2º Bi - Notas	Conceito (nota) do aluno no referido bimestre	Caractere
3º Bi - Faltas	Número de faltas do aluno no referido bimestre	Numérico
3º Bi - Notas	Conceito (nota) do aluno no referido bimestre	Caractere

4º Bi - Faltas	Número de faltas do aluno no referido bimestre	Numérico
4º Bi - Notas	Conceito (nota) do aluno no referido bimestre	Caractere
Total de Faltas	Número de faltas que aluno o aluno teve no ano	Numérico
Total de Notas	Conceito (nota) que o aluno teve como resultado final	Caractere
Resultado Final	Mostra se o aluno foi aprovado ou retido	Caractere

Tabela 68: Descrição dos campos – Tela de visualização de Notas e Frequências.

Nome do Botão	Descrição
Voltar	Redireciona o usuário Aluno para a tela inicial do Painel do Aluno

Tabela 69: Descrição das ações – Tela de visualização de Notas e Frequências.

## 6.23 Tela Painel do Professor

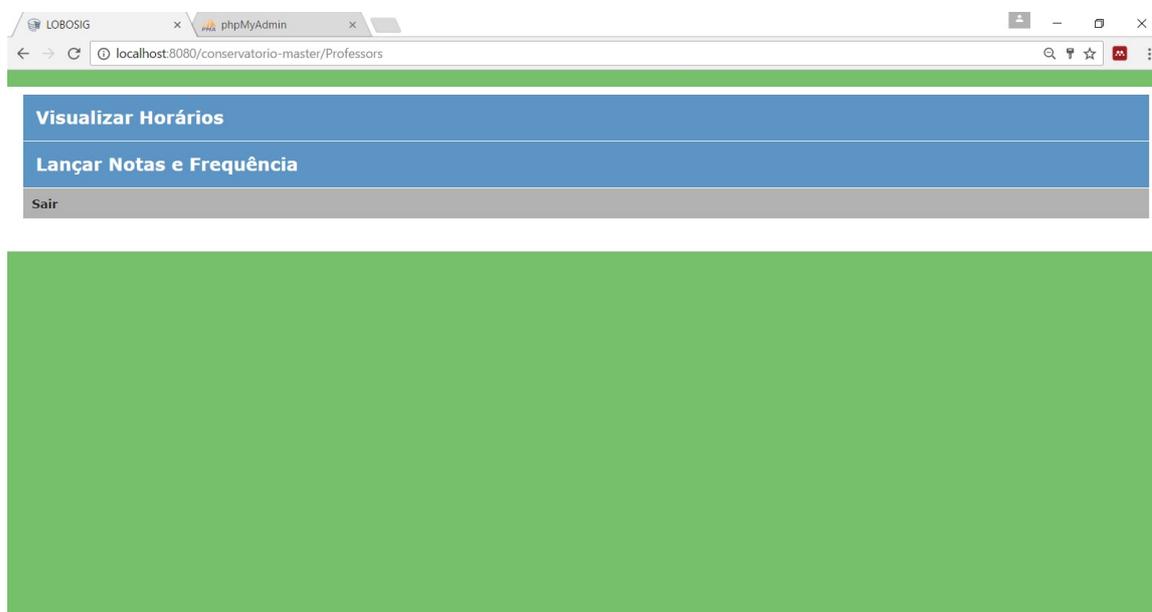


Figura 31: Tela Painel do Professor.

Nome do Botão	Descrição
Visualizar Horários	Redireciona o usuário Professor para a tela que exhibe as turmas e horários das aulas em que está vinculado
Lançar Notas e Frequências	Redireciona o usuário Professor para a tela que exhibe campos para lançar notas e frequências das turmas em que está vinculado
Sair	Redireciona o usuário Professor para a tela de login do Sistema

Tabela 70: Descrição das ações – Tela Painel do Professor.

## 6.24 Tela de seleção da Turma

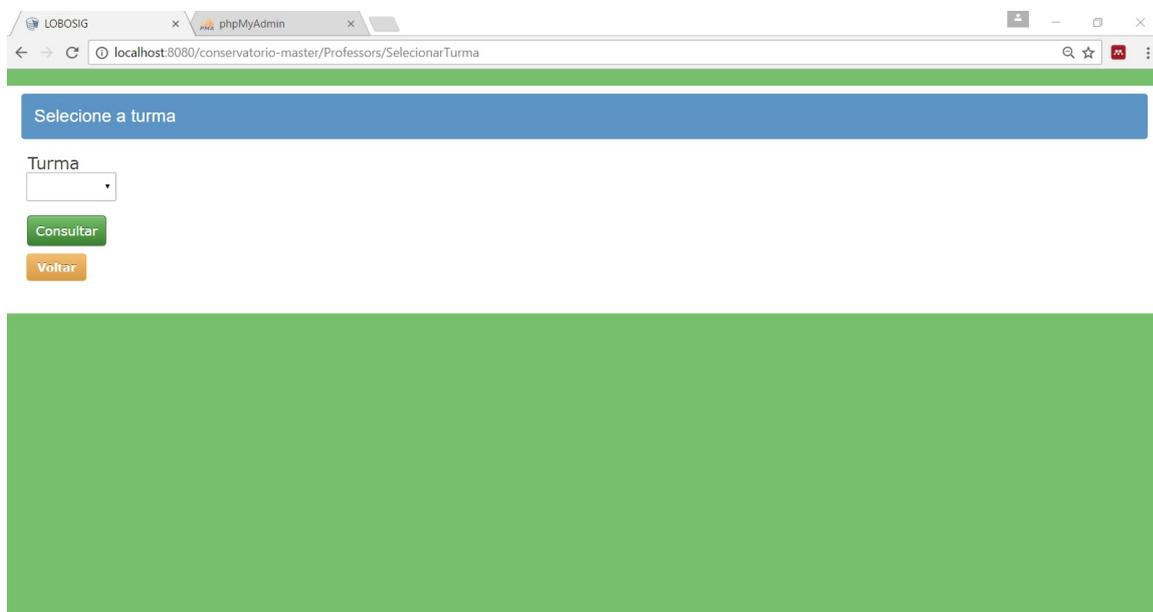


Figura 32: Tela de seleção da Turma.

<b>Campo</b>	<b>Descrição</b>	<b>Tipos de Dados</b>
Turma	Seleção da turma na qual o Professor deseja lançar notas e frequências	Caractere

Tabela 71: Descrição dos campos – Tela de seleção da Turma.

<b>Nome do Botão</b>	<b>Descrição</b>
Cadastrar	Redireciona o usuário Professor para a tela que permite lançar notas e frequências dos alunos matriculados
Voltar	Redireciona o usuário Professor para a tela inicial do Painel do Professor

Tabela 72: Descrição das ações – Tela de seleção da Turma.

## 6.25 Tela de visualização de Horário das Aulas do Professor

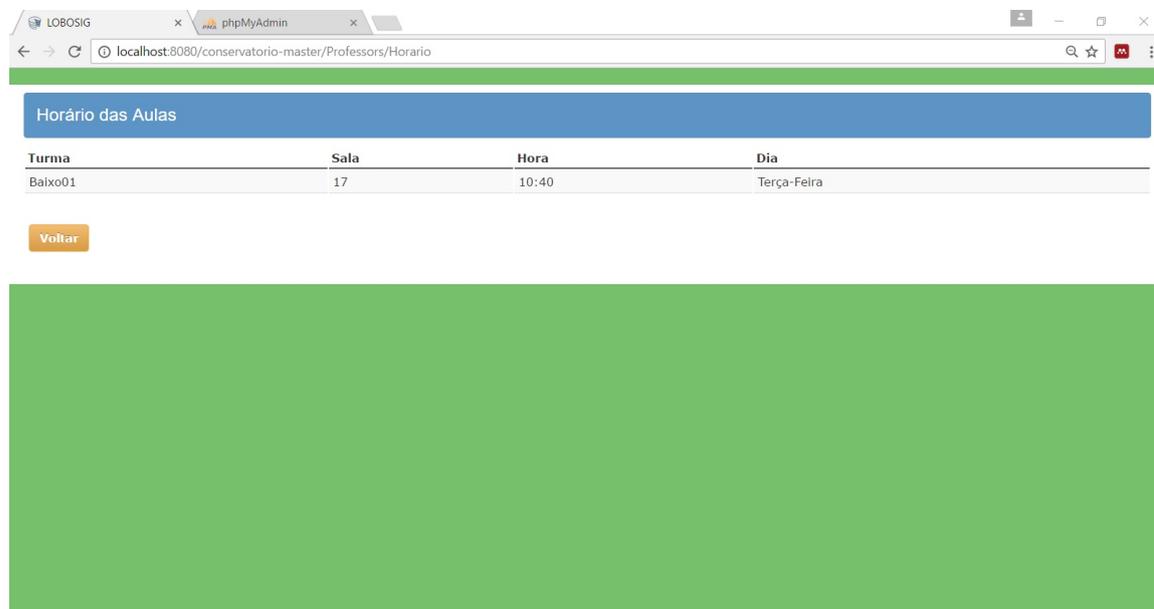


Figura 33: Tela de visualização de Horário das Aulas.

Campo	Descrição	Tipos de Dados
Nome	Nome dos alunos matriculados na turma	Caractere
Sala	Número da sala onde ocorrerá a aula da turma	Numérico
Hora	Horário em que ocorrerá a aula da turma	Caractere
Dia	Dia em que ocorrerá a aula da turma	Caractere

Tabela 73: Descrição dos campos – Tela de visualização de Horário das Aulas.

Nome do Botão	Descrição
Voltar	Redireciona o usuário Professor para a tela inicial do Painel do Professor

Tabela 74: Descrição das ações – Tela de visualização de Horário das Aulas.

## 6.26 Tela de lançar Notas e Frequências

The screenshot shows a web browser window with the URL `localhost:8080/conservatorio-master/Professors/NotaFrequencia/17`. The page title is "Lançar Notas e Frequências". The main content is a table with the following columns: Nome, 1º Bi - Faltas, 1º Bi - Notas, 2º Bi - Faltas, 2º Bi - Notas, 3º Bi - Faltas, 3º Bi - Notas, 4º Bi - Faltas, 4º Bi - Notas, Total de Faltas, Total de Notas, and Resultado Final. Two rows of student data are visible, each with input fields for the respective values and a "Lançar Notas" button. Below the table is a "Voltar" button. The bottom of the page is a solid green bar.

Figura 34: Tela de lançar de Notas e Frequências.

Campo	Descrição	Tipos de Dados
Nome	Nome do aluno o qual o professor lançará notas e frequências	Caractere
1º Bi - Faltas	Digitar número de faltas do aluno no referido bimestre	Numérico
1º Bi - Notas	Digitar conceito (nota) do aluno no referido bimestre	Caractere
2º Bi - Faltas	Digitar número de faltas do aluno no referido bimestre	Numérico
2º Bi - Notas	Digitar conceito (nota) do aluno no referido bimestre	Caractere
3º Bi - Faltas	Digitar número de faltas do aluno no referido bimestre	Numérico
3º Bi - Notas	Digitar conceito (nota) do aluno no referido bimestre	Caractere
4º Bi - Faltas	Digitar número de faltas do aluno no referido bimestre	Numérico
4º Bi - Notas	Digitar conceito (nota) do aluno no referido bimestre	Caractere
Total de Faltas	Digitar número de faltas que aluno o aluno teve no ano	Numérico
Total de Notas	Digitar conceito (nota) que o aluno teve como resultado final	Caractere
Resultado Final	Digitar se o aluno foi aprovado ou retido	Caractere

Tabela 75: Descrição dos campos – Tela de lançar de Notas e Frequências.

<b>Nome do Botão</b>	<b>Descrição</b>
Voltar	Redireciona o usuário Professor para a tela inicial do Painel do Professor

Tabela 76: Descrição das ações – Tela de lançar de Notas e Frequências.



## 7 Resultados e Discussões

Após a finalização da execução desse trabalho pode-se obter um Sistema de Gestão de dados simples, com interface descomplicada e que propicia muita intuitividade ao usuário, até mesmo para aqueles que possuem pouco domínio das tecnologias de informação.

Esse é o caso do Conservatório Estadual de Música Lobo de Mesquita, possui muitos funcionários que detém de pouca desenvoltura com relação às tecnologias computacionais. E tais pessoas desempenham funções que requerem muita responsabilidade por parte de quem a executa, afinal elas estão tratando de dados que são muito importantes para a instituição, ou seja, dados pessoais dos alunos e os dados escolares.

Não houve tempo hábil para a implantação do Sistema de Gestão na escola para que fosse feita a adaptação dos funcionários no período em que o Conservatório realiza as matrículas dos alunos, que é no final de cada ano. Esse é o período em que a direção da escola consegue fazer um melhor planejamento para o início do próximo ano letivo. Essa implantação do Sistema não foi realizada porque o mesmo não ficou pronto a tempo de ser instalado na Instituição antes dessa fase de matrícula, para que fossem realizados testes por parte de quem iria ser o responsável pelo cadastro dos dados, a fim de averiguar possíveis erros do Sistema, bem como fazer sugestões para melhoria do mesmo.

O Sistema de Gestão LoboSig está operando corretamente com as principais funções que ele oferece nessa versão. Mas precisa de pequenos ajustes em algumas restrições, bem como que sejam implementadas algumas funcionalidades já sugeridas pela direção da escola, que teve acesso ao uso do Sistema nessa versão e realizou pequenos testes. Tais sugestões para continuidade de codificação do aplicativo serão melhores detalhadas na Conclusão desse trabalho, na seção de "Trabalhos Futuros".

Atualmente o LoboSig possui as seguintes funcionalidades:

- Cadastro de alunos, funcionários e disciplinas.
- Edição de dados cadastrados de aluno, funcionários e disciplinas.
- Deleção de dados cadastrados de aluno, funcionários e disciplinas.
- Cadastro de Turmas vinculando um professor à determinada disciplina cadastrada.
- Matrícula de alunos em uma Turma cadastrada.
- Lançamento de notas e frequências dos alunos (Painel do Professor).
- Impressão de comprovante de matrícula, listagem de alunos e lista de presença (Painel do Administrador).

- Visualização dos dados (Paineis Aluno, Administrador e Professor).

O software não pode ser implantado, dentre a questão já citada, por mais questões adversas. Fica esse trabalho exposto em forma de documentação bem detalhada para que aqueles que despertem o interesse em continuar o projeto tenham em mãos um aporte valioso e completo sobre a problemática do Conservatório Estadual de Música Lobo de Mesquita.

## Conclusão

A elaboração do presente trabalho proporcionou uma avaliação de como um sistema feito sob medida poderá proporcionar uma melhora nos processos de trabalho realizados na secretaria do Conservatório Estadual de Música Lobo de Mesquita. Ademais, também permitiu a realização de pesquisa de campo para obter dados mais consistentes sobre as etapas do processo de trabalho, partes mais demoradas da realização do trabalho e grau de conhecimento em informática dos responsáveis pelo uso do sistema.

De maneira geral, os usuários são, na maior parte, do sexo feminino e possuem entendimento de informática muito básico. Porém, argumentam que as matrículas de alunos e o lançamento de notas e frequências são as partes que mais demoram no processo de trabalho. Outras atividades que devem ser executadas pelos profissionais ficam atrasadas em decorrência de perdas constantes que causam um frequente retrabalho.

Ao realizar uma análise no ambiente de trabalho, verificou-se que a parte mais complexa e desgastante do processo é o cadastro de alunos em softwares que não atendem corretamente a demanda da escola, que possui um número enorme de alunos. Utilizar de sistemas que não são apropriados para gerenciamento dados causa grande dor de cabeça aos profissionais responsáveis pelos processos de trabalho.

A análise de requisitos realizada por meio de um questionário formulado com perguntas abertas pode demonstrar a situação dos profissionais em relação a cada atividade. Além do que, também foi evidenciado que os usuários em questão não possuíam conhecimento em informática muito aprofundado, o que de certa forma justifica o fato de não terem procurado ainda por uma forma mais fácil e rápida de facilitar e melhorar os processos de trabalho. Todavia contribuiu para o desenvolvimento do sistema de forma a possibilitar o acesso a todos os envolvidos na comunidade escolar.

Além do mais, por ser uma escola regida por leis estaduais e por ter que seguir as diretrizes enviadas pela Secretaria de Educação do Estado, o Conservatório não possui recursos financeiros para disponibilizar de um Sistema de Gestão que possa sanar os problemas que a escola possui. E não é uma tarefa fácil passar por um projeto de licitação para esse propósito, pois o mesmo pode ser barrado pelo Estado com o argumento de que não há verbas para tais gastos. Nesse sentido, é visível que o projeto terá grande valia para a comunidade escolar num geral, pois trará benefícios sem custar nenhum valor financeiro para o caixa escolar.

Haja vista a relevância do assunto, tornou-se essencial o desenvolvimento de um sistema que proporcionasse meios para agilizar as partes mais demoradas do trabalho e torná-las fáceis de serem realizadas digitalmente por pessoas sem conhecimento mais

aprofundado em informática. Assim sendo, é possível economizar não só o tempo, como recursos humanos que são necessários para que as atividades sejam concluídas.

Nesse sentido, a utilização do sistema de gerenciamento de dados permitirá que os funcionários realizem seu trabalho de forma mais rápida e eficiente. Além disso, diminuirá o tempo de realização e perda de dados, que são constante, motivando mais as partes envolvidas nos processos de trabalho a realizarem suas atividades.

Esse trabalho proporcionou grande experiência para a formação acadêmica e pessoal, visto que é visível que o projeto idealizado aqui pode agregar grande conhecimento técnico e prático de todos os conteúdos vivenciados durante a graduação, enfim, trouxe grande experiência profissional para o programador. Também poderá trazer um grande valor à comunidade escolar do Conservatório Estadual de Música Lobo de Mesquita que envolve a vida de centenas de pessoas, já que poderá proporcionar algo que a escola almeja, porém não possui condições para adquirir. Escola tal que possui grande importância cultural para toda a comunidade da cidade de Diamantina e região.

## Trabalhos Futuros

Como já foi exposto, ainda há questões a serem desenvolvidas nesse trabalho. Para isso, é preciso que esse importante passo que foi dado em prol de ajudar o Conservatório Estadual de Música Lobo de Mesquita não seja apenas um passo, mas sim o início de um trabalho que possa ter um final satisfatório que possa atender a escola de música.

Ficam descritas nessa seção as sugestões para futuros trabalhos, a fim de que o programa possa atender plenamente às necessidades do Conservatório:

- As senhas dos usuários do Sistema LoboSig necessitam de criptografia para que nem mesmo a pessoa que possuir o acesso aos dados do SGBD possa visualizá-las.
- Implementar uma função que possa colocar um aluno desistente como um ex aluno, ou seja, uma forma de deixar registrado e de visualizar no programa os alunos que já passaram pela escola.
- Codificar mensagem de tempo de login excedido.
- Possibilitar que o professor também possa imprimir suas listagens de suas turmas e as listas de presença sem o controle do usuário Administrador para isso.
- Incluir a opção no Painel Administrador de visualizar os dados dos alunos ou funcionários sem a necessidade de entrar na aba "Edição".

- Implementar uma função para que o próprio Sistema possa efetuar as somas dos conceitos e faltas de cada aluno e armazenar essa soma no campo de Conceitos Finais e Frequências Finais.
- Implementar uma função para que o usuário que esquecer a senha possa recuperá-la.



## Referências

- AGUIAR, P. H. Sistema de Informação para Gestão Educacional: sistematização de uma proposta de modelo e avaliação do processo de sua construção. 2004. Citado na página 33.
- ALECRIM, E. *Conhecendo o Servidor Apache*. 2006. Disponível em: <[www.infowester.com/servapach.php](http://www.infowester.com/servapach.php)>. Citado na página 55.
- Allan Douglas. *Por que usar o CakePHP?* 2014. Disponível em: <<http://www.devmedia.com.br/por-que-usar-o-cakephp/24437>>. Citado na página 50.
- BALTHAZAR, G. d. R. et al. Uma Abordagem Prática sobre a Aplicação do padrão MVC com o Framework Struts. Nenhuma citação no texto.
- BERNERS-LEE, T. *The World-Wide Web*. New York: Communications of the ACM, 1994. v. 37. 76-82 p. Citado na página 32.
- CAKEPHP. *Bem-vindo*. 2016. Disponível em: <<https://book.cakephp.org/3.0/pt/index.html>>. Citado 2 vezes nas páginas 39 e 51.
- CANALTECH. *O que é servidor Apache*. Disponível em: <<https://canaltech.com.br/o-que-e/internet/O-que-e-servidor-Apache/>>. Nenhuma citação no texto.
- CARVALHO, F. P. D. Wampserver - Passo-a-Passo para Criar Formulários HTML usando PHP para acessar o Banco de Dados MySQL. p. 10. Nenhuma citação no texto.
- COOKBOOK. *Contents*. 2016. Disponível em: <<http://book.cakephp.org/1.2/pt/contents.html>>. Citado 2 vezes nas páginas 50 e 51.
- CORDEIRO, E. d. C. A.; JOAQUIM, C. H.; CEDRAN, D. H. Tutorial para uso do Latex para escrita científica. p. 95, 2013. Nenhuma citação no texto.
- DALL’OGLIO, P. [<http://www.adianti.com.br/phpoo>] *PHP Programando com Orientação a Objetos*. [S.l.]: Novatec, 2007. ISBN 9788575224656. Citado na página 40.
- Diego Eis. *O que é html e para que serve?* 2011. Disponível em: <<https://tableless.com.br/o-que-html-basico/>>. Citado 2 vezes nas páginas 52 e 53.
- DUARTE, M. d. R. D. Mendeley. p. 38, 2012. Nenhuma citação no texto.
- Edivaldo Brito. *WampServer | TechTudo*. 2013. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/wampserver.html>>. Citado na página 54.
- Edivaldo Brito. *Windows 10 | TechTudo*. 2016. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/windows-10.html>>. Citado na página 60.
- GONÇALVES, J. E. L. Os impactos das novas tecnologias nas empresas prestadoras de serviços. *Revista de Administração de Empresas*, v. 34, p. 19, 1994. Nenhuma citação no texto.

HAMMER, M.; CHAMPY, J. *Reengenharia Revolucionando a Empresa: Em função dos clientes, da concorrência e das grandes mudanças da gerência*. 11. ed. Rio de Janeiro: Briquet de Lemos/Livros, 1994. 189 p. Citado na página 31.

HOSTNET. 2000. Disponível em: <<https://www.hostnet.com.br/info/phpmyadmin/>>. Citado na página 57.

Ivan Luiz Marques Ricarte. *Programação Orientada a Objetos: Uma Abordagem com Java*. 2001. Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>>. Citado 4 vezes nas páginas 41, 43, 44 e 45.

Jair Rabelo. *PHP Orientado a Objetos: MVC em PHP | Criatividade, Tutoriais Photoshop e Templates*. 2012. Disponível em: <<http://www.escolacriatividade.com/php-orientado-a-objetos-mvc-em-php/>>. Nenhuma citação no texto.

Jason Legstorf. *PHP Orientado a Objetos para Iniciantes*. 2011. Disponível em: <<https://code.tutsplus.com/pt/tutorials/object-oriented-php-for-beginners--net-12762>>. Citado na página 42.

João Otávio. *Sublime Text IDE: Introdução a melhor IDE para desenvolvimento*. 2016. Disponível em: <<http://www.devmedia.com.br/sublime-text-ide-introducao-a-melhor-ide-para-desenvolvimento/34117>>. Citado na página 58.

José Carlos Macoratti. *Conceitos Básicos de Orientação a Objetos*. 2010. Disponível em: <[http://www.macoratti.net/net{\\\_}oocb](http://www.macoratti.net/net{\_}oocb)>. Citado na página 44.

José Malcher Jr. *Sobrecarga no PHP*. 2015. Disponível em: <<http://josemalcher.net/sobrecarga-php/>>. Citado 2 vezes nas páginas 46 e 47.

JUNIOR, L. A. Z.; VIDAL, A. G. d. R. *Construção de sistemas de informação baseados na Tecnologia Web*. p. 232–244, 2006. Citado 2 vezes nas páginas 31 e 32.

LAUDON, K.; LAUDON, J. *Sistemas de informação gerenciais*. [S.l.]: Pearson, 2011. 448 p. Citado na página 31.

LIMA, J. E. G. *Os impactos das novas tecnologias nas empresas prestadoras de serviços*. São Paulo: RAE, 1994. v. 34. 663-681 p. Citado na página 34.

Luiz Otávio Miranda. *Utilizando estrutura MVC em PHP - Parte 1 - Todo Espaço Online*. 2014. Disponível em: <<https://www.todoespacoonline.com/w/2014/09/mvc-em-php-parte-1/>>. Nenhuma citação no texto.

MACGARRY, L. *The next generation of business data processing*. New York: Communications of the ACM, 1999. v. 42. 13-16 p. Citado na página 31.

MCGARRY, K. *O contexto dinâmico da informação: uma análise introdutória*. Brasília: Briquet de Lemos/Livros, 1999. Citado na página 31.

OLIVEIRA, L. F. de. *Entendendo a tríade Model-View-Controller (MVC) utilizando padrões de projeto de software orientado a objetos*. v. 1, n. Mvc, 2013. Nenhuma citação no texto.

OLIVEIRA, S. M. de; FREITAS, M. d. C. D.; TONO, C. C. P. Gestão da Informação na Escola Pública Estadual do Paraná : Aplicabilidade Pedagógica do Sistema de Informação do Registro Escolar. n. Sbie, p. 309–319, 2008. Citado na página 31.

PHP Group. *PHP: História do PHP - Manual*. 2001. Disponível em: <<http://php.net/history.php>>. Citado 5 vezes nas páginas 39, 41, 45, 46 e 47.

PISA, P. *O que é e como usar o MySQL?* 2012. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>>. Citado na página 56.

QUEIROZ, A. L. V.; MUCELINI, H. E. M. Gestão por processos e gestão do conhecimento: o caso da superintendência de materiais de Itaipu Binacional. 2010. Citado na página 31.

Rafaela Pozzebom. *O que GitHub?* 2015. Disponível em: <<https://www.oficinadanet.com.br/post/14791-o-que-github>>. Citado na página 59.

RED, B.; GREEN, B. Model-View-Controller. *Structure*, n. Mvc, p. 19, 2011. Nenhuma citação no texto.

Ricardo Rezende. *Banco de dados: Conceitos Fundamentais*. 2006. Disponível em: <<http://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Citado na página 56.

Rodrigo Almeida. *O que é um Modelo Relacional?* 2008. Disponível em: <<http://www.rodrigoalmeida.net/e-um-modelo-relacional/>>. Citado na página 56.

RODRIGUES, J. *MER Modelo Entidade Relacionamento e DER Diagrama Entidade Relacionamento*. 2015. Disponível em: <<http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>. Citado na página 80.

ROSINI, A. O uso da tecnologia da informática na educação. Uma reflexão no ensino com crianças. 2003. Disponível em: <<http://www.ipv.pt/millennium/Millennium27/15.htm>>. Citado na página 34.

SANTIAGO, F. *Orientação a Objetos - Simples assim*. 2016. Disponível em: <<https://devmedia.com.br/orientacao-a-objetos-simples-assim/3254>>. Citado na página 42.

SANTOS, R. J. Introdução ao LaTeX. 2012. Nenhuma citação no texto.

SERVER, V. Virtual Machine Guide. 2006. Nenhuma citação no texto.

SOFTWARE, F. C. CakePHP Cookbook Documentation. p. 1–760, 2015. Disponível em: <[http://book.cakephp.org/2.0/{\\\_\\\_}downloads/pt/CakePHPCookbook.>](http://book.cakephp.org/2.0/{\_\_}downloads/pt/CakePHPCookbook.>) Nenhuma citação no texto.

STAIR, R. M.; REYNOLDS, G. W. *Princípios de sistemas de informação*. [s.n.], 2005. 646 p. ISBN 9788522104819. Disponível em: <<http://books.google.com.br/books?id=sOMLBAAACAAJ>>. Citado na página 31.

TANENBAUM, A. S. Redes de computadores. *Editora Campus*, v. 18, n. LXIII, p. 1–632, 2003. ISSN 13891286. Disponível em: <<http://www.prenhall.com/tanenbaum>>. Citado na página 31.

United Nations Development Programme. *United Nations Development Report*. [S.l.]: United Nations, 1999. 1–130 p. ISBN 0195215613. Nenhuma citação no texto.

VENTURA, P. *Requisito Funcional*. 2016. Disponível em: <<http://www.ateomomento.com.br/o-que-e-requisito-funcional/>>. Citado na página 65.

WEBDEVBR. *MVC na prática - Entendendo o padrão MVC na prática*. Disponível em: <<https://www.webdevbr.com.br/mvc-na-pratica-entendendo-o-padrao-mvc-na-pratica>>. Nenhuma citação no texto.

WIKIPÉDIA. *phpMyAdmin*. 2014. Disponível em: <<https://pt.wikipedia.org/wiki/PhpMyAdmin>>. Citado na página 65.

WINDOWS\_MICROSOFT. *Ficha informativa do ciclo de vida do Windows*. Windows - Microsoft. Disponível em: <<http://windows.microsoft.com/pt-br/windows/lifecycle>>. Nenhuma citação no texto.

# Anexos



# ANEXO A – Banco de Dados

---

```

1 <?php
2 /**
3  *
4  *
5  * CakePHP(tm) : Rapid Development Framework (http://cakephp.org)
6  * Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
7  *
8  * Licensed under The MIT License
9  * For full copyright and license information, please see the LICENSE.txt
10 * Redistributions of files must retain the above copyright notice.
11 *
12 * @copyright      Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
13 * @link           http://cakephp.org CakePHP(tm) Project
14 * @package        app.Config
15 * @since          CakePHP(tm) v 0.2.9
16 * @license        http://www.opensource.org/licenses/mit-license.php MIT
17 *                License
18 */
19 /**
20 * Database configuration class.
21 *
22 * You can specify multiple configurations for production, development and
23 * testing.
24 *
25 * datasource => The name of a supported datasource; valid options are as
26 * follows:
27 * Database/Mysql – MySQL 4 & 5,
28 * Database/Sqlite – SQLite (PHP5 only),
29 * Database/Postgres – PostgreSQL 7 and higher,
30 * Database/Sqlserver – Microsoft SQL Server 2005 and higher
31 *
32 * You can add custom database datasources (or override existing
33 * datasources) by adding the
34 * appropriate file to app/Model/Datasource/Database. Datasources should be
35 * named 'MyDatasource.php',
36 *
37 *
38 * persistent => true / false
39 *
40 * Determines whether or not the database should use a persistent
41 * connection
42 *
43 */

```

```

37 * host =>
38 * the host you connect to the database. To add a socket or port number,
    use 'port' => #
39 *
40 * prefix =>
41 * Uses the given prefix for all the tables in this database. This setting
    can be overridden
42 * on a per-table basis with the Model::$tablePrefix property.
43 *
44 * schema =>
45 * For Postgres/Sqlserver specifies which schema you would like to use the
    tables in.
46 * Postgres defaults to 'public'. For Sqlserver, it defaults to empty and
    use
47 * the connected user's default schema (typically 'dbo').
48 *
49 * encoding =>
50 * For MySQL, Postgres specifies the character encoding to use when
    connecting to the
51 * database. Uses database default not specified.
52 *
53 * unix_socket =>
54 * For MySQL to connect via socket specify the 'unix_socket' parameter
    instead of 'host' and 'port'
55 *
56 * settings =>
57 * Array of key/value pairs, on connection it executes SET statements for
    each pair
58 * For MySQL : http://dev.mysql.com/doc/refman/5.6/en/set-statement.html
59 * For Postgres : http://www.postgresql.org/docs/9.2/static/sql-set.html
60 * For Sql Server : http://msdn.microsoft.com/en-us/library/ms190356.aspx
61 *
62 * flags =>
63 * A key/value array of driver specific connection options.
64 */
65 class DATABASE_CONFIG {
66
67     public $default = array(
68         'datasource' => 'Database/Mysql',
69         'persistent' => false,
70         'host' => 'localhost',
71         'login' => 'root',
72         'password' => '',
73         'database' => 'conservatorio',
74         'prefix' => '',
75         //'encoding' => 'utf8',
76     );

```

```
77
78     public $test = array(
79         'datasource' => 'Database/Mysql',
80         'persistent' => false,
81         'host' => 'localhost',
82         'login' => 'user',
83         'password' => 'password',
84         'database' => 'test_database_name',
85         'prefix' => '',
86         // 'encoding' => 'utf8',
87     );
88 }
```

---



# ANEXO B – View

## B.1 View Alunos

### B.1.1 Horário

---

```

1 <?php
2     echo '<h3>Hor rio das Aulas</h3>', '<table>',
3     $this->Html->tableHeaders( array( 'Turma', 'Sala', 'Hora', 'Dia' ));
4     $count = count($idturm);
5
6     foreach ( $turma as $k => $turma ){
7         if ( $count == $k ) break;    //PARA QND O TAMANHO DO VETOR ATINGE
8         if($turma[ 'Turma' ][ 'id' ] == $idturm[ $k ])
9             echo $this->Html->tableCells ( array (
10                array (
11                    $turma [ 'Turma' ] [ 'nome' ],
12                    $turma [ 'Turma' ] [ 'sala' ],
13                    $turma [ 'Turma' ] [ 'horaaula' ],
14                    $turma [ 'Turma' ] [ 'diaaula' ]
15                )
16            ) );
17     }
18     echo
19     '</table><br />',
20     $this->Html->link ( 'Voltar', array (
21         'controller' => 'Alunos',
22         'action' => 'Index'
23     ), array (
24         'class' => 'btn btn-back'
25     ) );
26 ?>

```

---

### B.1.2 Index

---

```

1 <?php
2     echo
3     $this->Html->link ( 'Visualizar Hor rios', array (
4         'controller' => 'Alunos',
5         'action' => 'Horario'
6     ), array (
7         'class' => 'menu'
8     ) ), $this->Html->link ( 'Visualizar Notas e Frequ ncia', array (
9         'controller' => 'Alunos',

```



```
33         $avaliacao[ 'Avaliacao' ][ 'fnotas' ],
34         $avaliacao[ 'Avaliacao' ][ 'resultadofinal' ],
35     ) );
36 }
37 echo '</table><br />',
38     $this->Html->link ( 'Voltar', array (
39         'controller' => 'Alunos',
40         'action' => 'Index'
41     ), array (
42         'class' => 'btn btn-back'
43     ) );
44 ?>
```

---

## B.2 View Autenticações

### B.2.1 Login

---

```
1 <?php
2     echo
3     $this->Form->create ( 'Login', array( 'style' => 'text-align: center;
4         width:100%' ) ),
5     $this->Html->image( 'logo.png', array( 'width' => '300px', 'height' => '
6         300px', 'style' => 'text-align: center;' ) ),
7     '<h1>LOBOSIG</h1>',
8     $this->Form->input ( 'usuario', array (
9         'label' => false ,
10        'type' => 'text',
11        'placeholder' => 'Usu rio'
12    ) ), $this->Form->input ( 'senha', array (
13        'label' => false ,
14        'type' => 'password',
15        'placeholder' => 'Senha'
16    ) ), $this->Form->end ( 'Entrar' );
17 ?>
```

---

## B.3 View Coordenação

### B.3.1 Aluno

---

```
1 <?php
2     echo
3     $this->Html->link ( 'Novo Aluno', array (
4         'controller' => 'Coordenacaos',
5         'action' => 'NovoAluno'
```

```

6         ), array (
7             'class' => 'btn btn-primary'
8         ) ),
9     '<br /><br /><table>',
10    $this->Html->tableHeaders(array('Nome', 'Matricula', 'Opções', '', ''
11    ));
12    foreach ( $aluno as $aluno ) :
13        echo
14            $this->Html->tableCells ( array (
15                array (
16                    $aluno ['Aluno'] ['nome'],
17                    $aluno ['Aluno'] ['matricula'],
18                    $this->Html->link ( 'Editar Aluno', array (
19                        'controller' => 'Coordenacaos',
20                        'action' => 'EditarAluno',
21                        $aluno ['Aluno'] ['id']
22                    ) ),
23                    $this->Html->link ( 'Comprovante de Matrícula',
24                        array (
25                            'controller' => 'Coordenacaos',
26                            'action' => 'ComprovanteMatricula',
27                            $aluno ['Aluno'] ['id']
28                        ) ),
29                    $this->Html->link ( 'Excluir Aluno', array (
30                        'controller' => 'Coordenacaos',
31                        'action' => 'ExcluirAluno',
32                        $aluno ['Aluno'] ['id']
33                    ) )
34                )
35            ) );
36    endforeach;
37    echo
38        '</table>',
39        $this->Html->link ( 'Voltar', array (
40            'controller' => 'Coordenacaos',
41            'action' => 'Index'
42        ) ), array (
43            'class' => 'btn btn-back'
44        ) );
45 ?>

```

---

### B.3.2 Comprovante de Matrícula

---

```

1 <table>
2     <tr>
3         <td><?php echo $this->Html->image('logo.png', array('width' => '100
4             px', 'height' => '100px', 'style' => 'text-align: center;'))

```

```

        ?></td>
4      <td>
5          <br />
6          Conservat rio Estadual de M sica Lobo de Mesquita<br />
7          Pra a Coronel Cosme Alves Souto, s/n – Centro<br />
8          Diamantina – MG (39100–000) / Brasil <br />
9          Telefone: (38) 3531–1774
10     </td>
11 </tr>
12 </table>
13 <p style="text-align: center; font-size: 30px; font-weight: bold;">
    Comprovante de Matr cula</p>
14 <br />
15 <p>Declaro, para os devidos fins, que <?php echo strtoupper($aluno ['Aluno'
    ] ['nome']); ?>, filho(a) de <?php echo strtoupper($aluno ['Aluno' ] ['
    mae']); ?> e de <?php echo strtoupper($aluno ['Aluno' ] ['pai']); ?>,
    matr cula <?php echo strtoupper($aluno ['Aluno' ] ['matricula']); ?>,
    nascido em <?php echo date( 'd/m/Y', strtotime($aluno ['Aluno' ] ['
    datanasc'])); ?> em <?php echo strtoupper($aluno ['Aluno' ] ['
    naturalidade']); ?> – <?php echo strtoupper($aluno ['Aluno' ] ['
    ufnaturalidade']); ?>, encontra-se matriculado(a) neste estabelecimento
    de ensino no corrente ano letivo.</p>
16 <br /><br class="show"><br class="show"><br class="show"><br class="show"><br
    class="show">
17 <p style="text-align: right;">Diamantina, <?php setlocale(LC_ALL, 'pt_BR',
    'pt_BR.utf-8', 'pt_BR.utf-8', 'portuguese'); date_default_timezone_set('
    America/Sao_Paulo'); echo strftime('%A, %d de %B de %Y', strtotime('
    today')); ?>.</p>
18 <br /><br class="show"><br class="show"><br class="show"><br class="show">
19 <p style="text-align: center;">
    _____</p>
20 <p style="text-align: center;">Assinatura do Funcion rio(a) Respons vel</
    p>
21
22 <?php
23     echo $this->Html->link ( 'Imprimir', '#', array (
24         'class' => 'btn btn-primary hidden',
25         'onclick' => 'window.print()'
26     ) );
27     echo $this->Html->link ( 'Voltar', array (
28         'controller' => 'Coordenacaos',
29         'action' => 'Aluno'
30     ), array (
31         'class' => 'btn btn-back hidden'
32     ) );
33 ?>

```

### B.3.3 Disciplina

---

```

1 <?php
2     echo
3         $this->Html->link ( 'Nova Disciplina', array (
4             'controller' => 'Coordenacaos',
5             'action' => 'NovaDisciplina'
6         ), array (
7             'class' => 'btn btn-primary'
8         ) ),
9         '<br /><br /><table>',
10    $this->Html->tableHeaders(array('Disciplinas', 'Curso', 'Ciclo', 'Fase'
11        , 'Opções', ''));
12    foreach ( $disciplina as $disciplina ) :
13        echo
14            $this->Html->tableCells ( array (
15                array (
16                    $disciplina ['Disciplina'] ['nome'],
17                    $disciplina ['Disciplina'] ['curso'],
18                    $disciplina ['Disciplina'] ['ciclo'],
19                    $disciplina ['Disciplina'] ['fase'],
20                    $this->Html->link ( 'Editar Disciplina', array (
21                        'controller' => 'Coordenacaos',
22                        'action' => 'EditarDisciplina',
23                        $disciplina ['Disciplina'] ['id']
24                    ) ),
25                    $this->Html->link ( 'Excluir Disciplina', array (
26                        'controller' => 'Coordenacaos',
27                        'action' => 'ExcluirDisciplina',
28                        $disciplina ['Disciplina'] ['id']
29                    ) )
30                )
31            ) );
32    endforeach;
33    echo
34        '</table>',
35        $this->Html->link ( 'Voltar', array (
36            'controller' => 'Coordenacaos',
37            'action' => 'Index'
38        ), array (
39            'class' => 'btn btn-back'
40        ) );
41 ?>

```

---

### B.3.4 Editar Aluno

---

```

1 <?php

```

```
2      echo
3      $this->Form->create ( 'Aluno' ),
4      '<h3>Dados Pessoais</h3>',
5      $this->Form->input ( 'id', array (
6          'type' => 'hidden',
7          'value' => $aluno['Aluno']['id']
8      ) ), $this->Form->input ( 'nome', array (
9          'label' => 'Nome',
10         'type' => 'text',
11         'value' => $aluno['Aluno']['nome']
12     ) ), $this->Form->input ( 'rg', array (
13         'label' => 'RG',
14         'type' => 'text',
15         'value' => $aluno['Aluno']['rg']
16     ) ), $this->Form->input ( 'cpf', array (
17         'label' => 'CPF',
18         'type' => 'number',
19         'value' => $aluno['Aluno']['cpf']
20     ) ), $this->Form->input ( 'datanasc', array(
21         'label' => 'Data de Nascimento',
22         'type' => 'date',
23         'dateFormat' => 'DMY',
24         'minYear' => date('Y') - 50,
25         'maxYear' => date('Y') - 10,
26         'value' => $aluno['Aluno']['datanasc']
27     ) ), $this->Form->label ( 'Sexo' ),
28     $this->Form->select ( 'sexo', array (
29         'M' => 'Masculino',
30         'F' => 'Feminino',
31     ) ), $this->Form->input ( 'naturalidade', array (
32         'label' => 'Naturalidade',
33         'type' => 'text',
34         'value' => $aluno['Aluno']['naturalidade']
35     ) ), $this->Form->input ( 'ufnaturalidade', array (
36         'label' => 'UF - Naturalidade',
37         'type' => 'text',
38         'value' => $aluno['Aluno']['ufnaturalidade']
39     ) ), $this->Form->input ( 'mae', array (
40         'label' => 'M e',
41         'type' => 'text',
42         'value' => $aluno['Aluno']['mae']
43     ) ), $this->Form->input ( 'pai', array (
44         'label' => 'Pai',
45         'type' => 'text',
46         'value' => $aluno['Aluno']['pai']
47     ) ),
48     $this->Form->label ( 'Turno' ),
```

```

49     $this->Form->select ( 'turno', array (
50         'M' => 'Matutino',
51         'V' => 'Vespertino',
52         'I' => 'Integral',
53         'N' => 'Noturno'
54     ) ), $this->Form->input ( 'escola', array (
55         'label' => 'Escola',
56         'type' => 'text',
57         'value' => $aluno['Aluno']['escola']
58     ) ), $this->Form->input ( 'anoescolar', array (
59         'label' => 'Escolaridade',
60         'type' => 'text',
61         'value' => $aluno['Aluno']['anoescolar']
62     ) ), $this->Form->input ( 'obs', array (
63         'label' => 'Observação',
64         'type' => 'textarea',
65         'value' => $aluno['Aluno']['obs']
66
67     ) ), '<h3>Endereço</h3>',
68     $this->Form->input ( 'rua', array (
69         'label' => 'Rua',
70         'type' => 'text',
71         'value' => $aluno['Aluno']['rua']
72     ) ), $this->Form->input ( 'numero', array (
73         'label' => 'Número',
74         'type' => 'number',
75         'value' => $aluno['Aluno']['numero']
76     ) ), $this->Form->input ( 'bairro', array (
77         'label' => 'Bairro',
78         'type' => 'text',
79         'value' => $aluno['Aluno']['bairro']
80     ) ), $this->Form->input ( 'cidade', array (
81         'label' => 'Cidade',
82         'type' => 'text',
83         'value' => $aluno['Aluno']['cidade']
84     ) ), $this->Form->input ( 'uf', array (
85         'label' => 'UF',
86         'type' => 'text',
87         'value' => $aluno['Aluno']['uf']
88     ) ), '<h3>Contato</h3>',
89     $this->Form->input ( 'telefone', array (
90         'label' => 'Telefone',
91         'type' => 'text',
92         'value' => $aluno['Aluno']['telefone']
93     ) ), $this->Form->input ( 'celular', array (
94         'label' => 'Celular',
95         'type' => 'text',

```

```

96         'value' => $aluno['Aluno']['celular']
97     ) ), $this->Form->input ( 'email', array (
98         'label' => 'Email',
99         'type' => 'email',
100        'value' => $aluno['Aluno']['email']
101    ) ), $this->Form->end ( 'Atualizar' ),
102    $this->Html->link ( 'Voltar', array (
103        'controller' => 'Coordenacaos',
104        'action' => 'Aluno'
105    ), array (
106        'class' => 'btn btn-back'
107    ) );
108 ?>

```

### B.3.5 Editar Disciplina

```

1 <?php
2     echo
3     $this->Form->create ( 'Disciplina' ),
4     $this->Form->input ( 'id', array (
5         'type' => 'hidden',
6         'value' => $disciplina['Disciplina']['id']
7     ) ), $this->Form->label ( 'Disciplina' ),
8     $this->Form->select ( 'nome', array (
9         'Atividade Art stica Complementar' => 'Atividade
10         Art stica Complementar',
11         'Canto Coral' => 'Canto Coral',
12         'Musicaliza o' => 'Musicaliza o',
13         'Oficina Multimeios' => 'Oficina Multimeios',
14         'Percep o Musical' => 'Percep o Musical',
15         'Pr tica de Conjunto' => 'Pr tica de Conjunto',
16         'Bateria' => 'Bateria',
17         'Cavaquinho' => 'Cavaquinho',
18         'Contrabaixo el trico' => 'Contrabaixo el trico',
19         'Flauta doce' => 'Flauta doce',
20         'Flauta transversal' => 'Flauta transversal',
21         'Guitarra' => 'Guitarra',
22         'Percuss o' => 'Percuss o',
23         'Piano' => 'Piano',
24         'Saxofone' => 'Saxofone',
25         'Teclado' => 'Teclado',
26         'Trombone' => 'Trombone',
27         'Trompa' => 'Trompa',
28         'Trompete' => 'Trompete',
29         'Viol o' => 'Viol o',
30         'Violino' => 'Violino',
31         'Violoncelo' => 'Violoncelo',

```

```

31     ) ), '<br /><br />',
32     $this->Form->label ( 'Curso' ),
33     $this->Form->select ( 'curso', array (
34         'Educação Musical' => 'Educação Musical',
35         'Técnico' => 'Técnico'
36     ) ), '<br /><br />',
37     $this->Form->label ( 'Ciclo - Ano' ),
38     $this->Form->select ( 'ciclo', array (
39         'Inicial' => 'Inicial',
40         'Intermediário' => 'Intermediário',
41         'Complementar' => 'Complementar',
42         'Primeiro Ano' => 'Primeiro Ano',
43         'Segundo Ano' => 'Segundo Ano',
44         'Terceiro Ano' => 'Terceiro Ano',
45     ) ), '<br /><br />',
46     $this->Form->label ( 'Fase' ),
47     $this->Form->select ( 'fase', array (
48         'I' => 'I',
49         'II' => 'II',
50         'III' => 'III'
51     ) ), '<br /><br />', $this->Form->end ( 'Atualizar' ),
52     $this->Html->link ( 'Voltar', array (
53         'controller' => 'Coordenacaos',
54         'action' => 'Disciplina'
55     ), array (
56         'class' => 'btn btn-back'
57     ) );
58 ?>

```

### B.3.6 Editar Funcionário

```

1 <?php
2     echo
3         $this->Html->script ( 'senha' ),
4         $this->Form->create ( 'Funcionario' ),
5         '<h3>Dados Pessoais</h3>',
6         $this->Form->input ( 'id', array (
7             'type' => 'hidden',
8             'value' => $funcionario [ 'Funcionario' ] [ 'id' ]
9         ) ),
10        $this->Form->input ( 'nome', array (
11            'label' => 'Nome',
12            'type' => 'text',
13            'value' => $funcionario [ 'Funcionario' ] [ 'nome' ]
14        ) ), $this->Form->label ( 'Cargo' ),
15        $this->Form->select ( 'cargo', array (
16            'Coordenacaos' => 'Coordenador(a)',

```

```
17         'Professors' => 'Professor(a)',
18         'Coordenacaos' => 'Auxiliar de Servi os Gerais'
19     ) ), $this->Form->input ( 'masp', array (
20         'label' => 'MASP',
21         'type' => 'number',
22         'value' => $funcionario[ 'Funcionario' ][ 'masp' ]
23     ) ), $this->Form->input ( 'datanasc', array(
24         'label' => 'Data de Nascimento',
25         'type' => 'date',
26         'dateFormat' => 'DMY',
27         'minYear' => date( 'Y' ) - 50,
28         'maxYear' => date( 'Y' ) - 10,
29     ) ), $this->Form->label ( 'Sexo' ),
30     $this->Form->select ( 'sexo', array (
31         'M' => 'Masculino',
32         'F' => 'Feminino'
33     ) ), '<h3>Endere o</h3>',
34     $this->Form->input ( 'rua', array (
35         'label' => 'Rua',
36         'type' => 'text',
37         'value' => $funcionario[ 'Funcionario' ][ 'rua' ]
38     ) ), $this->Form->input ( 'numero', array (
39         'label' => 'Numero',
40         'type' => 'number',
41         'value' => $funcionario[ 'Funcionario' ][ 'numero' ]
42     ) ), $this->Form->input ( 'bairro', array (
43         'label' => 'Bairro',
44         'type' => 'text',
45         'value' => $funcionario[ 'Funcionario' ][ 'bairro' ]
46     ) ), $this->Form->input ( 'cidade', array (
47         'label' => 'Cidade',
48         'type' => 'text',
49         'value' => $funcionario[ 'Funcionario' ][ 'cidade' ]
50     ) ), $this->Form->input ( 'uf', array (
51         'label' => 'UF',
52         'type' => 'text',
53         'value' => $funcionario[ 'Funcionario' ][ 'uf' ]
54     ) ), '<h3>Contato</h3>',
55     $this->Form->input ( 'telefone', array (
56         'label' => 'Telefone',
57         'type' => 'text',
58         'value' => $funcionario[ 'Funcionario' ][ 'telefone' ]
59     ) ), $this->Form->input ( 'celular', array (
60         'label' => 'Celular',
61         'type' => 'text',
62         'value' => $funcionario[ 'Funcionario' ][ 'celular' ]
63     ) ), $this->Form->input ( 'email', array (
```

```

64         'label' => 'Email',
65         'type' => 'email',
66         'value' => $funcionario ['Funcionario'] ['email']
67     ) ), '<h3>Acesso</h3>',
68     $this->Form->input ( 'senha', array (
69         'label' => 'Senha de Acesso',
70         'type' => 'password'
71     ) ), $this->Form->input ( 'repsenha', array (
72         'label' => 'Repita a Senha de Acesso',
73         'type' => 'password'
74     ) ), $this->Form->end ( 'Atualizar' ),
75
76     $this->Html->link ( 'Voltar', array (
77         'controller' => 'Coordenacaos',
78         'action' => 'Funcionario'
79     ), array (
80         'class' => 'btn btn-back'
81     ) );
82 ?>

```

---

### B.3.7 Funcionário

---

```

1 <?php
2     echo
3         $this->Html->link ( 'Novo Funcionario', array (
4             'controller' => 'Coordenacaos',
5             'action' => 'NovoFuncionario'
6         ), array (
7             'class' => 'btn btn-primary'
8         ) ), '<br /><br /><table>',
9     $this->Html->tableHeaders(array('Nome', 'Opções', ''));
10    foreach ( $funcionario as $funcionario ) :
11        echo
12            $this->Html->tableCells ( array (
13                array (
14                    $funcionario ['Funcionario'] ['nome'],
15                    $this->Html->link ( 'Editar Funcionario', array (
16                        'controller' => 'Coordenacaos',
17                        'action' => 'EditarFuncionario',
18                        $funcionario ['Funcionario'] ['id']
19                    ) ),
20                    $this->Html->link ( 'Excluir Funcionario', array (
21                        'controller' => 'Coordenacaos',
22                        'action' => 'ExcluirFuncionario',
23                        $funcionario ['Funcionario'] ['id']
24                    ) )
25                )

```

```
26     ) );
27 endforeach;
28 echo
29     '</table>',
30     $this->Html->link ( 'Voltar', array (
31         'controller' => 'Coordenacaos',
32         'action' => 'Index'
33     ), array (
34         'class' => 'btn btn-back'
35     ) );
36 ?>
```

---

### B.3.8 Index

---

```
1 <?php
2     echo
3         $this->Html->link ( 'Aluno', array (
4             'controller' => 'Coordenacaos',
5             'action' => 'Aluno'
6         ), array (
7             'class' => 'menu'
8         ) ),
9         $this->Html->link ( 'Funcion rio', array (
10            'controller' => 'Coordenacaos',
11            'action' => 'Funcionario'
12        ), array (
13            'class' => 'menu'
14        ) ),
15        $this->Html->link ( 'Disciplina', array (
16            'controller' => 'Coordenacaos',
17            'action' => 'Disciplina'
18        ), array (
19            'class' => 'menu'
20        ) ),
21        $this->Html->link ( 'Vincular Professor/Disciplina', array (
22            'controller' => 'Coordenacaos',
23            'action' => 'Vinculo'
24        ), array (
25            'class' => 'menu'
26        ) ),
27        $this->Html->link ( 'Turma', array (
28            'controller' => 'Coordenacaos',
29            'action' => 'Turma'
30        ), array (
31            'class' => 'menu'
32        ) ),
33        $this->Html->link ( 'Sair', array (
```

```

34         'controller' => 'Autenticacao',
35         'action' => 'logout'
36     ), array (
37         'class' => 'logout'
38     ) );
39 ?>

```

---

### B.3.9 Lista de presença

---

```

1 <?php
2
3     echo
4     '<h2 class="show">Lista de Presen a - Turma: '.str_replace('0',' ',
5     explode('/', $this->here)[4]).'</h2><br /><br class="show">
6     <h1 class="show">Professor: '.str_replace('0',' ',explode('/', $this->
7     here)[5]).'</h1><br class="show"><br class="show">
8     <table>',
9     $this->Html->tableHeaders(array('Nome',
10     $this->Form->input('bimestre', array(
11         'label' => false,
12         'placeholder' => 'Mes',
13         'type' => 'text',
14         'class' => 'printInput'
15     ))),
16     $this->Form->input('bimestre', array(
17         'label' => false,
18         'placeholder' => 'Mes',
19         'type' => 'text',
20         'class' => 'printInput'
21     ))),
22     $this->Form->input('bimestre', array(
23         'label' => false,
24         'placeholder' => 'Mes',
25         'type' => 'text',
26         'class' => 'printInput'
27     ))),
28     ));
29
30     if (empty($tab_matricula)) {
31         echo $this->Html->div('error', 'N o h alunos cadastrados nessa
32         turma!');
33     }
34     else {
35         echo '<table>';
36         foreach ($tab_matricula as $tab_matricula) :
37             $i = 0;
38             echo $this->Html->tableCells(array(

```

```
36         array (
37             $stab_matricula [ 'Matricula' ][ 'nomealuno' ],
38             $this->Form->input ( 'dia', array (
39                 'label' => false ,
40                 'type' => 'text'
41             ) ),
42             $this->Form->input ( 'dia', array (
43                 'label' => false ,
44                 'type' => 'text'
45             ) ),
46             $this->Form->input ( 'dia', array (
47                 'label' => false ,
48                 'type' => 'text'
49             ) ),
50             $this->Form->input ( 'dia', array (
51                 'label' => false ,
52                 'type' => 'text'
53             ) ),
54             $this->Form->input ( 'dia', array (
55                 'label' => false ,
56                 'type' => 'text'
57             ) ),
58             '_' ,
59             $this->Form->input ( 'dia', array (
60                 'label' => false ,
61                 'type' => 'text'
62             ) ),
63             $this->Form->input ( 'dia', array (
64                 'label' => false ,
65                 'type' => 'text'
66             ) ),
67             $this->Form->input ( 'dia', array (
68                 'label' => false ,
69                 'type' => 'text'
70             ) ),
71             $this->Form->input ( 'dia', array (
72                 'label' => false ,
73                 'type' => 'text'
74             ) ),
75             $this->Form->input ( 'dia', array (
76                 'label' => false ,
77                 'type' => 'text'
78             ) ),
79             '_' ,
80             $this->Form->input ( 'dia', array (
81                 'label' => false ,
82                 'type' => 'text'
```

```

83         ) ),
84     $this->Form->input ( 'dia', array (
85         'label' => false,
86         'type' => 'text'
87     ) ),
88     $this->Form->input ( 'dia', array (
89         'label' => false,
90         'type' => 'text'
91     ) ),
92     $this->Form->input ( 'dia', array (
93         'label' => false,
94         'type' => 'text'
95     ) ),
96     $this->Form->input ( 'dia', array (
97         'label' => false,
98         'type' => 'text'
99     ) ),
100
101     )
102     ) );
103     endforeach;
104     echo '</table><br />';
105 }
106 echo $this->Html->link ( 'Imprimir', '#', array (
107     'class' => 'btn btn-primary hidden',
108     'onclick' => 'window.print()'
109 ) );
110 echo $this->Html->link ( 'Voltar', array (
111     'controller' => 'Coordenacaos',
112     'action' => 'Turma'
113 ), array (
114     'class' => 'btn btn-back hidden'
115 ) );
116 ?>

```

---

### B.3.10 Matrícula

```

1 <?php
2     echo
3         '<h3>Matricular Aluno</h3>',
4         $this->Form->create ( 'Matricula' ),
5         $this->Form->label ( 'Aluno' ),
6         '<select class="select" name="data[Matricula][idaluno]" id="
7             MatriculaIdaluno">
8             <option value=" "></option>';
9         foreach ( $nomealuno as $aluno )
10            if (!empty($aluno)){

```

```

10         echo '<option value="'. $aluno[ 'Aluno' ][ 'id' ]. '>' . $aluno[ '
11             Aluno' ][ 'nome' ]. '</option>';
12     }
13     echo '</select>',
14     $this->Form->end ( 'Matricular' );
15
16     echo
17     $this->Html->link ( 'Voltar', array (
18         'controller' => 'Coordenacaos',
19         'action' => 'Turma'
20     ), array (
21         'class' => 'btn btn-back'
22     ) );
23 ?>

```

---

### B.3.11 Nova Disciplina

---

```

1 <?php
2     echo
3     '<h3>Nova Disciplina</h3>',
4     $this->Form->create ( 'Disciplina' ),
5     $this->Form->label ( 'Disciplina' ),
6     '<select class="select" name="data[Disciplina][nome]" id="
7         DisciplinaNome">
8         <option value=""> </option>
9         <option value="Atividade Art stica Complementar">Atividade
10             Art stica Complementar</option>
11         <option value="Canto Coral">Canto Coral</option>
12         <option value="Musicaliza o">Musicaliza o</option>
13         <option value="Oficina Multimeios">Oficina Multimeios</option>
14         <option value="Percep o Musical">Percep o Musical</option
15         >
16         <option value="Pr tica de Conjunto">Pr tica de Conjunto</
17         option>
18         <optgroup label="Instrumento">
19         <option style="margin-left:20px;" value="Bateria">Bateria</
20         option>
21         <option style="margin-left:20px;" value="Cavaquinho">
22         Cavaquinho</option>
23         <option style="margin-left:20px;" value="Contrabaixo
24         el trico">Contrabaixo el trico</option>
25         <option style="margin-left:20px;" value="Flauta doce">Flauta
26         doce</option>
27         <option style="margin-left:20px;" value="Flauta transversal
28         ">Flauta transversal</option>
29         <option style="margin-left:20px;" value="Guitarra">Guitarra
30         </option>

```

```

21         <option style="margin-left:20px;" value="Percuss o">
           Percuss o </option>
22         <option style="margin-left:20px;" value="Piano">Piano</
           option>
23         <option style="margin-left:20px;" value="Saxofone">Saxofone
           </option>
24         <option style="margin-left:20px;" value="Teclado">Teclado</
           option>
25         <option style="margin-left:20px;" value="Trombone">Trombone
           </option>
26         <option style="margin-left:20px;" value="Trompa">Trompa</
           option>
27         <option style="margin-left:20px;" value="Trompete">Trompete
           </option>
28         <option style="margin-left:20px;" value="Viol o">Viol o</
           option>
29         <option style="margin-left:20px;" value="Violino">Violino</
           option>
30         <option style="margin-left:20px;" value="Violoncelo">
           Violoncelo</option>
31     </optgroup>
32 </select><br /><br />',
33 $this->Form->label ( 'Curso' ),
34 $this->Form->select ( 'curso', array (
35     'Educa o Musical' => 'Educa o Musical',
36     'T cnico' => 'T cnico'
37 ) ), '<br /><br />',
38 $this->Form->label ( 'Ciclo - Ano' ),
39 $this->Form->select ( 'ciclo', array (
40     'Inicial' => 'Inicial',
41     'Intermedi rio' => 'Intermedi rio',
42     'Complementar' => 'Complementar',
43     'Primeiro Ano' => 'Primeiro Ano',
44     'Segundo Ano' => 'Segundo Ano',
45     'Terceiro Ano' => 'Terceiro Ano',
46 ) ), '<br /><br />',
47 $this->Form->label ( 'Fase' ),
48 $this->Form->select ( 'fase', array (
49     'I' => 'I',
50     'II' => 'II',
51     'III' => 'III'
52 ) ), '<br /><br />', $this->Form->end ( 'Cadastrar' ),
53 $this->Html->link ( 'Voltar', array (
54     'controller' => 'Coordenacaos',
55     'action' => 'Disciplina'
56 ) ), array (
57     'class' => 'btn btn-back'

```

```

58         ) );
59     ?>

```

---

### B.3.12 Nova Turma

---

```

1 <?php
2
3     if ( empty($professor) || empty($disciplina) ){
4     }
5
6     else {
7
8     echo
9         '<h3>Nova Turma</h3>';
10    $this->Form->create ( 'Turma' ),
11    $this->Form->input ( 'nome', array (
12        'label' => false ,
13        'type' => 'text' ,
14        'placeholder' => 'Nome'
15    ) ), $this->Form->label ( 'Professor' ),
16    '<select class="select" name="data[Turma][masp]" id="TurmaMasp">
17        <option value=""></option>';
18    foreach ( $professor as $professor )
19        echo '<option value="'. $professor [ 'Funcionario' ] [ 'masp' ] . '">'
20            . $professor [ 'Funcionario' ] [ 'nome' ] . '</option>';
21    echo '</select><br />',
22
23    $this->Form->label ( 'Disciplina' ),
24    '<select class="select" name="data[Turma][coddisc]" id="
25        TurmaCoddisc">
26        <option value=""></option>';
27    foreach ( $disciplina as $disciplina )
28        if (!empty($disciplina)){
29            echo '<option value="'. $disciplina [ 'Disciplina' ] [ 'id' ] . '">' .
30                $disciplina [ 'Disciplina' ] [ 'nome' ] . ' - ' . $disciplina [ '
31                Disciplina' ] [ 'ciclo' ] . ' - ' . $disciplina [ 'Disciplina' ] [ 'fase' ]
32                . '</option>';
33        }
34    echo '</select><br />',
35    $this->Form->input ( 'sala', array (
36        'label' => false ,
37        'type' => 'number' ,
38        'placeholder' => 'N mero da sala'
39    ) ), $this->Form->input ( 'horaaula', array (
40        'label' => false ,
41        'type' => 'text' ,
42        'placeholder' => 'Hor rio da Aula - exemplo: 22:30'

```

```

38     ) ); $options = array(
39         'Segunda-Feira' => 'Segunda-Feira',
40         'Ter a-Feira' => 'Ter a-Feira',
41         'Quarta-Feira' => 'Quarta-Feira',
42         'Quinta-Feira' => 'Quinta-Feira',
43         'Sexta-Feira' => 'Sexta-Feira',
44     ); echo $this->Form->label ( 'Dia da Semana' ),
45     $this->Form->select ( 'diaaula', $options ),
46     '<br /><br />', $this->Form->end ( 'Cadastrar' );
47
48 }
49 echo $this->Html->link ( 'Voltar', array (
50     'controller' => 'Coordenacaos',
51     'action' => 'Turma'
52 ), array (
53     'class' => 'btn btn-back'
54 ) );
55 ?>

```

---

### B.3.13 Novo Aluno

---

```

1 <?php
2     echo
3     $this->Form->create ( 'Aluno' ),
4     '<h3>Dados Pessoais</h3>',
5     $this->Form->input ( 'nome', array (
6         'label' => false,
7         'required' => true,
8         'type' => 'text',
9         'placeholder' => 'Nome Completo'
10    ) ), $this->Form->input ( 'rg', array (
11        'label' => false,
12        'type' => 'text',
13        'placeholder' => 'Identidade'
14    ) ), $this->Form->input ( 'cpf', array (
15        'label' => false,
16        'type' => 'number',
17        'placeholder' => 'CPF'
18    ) ), $this->Form->label ( 'Data de Nascimento' ),
19    $this->Form->input ( 'datanasc', array(
20        'label' => false,
21        'required' => true,
22        'type' => 'date',
23        'dateFormat' => 'DMY',
24        'minYear' => date( 'Y' ) - 50,
25        'maxYear' => date( 'Y' ) - 10,
26        'style' => 'margin-left: -0.5px;'

```

```
27     ) ), $this->Form->label ( 'Sexo' ),
28     $this->Form->select ( 'sexo', array (
29         'M' => 'Masculino',
30         'F' => 'Feminino'
31     ) ), $this->Form->input ( 'naturalidade', array (
32         'label' => false ,
33         'required' => true ,
34         'type' => 'text' ,
35         'placeholder' => 'Naturalidade'
36     ) ), $this->Form->input ( 'ufnaturalidade', array (
37         'label' => false ,
38         'required' => true ,
39         'type' => 'text' ,
40         'placeholder' => 'UF - Naturalidade'
41     ) ), $this->Form->input ( 'mae', array (
42         'label' => false ,
43         'required' => true ,
44         'type' => 'text' ,
45         'placeholder' => 'Nome da Mãe'
46     ) ), $this->Form->input ( 'pai', array (
47         'label' => false ,
48         'type' => 'text' ,
49         'placeholder' => 'Nome do Pai'
50     ) ),
51     $this->Form->label ( 'Turno' ),
52     $this->Form->select ( 'turno', array (
53         'Matutino' => 'Matutino' ,
54         'Vespertino' => 'Vespertino' ,
55         'Noturno' => 'Noturno'
56     ) ), $this->Form->input ( 'escola', array (
57         'label' => false ,
58         'type' => 'text' ,
59         'placeholder' => 'Escola'
60     ) ), $this->Form->input ( 'anoescolar', array (
61         'label' => false ,
62         'type' => 'text' ,
63         'placeholder' => 'Escolaridade'
64     ) ), $this->Form->input ( 'obs', array (
65         'label' => false ,
66         'type' => 'textarea' ,
67         'placeholder' => 'Observação'
68     ) ), '<br /><h3>Endereço</h3>',
69     $this->Form->input ( 'rua', array (
70         'label' => false ,
71         'required' => true ,
72         'type' => 'text' ,
73         'placeholder' => 'Rua'
```

```

74     ) ), $this->Form->input ( 'numero', array (
75         'label' => false,
76         'required' => true,
77         'type' => 'number',
78         'placeholder' => 'Numero'
79     ) ), $this->Form->input ( 'bairro', array (
80         'label' => false,
81         'required' => true,
82         'type' => 'text',
83         'placeholder' => 'Bairro'
84     ) ), $this->Form->input ( 'cidade', array (
85         'label' => false,
86         'required' => true,
87         'type' => 'text',
88         'placeholder' => 'Cidade'
89     ) ), $this->Form->input ( 'uf', array (
90         'label' => false,
91         'required' => true,
92         'type' => 'text',
93         'placeholder' => 'UF'
94     ) ), '<br /><br /><h3>Contato</h3>',
95     $this->Form->input ( 'telefone', array (
96         'label' => false,
97         'type' => 'text',
98         'placeholder' => 'Telefone'
99     ) ), $this->Form->input ( 'celular', array (
100        'label' => false,
101        'type' => 'text',
102        'placeholder' => 'Celular'
103    ) ), $this->Form->input ( 'email', array (
104        'label' => false,
105        'type' => 'email',
106        'placeholder' => 'Email'
107    ) ), $this->Form->end ( 'Cadastrar' ),
108    $this->Html->link ( 'Voltar', array (
109        'controller' => 'Coordenacaos',
110        'action' => 'Aluno'
111    ) ), array (
112        'class' => 'btn btn-back'
113    ) );
114 ?>

```

---

### B.3.14 Novo Funcionário

```

1 <?php
2     echo
3     $this->Html->script ( 'senha' ),

```

```

4     $this->Form->create ( 'Funcionario' ),
5     '<h3>Dados Pessoais</h3>',
6     $this->Form->input ( 'nome', array (
7         'label' => false ,
8         'required' => true ,
9         'type' => 'text' ,
10        'placeholder' => 'Nome'
11    ) ), $this->Form->label ( 'Cargo' ),
12    $this->Form->select ( 'cargo', array (
13        'Coordenacaos' => 'Coordenador(a)',
14        'Professors' => 'Professor(a)',
15        'Coordenacaos' => 'Auxiliar de Secretaria (ATB)'
16    ) ), $this->Form->input ( 'masp', array (
17        'label' => false ,
18        'required' => true ,
19        'type' => 'number' ,
20        'placeholder' => 'MASP'
21    ) ), $this->Form->input ( 'rg', array (
22        'label' => false ,
23        'required' => true ,
24        'type' => 'text' ,
25        'placeholder' => 'Identidade'
26    ) ), $this->Form->input ( 'cpf', array (
27        'label' => false ,
28        'required' => true ,
29        'type' => 'number' ,
30        'placeholder' => 'CPF'
31    ) ), $this->Form->input ( 'datanasc', array(
32        'label' => 'Data de Nascimento',
33        'type' => 'date' ,
34        'dateFormat' => 'DMY' ,
35        'minYear' => date( 'Y' ) - 50 ,
36        'maxYear' => date( 'Y' ) - 10 ,
37        'style' => 'margin-left: -0.5px;'
38    ) ), $this->Form->label ( 'Sexo' ),
39    $this->Form->select ( 'sexo', array (
40        'M' => 'Masculino' ,
41        'F' => 'Feminino'
42    ) ), '<br /><br /><h3>Endere o</h3>',
43    $this->Form->input ( 'rua', array (
44        'label' => false ,
45        'required' => true ,
46        'type' => 'text' ,
47        'placeholder' => 'Rua'
48    ) ), $this->Form->input ( 'numero', array (
49        'label' => false ,
50        'required' => true ,

```

```

51         'type' => 'number',
52         'placeholder' => 'Numero'
53     ) ), $this->Form->input ( 'bairro', array (
54         'label' => false,
55         'required' => true,
56         'type' => 'text',
57         'placeholder' => 'Bairro'
58     ) ), $this->Form->input ( 'cidade', array (
59         'label' => false,
60         'required' => true,
61         'type' => 'text',
62         'placeholder' => 'Cidade'
63     ) ), $this->Form->input ( 'uf', array (
64         'label' => false,
65         'required' => true,
66         'type' => 'text',
67         'placeholder' => 'UF'
68     ) ), '<br /><br /><h3>Contato</h3>',
69     $this->Form->input ( 'telefone', array (
70         'label' => false,
71         'type' => 'text',
72         'placeholder' => 'Telefone'
73     ) ), $this->Form->input ( 'celular', array (
74         'label' => false,
75         'type' => 'text',
76         'placeholder' => 'Celular'
77     ) ), $this->Form->input ( 'email', array (
78         'label' => false,
79         'type' => 'email',
80         'placeholder' => 'Email'
81     ) ), '<br /><br /><h3>Acesso</h3>',
82     $this->Form->input ( 'senha', array (
83         'label' => false,
84         'required' => true,
85         'type' => 'password',
86         'placeholder' => 'Senha de Acesso'
87     ) ), $this->Form->input ( 'repsenha', array (
88         'label' => false,
89         'required' => true,
90         'type' => 'password',
91         'placeholder' => 'Repita Senha de Acesso'
92     ) ), $this->Form->end ( 'Cadastrar' ),
93
94     $this->Html->link ( 'Voltar', array (
95         'controller' => 'Coordenacaos',
96         'action' => 'Funcionario'
97     ) ), array (

```

```

98         'class' => 'btn btn-back'
99     ) );
100 ?>

```

---

### B.3.15 Novo Vínculo

---

```

1 <?php
2     echo
3     '<h3>Vincular Professor/Disciplina</h3>',
4     $this->Form->create ( 'Vinculo' ),
5     $this->Form->label ( 'Professor' ),
6     '<select class="select" name="nomeprof">
7         <option value=""></option>';
8     foreach ( $prof as $prof )
9         echo '<option value="' . $prof[ 'Funcionario' ][ 'nome' ] . '">
10             ' . $prof[ 'Funcionario' ][ 'nome' ] . '</option>';
11
12     echo $this->Form->label ( 'Disciplina' ),
13     '<select class="select" name="nomedisc">
14         <option value=""></option>';
15     foreach ( $disciplina as $disciplina )
16         if (!empty($disciplina)){
17             echo '<option value="' . $disciplina[ 'Disciplina' ][ 'nome' ] . '
18                 '>' . $disciplina[ 'Disciplina' ][ 'nome' ] . ' - ' . $disciplina[
19                 'Disciplina' ][ 'ciclo' ] . ' - ' . $disciplina[ 'Disciplina' ][ '
20                 fase' ] . '</option>';
21         }
22     echo '</select>';
23
24     echo '<br />', $this->Form->end ( 'Cadastrar' ),
25     $this->Html->link ( 'Voltar', array (
26         'controller' => 'Coordenacaos',
27         'action' => 'Vinculo'
28     ), array (
29         'class' => 'btn btn-back'
30     ) );
31 ?>

```

---

### B.3.16 Turma

---

```

1 <?php
2
3     echo
4     $this->Html->link ( 'Nova Turma', array (
5         'controller' => 'Coordenacaos',

```

```

6         'action' => 'NovaTurma'
7     ), array (
8         'class' => 'btn btn-primary'
9     ) ),
10 '<br /><br /><table>',
11 $this->Html->tableHeaders(array('Turma', 'Professor', 'Opções', ''));
12 foreach ( $turma as $turma ) :
13     echo
14         $this->Html->tableCells ( array (
15             array (
16                 $turma ['Turma'] ['nome'],
17                 $turma ['Turma'] ['profnome'],
18                 $this->Html->link ( 'Excluir Turma', array (
19                     'controller' => 'Coordenacaos',
20                     'action' => 'ExcluirTurma',
21                     $turma ['Turma'] ['id']
22                 ) ),
23                 $this->Html->link ( 'Consultar', array (
24                     'controller' => 'Coordenacaos',
25                     'action' => 'VisualizarMatricula',
26                     $turma ['Turma'] ['id'],
27                     str_replace(' ', '0', $turma ['Turma'] ['
28                         nome']),
29                     str_replace(' ', '0', $turma ['Turma'] ['
30                         profnome'])
31                 ) )
32             )
33         ) );
34 endforeach;
35 echo
36     '</table>',
37     $this->Html->link ( 'Voltar', array (
38         'controller' => 'Coordenacaos',
39         'action' => 'Index'
40     ), array (
41         'class' => 'btn btn-back'
42     ) );
43 ?>

```

### B.3.17 Vínculo

```

1 <?php
2     echo
3         $this->Html->link ( 'Novo Vínculo', array (
4             'controller' => 'Coordenacaos',
5             'action' => 'NovoVinculo'
6         ), array (

```

```

7         'class' => 'btn btn-primary'
8     ) ),
9     '<br /><br /><table>',
10    $this->Html->tableHeaders( array( 'Professor', 'Disciplina', 'Opções' ) )
11    ;
12    foreach ( $vinculo as $vinculo ) :
13        echo
14            $this->Html->tableCells ( array (
15                array (
16                    $vinculo [ 'Vinculo' ] [ 'nomeprof' ],
17                    $vinculo [ 'Vinculo' ] [ 'nomedisc' ],
18                    $this->Html->link ( 'Excluir Vinculo', array (
19                        'controller' => 'Coordenacao',
20                        'action' => 'ExcluirVinculo',
21                        $vinculo [ 'Vinculo' ] [ 'id' ]
22                    ) )
23                ) )
24            );
25        endforeach;
26        echo
27            '</table>',
28
29
30
31        $this->Html->link ( 'Voltar', array (
32            'controller' => 'Coordenacao',
33            'action' => 'Index'
34        ), array (
35            'class' => 'btn btn-back'
36        ) );
37    ?>

```

---

### B.3.18 Visualizar Matrícula

---

```

1 <?php
2
3     echo $this->Html->link ( 'Matricular Aluno', array (
4         'controller' => 'Coordenacao',
5         'action' => 'Matricula',
6         $id
7     ), array (
8         'class' => 'btn btn-primary hidden'
9     ) );
10    echo $this->Html->link ( 'Imprimir', '#', array (
11        'class' => 'btn btn-primary hidden',
12        'onclick' => 'window.print()'

```

```

13     ) );
14     echo $this->Html->link ( 'Lista de Presen a', array (
15         'controller' => 'Coordenacaos',
16         'action' => 'ListaPresenca',
17         $id,
18         explode('/', $this->here)[4],
19         explode('/', $this->here)[5]
20     ), array (
21         'class' => 'btn btn-primary hidden'
22     ) ),
23     '<br /><br />
24     <h2 class="show">Turma: '.str_replace('0', ' ', explode('/', $this->here)
25         [4]). '</h2><br />
26     <h2 class="show">Professor: '.str_replace('0', ' ', explode('/', $this->
27         here)[5]). '</h2>
28     <table>',
29     $this->Html->tableHeaders(array('Nome', 'Op es'), array ('class' =>
30         'hidden'));
31
32     if (empty($tab_matricula)) {
33         echo $this->Html->div('error', 'N o h alunos cadastrados nessa
34         turma!');
35     }
36     else {
37         echo '<table>';
38         foreach ( $tab_matricula as $tab_matricula ) :
39             echo $this->Html->tableCells ( array (
40                 array (
41                     $tab_matricula ['Matricula'] ['nomealuno'],
42                     $this->Html->link ( 'Desmatricular', array (
43                         'controller' => 'Coordenacaos',
44                         'action' => 'Desmatricular',
45                         $tab_matricula ['Matricula'] ['idaluno'],
46                         $id
47                     ), array (
48                         'class' => 'hidden'
49                     ) )
50                 )
51             ) );
52         endforeach;
53         echo '</table><br />';
54     }
55     echo $this->Html->link ( 'Voltar', array (
56         'controller' => 'Coordenacaos',
57         'action' => 'Turma'
58     ), array (

```

```

56         'class' => 'btn btn-back hidden'
57     ) );
58 ?>

```

---

## B.4 View Professores

### B.4.1 Horário

```

1 <?php
2     echo '<h3>Hor rio das Aulas</h3>', '<table>',
3     $this->Html->tableHeaders(array('Turma', 'Sala', 'Hora', 'Dia'));
4     foreach ( $turmas as $turma ) :
5         echo
6             $this->Html->tableCells ( array (
7                 array (
8                     $turma ['Turma'] ['nome'],
9                     $turma ['Turma'] ['sala'],
10                    $turma ['Turma'] ['horaaula'],
11                    $turma ['Turma'] ['diaaula']
12                )
13            ) );
14     endforeach;
15     echo
16     '</table><br />',
17     $this->Html->link ( 'Voltar', array (
18         'controller' => 'Professors',
19         'action' => 'Index'
20     ), array (
21         'class' => 'btn btn-back'
22     ) );
23 ?>

```

---

### B.4.2 Index

```

1 <?php
2     echo
3     $this->Html->link ( 'Visualizar Hor rios', array (
4         'controller' => 'Professors',
5         'action' => 'Horario'
6     ), array (
7         'class' => 'menu'
8     ) ), $this->Html->link ( 'Lan ar Notas e Frequ ncia', array (
9         'controller' => 'Professors',
10        'action' => 'SelecionarTurma'
11    ), array (

```

```

12         'class' => 'menu'
13     ) ), $this->Html->link ( 'Sair', array (
14         'controller' => 'Autenticacao',
15         'action' => 'logout'
16     ), array (
17         'class' => 'logout'
18     ) );
19 ?>

```

### B.4.3 Notas e Frequência

```

1 <?php
2     echo '<h3>Lançar Notas e Frequências</h3>', '<table>',
3     $this->Html->tableHeaders ( array (
4         'Nome',
5         '1 - Bi - Faltas',
6         '1 - Bi - Notas',
7         '2 - Bi - Faltas',
8         '2 - Bi - Notas',
9         '3 - Bi - Faltas',
10        '3 - Bi - Notas',
11        '4 - Bi - Faltas',
12        '4 - Bi - Notas',
13        'Total de Faltas',
14        'Total de Notas',
15        'Resultado Final',
16    ) );
17    $count = count($idaluno);
18    foreach ( $avaliacao as $k => $avaliacao ){
19        if ($count == $k) break; //PARA QND O TAMANHO DO VETOR ATINGE
20        if($avaliacao['Avaliacao'][$idaluno] == $idaluno[$k])
21            echo $this->Form->create ( 'Avaliacao' ),
22            $this->Form->input ( 'Avaliacao.id', array(
23                'value' => $avaliacao['Avaliacao'][$id],
24                'type' => 'hidden',
25            ) ),
26            $this->Form->input ( 'Avaliacao.idaluno', array(
27                'value' => $avaliacao['Avaliacao'][$idaluno],
28                'type' => 'hidden',
29            ) ),
30            $this->Html->tableCells ( array (
31                $aluno[$k]['Aluno']['nome'],
32                $this->Form->input ( 'Avaliacao.b1faltas', array(
33                    'value' => $avaliacao['Avaliacao'][$b1faltas],
34                    'type' => 'text',
35                    'div' => false,
36                    'label' => false

```

```
37         ) ), $this->Form->input( 'Avaliacao.b1notas', array(
38             'value' => $avaliacao[ 'Avaliacao' ][ 'b1notas' ],
39             'type' => 'text',
40             'div' => false,
41             'label' => false
42         ) ), $this->Form->input( 'Avaliacao.b2faltas', array(
43             'value' => $avaliacao[ 'Avaliacao' ][ 'b2faltas' ],
44             'type' => 'text',
45             'div' => false,
46             'label' => false
47         ) ), $this->Form->input( 'Avaliacao.b2notas', array(
48             'value' => $avaliacao[ 'Avaliacao' ][ 'b2notas' ],
49             'type' => 'text',
50             'div' => false,
51             'label' => false
52         ) ), $this->Form->input( 'Avaliacao.b3faltas', array(
53             'value' => $avaliacao[ 'Avaliacao' ][ 'b3faltas' ],
54             'type' => 'text',
55             'div' => false,
56             'label' => false
57         ) ), $this->Form->input( 'Avaliacao.b3notas', array(
58             'value' => $avaliacao[ 'Avaliacao' ][ 'b3notas' ],
59             'type' => 'text',
60             'div' => false,
61             'label' => false
62         ) ), $this->Form->input( 'Avaliacao.b4faltas', array(
63             'value' => $avaliacao[ 'Avaliacao' ][ 'b4faltas' ],
64             'type' => 'text',
65             'div' => false,
66             'label' => false
67         ) ), $this->Form->input( 'Avaliacao.b4notas', array(
68             'value' => $avaliacao[ 'Avaliacao' ][ 'b4notas' ],
69             'type' => 'text',
70             'div' => false,
71             'label' => false
72         ) ), $this->Form->input( 'Avaliacao.ffaltas', array(
73             'value' => $avaliacao[ 'Avaliacao' ][ 'ffaltas' ],
74             'type' => 'text',
75             'div' => false,
76             'label' => false
77         ) ), $this->Form->input( 'Avaliacao.fnotas', array(
78             'value' => $avaliacao[ 'Avaliacao' ][ 'fnotas' ],
79             'type' => 'text',
80             'div' => false,
81             'label' => false
82         ) ), $this->Form->input( 'Avaliacao.resultadofinal',
            array(
```

```

83         'value' => $avaliacao[ 'Avaliacao' ][ '
            resultadofinal' ],
84         'type' => 'text',
85         'div' => false,
86         'label' => false
87     ) ),
88     $this->Form->end( 'Lan ar Notas' ),
89     ) );
90 }
91 echo '</table><br />',
92     $this->Html->link ( 'Voltar', array (
93         'controller' => 'Professors',
94         'action' => 'Index'
95     ), array (
96         'class' => 'btn btn-back'
97     ) );
98 ?>

```

---

#### B.4.4 Selecionar Turma

---

```

1 <?php
2 echo '<h3>Selecione a turma</h3>',
3     $this->Form->create( 'Turma' ),
4     $this->Form->label( 'Turma' ),
5     '<select class="select" name="data[Turma][id]" id="TurmaId">
6         <option value=""></option>;
7         foreach ( $turma as $turma )
8             echo '<option value="' . $turma[ 'Turma' ][ 'id' ] . '">' .
                $turma[ 'Turma' ][ 'nome' ] . '</option>;
9         echo '</select>',
10
11     $this->Form->end( 'Consultar' ),
12     $this->Html->link( 'Voltar', array (
13         'controller' => 'Professors',
14         'action' => 'Index'
15     ), array (
16         'class' => 'btn btn-back'
17     ) );
18 ?>

```

---

# ANEXO C – Controller

## C.1 Controller Alunos

---

```

1 <?php
2 class AlunosController extends ApplicationController {
3     public $name = 'Alunos';
4     public $uses = array ('Aluno', 'Turma', 'Avaliacao', 'Matricula');
5
6     public function Index(){
7     }
8
9     public function Horario(){
10        $this->VerificaLogin();
11
12        $sessao = $this->Session->read(); //PEGA A MATRICULA
13        $aluno = $this->Aluno->findByMatricula($sessao['session_usuario']);
14        // PEGA O ALUNO REFERENTE A MATRICULA
15        $sid = $this->Matricula->findAllByIdaluno($aluno['Aluno']['id']); //
16        PEGA AS TUPLAS COM ID DO ALUNO
17        $idturn = $this->RetornaIndice($sid, 'Matricula', 'idturn');
18
19        $turma = $this->Turma->find('all');
20
21        $this->set(compact('turma'));
22        $this->set(compact('idturn'));
23    }
24    public function Turma(){
25        $this->VerificaLogin();
26
27        $sessao = $this->Session->read(); //PEGA A MATRICULA
28        $aluno = $this->Aluno->findByMatricula($sessao['session_usuario']);
29        // PEGA O ALUNO REFERENTE A MATRICULA
30        $avaliacao = $this->Avaliacao->findAllByIdaluno($aluno['Aluno']['id
31        ']); //PEGA AS TUPLAS COM ID DO ALUNO
32
33        $idturn = $this->RetornaIndice($avaliacao, 'Avaliacao', 'idturn');
34        $turma = $this->Turma->find('all');
35
36        $this->set(compact('avaliacao'));
37        $this->set(compact('turma'));
38        $this->set(compact('idturn'));
39    }
40 }

```

37 ?&gt;

---

## C.2 Controller App

---

```

1 <?php
2 /**
3  * Application level Controller
4  *
5  * This file is application-wide controller file. You can put all
6  * application-wide controller-related methods here.
7  *
8  * CakePHP(tm) : Rapid Development Framework (http://cakephp.org)
9  * Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
10 *
11 * Licensed under The MIT License
12 * For full copyright and license information, please see the LICENSE.txt
13 * Redistributions of files must retain the above copyright notice.
14 *
15 * @copyright      Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
16 * @link           http://cakephp.org CakePHP(tm) Project
17 * @package        app.Controller
18 * @since          CakePHP(tm) v 0.2.9
19 * @license        http://www.opensource.org/licenses/mit-license.php MIT
20 *                License
21 */
22 App::uses('Controller', 'Controller');
23
24 /**
25  * Application Controller
26  *
27  * Add your application-wide methods in the class below, your controllers
28  * will inherit them.
29  *
30  * @package        app.Controller
31  * @link           http://book.cakephp.org/2.0/en/controllers.html#the-app-controller
32 */
33 class AppController extends Controller {
34
35     public function CriaAvaliacao($id, $turma){
36         $data = array(
37             'idaluno' => $id,
38             'idturm'  => $turma
39         );

```

```

40
41     $this->Avaliacao->save($data);
42     return $this->Avaliacao->id;
43 }
44
45 public function VerificaLogin(){
46     $dados_sessao = $this->Session->read('session_id');
47
48     if($dados_sessao==null){
49         $this->redirect(array('controller' => 'Autenticacao', 'action'
50             => 'logout')); // REDIRECIONANDO
51     }
52
53     public function RetornaIndice($id, $banco, $coluna){
54         $indices = array();
55         foreach ( $id as $id ):
56             array_push($indices, $id[$banco][$coluna]);
57         endforeach;
58         return $indices;
59     }
60 }

```

## C.3 Controller Autenticações

```

1 <?php
2 class AutenticacaosController extends ApplicationController {
3     public $name = 'Autenticacaos';
4     public $uses = array ( 'Autenticacao' );
5     public function login() {
6         if ($this->request->is ( 'post' )) {
7             $usuario = $this->Autenticacao->findByUsuario($this->request->
8                 data['Login']['usuario']);
9             if ($usuario) {
10                if ($usuario ['Autenticacao'] ['senha'] == $this->request->
11                    data['Login']['senha']) {
12                    $this->Session->write ( 'session_id', $usuario [ '
13                        Autenticacao' ] [ 'id' ] );
14                    $this->Session->write ( 'session_usuario', $usuario [ '
15                        Autenticacao' ] [ 'usuario' ] );
16                    $this->redirect ( array (
17                        'controller' => $usuario [ 'Autenticacao' ] [ '
18                            contracess' ],
19                        'action' => 'index'
20                    ) );
21                } else {

```

```

17         $this->Session->setFlash ( 'Usu rio ou Senha incorreto
18             (a) !' );
19     }
20     } else {
21         $this->Session->setFlash ( 'Usu rio ou Senha incorreto(a)
22             !' );
23     }
24 }
25 public function logout() {
26     if ($this->Session->valid ()) {
27         $this->Session->destroy ();
28         $this->redirect ( array (
29             'controller' => 'Autenticacao',
30             'action' => 'login'
31         ) );
32     }
33 }
34 ?>

```

## C.4 Controller Coordenação

```

1 <?php
2 class CoordenacaoController extends ApplicationController {
3     public $name = 'Coordenacao';
4     public $uses = array ( 'Aluno', 'Vinculo', 'Funcionario', 'Disciplina',
5         'Turma', 'Avaliacao', 'Matricula', 'Autenticacao' );
6     public function Index() {
7         $this->VerificaLogin ();
8     }
9     public function Aluno(){
10        $this->VerificaLogin ();
11        $aluno = $this->Aluno->find ( 'all', array ( 'order' => array ( 'Aluno.
12            nome' => 'asc' ) ) );
13        $this->set ( compact ( 'aluno' ) );
14    }
15    public function NovoAluno(){
16        $this->VerificaLogin ();
17
18        if ( $this->request->is ( 'post' ) ) {
19            $this->request->data [ 'Aluno' ] [ 'matricula' ] = substr ( $this->
20                request->data [ 'Aluno' ] [ 'nome' ], 0, 4 ) . date ( 'i' ) . date ( 's' );
21            if ( $this->Aluno->save ( $this->request->data ) ) {

```

```
21         $data = array(  
22             'usuario'      => $this->request->data['Aluno']['  
                matricula'],  
23             'senha'       => $this->request->data['Aluno']['  
                matricula'],  
24             'contracess'  => 'Alunos'  
25         );  
26  
27         $this->Autenticacao->save($data);  
28  
29         $this->Session->setFlash('Aluno cadastrado com sucesso!', '  
                default', array('class' => 'success'));  
30         $this->redirect(array('controller' => 'Coordenacao', '  
                action' => 'Aluno')); // REDIRECIONANDO  
31     }  
32     else {  
33         $this->Session->setFlash('N o foi poss vel cadastrar o  
                aluno!');  
34     }  
35 }  
36 }  
37 public function EditarAluno($id){  
38     $this->VerificaLogin();  
39  
40     $this->set('aluno', $this->Aluno->findById($id));  
41  
42     if($this->request->data){  
43         if($this->Aluno->save($this->request->data) {  
44             $this->Session->setFlash('Aluno alterado com sucesso!', '  
                default', array('class' => 'success'));  
45             $this->redirect(array('controller' => 'Coordenacao', '  
                action' => 'Aluno')); // REDIRECIONANDO  
46         }  
47         else{  
48             $this->Session->setFlash('N o foi poss vel alterar o  
                aluno!');  
49         }  
50     }  
51 }  
52 public function ExcluirAluno($id){  
53     $this->VerificaLogin();  
54  
55     if($this->Aluno->delete($id) {  
56         $this->Session->setFlash('Aluno deletado com sucesso!', '  
                default', array('class' => 'success'));  
57         $this->redirect(array('controller' => 'Coordenacao', 'action'  
                => 'Aluno'));
```

```

58     }
59     else{
60         $this->Session->setFlash('N o foi poss vel deletar o aluno!')
        ;
61     }
62 }
63
64 public function Funcionario(){
65     $this->VerificaLogin();
66     $funcionario = $this->Funcionario->find('all', array('order' =>
        array('Funcionario.nome' => 'asc')));
67     $this->set(compact('funcionario'));
68 }
69 public function NovoFuncionario(){
70     $this->VerificaLogin();
71     if ($this->request->is('post')) {
72
73         $id = $this->request->data['Funcionario']['masp'];
74         $verifica = $this->Funcionario->find ( 'all', array (
75             'conditions' => array (
76                 'Funcionario.masp' => $id ,
77             )
78         ) );
79
80         $senha = $this->request->data['Funcionario']['senha'];
81         $repsenha = $this->request->data['Funcionario']['repsenha'];
82
83         if ($senha == $repsenha) {
84             if ($verifica){
85                 $this->Session->setFlash('Funcionario(a) j cadastrado
            !');
86             }
87             else {
88                 if ($this->Funcionario->save($this->request->data)) {
89
90                     $data = array(
91                         'usuario' => $this->request->data[ '
            Funcionario' ][ 'masp' ],
92                         'senha' => $this->request->data[ '
            Funcionario' ][ 'senha' ],
93                         'contracess' => $this->request->data[ '
            Funcionario' ][ 'cargo' ]
94                     );
95
96                     $this->Autenticacao->save($data);
97                     $this->Session->setFlash('Funcionario cadastrado
            com sucesso!', 'default', array('class' => '

```

```

100         success');
98         $this->redirect(array('controller' => 'Coordenacoes
101         ', 'action' => 'Funcionario'));
99     }
100     else {
101         $this->Session->setFlash('N o foi poss vel
102         cadastrar o funcionario!');
103     }
104 }
105 else {
106     $this->Session->setFlash('As senhas n o coincidem!');
107 }
108 }
109 }
110 public function EditarFuncionario($id){
111     $this->VerificaLogin();
112
113     $this->set('funcionario', $this->Funcionario->findById($id));
114     if($this->request->data){
115         if($this->Funcionario->save($this->request->data)) {
116             $this->Session->setFlash('Funcionario alterado com sucesso!
117             ', 'default', array('class' => 'success'));
118             $this->redirect(array('controller' => 'Coordenacoes', '
119             action' => 'Funcionario')); // REDIRECIONANDO
120         }
121         else{
122             $this->Session->setFlash('N o foi possivel alterar o
123             funcionario!');
124         }
125     }
126 }
127 public function ExcluirFuncionario($id){
128     $this->VerificaLogin();
129
130     $funcionario = $this->Funcionario->findById($id);
131     $autenticacao_funcionario = $this->Autenticacao->findByUsuario(
132     $funcionario['Funcionario']['masp']);
133
134     if($this->Funcionario->delete($id)) {
135         $this->Autenticacao->delete($autenticacao_funcionario['
136         Autenticacao']['id']);
137         $this->Session->setFlash('Funcionario deletado com sucesso!', '
138         default', array('class' => 'success'));
139         $this->redirect(array('controller' => 'Coordenacoes', 'action'
140         => 'Funcionario'));
141     }

```

```

135     else{
136         $this->Session->setFlash('N o foi possivel deletar o
            funcionario!');
137         $this->redirect(array('controller' => 'Coordenacaos', 'action'
            => 'Funcionario'));
138     }
139 }
140
141 public function Disciplina(){
142     $this->VerificaLogin();
143     $disciplina = $this->Disciplina->find('all');
144     $this->set(compact('disciplina'));
145 }
146 public function NovaDisciplina(){
147     $this->VerificaLogin();
148     if ($this->request->is('post')) {
149         if ($this->Disciplina->save($this->request->data)) {
150             $this->Session->setFlash('Disciplina cadastrada com sucesso
                !', 'default', array('class' => 'success'));
151             $this->redirect(array('controller' => 'Coordenacaos', '
                action' => 'Disciplina'));
152         }
153         else {
154             $this->Session->setFlash('N o foi poss vel cadastrar a
                disciplina!');
155         }
156     }
157 }
158 public function EditarDisciplina($id){
159     $this->VerificaLogin();
160
161     $this->set('disciplina', $this->Disciplina->findById($id));
162     if($this->request->data){
163         if($this->Disciplina->save($this->request->data)) {
164             $this->Session->setFlash('Disciplina alterada com sucesso!'
                , 'default', array('class' => 'success'));
165             $this->redirect(array('controller' => 'Coordenacaos', '
                action' => 'Disciplina')); // REDIRECIONANDO
166         }
167         else{
168             $this->Session->setFlash('N o foi poss vel alterar a
                disciplina!');
169         }
170     }
171 }
172 public function ExcluirDisciplina($id){
173     $this->VerificaLogin();

```

```
174
175     if ($this->Disciplina->delete($id)) {
176         $this->Session->setFlash('Disciplina deletada com sucesso!', '
177             default', array('class' => 'success'));
178         $this->redirect(array('controller' => 'Coordenacao', 'action'
179             => 'Disciplina'));
180     }
181     else{
182         $this->Session->setFlash('N o foi poss vel deletar a
183             Disciplina!');
184     }
185 }
186
187 public function Vinculo(){
188     $this->VerificaLogin();
189     $vinculo = $this->Vinculo->find('all', array('order' => array('
190         Vinculo.nomeprof' => 'asc')));
191     $this->set(compact('vinculo'));
192 }
193
194 public function NovoVinculo(){
195     $this->VerificaLogin();
196
197     $prof = $this->Funcionario->find('all');
198     $this->set(compact('prof'));
199
200     $disciplina = $this->Disciplina->find('all');
201     $this->set(compact('disciplina'));
202
203     if ($this->request->is('post')) {
204         if ($this->Vinculo->save($this->request->data)) {
205             $this->Session->setFlash('Professor cadastrado com sucesso!
206                 ', 'default', array('class' => 'success'));
207             $this->redirect(array('controller' => 'Coordenacao', '
208                 action' => 'Vinculo'));
209         }
210         else {
211             $this->Session->setFlash('N o foi poss vel cadastrar o
212                 professor!');
213         }
214     }
215 }
216
217 public function ExcluirVinculo($id){
218     $this->VerificaLogin();
219
220     if($this->Vinculo->delete($id)) {
221         $this->Session->setFlash('Professor desvinculado com sucesso!',
222             'default', array('class' => 'success'));
223     }
```

```

213         $this->redirect(array('controller' => 'Coordenacaos', 'action'
214             => 'Vinculo'));
215     }
216     else{
217         $this->Session->setFlash('N o foi poss vel desvincular o
218             professor!');
219     }
220 }
221
222 public function Turma(){
223     $this->VerificaLogin();
224     $turma = $this->Turma->find('all');
225     $professor = $this->Funcionario->find('all');
226
227     foreach($turma as $key => $value){
228         foreach($professor as $keyI => $valueI){
229             if($value['Turma']['masp'] == $valueI['Funcionario']['masp']
230                 )){
231                 $turma[$key]['Turma']['profnome'] = $valueI['
232                     Funcionario']['nome'];
233             }
234         }
235     }
236
237     $this->set(compact('turma'));
238 }
239
240 public function NovaTurma(){
241     $this->VerificaLogin();
242
243     $professor = $this->Funcionario->find('all');
244     $this->set(compact('professor'));
245
246     $disciplina = $this->Disciplina->find('all');
247     $this->set(compact('disciplina'));
248
249     if ( empty($professor) || empty($disciplina) ){
250         echo $this->Session->setFlash('N o h professor/disciplina
251             cadastrado(a) no sistema');
252     }
253
254     if ($this->request->is('post')) {
255
256         if ($this->Turma->save($this->request->data)) {
257             $this->Session->setFlash('Turma cadastrada com sucesso!', '
258                 default', array('class' => 'success'));
259             $this->redirect(array('controller' => 'Coordenacaos', '
260                 action' => 'Turma'));

```

```
253     }
254     else {
255         $this->Session->setFlash('N o foi poss vel cadastrar a
                turma!');
256     }
257 }
258 }
259 public function ExcluirTurma($id){
260     $this->VerificaLogin();
261
262     if($this->Turma->delete($id)) {
263         $this->Session->setFlash('Turma deletada com sucesso!', '
                default', array('class' => 'success'));
264         $this->redirect(array('controller' => 'Coordenacaos', 'action'
                => 'Turma'));
265     }
266     else{
267         $this->Session->setFlash('N o foi poss vel deletar a turma!')
                ;
268     }
269 }
270
271 public function Matricula($idTURMA){
272     $this->VerificaLogin();
273
274     $nomealuno = $this->Aluno->find('all', array('order' => array('
                Aluno.nome' => 'asc')));
275
276     $this->set(compact('nomealuno'));
277
278     if ($this->request->is('post')) {
279
280
281         $id = $this->request->data['Matricula']['idaluno'];
282         $turma = $idTURMA;
283
284         $this->request->data['Matricula']['idavalia'] = $this->
                CriaAvaliacao($id, $turma);
285
286         $NomeAlunoToSave = $this->Aluno->findById($id);
287         $this->request->data['Matricula']['nomealuno'] =
                $NomeAlunoToSave['Aluno']['nome'];
288         $this->request->data['Matricula']['idturma'] = $turma;
289
290         $verifica = $this->Matricula->find ('all', array (
291             'conditions' => array (
292                 'Matricula.idaluno' => $id ,
```

```

293         'Matricula.idturma' => $turma
294     )
295 ) );
296
297     if ($verifica){
298         $this->Session->setFlash('Aluno(a) j matriculado!');
299     }
300     else {
301
302         if ($this->Matricula->save($this->request->data)) {
303             $this->Session->setFlash('Matricula realizada com
304                 sucesso!', 'default', array('class' => 'success'));
305             $this->redirect(array('controller' => 'Coordenacaos', '
306                 action' => 'Matricula', $idTURMA));
307         }
308         else {
309             $this->Session->setFlash('N o foi poss vel realizar a
310                 matr cula!');
311         }
312     }
313 }
314
315 public function VisualizarMatricula($id){ // RECEBE O
316     $ID DA TURMA SELECIONADA
317     $this->VerificaLogin();
318 // PROCURA
319     O $ID NA
320     TAB
321     MATRICULA
322
323     $this->set(compact('id'));
324     $tab_matricula = $this->Matricula->find ('all', array (
325         'order' => array (
326             'Matricula.nomealuno' => 'asc'
327         ),
328         'conditions' => array (
329             'Matricula.idturma' => $id
330         )
331     ) );
332     $this->set(compact('tab_matricula')); // COMPARA
333     COM O $ID DA TAB ALUNO
334 }
335
336 public function Desmatricular($id, $idturma ){
337     $this->VerificaLogin();
338
339     $id_delete = $this->Matricula->findByidaluno ($id);
340     $id_delete_avalicao = $this->Avalicao->findByidaluno($id);

```

```

331
332     if($this->Matricula->delete($id_delete['Matricula'][$id])) {
333         $this->Avaliacao->delete($id_delete_avalicao['Avaliacao'][$id]
334             );
335         $this->Session->setFlash('Aluno desmatriculado com sucesso!', '
336             default', array('class' => 'success'));
337         $this->redirect(array('controller' => 'Coordenacoes', 'action'
338             => 'Turma', $idturma));
339     }
340     else{
341         $this->Session->setFlash('N o foi poss vel desmatricular o
342             aluno!');
343         $this->redirect(array('controller' => 'Coordenacoes', 'action'
344             => 'Turma', $idturma));
345     }
346 }
347
348 public function ListaPresenca($id){
349     $this->VerificaLogin();
350     $this->set(compact('id'));
351     $tab_matricula = $this->Matricula->find ('all', array (
352         'order' => array (
353             'Matricula.nomealuno' => 'asc'
354         ),
355         'conditions' => array (
356             'Matricula.idturm' => $id
357         )
358     ) );
359     $this->set(compact('tab_matricula'));
360 }
361
362 public function ComprovanteMatricula($id){
363     $this->VerificaLogin();
364     $this->set('aluno', $this->Aluno->findById($id));
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

## C.5 Controller Paginação

```

1 <?php
2 /**
3  * Static content controller.
4  *
5  * This file will render views from views/pages/

```

```

6  *
7  * CakePHP(tm) : Rapid Development Framework (http://cakephp.org)
8  * Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
9  *
10 * Licensed under The MIT License
11 * For full copyright and license information, please see the LICENSE.txt
12 * Redistributions of files must retain the above copyright notice.
13 *
14 * @copyright      Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
15 * @link           http://cakephp.org CakePHP(tm) Project
16 * @package        app.Controller
17 * @since          CakePHP(tm) v 0.2.9
18 * @license        http://www.opensource.org/licenses/mit-license.php MIT
19 *               License
20 */
21 App::uses('AppController', 'Controller');
22
23 /**
24  * Static content controller
25  *
26  * Override this controller by placing a copy in controllers directory of
27  * an application
28  *
29  * @package        app.Controller
30  * @link http://book.cakephp.org/2.0/en/controllers/pages-controller.html
31  */
32 class PagesController extends AppController {
33     /**
34      * This controller does not use a model
35      *
36      * @var array
37      */
38     public $uses = array();
39
40     /**
41      * Displays a view
42      *
43      * @param mixed What page to display
44      * @return void
45      * @throws NotFoundException When the view file could not be found
46      * or MissingViewException in debug mode.
47      */
48     public function display() {
49         $path = func_get_args();

```

```

50
51     $count = count($path);
52     if (!$count) {
53         return $this->redirect('/');
54     }
55     $page = $subpage = $title_for_layout = null;
56
57     if (!empty($path[0])) {
58         $page = $path[0];
59     }
60     if (!empty($path[1])) {
61         $subpage = $path[1];
62     }
63     if (!empty($path[$count - 1])) {
64         $title_for_layout = Inflector::humanize($path[$count - 1]);
65     }
66     $this->set(compact('page', 'subpage', 'title_for_layout'));
67
68     try {
69         $this->render(implode('/', $path));
70     } catch (MissingViewException $e) {
71         if (Configure::read('debug')) {
72             throw $e;
73         }
74         throw new NotFoundException();
75     }
76 }
77 }

```

---

## C.6 Controller Professor

---

```

1 <?php
2 class ProfessorsController extends AppController {
3     public $name = 'Professors';
4     public $uses = array('Turma', 'Avaliacao', 'Aluno');
5     public function index() {
6         $this->VerificaLogin();
7     }
8
9     public function horario() {
10        $this->VerificaLogin();
11
12        $sessao = $this->Session->read();
13        $turma = $this->Turma->find('all', array('conditions' => array('
14            Turma.masp' => $sessao['session_usuario'])));
15        $this->set(compact('turma'));

```

```

15     }
16
17     public function NotaFrequencia($id) {
18         $this->VerificaLogin();
19
20         $sessao = $this->Session->read();
21
22         $avaliacao = $this->Avaliacao->findAllByIdturma($id);
23         $this->set(compact('avaliacao'));
24
25         $idaluno = $this->RetornaIndice($avaliacao, 'Avaliacao', 'idaluno')
26             ;
27         $aluno = $this->Aluno->find('all');
28
29         $this->set(compact('avaliacao'));
30         $this->set(compact('aluno'));
31         $this->set(compact('idaluno'));
32
33         if ($this->request->is('post')) {
34             $this->request->data['Avaliacao']['idturma'] = $id;
35
36             if ($this->Avaliacao->save($this->request->data)) {
37                 $this->Session->setFlash('Lanamento bem sucedido!', '
38                     default', array('class' => 'success'));
39                 $this->redirect ( array ('controller' => 'Professors', '
40                     action' => 'NotaFrequencia', $id));
41             }
42             else {
43                 $this->Session->setFlash('N o foi possivel realizar o
44                     lanamento!');
45             }
46         }
47     }
48
49     public function SelecionarTurma() {
50         $this->VerificaLogin();
51
52         $sessao = $this->Session->read();
53         $turma = $this->Turma->findAllByMasp($sessao['session_usuario']);
54         $this->set(compact('turma'));
55
56         if ($this->request->is('post')) {
57             $id = $this->request->data['Turma']['id'];
58             $this->redirect ( array (
59                 'controller' => 'Professors',
60                 'action' => 'NotaFrequencia',

```

---

```
58         $id
59     ) );
60     }
61
62     }
63
64     }
65 ?>
```

---



# ANEXO D – Model

---

```
1 <?php
2 /**
3  * Application model for CakePHP.
4  *
5  * This file is application-wide model file. You can put all
6  * application-wide model-related methods here.
7  *
8  * CakePHP(tm) : Rapid Development Framework (http://cakephp.org)
9  * Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
10 *
11 * Licensed under The MIT License
12 * For full copyright and license information, please see the LICENSE.txt
13 * Redistributions of files must retain the above copyright notice.
14 *
15 * @copyright      Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
16 * @link           http://cakephp.org CakePHP(tm) Project
17 * @package        app.Model
18 * @since          CakePHP(tm) v 0.2.9
19 * @license        http://www.opensource.org/licenses/mit-license.php MIT
20 *               License
21 */
22 App::uses('Model', 'Model');
23
24 /**
25  * Application model for Cake.
26  *
27  * Add your application-wide methods in the class below, your models
28  * will inherit them.
29  *
30  * @package        app.Model
31  */
32 class AppModel extends Model {
33 }
```

---