

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
FACULDADE DE CIÊNCIAS EXATAS
CURSO DE SISTEMAS DE INFORMAÇÃO

ROCHELLY FERNANDES ANDRADE

SISTEMA DE GERENCIAMENTO DE CONSERVATÓRIOS DE MÚSICA

Diamantina - MG

2014

ROCHELLY FERNANDES ANDRADE

**SISTEMA DE GERENCIAMENTO DE GERENCIAMENTO DE CONSERVATÓRIOS
DE MÚSICA**

**Trabalho de conclusão de curso de
graduação apresentado à Universidade
Federal dos vales do Jequitinhonha e Mucuri,
como requisito parcial para obtenção do grau
de Bacharel em Sistemas de Informação.**

Orientador: Alessandro Vivas Andrade

Diamantina - MG

2014

LISTA DE SIGLAS

| | |
|------------|---|
| JEE ----- | <i>Java Enterprise Edition</i> |
| POO ----- | <i>Programação Orientada a Objetos</i> |
| JDBC ----- | <i>Java Database Connectivity</i> |
| JSF ----- | <i>Java Server Faces,</i> |
| JPA ----- | <i>Java Persistence API</i> |
| ORM ----- | <i>Object Relational Mapper</i> |
| MVC ----- | <i>Model View Controller</i> |
| IDE ----- | <i>Integrated Development Environment</i> |
| WWW ----- | <i>World Wide Web</i> |

SUMÁRIO

| | |
|--|-----------|
| 1 - CAPÍTULO 1 - INTRODUÇÃO ----- | 5 |
| 2 - CAPÍTULO 2 ----- | 6 |
| 2.1 REFERENCIAL TEÓRICO----- | 6 |
| 2.2 DESCRIÇÃO DO PROBLEMA ----- | 8 |
| 2.3 PROPOSTA DE SOLUÇÃO ----- | 10 |
| 3 - CAPÍTULO 3 - FERRAMENTAS UTILIZADAS ----- | 11 |
| 3.1 PROGRAMAÇÃO ORIENTADA A OBJETOS----- | 12 |
| 3.1.1 OBJETOS ----- | 12 |
| 3.1.2 CLASSES----- | 12 |
| 3.1.3 ENCAPSULAMENTO ----- | 13 |
| 3.1.4 HERANÇA E POLIMORFISMO ----- | 13 |
| 3.2 PLATAFORMA JAVA ----- | 15 |
| 3.2.1 JAVA EE ----- | 16 |
| 3.3 BANCO DE DADOS ----- | 17 |
| 3.3.1 MySQL ----- | 18 |
| 3.3.2 JDBC ----- | 18 |
| 3.3.3 JPA2 ----- | 19 |
| 3.3.3.1 FERRAMENTAS ORMs ----- | 20 |
| 3.3.3.2 HIBERNATE ----- | 21 |

| | |
|---|-----------|
| 3.4 MVC ----- | 22 |
| 4 - CAPÍTULO 04 - SISTEMA DESENVOLVIDO ----- | 23 |
| 4.1 MENU PRINCIPAL | 28 |
| 4.2 CADASTRO DE ALUNO | 29 |
| 4.3 CONSULTA DE ALUNO | 32 |
| 4.4 TELA DE EDIÇÃO DE DADOS DE ALUNO | 34 |
| 4.5 TELA DE LANÇAMENTO DE NOTAS | 36 |
| 4.6 TELA DE MATRICULA DE ALUNO | 37 |
| 4.7 CADASTRO DE PROFESSOR | 41 |
| 4.8 CONSULTA PROFESSOR | 42 |
| 4.9 TELA DE EDIÇÃO DE DADOS DE PROFESSOR | 42 |
| 4.10 CADASTRO DE DISCIPLINAS | 44 |
| 4.11 CONSULTA DE DISCIPLINAS | 46 |
| 4.12 ALTERAÇÃO DE DISCIPLINA | 48 |
| 4.13 CADASTRO DE TURMAS | 49 |
| 4.14 CONSULTA DE TURMAS | 51 |
| 4.16 CADASTRO DE AULAS | 53 |
| 5 - CAPÍTULO 5 - TESTES REALIZADOS ----- | |
| 6- CAPÍTULO 6 - CONCLUSÃO ----- | 24 |

RESUMO

A TI (Tecnologia da Informação) é cada vez mais presente nas organizações das mais diferentes naturezas, para a melhoria de seus processos internos e vem se intensificando ao longo dos anos, por vários motivos, dentre eles a diminuição de custos de hardware e aumento de técnicas e ferramentas eficientes de desenvolvimento de software. Esta monografia tem como objetivo de desenvolver a primeira iteração de um sistema baseado na Web para o gerenciamento de conservatórios de música (escolas voltadas ao ensino de música e matérias a fim). Para desenvolvimento deste sistema, foi utilizado ferramentas da plataforma de desenvolvimento Java sob as especificações contidas na JavaEE (Java Enterprise Edition), uma plataforma para desenvolvimento de sistemas Java que oferece funcionalidades e padronizações para o desenvolvimento de aplicações de multicamadas baseadas na Web. O Sistema desenvolvido, foi nomeado de SisGTC - Sistema de Gerenciamento de Turmas de Conservatórios, apresenta funcionalidades de cadastros, consulta e alteração para todos os atores(entidades reais, modeladas de forma computacional) identificados durante a etapa de levantamento de requisitos. Após a codificação do sistema, foram realizados testes “caixa preta” em todos os pontos de função identificados, observando o correto funcionamento de botões de ações, campos de entrada e relações de persistência de dados em todo o sistema.

CAPITULO 1- INTRODUÇÃO

Diante do cenário atual, a tecnologia da informação é cada vez mais utilizada em diversos tipos de organizações com as mais diversas finalidades, para melhoria nas realizações de seus processos. A automatização de processos através de softwares se intensificou ao longo dos anos pela diminuição dos custos de hardware e aumento das técnicas de desenvolvimento ágil de sistemas, juntamente com a difusão dos sistemas baseados na Web. Com esse cenário, surge a necessidade nas organizações de se atualizar mediante ao estado tecnológico atual do mundo, para se manter competitiva e eficiente quando comparada as outras organizações.

Os conservatórios estaduais de musica, que são escolas direcionadas ao ensino musical e disciplinas relacionadas a musica se enquadram no perfil das organizações de caráter educacional, e cada vez mais, necessitam de transformar processos manuais em processos automatizados para redução de tempo e retrabalho em suas atividades administrativas rotineiras.

Este trabalho propõe e realizar a primeira iteração de um processo de desenvolvimento de software utilizando a plataforma Java de desenvolvimento para o gerenciamento do Conservatório Estadual de Musica Lobo de Mesquita situado na cidade de Diamantina, Minas Gerais. Sendo a proposta deste trabalho apenas a primeira iteração de desenvolvimento o mesmo deverá ser continuado por acadêmicos do curso de Sistemas de Informação da UFVJM.

CAPITULO 2

2.1 REFERENCIAL TEÓRICO

A tecnologia da informação tem se tornado crucial em todas organizações do mundo contemporâneo, seja na área comercial, seja na área educacional. Os sistemas de informação que são “conjunto de partes, componentes, que interagem entre si, de forma ordenada, a fim de atingir um objetivo comum” (STAIR, 1998; LAUDON & LAUDON, 2004) são aplicados na automação de processos e consistem na administração dos fluxos de informações geradas em um determinado ambiente sobre um determinado contexto provendo a organização das informações. Entende-se como informação “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.” (McGarry, 1999).

Um sistema de informação também é caracterizado pelo o processamento, armazenamento, recuperação da informação segundo os critérios e fluxos estabelecidos pelos processos existentes na organização, acarretando em melhoria de desempenho, eficiência e racionalização de processos na qual entende-se como processos “Um grupo de atividades realizadas numa sequencia lógica com objetivo de produzir um bem ou serviço que tem valor para um grupo específico de clientes” (Hammer e Champy, 1994) .

A difusão da Word Wide Web(WWW ou Web) acarretou em um grande crescimento em numero e complexidade dos sistemas de informação baseados na Web, na qual Press(1999, p.13) ainda classificou 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 .

Uma ressalta deve ser feita na definição de Press(1999, p.13) onde deve-se se considerar que muitos sistemas de informação baseados na Web são modelos cliente-servidor.

Berners-Lee (1996) 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 [...].

Outra vantagem dos Sistemas Baseados na Web é a busca constante de características positivas de sistemas tradicionais desktop como o uso de hipermídia onde Bieber(1997, p.35) diz:

Os benefícios de adicionar funcionalidade hipermídia às aplicações de sistemas de informação são que a hipermídia proporciona acesso navegacional, contextual para ver informação e que representa conhecimento em uma forma relativamente próxima das estruturas cognitivas organizacionais que as pessoas usam. Assim, a hipermídia apóia entendimento.

Sobre essa perspectiva, segundo Gosciola (2003, p.34) podemos Definir Hipermídia como :

Hipermídia é o conjunto de meios que permite acesso simultâneo a textos, imagens e sons de modo interativo e não-linear, possibilitando fazer links entre elementos de mídia, controlar a própria navegação e, até, extrair textos, imagens e sons cuja sequencia constituirá uma versão pessoal desenvolvida pelo usuário.

E Machado (1997, p.252):

Com base na arquitetura não linear das memórias de computador, pode-se hoje conceber obras em que textos, sons e imagens estariam ligados entre si por elos probabilísticos e móveis, podendo ser configurados pelos receptores de diferentes maneiras, de modo a compor possibilidades instáveis em quantidades infinitas. Isso é justamente o que chamamos de hipermídia.

Em um contexto educacional a aplicação da tecnologia da informação cada vez mais se destaca com como um quesito de muita importância, segundo 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.

2.2 DESCRIÇÃO DO PROBLEMA

Conservatórios de Musicas são escolas voltadas para o ensino de música e canto e matérias relacionadas (História da Arte, História da Música e Apreciação Musical, Percepção, Estruturação, Folclore Regional e Música Popular, Música de Câmara e Prática de ensino). O Conservatório Estadual de Musica 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 musica.

O Conservatório Estadual de Musica Lobo de Mesquita possui cerca de 1525 alunos como idades diversas, que variam desde crianças (a partir de 6 anos) a adultos, possui em seu grupo docente 48 professores que são auxiliados por 2 supervisores. 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. Trompam,
- J. Trombone,
- K. Trompete,
- L. Bateria,
- M. Cavaquinho,
- N. Contrabaixo Elétrico.

O Conservatório oferece duas modalidades de curso, o curso Educação Musical (conforme as Tabelas 2.1, 2.2 e 2.3 o plano curricular do curso de Educação Musical) e o curso de Formação Profissional (conforme a Tabela 2.4 o plano curricular do curso de Formação Profissional).

Tabela 2.1: Plano curricular do curso de Educação Musical - Ciclo Inicial

| I | II | III |
|---------------|-------------------------|----------------------------------|
| Musicalização | Musicalização Percepção | Musicalização Percepção |
| Canto Coral | Canto Coral | Canto Coral |
| Instrumento | Instrumento | Instrumento |
| | | Atividade Artística Complementar |

Tabela 2.2: Plano curricular do curso de Educação Musical - Ciclo Intermediário

| I | II | III |
|----------------------------------|-------------------------|---------------------|
| Musicalização Percepção | Musicalização Percepção | Percepção |
| Canto Coral | Canto Coral | Canto Coral |
| Instrumento | Instrumento | Instrumento |
| Atividade Artística Complementar | | Prática de Conjunto |

Tabela 4.3: Plano curricular do curso de Educação Musical- Ciclo Complementar

| I | II | III |
|---------------------|---------------------|----------------------------------|
| Percepção | Percepção | Percepção |
| Canto Coral | Oficina Multimeio | Oficina Multimeio |
| Instrumento | Instrumento | Instrumento |
| Prática de Conjunto | Prática de Conjunto | Prática de Conjunto |
| | | Atividade Artística Complementar |

Tabela 4: Plano curricular do curso de Formação Profissional

| 1º Ano | 2º Ano | 3º Ano |
|----------------------------------|---|---|
| Atividade Artística Complementar | Estrutura Musical | Estrutura Musical |
| Folclore | Ética e Normas Técnicas | História da Música e Apreciação Musical |
| História da Arte | História da Música e Apreciação Musical | Instrumento |
| Instrumento | Instrumento | Oficina Multimeio |
| Noções de Educação Musical | Oficina Multimeio | Prática de Conjunto |
| Oficina Multimeio | Percepção Musical | Prática de Ensino |
| Percepção Musical | Prática de Conjunto | Produção Cultural e Empreendedorismo |
| Prática de Conjunto | Prática de Ensino | |

Visto que o Conservatório Estadual de Música Lobo de Mesquita é uma escola que se difere das escolas tradicionais de ensino não somente pelas modalidades de ensino e cursos ofertados, mas principalmente pela sua estrutura interna de graduação de alunos, fazendo assim surgir a necessidade de um tratamento diferente para esse contexto ainda mais com a diminuição de custos computacionais devido aos grandes avanços de hardware e software e o crescimento da necessidade de automatização e modernização de processos, a proposta deste trabalho é desenvolver um sistema para o gerenciamento de alunos e turmas do Conservatório Estadual de Música Lobo de Mesquita.

CAPÍTULO 3 - FERRAMENTAS UTILIZADAS

Para a implementação do sistema proposto, para gerenciamento de alunos e turmas, utilizou-se a linguagem de programação Java¹ que é obrigatoriamente acompanhada com seu paradigma de orientação a objetos. Para esse trabalho optou-se por utilizar os as padronizações JEE - *Java Enterprise Edition*², que envolve a especificação JPA2 - *Java Persistence API*³ para a persistência de dados e como implementação dessa especificação utilizou-se o Hibernate⁴ como framework para a implementação e o Mysql⁵ como sistema gerenciador de banco de dados. Como padrão de arquitetura utilizou-se MVC - *Model View Controller*⁶ e sua implementação o JSF- *Java Server Faces*⁷, com o framework Primefaces⁸. Como IDE- ambiente integrado para desenvolvimento de software, utilizou-se o Eclipse⁹, instalado em uma distribuição Linux, o Ubuntu 14.04. Por se tratar de uma aplicação desenvolvida para a Web, necessitou de com servidor de aplicação embutido, foi escolhido o Jboss¹⁰.

1RFC 2713, 1999

2JSR 316

3JSR 317

4HIBERNATE, Disponível em <<http://hibernate.org>>. Acesso em 20de mar. 2014

5MYSQL, Disponível em <<http://www.mysql.com>>. Acesso em 22 de mar. 2014.

6JSR 371

7JSR 372

8PRIMEFACES, Disponível em <<http://www.primefaces.org/documentation>>. Acesso em 15 de mar. 2014.

9ECLIPSE, Disponível em <<https://www.eclipse.org/>>. Acesso em 15 de mar. 2014.

10JBASS DEVELOPER, Disponível em <<http://www.jboss.org/>>. Acesso em 18 de mar. 2014.

3.1 PROGRAMAÇÃO ORIENTADA A OBJETOS

O paradigma de orientação a objetos enxerga o software como uma projeção do mundo real, na qual os componentes do mundo são representados por objetos que interagem entre si. Os sistemas que são construídos sobre esse paradigma são estruturados em módulos, cada módulo apresenta atributos e operações que interagem sobre esses módulos e atributos. Os módulos são chamados de classes e as operações presentes nas classes são chamados de métodos, o paradigma de programação orientada a objetos se diferencia dos demais pois o conceito de reaproveitamento de código também está muito presente através dos conceitos de herança e polimorfismo .

3.1.1 OBJETOS

Em POO (Programação Orientada a Objetos) entidades do mundo real que têm relevância para solucionar um problema são modeladas em objetos, os objetos devem promover o entendimento para a implementação computacional, sendo instâncias de classes e podendo conter informações e operações. As informações são armazenadas nos atributos de um objeto, ou seja, os atributos armazenam valores em si, que são manipulados por operações que são chamados de métodos. Os métodos de um objetos são funções ou procedimentos que realizam operações que espelham o comportamento dos objetos reais.

3.1.2 CLASSES

A representação de um conjunto de objetos que possuem mesmo comportamento e características é dado por meio das classes. Os objetos são definidos através das propriedades definidas nas classes a qual o objeto pertence, ou seja, as classes definem um conjunto de objetos com atributos e métodos similares. Pode se afirmar que classe é a implementação de um tipo de objeto e objeto é uma instância de um determinada classe. Segundo Correia (2006, p. 17), “classe é um modelo e todos os seus objetos têm os mesmos atributos (embora esses atributos possam ter valores diferentes) e os mesmos métodos”.

3.1.3 ENCAPSULAMENTO

O Encapsulamento é uma técnica que consiste em deixar visível/acessível apenas algumas partes dos dados do objeto, fazendo “esconder” os detalhes de implementação aos usuários da classe, 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 própria estrutura indicando quais e de qual maneira seus métodos e dados devem ser encapsulados. Abaixo um exemplo para maior compreensão:

Imagine uma pessoa (usuário) que vai utilizar um aparelho de som (sistema) para ouvir suas músicas, essa pessoa não precisa conhecer de toda a complexidade interna da construção do aparelho (circuitos elétricos, leitores ópticos etc.), ela apenas necessita conhecer as funcionalidades da interface do aparelho que são os botões de controle do aparelho de som.

3.1.4 HERANÇA E POLIMORFISMO

Em Orientação a Objetos existem dois grandes conceitos para se implementar o reaproveitamento de código, são eles herança e polimorfismo. Herança é meio que a Orientação a Objetos utiliza para uma classe obter características de outra classe a fim de expandir ou especializar de alguma forma. Pode se dizer que uma classe “Herda” de outra classe quando os métodos e atributos de uma classe “Pai” (Super Classe) estão presentes na classe “Filho” (Sub Classe) através de artifícios de Herança. Abaixo um exemplo:

Suponha que uma classe chamada animal que possui dois atributos, o atributo Nome e o atributo Idade e ainda os métodos Comer e Respirar e existe uma classe peixe que herda da classe animal, a classe peixe possui os atributos Nadadeiras e Escamas, pelo fato de que a classe peixe herdar da classe animal, ela também deverá conter os atributos Nome e Idade bem como os métodos comer e respirar que estão presentes na superclasse.

O conceito de Polimorfismo está fortemente relacionado com o conceito de herança e conseqüentemente de hierarquia de classes, o termo que é de origem grega e significa “Muitas Formas”, em Orientação a Objetos significa que um método definido em uma classe pai pode apresentar comportamento diferente nas classes filhas. Para melhor entendimento segue um exemplo:

Suponha novamente uma classe chamada animal (superclasse) e duas classes filhas, a classe anfíbia e a classe mamífero, na classe animal está presente o método respirar, o método respirar estará presente nas classes anfíbio e mamífero, porém cada classe apresentará uma implementação diferente desse método, a classe anfíbio pode realizar na sua implementação do método respirar a uma chamada a um método respiração_cutânea e a classe mamífero uma chamada a um método respiração_pulmonar, por exemplo.

3.2 PLATAFORMA JAVA

O Java é uma poderosa linguagem de programação que nos últimos anos está sendo amplamente utilizada para implementar aplicativos baseados na internet e sistemas que se comunicam em rede. O Java é uma linguagem de programação que implementa obrigatoriamente o paradigma de orientação a objetos, ou seja, implementa o conceito de criação de objetos a partir das estruturação e implementação das classes, bem como os demais conceitos descritos pelo paradigma de orientação a objetos.

Uma das grandes vantagens do uso da linguagem Java é a existência de uma gama de coleções de classes presentes nas bibliotecas Java que são chamadas de Java APIs (*Application Programming Interfaces*), na qual podem ser usadas em conjuntos com as classes criadas pelo programador Java. O uso das classes presentes nas APIs do Java, juntamente com os conceitos de reaproveitamento de código fazem que o processo de desenvolvimento de softwares, baseados na internet ou não, se tornem mais ágeis.

A linguagem Java é dependente de um ambiente de execução que faz com que as suas aplicações sejam multiplataformas. No ambiente de execução Java se tem o conceito de Máquina Virtual¹¹ que faz com que computadores de as que utilizam arquiteturas de processadores e sistemas operacionais diferentes, possam responder as chamadas de instruções feitas pela linguagem de programação.

De modo geral e simplista a Máquina Virtual do Java desacopla a aplicação de uma plataforma específica (sistema operacional + arquitetura de processador), ou seja, a Máquina Virtual do Java cria uma grande conceito de versatilidade, fazendo com que as aplicações desenvolvidas em Java se torne multiplataforma, implicando grandes vantagens em que quesito de versatilidade, mas aumentando o custo computacional de processamento, devido ao processamento gasto da execução da Máquina Virtual somado ao da aplicação, para compensar o custo adicional de processamento, as Máquinas Virtuais atuais implementam otimizações através de informações geradas

11 JSR 924

durante a execução de programas que podem inclusive fazer com que um programa que executa sobre uma Máquina Virtual, em um determinado momento durante a execução, ser mais eficiente em alguns casos do que um programa executado diretamente no sistema operacional. Atualmente existem dois exemplos de Maquinas Virtuais Java muito conceituados, a Oracle JVM (*Java Virtual Machine*)¹² e OpenJDK JVM¹³ que implementam tais otimizações.

3.2.1 JAVA EE (*JAVA ENTERPRISE EDITION*)

Java EE (*Java Enterprise Edition*)¹⁴ é uma plataforma para desenvolvimento de aplicações Java que oferece a funcionalidade para o desenvolvimento de aplicações de multicamadas, baseadas na Web. O Java EE que apresenta de um conjunto de especificações com alto nível de detalhamento, que descreve convenções e padronizações de implementações de softwares bem como usos dos recursos computacionais tais como conexão Banco de Dados, utilização de componentes de Web, acesso remoto, gerenciamento de threads, gerenciamento de conexões HTTP (*Hypertext Transfer Protocol*)¹⁵, gerenciamento da sessão Web dentro outros que são utilizadas por desenvolvedores de sistemas Web.

12 ORACLE, Disponível em <<http://www.oracle.com/br>>. Acesso em 20 de abr. 2014

13 OPENJDK, Disponível em <<http://openjdk.java.net/>>. Acesso em 25 de abr. 2014

14 JSR 316

15 RFC 2616

3.3 BANCO DE DADOS

As grandes maiorias dos softwares necessitam de armazenar dados aos quais trabalham, ou seja, necessitam de persistir seus dados para futuras consultas, inserções, exclusões e alterações dos mesmos, além de dessas necessidades, que são chamadas de CRUD (*Create, Read, Update e Delete*) esses dados devem se apresentar de maneira organizada para facilitar essas operações, para isso, em sua grande maioria, os softwares utilizam os chamados banco de dados. Os bancos de dados, por sua vez, são gerenciados por softwares chamados de SGBDs - Sistema Gerenciador de Banco de Dados, 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.

A maioria dos SGBD existentes no mercado utiliza como modelo de estruturação e organização de seus dados o modelo relacional. O Modelo Relacional de bando de dados surgiu em 1970 como uma revolução dos bancos de dados e é largamente utilizado até os tempos de hoje pelos SGBDs. No Modelo Relacional os dados são separados e organizados de acordo com suas correlações em tabelas que permitem relacionamento de registros (linhas de uma tabela) para que se possa realizar as operações de consultas, tais consultas são feitas através de atributos (colunas de uma tabela) chamados de chaves.

Uma das grandes vantagens do uso dos bancos de dados é a sua independência em relação as aplicações, ou seja, a existência banco de dados não é necessariamente atrelado a existência de uma aplicação.

No contexto atual dos SGBDs existem tanto pagos, quanto gratuitos, abaixo alguns do mais utilizados atualmente:

- A. SQLServer¹⁶
- B. Mysql¹⁷
- C. FirebirdSQL¹⁸

¹⁶MICROSOFT, Disponível em <<http://www.microsoft.com/pt-br/server-cloud/products/sql-server/>>, Acesso em 20 de mar. 2014.

¹⁷MYSQL, Disponível em <<http://www.mysql.com/>>. Acesso em 22 de mar. 2014.

¹⁸FIREBIRDSQL, Disponível em <<http://www.firebirdsql.org/>>. Acesso em 22 de mar. 2014.

- D. mSQL¹⁹
- E. Microsoft Access²⁰
- F. Oracle Database 12c²¹

Para este trabalho será utilizado o SGBD MySql por ser um SGBD gratuito, sem limitação de armazenamento e bem conceituado entre os desenvolvedores de aplicações WEB.

3.3.1 MySQL

O MySQL é um SGBD - Sistema Gerenciador de Banco de Dados relacional de código aberto usado na maioria das aplicações gratuitas. O MySQL utiliza, assim como a maioria dos Sistemas Gerenciadores de bancos de dados a linguagem SQL (*Structure Query Language* – Linguagem de Consulta Estruturada), foi criado em 1996 pela a empresa TcX²², com o intuito de se ter um banco de dados que apresentasse segurança, agilidade e ainda pouco custoso e que necessitasse de requisitos de hardware com baixo custo, hoje é atualmente desenvolvido pela MySQL AB, na qual apresenta uma versão gratuita 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.

3.3.2 JDBC (JAVA DATABASE CONNECTIVITY)

19 XOOPS BRASILI, Disponível em <<http://xoops.net.br/docs/php/manual/ref.mssql.php>>. Acesso em 22 de mar. 2014.

20 MICROSOFT, Disponível em<<http://office.microsoft.com/pt-br/access/>>. Acesso em 23 de mar. 2014.

21 ORACLE, Disponível em<<http://office.microsoft.com/pt-br/access/>>. Acesso em 23 de mar. 2014.

22 OFICINA DA NET, Disponível em<<http://office.microsoft.com/pt-br/access/>>. Acesso em 23 de mar. 2014.

O JDBC- *Java Database Connectivity* é uma especificação para drives de conexão entre os SGBDs e sistemas programados em Java. Os drivers funcionam como uma ponte de tradução (pois em geral, por uma questão de desempenho, os dados são recuperados utilizando protocolos binários) e conexão dos dados obtidos do banco de dados. Os drivers são geralmente fornecidos pelas empresas produtoras de SGBDs.

3.3.3 JPA2 (JAVA PERSISTENCE API)

Para que um sistema possa utilizar as informações presentes em um banco de dados é necessário existir comunicação com uma determinada sintaxe de consulta entre o banco de dados e o sistema, essa comunicação feita mediante a uma conexão utilizando um driver de conexão (como visto no tópico anterior). Para evitar a diversidade de tipos de sintaxes entre a linguagem Java e os vários tipos de SGBDs utiliza-se a *Java Persistence API* - JPA que é uma especificação da linguagem Java para a persistência de dados. A especificação além de padronizar as diferentes tipos sintaxe ainda provê recursos para problemas relacionados a incompatibilidade de paradigmas visto que a linguagem Java é uma linguagem orientada a objetos e Sistemas Gerenciadores de Banco de Dados, em sua maioria, utilizam o Modelo Relacional como paradigma. Para a “conversão” “Objeto-Relacional” (basicamente transformar instâncias de tabelas de bancos de dados em objetos) a especificação JPA2 utiliza as chamadas ferramentas ORMs-(*Object Relational Mapper*).

3.3.3.1 FERRAMENTAS ORMs (*OBJECT RELATIONAL MAPPER*)

Uma ferramenta ORMs (*Object Relational Mapper*) consiste em um framework ou conjunto de classes que auxiliam desenvolvedores de linguagens orientadas a objetos a comunicar-se com SGBDs que utilizam o modelo relacional. As ferramentas ORMs realizam um mapeamento entre os conceitos de paradigmas de maneira automática e transparente com a grande vantagem de ser independente da sintaxe da linguagem Sql utilizada. Atualmente o Hibernate é a ferramenta mais utilizada no mercado.

3.3.3.2 HIBERNATE

O Hibernate é uma ferramenta ORM, ou seja, um framework que implementa o mapeamento objeto relacional (mapeamento de tabelas em classes e classes em tabelas) para a Linguagem Java, e também provê recursos para a consulta, gravação e alteração de dados, sendo um mediador entre a aplicação e o banco de dados. Um fato interessante é o Hibernate, após sua difusão, deu origem a especificação JPA já citada anteriormente, e obviamente o implementa as regras dessa especificação, na qual o mapeamento objeto relacional é feito mediante as chamadas anotações. Uma anotação ou (Annotation) é um metadado que é ignorado pelo compilador, em Java as anotações são marcadas com um @ em frente a anotação .

A especificação JPA especifica uma série de anotações, dentre elas podemos citar:

@Entity: Mapeia uma tabela em uma classe, as colunas de uma tabela são convertidas em atributos de uma classe.

@Id: Identifica o atributo que corresponde a chave primária de uma tabela

@GeneratedValue: Representa um atributo que é auto-incrementado pelo banco de dados.

@Table: Utilizado quando a classe não possui o mesmo nome da tabela, representa o nome da tabela que a classe está mapeando.

@Column: Utilizado quando um atributo não possui o mesmo nome de uma coluna da tabela.

@Transient: Indica que um atributo não deve ser persistido.

@Temporal: Mapeia colunas que suportam objetos binários grandes para um atributo da classe.

O Hibernate implementa todas essas Annotations e outras em para o mapeamento objeto relacional .

3.4 MVC (*MODEL-VIEW-CONTROLLER*)

O sucesso no desenvolvimento de sistema que utilizam o paradigma de orientação a objetos esta ligado com os padrões utilizados para a construção do mesmo, com o aumento do tamanho e complexidade de tais sistemas tornou-se fundamental a utilização de padrões que organizem e separam partes dos sistemas. O padrão de arquitetura 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. Em uma implementação desse padrão temos :

- 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.

O padrão MVC, em resumo, divide e desacopla a interface e aplicação, permitido que alterações feitas em algumas na View da aplicação não altere o Model e vice-versa, implicando em independência entre as camadas, que aumenta a facilidade de manutenção da aplicação.

3.4.1 JSF

JSF - Java Server Faces é um framework utilizado principalmente para se implementar recursos de **View**, em um sistema estruturado sobre uma arquitetura MVC. O JSF foi incorporado na especificação JEE e tem como objetivo promover a agilidade no desenvolvimento de aplicações web pois além de permiti a utilização componentes prontos ainda possibilita a criação de novos componentes.

As configurações de um projeto de sistema web baseado em JSF são feitas basicamente através de arquivos XML (*eXtensible Markup Language*) (principalmente nos arquivos faces-config.xml e web.xml utilizados nos padrões de desenvolvimento) e de anotações já mencionadas anteriormente.

Java Server Faces em sua versão JSF2 permite a utilização de escopos, configurados das seguintes anotações :

@RequestScoped: O tempo de vida do escopo corresponde ao tempo entre uma requisição e outra, ou seja, os objetos do escopo sobrevivem apenas até que uma nova requisição seja feita.

@SessionScoped: O tempo de vida dos objetos tem duração durante a seção de usuário, esse tipo de escopo é recomendado para armazenar dados sobre o usuário durante o período que o mesmo está com a seção ativa.

@ApplicationScoped: Os objetos sobrevivem durante todo o tempo em que a aplicação estiver sendo executada.

@ViewScoped: Matem os dados em de uma aplicação durante quantas requisições forem necessárias desde que estejam utilizando a mesma View para isso.

Os principais frameworks utilizados em implementação JSF são o PrimeFaces , Apache Tobago e RichFaces, para esse trabalho será utilizado o PrimeFaces.

3.4.1.1 PRIMEFACES

O PrimeFaces é um framework para a criação de interfaces amplamente utilizado no mundo pelos desenvolvedores Java. Segundo Venturini e Marchi (2011, citado por Carmisini e Vahldick1(2012)):

O PrimeFaces oferece um conjunto de componentes com versões estáveis e de código aberto para o JSF 2.0 e permite que sejam inseridos em seu conjunto, outros componentes através de especificações em JSF. Ele está organizado em três módulos:

- a) User Interface (UI) Components – compreende os componentes que contém as funcionalidades encapsuladas de AJAX, Javascript e gráficos animados;
- b) Optimus – módulo que oferece soluções para facilitar desenvolvimento com JSF. Também contém componentes de extensões de segurança;
- c) FacesTrace – módulo encarregado das funções relacionadas ao desempenho das aplicações baseadas em JSF.

As vantagens de se usar o Primefaces para a criação de interfaces são claras, segundo PrimeFaces (2012 ,citado por Carmisini e Vahldick1(2012)):

- a) Simplicidade e desempenho: PrimeFaces é leve, todas as decisões tomadas são baseadas em manter o PrimeFaces tão leves quanto possível.
- b) Facilidade de uso: os componentes do PrimeFaces são desenvolvidos com um princípio de design que afirma que “Uma boa UI deve ocultar a complexidade, mas manter a flexibilidade”;
- c) Possui o Mobile UI kit para criar aplicações web móveis para dispositivos portáteis baseados em navegadores webkit, por exemplo, iPhone, Palm, Android Phones, Nokia S60 entre outros;
- d) Possui um rico conjunto de componentes de interface: DataTable, AutoComplete, HTMLEditor, gráficos e etc.;
- e) Não é necessária nenhuma configuração extra de XML e não há dependências;
- f) Os componentes são construídos com AJAX no padrão JSF 2.0;
- g) Possui mais de 30 temas de templates;
- h) Possui boa documentação com exemplos práticos.

3.5 ECLIPSE IDE

Para Santos(2007) as IDEs *-(Integrated Development Environment) -* (Ambientes de Desenvolvimento Integrado) são ferramentas utilizadas para auxiliar os desenvolvedores de software, garantindo a diminuição de erros, aumento a produtividade e garantia de melhores produtos. Existem vários tipos de IDEs, tanto gratuitas quanto pagas, abaixo estão IDEs utilizadas no desenvolvimento de aplicações em Java :

- A. Eclipse
- B. NetBeans
- C. Gel
- D. JEdit
- E. Optimalj

Nesse trabalho será utilizada a Eclipse com seus plugs para o desenvolvimento Java Web. Para YAW(2012) a IDE Eclipse :

Não é apenas uma das principais IDEs Java, bem além disso, o projeto é uma referência e um caso de sucesso no desenvolvimento de software de um modo geral. Atualmente com pouco mais de 10 anos de vida, o Eclipse hoje é um ecossistema composto por vários projetos de diversas características. Um ponto extremamente importante relacionado ao Eclipse é o modelo open source. O Eclipse atingiu o patamar atual, principalmente pela força e influência da comunidade open source. Indo um pouco além, o Eclipse é a prova real de que desenvolvimento de software open source funciona, e muito bem.

Uma das grandes vantagens da utilização da IDEs Eclipse é a sua adaptação a diversas maneiras e metodologias de desenvolvimento através dos vários plug-ins que podem ser incorporados a essa IDE, veja a estrutura da arquitetura dessa IDE na Figura 3.1.

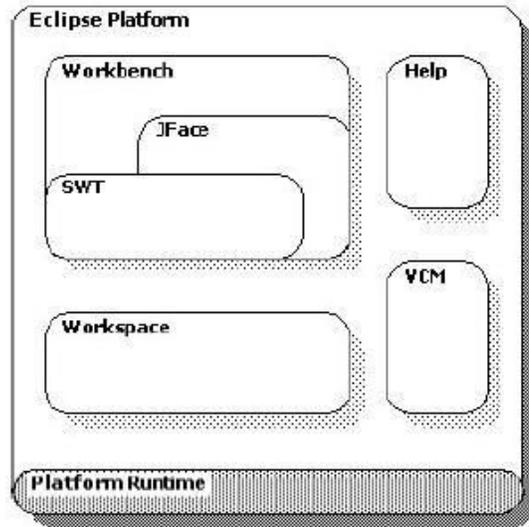


Figura 3.1 - Estrutura da Plataforma Eclipse

Tabela 3.1: Descrição dos subsistemas das estrutura da arquitetura do IDE Eclipse

| | |
|--|---|
| Plataforma Runtime | Define o ponto de extensão e modelo de plug-ins. Descobre Dinamicamente plug-ins e maténs informação em registro da plataforma. Plug-ins são carregados quando requeridos de acordo com a operação do usuário da plataforma. |
| Gerência de Recursos (workspace) | Define a API para criação e gerenciamento de recursos (projetos, arquivos e diretórios) que são produzidos por ferramentas e mantidos no sistema de arquivos. |
| Workbench UI | Implementa a interface do usuário para navegação entre os recursos e utilização dos plug-ins da ferramenta. Define pontos de extensão para adicionar componentes de UI tais como views ou ações de menu. Inclui conjunto de ferramentas adicionais (Jface e SWT) para o desenvolvimento de interfaces de usuário. |
| Sistema de Ajuda | Define pontos de extensão para que os plug-ins forneçam ajuda ou outra documentação como livros interativos. |
| Gerência de Versão e de Configuração(CVS) | Define um modelo de programação em equipe para o gerenciamento e controle versão dos recursos. |

CAPÍTULO 04 - SISTEMA DESENVOLVIDO

Como solução para o problema apresentado no Capítulo 1 desenvolveu-se um sistema WEB utilizando as ferramentas e metodologias descritas no capítulo anterior. O sistema foi denominado **SisGTC** - Sistema de Gerenciamento de Turmas de Conservatórios, feito com o propósito de auxiliar no gerenciamento de turmas e alunos do Conservatório Estadual de Música Lobo de Mesquita, o funcionamento do sistema segue a seguinte linha de raciocínio :

- A. Os alunos e professores são cadastrados no sistema fornecendo seus dados pessoais e dados relacionados com a instituição de ensino.
- B. Para alunos e professores cadastrados são fornecidas funcionalidades de consultas dos mesmos.
- C. Além da funcionalidade de consulta é fornecido as funcionalidades de alteração e exclusão de alunos e professores cadastrados.
- 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 de notas e frequência do aluno, de forma bimestral .
- I. Professores não podem ser excluídos se estiverem vinculados a uma turma.
- J. Disciplinas não podem ser excluídas se estiverem vinculadas a uma turma.

Nesse capítulo será apresentado de forma detalhada o funcionamento do sistema através de imagens do próprio sistema.

4.1 Menu principal

Veja na Figura 4.1 o que menu principal do sistema, que dispõe de links para todas as principais funcionalidades do sistema.

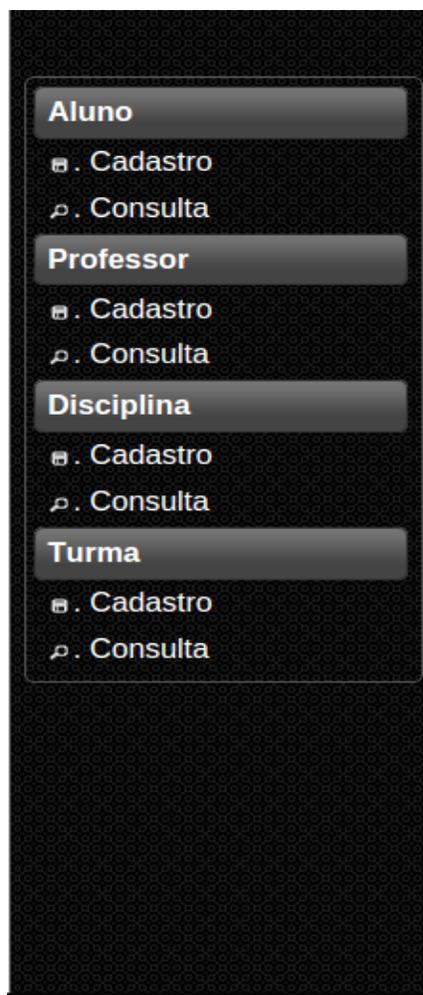


Figura 4.1: Menu Principal

4.2 - Cadastro de Aluno

] A tela de cadastro de aluno é representada na Figura 4.2, seus campos são descritos na Tabela 4.1 e as ações dos botões são explicados na Tabela 4.2.

| Cadastro Aluno | |
|--|----------------------|
| Nome: * | <input type="text"/> |
| Matricula: * | <input type="text"/> |
| Data de Nascimento * | <input type="text"/> |
| Nome do pai: | <input type="text"/> |
| Nome da mãe: | <input type="text"/> |
| CPF: | <input type="text"/> |
| Rua | <input type="text"/> |
| Número: | <input type="text"/> |
| Complemento: | <input type="text"/> |
| Bairro: | <input type="text"/> |
| Cidade: | <input type="text"/> |
| CEP: | <input type="text"/> |
| UF | <input type="text"/> |
| E-mail: * | <input type="text"/> |
| País: | <input type="text"/> |
| Telefone: | <input type="text"/> |
| <input type="button" value="Cadastrar"/> | |

Figura 4.2: Tela de cadastro de alunos

Tabela 4.1: Descrição de Campos - Cadastro de alunos

| Nome | Descrição | Tipo de Entrada |
|--------------------|---|------------------------|
| Nome | Nome do aluno a ser cadastrado | Alfanumérico |
| Matrícula | Matrícula do aluno a ser cadastrado do aluno | Alfanumérico |
| Data de Nascimento | Data de Nascimento do aluno a ser cadastrado | Data |
| Nome do Pai | Nome do Pai do aluno a ser cadastrado | Alfanumérico |
| Nome da Mãe | Nome da Mãe do aluno a ser cadastrado | Alfanumérico |
| CPF | CPF do aluno a ser cadastrado | número |
| Rua | Rua (endereço) do aluno a ser cadastrado | Alfanumérico |
| Número | Número(endereço) do aluno a ser cadastrado | Alfanumérico |
| Complemento | Complemento(endereço) do aluno a ser cadastrado | Alfanumérico |
| Bairro | Bairro(Endereço) do aluno a ser cadastrado | Alfanumérico |
| Cidade | Cidade (endereço) do aluno a ser cadastrado | Alfanumérico |
| CEP | CEP(endereço) do aluno a ser cadastrado | Alfanumérico |
| UF | UF(endereço) do do aluno a ser cadastrado | Alfanumérico |
| E-mail | E-mail do aluno a ser cadastrado | Alfanumérico |
| Pais | Pais do aluno a ser cadastrado | Alfanumérico |
| Telefone | Telefone do aluno a ser cadastrado | Número |

Tabela 4.2: Descrição de ações - Tela Cadastro Alunos

| Nome | Ação |
|-------------|--|
| Cadastrar | Realiza o cadastramento do aluno e redireciona para a página de consulta de aluno. |

4.3 - Consulta de aluno

A tela de consulta de aluno é representada na Figura 4.3., seus campos são descritos na Tabela 4.3 e as ações dos botões são explicados na Tabela 4.4.

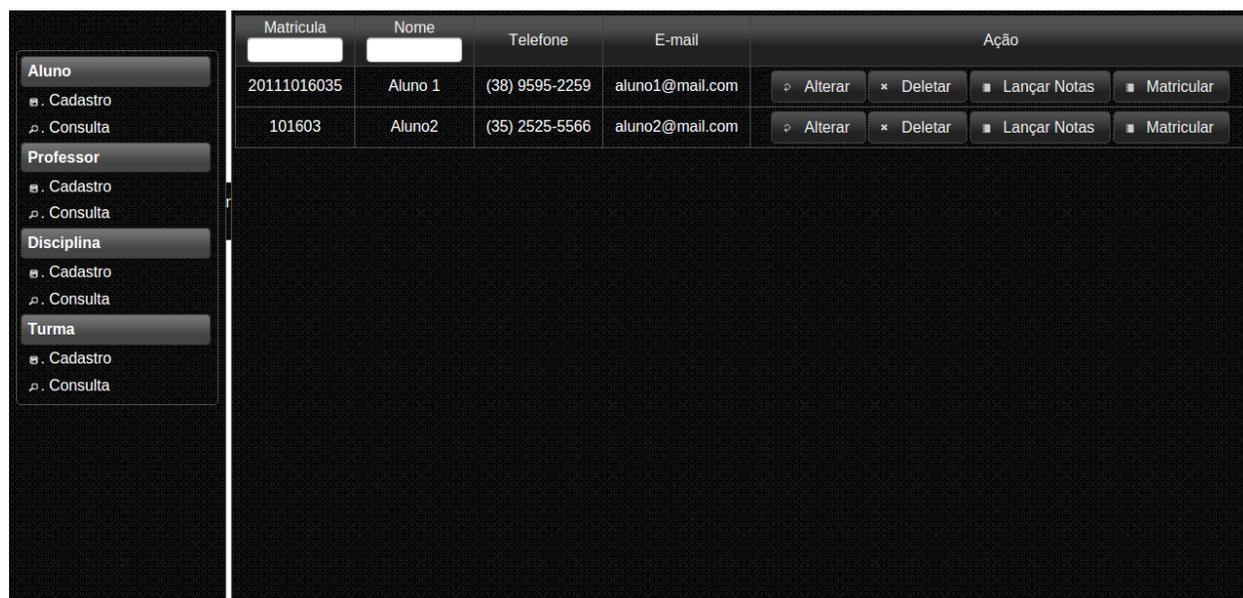


Figura 4.3 : Tela de Consulta de Alunos

Tabela 4.3: Descrição de Campos - Tela de Consulta de Alunos

| Nome | Descrição | Tipo de Entrada |
|-----------|-------------------------------------|-----------------|
| Matricula | Matricula do aluno a ser Consultado | Alfanumérico |
| Nome | Nome do aluno a ser Consultado | Alfanumérico |

Tabela 4.4: Descrição de ações - Tela Consulta Alunos

| Nome | Ação |
|--------------|---|
| Alterar | Redireciona para a página de alteração dos dados do alunos. |
| Deletar | Realiza a exclusão do aluno correspondente ao botão. |
| Lançar Notas | Redireciona a tela de lançamento de notas. |
| Matricular | Redireciona para a página de matrícula de aluno. |

4.4 -Tela de Edição de dados de aluno

A tela de Edição de dados de aluno é representada na Figura 4.4, seus campos são descritos na Tabela 4.5 e as ações dos botões são explicados na Tabela 4.6.

Figura 4.4 :Tela de Edição de Alunos

Tabela 4.5:Descrição de Campos - Consulta de alunos

| Nome | Descrição | Tipo de Entrada |
|--------------------|--|-----------------|
| Nome | Edição do nome do aluno cadastrado | Alfanumérico |
| Matrícula | Edição da matrícula do aluno cadastrado | Alfanumérico |
| Data de Nascimento | Edição do Data de Nascimento do aluno cadastrado | Data |
| Nome do Pai | Edição do nome do Pai do aluno cadastrado | Alfanumérico |
| Nome da Mãe | Edição do Nome da Mãe do aluno cadastrado | Alfanumérico |
| CPF | Edição do CPF do aluno cadastrado | número |

| | | |
|-------------|--|--------------|
| Rua | Edição do Rua (endereço) do aluno a ser cadastrado | Alfanumérico |
| Número | Edição do Número(endereço) do aluno cadastrado | Alfanumérico |
| Complemento | Edição do Complemento (endereço) do aluno cadastrado | Alfanumérico |
| Bairro | Edição do Bairro(Endereço) do aluno cadastrado | Alfanumérico |
| Cidade | Edição da Cidade (endereço) do aluno cadastrado | Alfanumérico |
| CEP | Edição do CEP(endereço) do aluno cadastrado | Alfanumérico |
| UF | UF(endereço) do do aluno a ser cadastrado | Alfanumérico |
| E-mail | Edição do E-mail do aluno cadastrado | Alfanumérico |
| Pais | Edição do Pais do aluno cadastrado | Alfanumérico |
| Telefone | Edição do Telefone do aluno a ser cadastrado | Número |

Tabela 4.6: Descrição de ações - Tela Cadastro Alunos

| Nome | Ação |
|-----------------|--|
| Atualizar dados | Realiza o a atualização dos dados do aluno e redireciona para a página de consulta de aluno. |

4.5-Tela de lançamento de notas

A tela de lançamento de notas aluno é representada na Figura 4.5, seus campos são descritos na Tabela 4.7 e as ações dos botões são explicados na Tabela 4.8.

| Aluno: Aluno 1 | | | | | |
|----------------|-------------|-------------|-------------|-------------|--------------|
| Nome da Turma | 1º Bimestre | 2º Bimestre | 3º Bimestre | 4º Bimestre | Salvar Notas |
| Guitarra01 | A | B | A | A | Salvar Notas |
| Piano03 | B | B | B | B | Salvar Notas |
| Bateria01 | D | D | D | D | Salvar Notas |
| Guitarra05 | A | A | A | A | Salvar Notas |
| Flauta01 | Selezione | Selezione | Selezione | F | Salvar Notas |

Figura 4.5: Tela de lançamento de notas.

Tabela 4.7: Descrição de Campos - Lançamento de notas.

| Nome | Descrição | Tipo de Entrada |
|---------------|---|-----------------|
| Nome da turma | Exibe o nome da turma que o aluno está cadastrado para que se possa lançar as notas correspondente a essa turma | - |
| 1º Bimestre | Nota/Conceito do aluno do 1º Bimestre na disciplina correspondente. | Caracter |
| 2º Bimestre | Nota/Conceito do aluno do 2º Bimestre na disciplina correspondente. | Caracter |

| | | |
|-------------|---|----------|
| 3º Bimestre | Nota/Conceito do aluno do 3º Bimestre na disciplina correspondente. | Caracter |
| 4º Bimestre | Nota/Conceito do aluno do 4º Bimestre na disciplina correspondente. | Caracter |

Tabela 4.8: Descrição de ações - Lançamento de notas

| Nome | Ação |
|--------------|--|
| Salvar Notas | Realiza o lançamento de notas e redireciona a página de consulta de aluno. |

4.6-Tela de Matricula de aluno

A tela de Matricula de aluno é representada na Figura 4.6, seus campos são descritos na Tabela 4.9 e as ações dos botões são explicados na Tabela 4.10 .

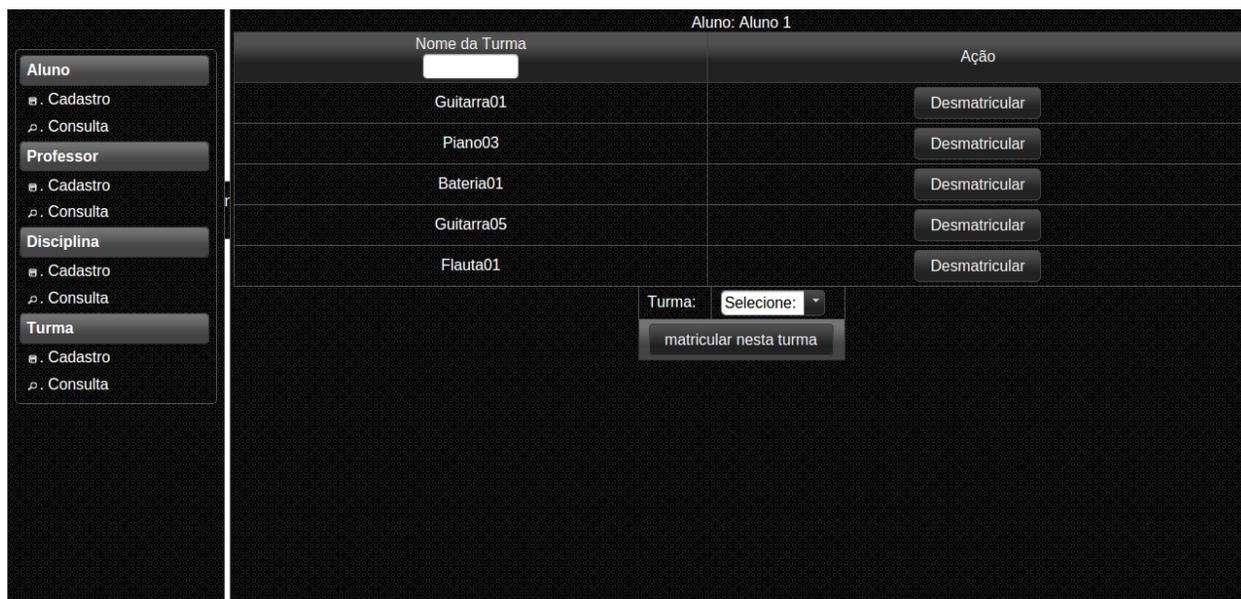


Figura 4.6:Tela de Matricula de Aluno.

Tabela 4.9: Descrição de Campos - Matricula de Aluno.

| Nome | Descrição | Tipo de Entrada |
|---------------|---|-----------------|
| Nome da turma | Nome da turma para consulta | Alfanumérico |
| Turma | Seleção da turma para se efetuar a matricula. | Seleção |

Tabela 4.10: Descrição de ações - Lançamento de notas.

| Nome | Ação |
|------------------------|--|
| Matricular Nessa Turma | Realiza a matricula do aluno na turma selecionada. |

4.7 Cadastro de Professor

A tela de cadastro de professor é representada na Figura 4.7, seus campos são descritos na Tabela 4.11 e as ações dos botões são explicados na Tabela 4.12.

Figura 9:Tela de cadastro de professor

Tabela 4.11: Descrição de Campos - Tela de Cadastro de professor

| Nome | Descrição | Tipo de Entrada |
|-------------|---|-----------------|
| Nome | Nome do Professor a ser cadastrado | Alfanumérico |
| CPF | CPF do Professor a ser cadastrado | número |
| Rua | Rua (endereço) do Professor a ser cadastrado | Alfanumérico |
| Número | Número(endereço) do Professor a ser cadastrado | Alfanumérico |
| Complemento | Complemento(endereço) do Professor a ser cadastrado | Alfanumérico |
| Bairro | Bairro(Endereço) do Professor a ser cadastrado | Alfanumérico |
| Cidade | Cidade (endereço) do a | Alfanumérico |

| | | |
|----------|---|--------------|
| | Professor luno a ser cadastrado | |
| CEP | CEP(endereço) do Professor a ser cadastrado | Alfanumérico |
| UF | UF(endereço) do do aProfessor luno a ser cadastrado | Alfanumérico |
| E-mail | E-mail do Professor no a ser cadastrado | Alfanumérico |
| Pais | Pais do Professor a ser cadastrado | Alfanumérico |
| Telefone | Telefone do Professor a ser cadastrado | Número |

Tabela 4.12: Descrição de ações - Cadastro de professor

| Nome | Ação |
|-------------|--|
| Cadastrar | Realiza o cadastramento do Professor e redireciona para a página de consulta de Professor. |

4.8 - Consulta professor

A tela de Consulta de professor é representada na Figura 4.8, seus campos são descritos na Tabela 4.13 e as ações dos botões são explicados na Tabela 4.14.

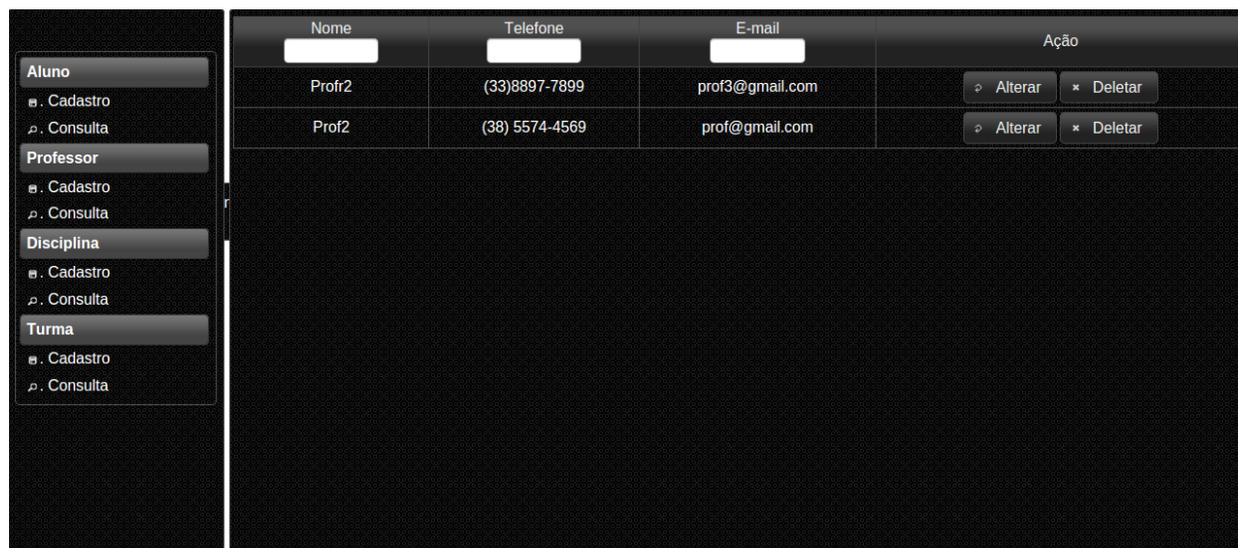


Figura 4.8:Tela de Consulta de Professor

Tabela 4.13: Descrição de Campos - Tela de Consulta de Professor

| Nome | Descrição | Tipo de Entrada |
|-----------|---|-----------------|
| Matricula | Matricula do Professor a ser Consultado | Alfanumérico |
| Telefone | Nome do Professor a ser Consultado | Alfanumérico |
| E-mail | Nome do Professor a ser Consultado | Alfanumérico |

Tabela 4.14: Descrição de ações- Cadastro de professor

| Nome | Ação |
|-------------|--|
| Alterar | Redireciona para a página de alteração dos dados do Professor. |
| Deletar | Realiza a exclusão do professor correspondente ao botão. |

4.9 Tela de Edição de Dados de Professor

A tela de Edição dados do professor é representada na Figura 4.9, seus campos são descritos na Tabela 4.15, e as ações dos botões são explicados na Tabela 4.16.

The screenshot shows a web application interface for editing professor data. On the left, a navigation menu lists categories: Aluno, Professor, Disciplina, and Turma. Each category has sub-options for 'Cadastro' and 'Consulta'. The 'Professor' category is highlighted. The main content area is titled 'Cadastro professor' and contains a form with the following fields and values: Nome: Prof2, CPF: 10160685995, Rua: Rua1, Número: 100, Complemento: Complemento, Bairro: Centro, Cidade: Cidade1, CEP: 00000-000, UF: UF, E-mail: prof@gmail.com, País: Brasil, and Telefone: (38) 5574-4569. A button labeled 'Atualizar dados' is located at the bottom of the form.

Figura 4.9: Tela de Edição de dados de Professor

Tabela 4.15: Descrição de Campos - Tela de Edição de dados de Professor

| Nome | Descrição | Tipo de Entrada |
|-------------|---|-----------------|
| Nome | Edição do nome do Professor cadastrado | Alfanumérico |
| CPF | Edição do CPF do Professor cadastrado | número |
| Rua | Edição do Rua (endereço) do Professor cadastrado | Alfanumérico |
| Número | Edição do Número(endereço) do Professor cadastrado | Alfanumérico |
| Complemento | Edição do Complemento(endereço) do Professor cadastrado | Alfanumérico |
| Bairro | Edição do Bairro(Endereço) do Professor cadastrado | Alfanumérico |

| | | |
|----------|---|--------------|
| Cidade | Edição do Cidade (endereço) do a Professor cadastrado | Alfanumérico |
| CEP | Edição do CEP(endereço) do Professor cadastrado | Alfanumérico |
| UF | Edição do UF(endereço) do do Professor cadastrado | Alfanumérico |
| E-mail | Edição do E-mail do Professor no cadastrado | Alfanumérico |
| Pais | Edição do Pais do Professor cadastrado | Alfanumérico |
| Telefone | Edição do Telefone do Professor cadastrado | Número |

Tabela 4.16: Descrição de ações - Edição de dados de Professor

| Nome | Ação |
|-----------------|--|
| Atualizar dados | Realiza o a atualização dos dados do professor e redireciona para a página de consulta de professor. |

4.10 Cadastro de disciplinas

A tela de Cadastro de Disciplinas é representada na Figura 4.10, seus campos são descritos na tabela 4.17 e as ações dos botões são explicados na tabela 4.18.

The screenshot shows a web application interface for 'Cadastro de Disciplinas'. On the left, there is a sidebar with a menu containing 'Aluno', 'Professor', 'Disciplina', and 'Turma'. The 'Disciplina' menu item is highlighted. The main area contains a form with the following fields: 'Nome : *', 'Codigo :', 'Curso :', 'Serie :', and 'Carga Horaria :'. Each field has a corresponding text input box. At the bottom of the form, there is a 'Cadastrar' button.

Figura 4.10:Tela de Cadastro de Disciplinas.

Tabela 4.17: Descrição de Campos -Tela de Cadastro de Disciplinas.

| Nome | Descrição | Tipo de Entrada |
|---------------|--|-----------------|
| Nome | Nome da disciplina da disciplina a ser cadastrada | Alfanumérico |
| Código | Código da disciplina da disciplina a ser cadastrada | Alfanumérico |
| Curso | Curso da disciplina a ser cadastrada | Alfanumérico |
| Serie | Serie da disciplina da disciplina a ser cadastrada | Alfanumérico |
| Carga Horária | Carga Horária da disciplina da disciplina a ser cadastrada | Alfanumérico |

Tabela 4.18: Descrição de ações - Cadastro de Disciplinas.

| Nome | Ação |
|-------------|---|
| Cadastrar | Realiza o cadastramento da disciplina e redireciona a página de consulta de disciplina. |

4.11 Consulta de disciplinas

A tela de Consulta de Disciplinas é representada na Figura 4.11, seus campos são descritos na Tabela 4.19e as ações dos botões são explicados na Tabela 4.20.



Figura 4.11: Tela de Consulta de Disciplinas.

Tabela 4.19: Descrição de Campos -Tela de Consulta de Disciplinas.

| Nome | Descrição | Tipo de Entrada |
|--------|---|-----------------|
| Nome | Nome da disciplina da disciplina a ser consultada | Alfanumérico |
| Código | Código da disciplina da disciplina a ser consultada | Alfanumérico |

Tabela 4.20: Descrição de ações - Consulta de Disciplinas.

| Nome | Ação |
|---------|--|
| Alterar | Redireciona a página de alteração da disciplina. |
| Deletar | Realiza a exclusão da disciplina. |

4.12 Alteração de disciplina

A tela de Alteração de Disciplinas é representada na Figura 4.12, seus campos são descritos na Tabela 4.21 e as ações dos botões são explicados na Tabela 4.22.

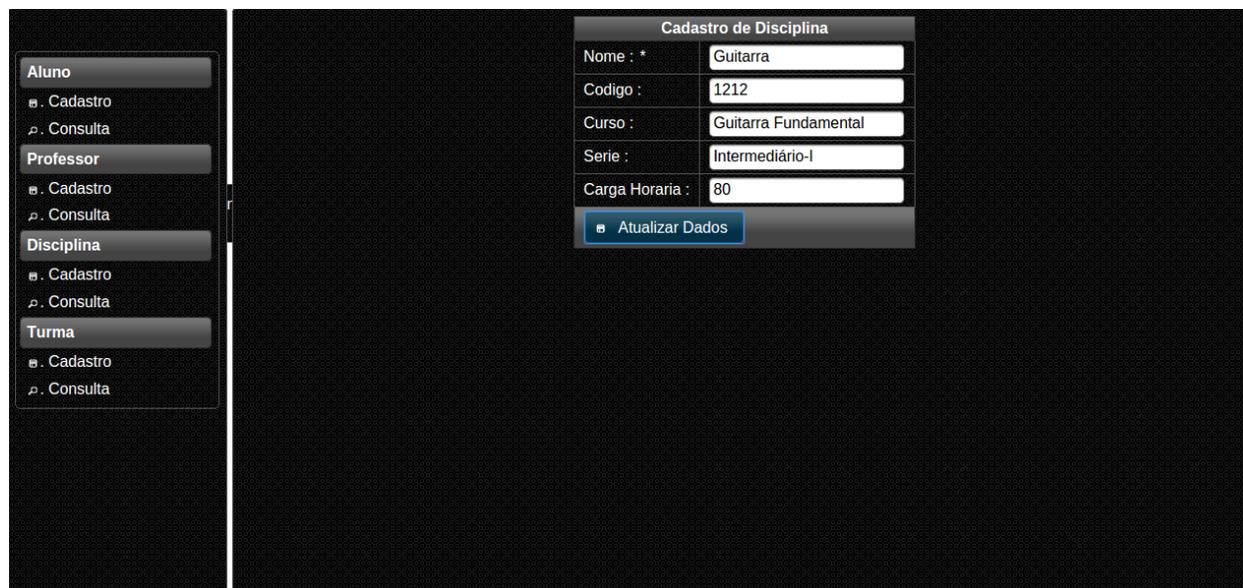


Figura 4.12: Tela de Alteração de Disciplinas.

Tabela 4.21: Descrição de Campos -Tela de Alteração de Disciplinas.

| Nome | Descrição. | Tipo de Entrada |
|---------------|---|-----------------|
| Nome | Edição do nome da disciplina. | Alfanumérico |
| Código | Edição do código da disciplina. | Alfanumérico |
| Curso | Edição do Curso da disciplina cadastrada. | Alfanumérico |
| Serie | Edição do serie da disciplina cadastrada. | Alfanumérico |
| Carga Horária | Edição Carga Horária da disciplina da cadastrada. | Alfanumérico |

Tabela 4.22: Descrição de ações - Alteração de Disciplinas.

| Nome | Ação |
|-----------------|--|
| Atualizar dados | Atualiza os dados da disciplina e redireciona á página de consulta de disciplinas. |

4.13- Cadastro de turmas

A tela de cadastro de turmas é representada na Figura 4.13, seus campos são descritos na Tabela 4.23 e as ações dos botões são explicados na Tabela 4.24.

Figura 4.13: Tela de Cadastro de Turmas.

Tabela 4.23: Descrição de Campos -Tela de Cadastro de Turmas.

| Nome | Descrição | Tipo de Entrada |
|------------|---|-----------------|
| Nome | Nome da Turma a ser cadastrada. | Alfanumérico |
| Disciplina | Nome Disciplina a ser cadastrada. | Seleção |
| Professor | Nome do professor que leciona a disciplina. | Seleção |

Tabela 4.24: Descrição de ações - Cadastro de Turmas.

| Nome | Ação |
|-----------|--|
| Cadastrar | Realiza o cadastramento da turma e redireciona a página de consulta de turmas. |

4.14 Consulta de turmas

A tela de consulta de turmas é representada na Figura 4.14, seus campos são descritos na Tabela 4.25 e as ações dos botões são explicados na Tabela 4.26.



Figura 4.14:Tela de Consulta de Turmas.

Tabela 4.25: Descrição de Campos -Tela de Consulta de Turmas.

| Nome | Descrição | Tipo de Entrada |
|------|---------------------------------|-----------------|
| Nome | Nome da Turma a ser consultada. | Alfanumérico |

Tabela 4.26: Descrição de ações - Cadastro de Turmas.

| Nome | Ação |
|---------|--|
| Alterar | Redireciona a página de alteração de turma. |
| Deletar | Realiza exclusão da turma |
| Aulas | Redireciona a página de cadastramento de aula para a disciplina selecionada. |

4.15 Edição de turmas.

A tela de edição de turmas é representada na Figura 4.15, seus campos são descritos na Tabela 4.27 e as ações dos botões são explicados na Tabela 4.28.

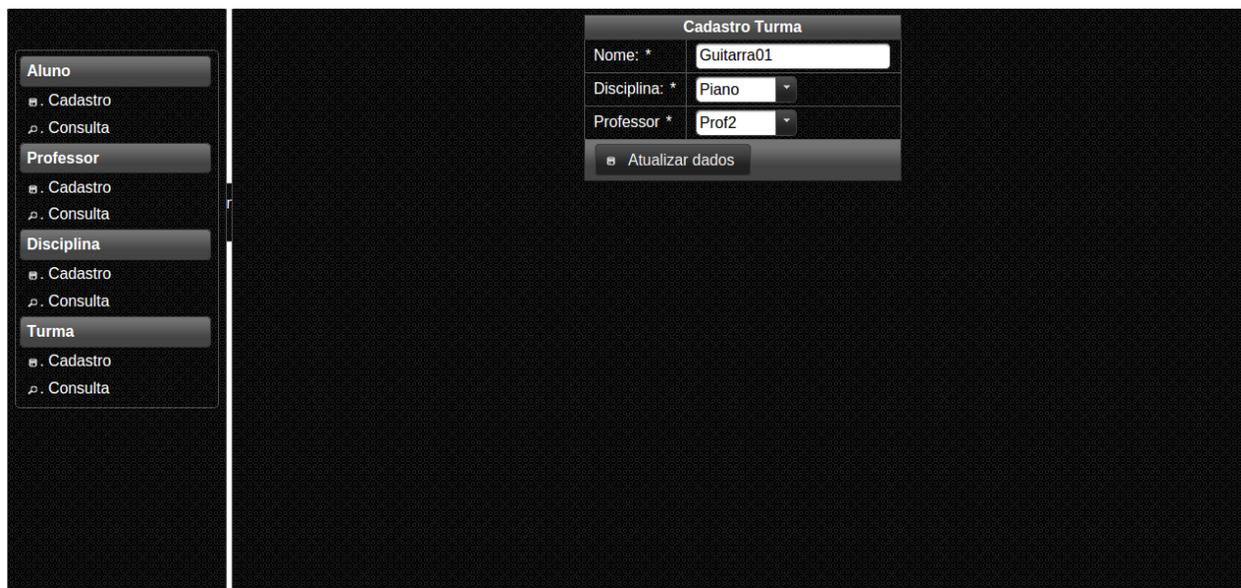


Figura 4.15: Tela de Edição de Turmas.

Tabela 4.27: Descrição de Campos -Tela de Edição de Turmas.

| Nome | Descrição | Tipo de Entrada |
|------------|---|-----------------|
| Nome | Edição do Nome da Turma cadastrada. | Alfanumérico |
| Disciplina | Edição do Nome Disciplina cadastrada. | Seleção |
| Professor | Edição do Nome do professor que leciona a disciplina. | Seleção |

Tabela 4.28: Descrição de ações - Edição de Turmas.

| Nome | Ação |
|-----------------|--|
| Atualizar dados | Realiza a alteração dos dados da turma e redireciona a página de consulta de turmas. |

4.16 Cadastro de aulas

A tela de cadastro de aulas é representada na Figura 4.16, seus campos são descritos na Tabela 4.29 e as ações dos botões são explicados na Tabela 4.30.



Figura 4.16: Tela de Cadastro de Aulas.

Tabela 4.29: Descrição de Campos -Tela de Cadastro de Aulas.

| Nome | Descrição | Tipo de Entrada |
|---------------|---|-----------------|
| Local | Local /sala onde a aula será realizada. | Alfanumérico |
| Horário | Horário da aula. | Alfanumérico |
| Dia da semana | Dia da semana que a aula será realizada | Seleção |

Tabela 4.30: Descrição de ações - Cadastro de Aulas

| Nome | Ação |
|----------------|---|
| Cadastrar aula | Realiza o cadastramento de uma aula para uma determinada turma. |
| Remover | Exclui a aula da lista de aulas da turma correspondente. |

CAPÍTULO 5 - TESTES REALIZADOS

Para o sistema **SisGTC** - Sistema de Gerenciamento de Turmas de Conservatórios realizou-se testes para verificar o funcionamento correto do sistema, com foco nas relações de persistência de dados em cadastros funcionais. Neste capítulo são descritos e representados (através de imagens) todos os testes bem como os resultados de cada um. Todos os testes do sistema são representados na tabela 5.1.

Tabela 5.1: Operações de Teste

| Ponto de Função | Operação de Teste | Resultado | Avaliação |
|----------------------|---|---------------------------------|-----------|
| Cadastro de Alunos | Preenchimento de todos os campos seguindo do clique no botão "Cadastrar". | Aluno Cadastrado com Sucesso. | Positiva |
| Edição de Aluno | Alteração de todos os campos seguindo do clique no botão "Alterar". | Operação realizada com sucesso. | Positiva |
| Exclusão de Aluno | Escolha de Aluno para a exclusão. Clique em "Deletar". | Operação realizada com sucesso. | Positiva |
| Lançamento de notas. | Seleção de todas as notas em todas as disciplinas cadastradas. Clique em "Salvar Notas", para cada disciplina correspondente. | Operação realizada com sucesso. | Positiva |
| Matricula | Seleção da Turma, e clique em "Matricular Nessa Turma". | Operação realizada com sucesso. | Positiva |

| | | | |
|-------------------------|--|-----------------------------------|----------|
| Cadastro de Professores | Preenchimento de todos os campos seguindo do clique no botão "Cadastrar". | Professor Cadastrado com Sucesso. | Positiva |
| Edição de Professores | Alteração de todos os campos seguindo do clique no botão "Alterar". | Operação realizada com sucesso. | Positiva |
| Exclusão de Professores | Escolha de Aluno para a exclusão. Clique em "Deletar". | Operação realizada com sucesso. | Positiva |
| Cadastro de Disciplina | Preenchimento de todos os campos seguindo do clique no botão "Cadastrar". | Operação realizada com sucesso. | Positiva |
| Edição de Disciplina | Alteração de todos os campos seguindo do clique no botão "Alterar". | Operação realizada com sucesso. | Positiva |
| Exclusão de Disciplina | Escolha da Disciplina para a exclusão. Clique em "Deletar". | Operação realizada com sucesso. | Positiva |
| Cadastro de turmas | Preenchimento de todos os campos e clique em "Cadastrar". | Operação Realizada com Sucesso. | Positiva |
| Alteração de Turmas | Preenchimento de todos os campos e clique em "Alterar". | Operação Realizada com Sucesso. | Positiva |
| Exclusão de Turmas | Escolha da Turma para a exclusão. Clique em deletar. Clique em "Sim" na Janela de Confirmação. | Operação Realizada com Sucesso. | Positiva |

| | | | |
|---|---|---|----------|
| Cadastro de Turmas | Preenchimento de todos os campos, clique no Botão "Cadastrar aulas". | Operação Realizada com Sucesso. | Positiva |
| Exclusão de Aulas | Escolha da Turma para a exclusão. Clique botão "Remover". Clique em "Sim" na Janela de Confirmação. | Operação Realizada com Sucesso. | Positiva |
| Exclusão de Disciplinas Vinculadas a um turma. | Escolha da Turma para a exclusão. Clique em deletar. Clique em "Sim" na Janela de Confirmação. | Não foi possível remover a disciplina pois está vinculada a uma Turma. | Positiva |
| Exclusão de Professores Vinculados a uma Turma. | Escolha da Turma para a exclusão. Clique em deletar. Clique em "Sim" na Janela de Confirmação. | Não foi possível remover a o Professor pois está vinculada a uma Turma. | Positiva |

CAPÍTULO 5 - CONCLUSÃO

5.1 - Conclusão Sobre a Solução

Tendo em vista as necessidades de automatização e modernização de processos e atividades de controle escolar do Conservatório Estadual de Musica Lobo de Mesquita de Diamantina, o sistema desenvolvido se mostra como a primeira iteração para que isso ocorra, para tanto, o sistema abertamente disponível para que outro desenvolvedor, seja acadêmico ou não, prossiga com o desenvolvimento do mesmo, para que o sistema se torne o mais completo, eficiente e eficaz possível.

5.2- Conclusão Técnica

Neste trabalho a linguagem de programação Java se mostrou muito eficiente de principalmente por seus artifícios de reaproveitamento de código utilizados mas apresentando como desafio a configuração do ambiente de desenvolvimento que demandou grande parte do tempo de desenvolvimento na integração das várias tecnologias e ferramentas utilizadas, principalmente a relacionado as bibliotecas de relacionadas ao modelo de persistência de dados.

CAPÍTULO 6 - BIBLIOGRAFIA

ALBUQUERQUE, F. Conceitos de Orientação a Objetos, Disponível em: <<http://www.cic.unb.br/~fernando/matdidatico/apostilas/resumo/conpoosp.pdf>> Acesso em: 12 de maio 2014.

BRITO, R.S. Conceitos de Orientação a Objetos 2012. Disponível em: <<http://social.technet.microsoft.com/wiki/pt-br/contents/articles/9644.conceitos-de-orientacao-a-objetos.aspx>> Acesso em: 28 de abril 2014.

CARMISINI, A.; VAHLDICK, A. (2012) “Comparativo Entre Frameworks de Javaserer Faces: Apache Tobago, Primefaces e Richfaces”. Santa Catarina.

CONTE, T., MENDES, E., TRAVASSOS G. H. () “Processos de Desenvolvimento para Aplicações Web: Uma Revisão Sistemática”. Rio de Janeiro.

CAMPOLI C., FREITAS M. C., OLIVEIRA, S. M.C. (2008) “Gestão da Informação na Escola Pública Estadual do Paraná: Aplicabilidade Pedagógica do Sistema de Informação do Registro Escolar”. Curitiba-PR.

COSTA, J. A., ROQUE, A. (2005) “ A Gestão da Informação no Contexto da Gestão Escolar”.

DEITEL, H. M.; DEITEL, P. J.(2005) “Java: Como Programar. 6. Ed. São Paulo: Pearson Education”.

FARINELLI, F. (2007) “Conceitos Básicos de Programação Orientada a Objetos”.

SANTOS, L. M. Conceitos Básicos de Programação Orientada a Objetos. Disponível em: <<http://pt.slideshare.net/leonardomelosantos/conceitos-bsicos-de-programao-orientada-a-objetos>> Acesso em: 14 maio. 2014.

SANTOS, R.(2003) “Introdução à Programação Orientada a Objetos Usando Java . 1. Ed. Rio de Janeiro: Campus”.

SIERRA, K.; BATES, B.(2007) "Use a Cabeça! Java. 2. Ed. Rio de Janeiro: Alta Books".

.

VOUILLAMOZ N. (2000) "Literatura e Hipermedia La irrupción de la literatura interactiva: precedentes y crítica. Barcelona; Paidós Papeles de comunicación".