

Sistema de controle acadêmico

Academic Control System

Vinícius Jonathan Santos Silva

Graduando do curso de Sistemas de Informação (UNIPAM).

E-mail: viniciusjns@live.com

Juliana Lilis da Silva

Professora orientadora (UNIPAM).

E-mail: juliana@unipam.edu.br

Resumo: Controlar as informações acadêmicas de uma escola não é nada fácil. São centenas de alunos, dezenas de professores, muitas notas, trabalhos, aulas e diversos outros aspectos que merecem uma atenção especial. Os muitos dados manipulados nos processos internos de uma instituição de ensino, muitas vezes, geram um esforço repetitivo e um gasto de tempo muito alto para realizar determinadas tarefas. Pensando nisso, foi desenvolvido um sistema de controle acadêmico que irá auxiliar secretárias, alunos e professores nas suas atividades diárias, como gerenciamento das aulas, gerenciamento de notas e trabalhos e geração dos boletins dos alunos. Como esse é um sistema *web*, os usuários poderão acessá-lo de qualquer lugar e a qualquer hora, gerando maior facilidade e comodidade para quem o estiver utilizando.

Palavras-chave: Sistema de Controle Acadêmico. *Software. Java. Web.*

Abstract: Check the academic information of a school is not easy. There are hundreds of students, dozens of teachers, many notes, papers, lectures and many other aspects that deserve special attention. Many data handled in the internal processes of an educational institution, often generates a repetitive strain and a very high time taken to complete certain tasks. Thinking about it, we developed an academic control system that will help secretaries, students and teachers in their daily activities, such as classes' management, notes and management jobs and generation of reports from students. Since this is a web system, users can access it from anywhere and at any time -day or night- creating greater ease and convenience for those using it.

Keywords: Academic Control System. *Software. Java. Web.*

1 INTRODUÇÃO

Nos últimos anos, houve um enorme avanço tecnológico em todas as áreas do conhecimento e em todo o mundo. Percebe-se que a cada dia surgem novidades nas mais diversas áreas de atuação, seja corporativa, comercial ou educacional. Essas novidades facilitam e agilizam o desenvolvimento das tarefas cotidianas.

Uma característica da sociedade neste início do século XXI é o fato das pessoas terem a necessidade de se conectarem à Internet o tempo todo, compartilhando vivências nas redes sociais, conhecendo pessoas e realizando seu trabalho de forma rápida e mais eficiente.

Nesse contexto, muitas universidades e escolas particulares já se preparam para essa nova realidade, pois elas enxergaram o rumo que o mundo está tomando e querem acompanhar todas as mudanças de perto. Muitas dessas escolas possuem um sistema informatizado que auxilia os professores a realizar as tarefas acadêmicas, como postar notas, receber trabalhos e realizar frequência dos alunos. De um outro lado, encontram-se as escolas públicas, que são escolas que têm o mesmo desejo de crescer, porém não possuem recursos suficientes para melhorar seu ambiente de ensino, muito menos para adquirir um sistema informatizado.

É para auxiliar uma dessas escolas - Escola Estadual Marcolino de Barros - que foi feito este trabalho. Ela não possui nenhum tipo de sistema informatizado que auxilie a gestão das informações para o funcionamento da mesma. Atualmente, tudo é feito de forma manual, o que pode demandar um maior período de tempo para realizar determinadas tarefas.

Nota-se um nível grande de complexidade que existe em alocar os alunos em turmas, os professores em aulas, controlar as notas de todos os alunos e imprimir o boletim ao final do período letivo. Isso se torna cansativo para todos que fazem parte da comunidade escolar, pois todo início de ano novos alunos chegam à escola e alunos veteranos precisam ser realocados para outras turmas.

Vendo a necessidade de um sistema de apoio aos processos internos da escola, foi desenvolvido um *software* para gerenciamento de notas e frequência dos alunos, por meio do qual os professores e alunos terão um maior controle de cada atividade e da frequência em cada aula.

2 REFERENCIAL TEÓRICO

Nesta seção, são abordados o processo e o padrão de projeto que foram utilizados no desenvolvimento do Sistema de Controle Acadêmico: o PU (Processo Unificado) e o padrão MVC (Model-View-Control).

2.1 Processo Unificado

Um processo de desenvolvimento de software é um conjunto de atividades relacionadas que levam à produção de um produto de software (SOMMERVILLE, 2011).

O Processo Unificado ou PU é um processo de desenvolvimento de software criado para que desenvolvedores possam criar sistemas orientados a objetos com uma documentação mais elaborada e códigos mais limpos. Ele é classificado como um processo iterativo e incremental. Isso quer dizer que o produto final será dividido em iterações (tempo necessário para desenvolver um artefato) de duração fixa e o sistema crescerá incrementalmente, iteração por iteração.

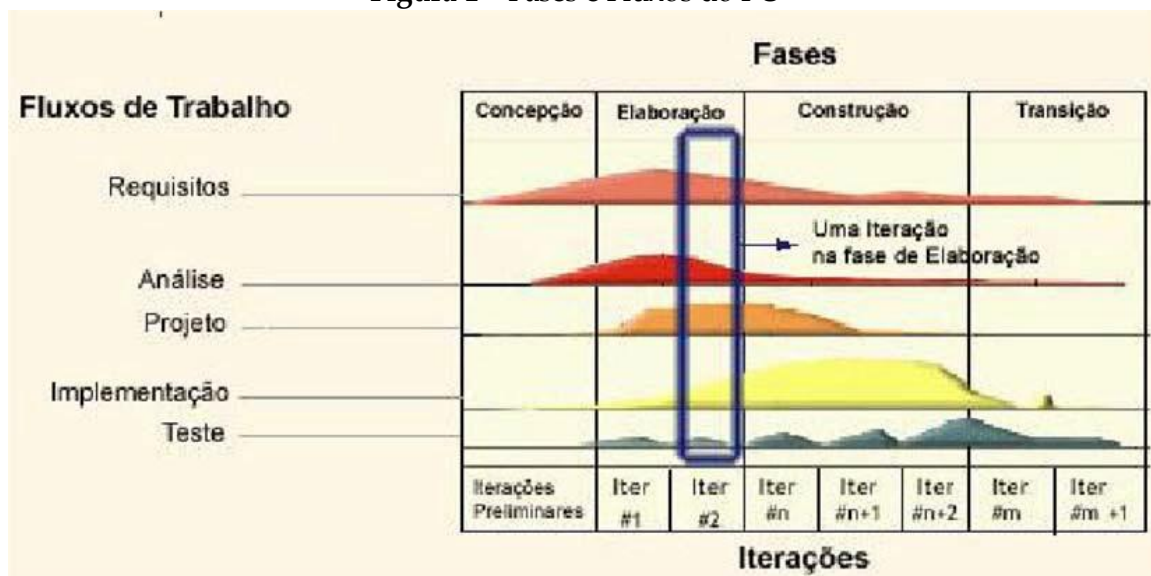
Segundo Larman *et al.* (2007), os benefícios de usar uma abordagem iterativa e incremental incluem:

- menos erros de projeto, maior produtividade e menor taxa de defeitos;
- mitigação precoce, em vez de tardia, de altos riscos;
- progresso visível desde o início;

- realimentação, envolvimento do usuário e adaptação imediatos, levando a um sistema refinado que atenda, de forma mais adequada, às reais necessidades dos interessados no projeto;
- administração da complexidade; a equipe não é sobrecarregada pela paralisia da análise ou por passos muito longos e complexos;
- uso metódico do aprendizado obtido em uma iteração para melhorar o próprio processo de desenvolvimento.

O PU é dividido em fases e fluxos. As fases são divididas em: concepção, elaboração, construção e transição; e os fluxos são divididos em: requisito, análise, projeto, implementação e teste. Cada fase possui uma quantidade n de iterações que, por sua vez, usam todos os fluxos. A Figura 1 mostra as fases e fluxos do PU.

Figura 1 – Fases e Fluxos do PU

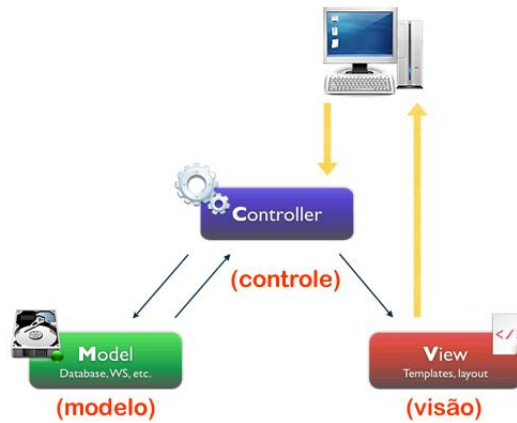


Fonte: Adaptado (LARMAN, 2005).

Como demais características do PU, podem-se citar a arquitetura e a utilização de diagramas de caso de uso. A arquitetura fornece a estrutura que guia o trabalho de cada iteração, enquanto os diagramas proporcionam aos usuários e demais interessados uma visão mais clara do que está sendo desenvolvido.

2.2 MVC

O MVC (*Model-View-Control*) é um padrão de projeto muito utilizado para sistemas orientados a objetos. Ele separa a lógica da aplicação das regras de negócio, gerando uma melhor manutenibilidade do código. Essa separação é feita em forma de camadas, as quais se dá o nome do padrão MVC (LUCKOW; MELO, 2010). A Figura 2 apresenta a estrutura do modelo MVC.

Figura 2 – Estrutura do MVC

Fonte: Criação dos autores deste estudo

Neste modelo, cada camada possui uma responsabilidade. A *model* é a camada de acesso aos dados, é ela que ficará responsável por salvar e buscar informações em um banco de dados. Ela também é responsável por gerenciar as regras de negócio do sistema. A *view* é a camada de apresentação, na qual o usuário terá acesso ao sistema, ou seja, às telas do sistema. A última camada, a *control*, é a camada que faz a ligação entre a *view* e a *model*, ela ficará responsável por filtrar os dados que vão da camada *view* para a *model* e vice-versa.

3 METODOLOGIA

O sistema de controle acadêmico foi desenvolvido usando o processo de desenvolvimento chamado PU, que é um processo para desenvolvimento de software composto por fases e fluxos, no qual cada fase possui todos os fluxos e é determinada por uma quantidade variável de iterações. A Tabela 1 descreve os artefatos desenvolvidos em cada fase.

Tabela 1 – Descrição dos Artefatos Desenvolvidos em Cada Fase do PU

Fase	Descrição
Concepção	<ul style="list-style-type: none"> Levantamento dos requisitos; Caso de uso que modela o contexto; Protótipos do sistema;
Elaboração	<ul style="list-style-type: none"> Especificação dos requisitos; Caso de uso que modela o requisito; Diagrama Entidade Relacionamento; Definição das tecnologias utilizadas para o desenvolvimento; Documento de Requisitos;
Construção	<ul style="list-style-type: none"> Codificação do sistema baseado no documento de requisitos; Criação do banco de dados;
Transição	<ul style="list-style-type: none"> Testes no ambiente de execução real para detecção de possíveis erros.

Fonte: Criação dos autores deste estudo

A Tabela 2 lista todas as ferramentas e tecnologias usadas no desenvolvimento do sistema de controle acadêmico.

Tabela 2 – Ferramentas e Tecnologias Utilizadas

Nome	Descrição
Eclipse	IDE de desenvolvimento que dá suporte a linguagem JAVA
Visual Paradigm for UML	Software utilizado para desenvolver os diagramas UML
MockupBuilder	Software para modelagem de protótipos
MySQL	Sistema gerenciador de banco de dados
MySQL Workbench	Ferramenta auxiliar para controle de dados
JEE	Plataforma de desenvolvimento JAVA voltada para internet
Hibernate/JPA	Frameworks utilizados no mapeamento de objetos para o banco de dados
Apache Tomcat	Container <i>web</i> utilizado para rodar aplicações JEE
Microsoft Office Word	Editor de texto utilizado para gerar a documentação do projeto

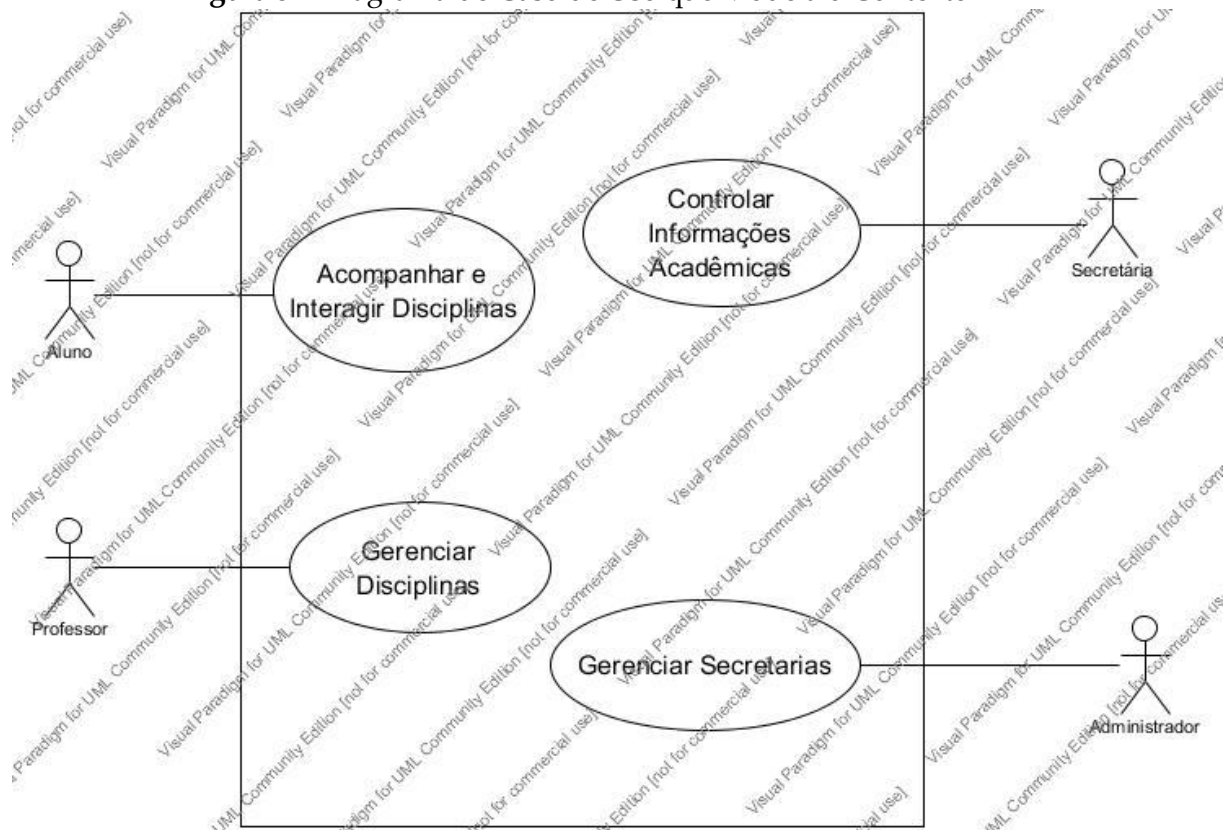
Fonte: Criação dos autores deste estudo

Na fase de concepção e elaboração, foram usadas as ferramentas *Microsoft Office Word* e *Visual Paradigm for UML* para documentação dos requisitos, construção dos diagramas de caso de uso, diagrama entidade relacionamento e diagrama de classes. Na fase de construção, foram usados o *Eclipse*, o *MySQL Workbench*, o *Hibernate* com *JPA* e o *Apache Tomcat* para codificação do *software*.

4 DESENVOLVIMENTO E RESULTADOS

Na fase de concepção, foi realizada uma pesquisa para saber como era o ambiente escolar. Após o levantamento dos requisitos, foi desenvolvido o diagrama de caso de uso que modela o contexto do sistema, o qual está representado na Figura 3.

Existem outros sistemas semelhantes, que hoje são utilizados por várias instituições educacionais, para melhor prover a organização e manutenção dos dados de alunos e secretaria. Para melhor especificar as funcionalidades do sistema proposto, foi analisado o Portal Acadêmico do UNIPAM (Centro Universitário de Patos de Minas), identificando quais funções poderiam ser implementadas, de forma a atender à demanda da escola. O Portal Acadêmico do UNIPAM é restrito a alunos e funcionários da instituição.

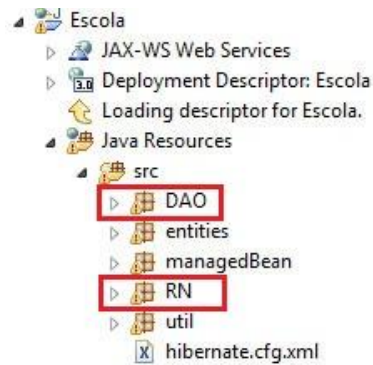
Figura 3 – Diagrama de Caso de Uso que Modela o Contexto

Fonte: Criação dos autores deste estudo

Na fase de elaboração do projeto, foi feito o detalhamento dos requisitos e apontados alguns pontos importantes para o funcionamento do sistema. Algumas das principais funcionalidades são:

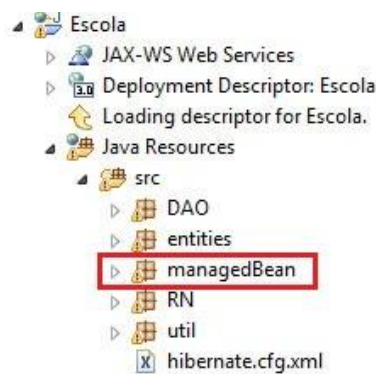
- manter informações acadêmicas: esse módulo tem a função de manter as informações básicas do sistema, como dados de alunos, professores, turmas, disciplinas e aulas.
- gerenciar trabalhos, notas e frequência: tem a função de gerenciar os trabalhos, notas e frequências cadastradas pelo professor.
- consultar trabalhos, notas e frequência: tem a função de apresentar aos alunos os trabalhos, notas e frequências cadastradas pelo professor.
- gerar boletim escolar: tem a função de emitir o boletim escolar dos alunos.

O sistema foi desenvolvido em uma arquitetura de três camadas, conhecida como MVC, na qual cada camada possui uma responsabilidade diferente. A Figura 4 apresenta os pacotes que contêm os códigos fonte da camada *model*. Ao pacote DAO pertencem as classes que fazem a comunicação com o banco de dados e ao pacote RN as classes responsáveis pelas regras de negócio do sistema.

Figura 4 – Camada *Model*

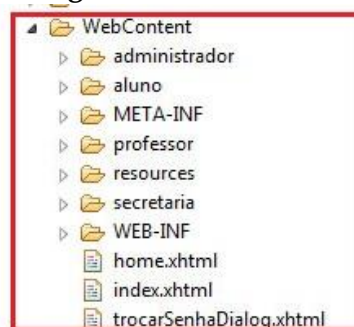
Fonte: Criação dos autores deste estudo

O pacote apresentado na Figura 5 representa a camada *control*. Nele estão os códigos que controlam quais informações são enviadas ao banco de dados.

Figura 5 – Camada *Control*

Fonte: Criação dos autores deste estudo

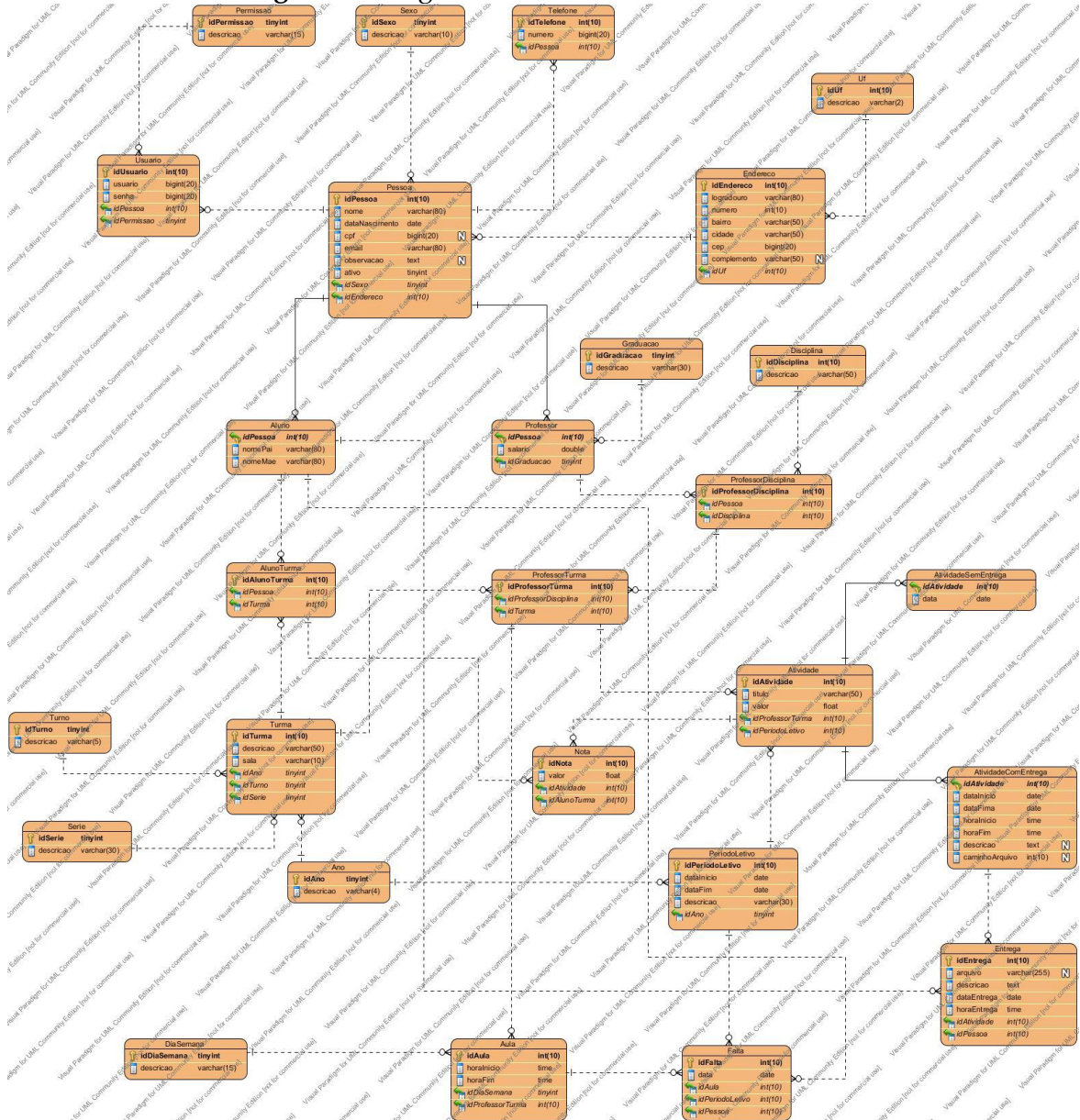
A camada *view* é representada na Figura 6. Nela, as telas do sistema são separadas por pasta, de acordo com cada perfil de usuário que tem acesso ao sistema. As telas *home.xhtml*, *index.xhtml* e *trocarSenhaDialog.xhtml* são comuns a todos os usuários.

Figura 6 – Camada *View*

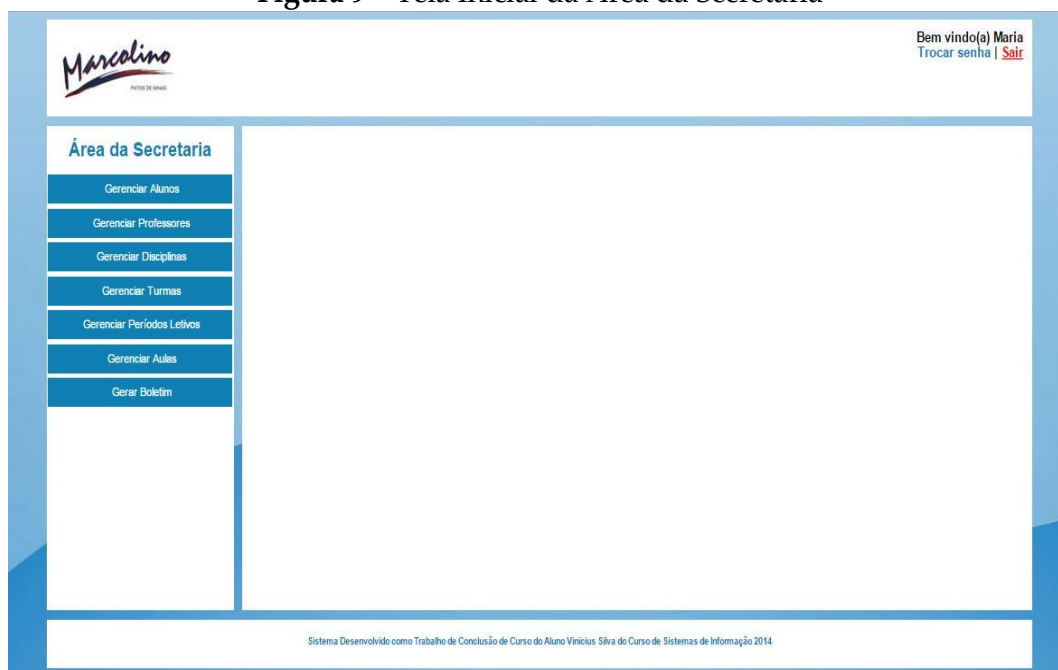
Fonte: Criação dos autores deste estudo

A Figura 7 apresenta o Diagrama Entidade Relacionamento (DER), contendo todas as tabelas com seus respectivos atributos e os relacionamentos. As tabelas 'Aluno', 'Professor', 'AlunoTurma', 'ProfessorDisciplina' e 'ProfessorDisciplinaTurma' representam as áreas centrais do sistema, assim como seus respectivos relacionamentos. Uma característica que pode ser observada é no relacionamento das tabelas 'Endereço' e 'Alunos', no qual tem-se um endereço para muitos alunos. Esse relacionamento foi desenvolvido dessa forma para que seja possível, posteriormente, implementar uma função que possibilite selecionar endereços já cadastrados no sistema para alunos que residam no mesmo local.

Figura 7 – Diagrama Entidade Relacionamento



Fonte: Criação dos autores deste estudo

Figura 9 – Tela Inicial da Área da Secretaria

Fonte: Criação dos autores deste estudo

A Figura 10 ilustra a tela de cadastro de novos alunos no sistema. A secretária irá informar os dados em seus respectivos campos e clicar no comando Salvar para que os dados do aluno sejam gravados no banco de dados. A secretária pode cancelar o cadastro clicando no comando Cancelar.

Figura 10 – Tela de Cadastro de Alunos

Fonte: Criação dos autores deste estudo

A tela ilustrada na Figura 11 apresenta a listagem de todos os alunos cadastrados no sistema e as ações para gerenciamento dos mesmos, como o comando Editar, o comando Excluir e o comando Consultar.

Figura 11 – Tela de Listagem de Alunos

Fonte: Criação dos autores deste estudo

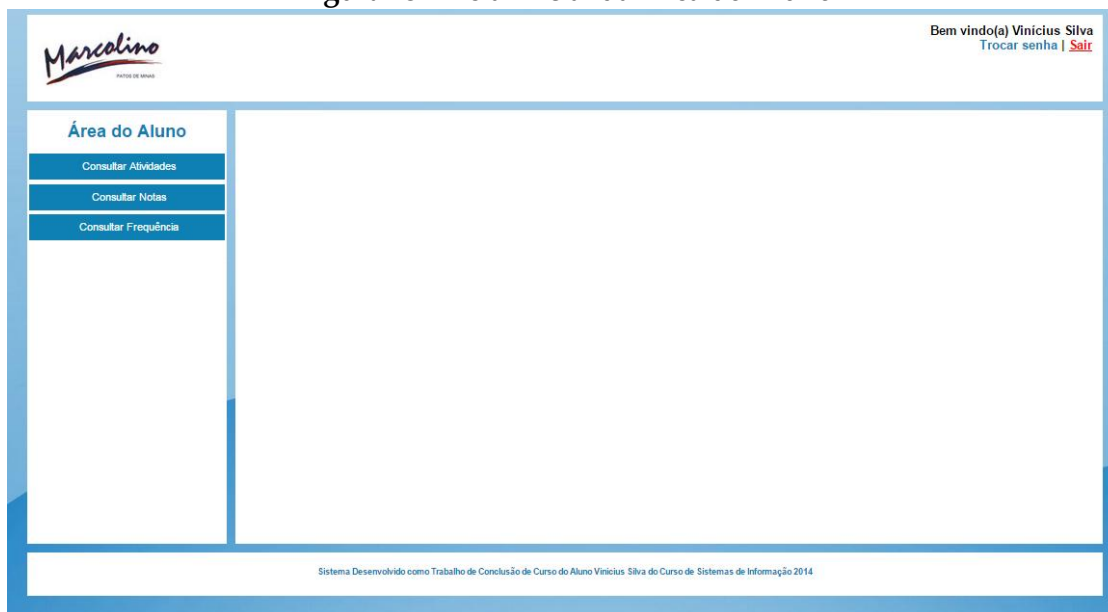
Além da área da secretaria, o sistema ainda apresenta a área dos professores, a área dos alunos e a área do administrador. A seguir, na Figura 12, é ilustrada a tela inicial da área do professor, por meio da qual o professor pode acessar as opções de Gerenciar Atividades, Gerenciar Notas e Gerenciar Frequência.

Figura 12 – Tela Inicial da Área do Professor

Fonte: Criação dos autores deste estudo

A Figura 13 ilustra a tela inicial da área do aluno, onde o mesmo tem as opções de consultar os trabalhos cadastrados pelo professor, as notas postadas e as faltas nas aulas durante o período letivo.

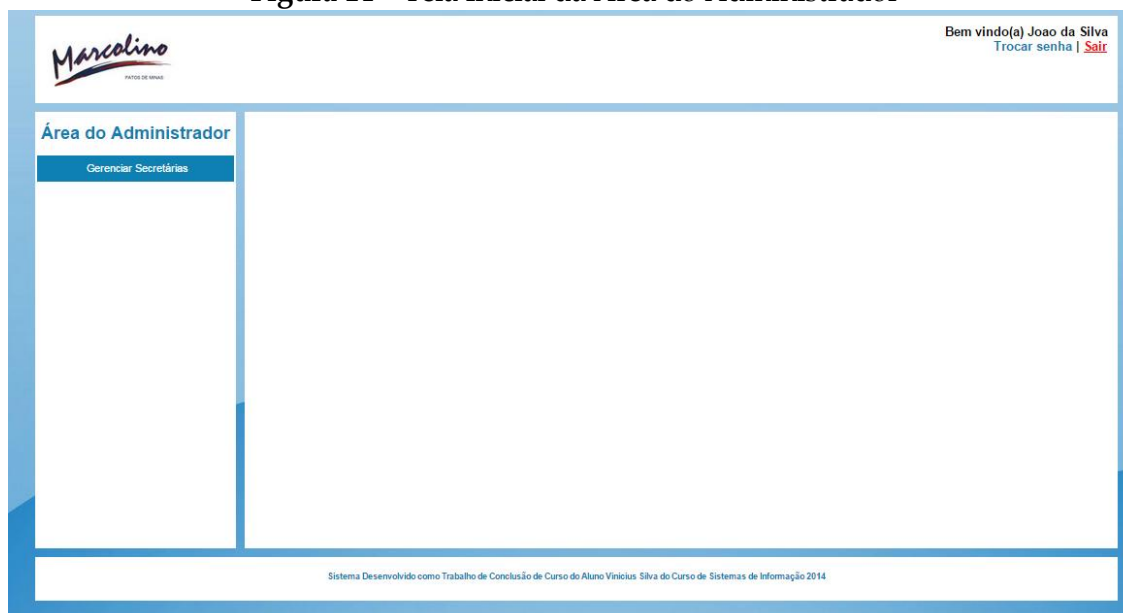
Figura 13 – Tela Inicial da Área do Aluno



Fonte: Criação dos autores deste estudo

A área do administrador do sistema é apresentada abaixo pela Figura 14.

Figura 14 – Tela Inicial da Área do Administrador



Fonte: Criação dos autores deste estudo

A imagem apresentada na Figura 15 é a representação de como será o boletim escolar dos alunos.

Figura 15 – Tela do Boletim

Boletim Escolar										
EE MARCOLINO DE BARROS										
AV GETÚLIO VARGAS, 367, CENTRO, PATOS DE MINAS, Telefone: (34)38212652										
SRE PATOS DE MINAS										
Nome: VINICIUS SILVA									Ano: 2014	
Nível de Ensino: 3º ANO										
Turno: MANHÃ			Turma: 309			Ano: ENSINO MÉDIO			Número: 2	
Divisões	1º bimestre		2º bimestre		3º bimestre		4º bimestre		Total	
	Apr	Fal	Apr	Fal	Apr	Fal	Apr	Fal	Apr	Fal
FISICA	0.0	0	0.0	0	0.0	0	0.0	0	0.0	0
MATEMATICA	0.0	0	0.0	0	0.0	0	0.0	0	0.0	0
ARTES	0.0	0	0.0	0	0.0	0	0.0	0	0.0	0
Observações										

Fonte: Criação dos autores deste estudo

Na Figura 16, é apresentado o código fonte referente ao comando *“salvar”*, responsável por inserir os dados dos alunos no banco de dados. Nesse trecho de código, é criado um objeto do tipo *“AlunoRN”* (linha 3), que é a classe que possui as regras de negócio da entidade *“Aluno”*. Na linha 4, o objeto *“alunoRN”* passa o *“id”* do sexo selecionado na tela de cadastro através do método *“buscarSexoPorId(int id)”*. Por sua vez, o método acessa a classe *“AlunoDAO”*, que é a classe responsável por todas as ações que envolvem o acesso ao banco de dados, e chama o método *“buscarSexoPorId(int id)”* da classe *“AlunoDAO”*, que irá retornar o objeto *“Sexo”* desejado.

As próximas linhas têm um comportamento bastante similar às anteriores. Na linha 7, o objeto *“enderecoRN”* é usado para acessar a classe *“EnderecoDAO”* e buscar o objeto *“Uf”* de acordo com o *“id”* da *“Uf”* selecionada na tela de cadastro e, então, na linha 9, o objeto *“endereco”* recebe a *“Uf”* encontrada para ser salva posteriormente. Logo mais abaixo, nas linhas 11 a 14, o objeto *“aluno”* recebe alguns outros objetos para, então, ser persistido no banco de dados a partir do método *“salvar(Aluno aluno)”* da classe *“AlunoRN”*, presente na linha 16.

Figura 16 – Trecho do Código de Cadastro de Alunos

```

1      public String salvar() {
2
3          AlunoRN alunoRN = new AlunoRN();
4          this.sexo = alunoRN.buscarSexoPorId(this.idSexo);
5
6          EnderecoRN enderecoRN = new EnderecoRN();
7          this.uf = enderecoRN.buscarUfPorId(idUf);
8
9          this.endereco.setUf(this.uf);
10
11         this.aluno.setEndereco(this.endereco);
12         this.aluno.addTelefone(this.telefone1);
13         this.aluno.addTelefone(this.telefone2);
14         this.aluno.setSexo(this.sexo);
15
16         alunoRN.salvar(this.aluno);
17
18         this.aluno = new Aluno();
19
20         this.alunoList = null;
21
22
23         return "gerenciarAlunos.jsf?faces-redirect=true";
24     }
25
26

```

Fonte: Criação dos autores deste estudo

Na Figura 17, encontra-se o código para a listagem dos alunos cadastrados no sistema. Mais uma vez a classe “AlunoRN”, presente na linha 5, é utilizada para chamar a classe de acesso aos dados, porém, dessa vez, será feita uma busca por todos os alunos cadastrados no sistema e o método responsável por tal ação se encontra na linha 6, denominado “buscarTodos()”. Um detalhe importante é que agora será usada uma lista de objetos para armazenar todos os alunos e antes mesmo disso acontecer há uma verificação na linha 3, apresentando um comando “if” para conferir se a lista está nula ou vazia (método “isEmpty()”). O método “getBuscarTodos()” se encerra na linha 10, quando é retornada a lista de alunos através do comando “return this.alunoList;”.

Figura 17 – Trecho do Código de Listagem dos Alunos

```

1      public List<Aluno> getBuscarTodos() {
2
3          if(this.alunoList == null || this.alunoList.isEmpty()) {
4
5              AlunoRN alunoRN = new AlunoRN();
6              this.alunoList = alunoRN.buscarTodos();
7
8          }
9
10         return this.alunoList;
11     }

```

Fonte: Criação dos autores deste estudo

A última fase do desenvolvimento começará a partir do momento em que o sistema for implantado na escola e os usuários tiverem contato com todas as funcionalidades do mesmo. Assim, será possível realizar um acompanhamento junto ao cliente com o propósito de obter um *feedback* sobre o comportamento do sistema no ambiente escolar. Dessa forma, será realizada a validação dos requisitos levantados nas primeiras fases do projeto, possibilitando melhorar a usabilidade do sistema através de correções evolutivas.

5 CONCLUSÃO

O artigo teve como propósito descrever a criação de um sistema informatizado para gerenciar os processos internos da Escola Estadual Marcolino de Barros, proporcionando mais facilidade e agilidade para manipular as informações relativas a alunos e professores.

O sistema foi dividido em três módulos referentes aos perfis dos usuários que utilizarão o sistema, sendo um módulo para a secretaria, um módulo para os professores e um módulo para os alunos, e cada módulo foi dividido em sub-módulos específicos para cada perfil. Assim, a área da secretaria conta com funções que auxiliarão a secretária a cadastrar alunos, professores, novas turmas e fazer a emissão dos boletins acadêmicos. A área dos professores conta com todos os recursos necessários para gerenciar trabalhos, provas, notas e fazer a frequência diária dos alunos. A área dos alunos apresenta os sub-módulos necessários para que eles possam ter melhor acompanhamento de suas notas e faltas e maior simplicidade na entrega dos trabalhos.

Sendo assim, o *software* pretende auxiliar no controle das informações nos diferentes ambientes dentro da escola, tendo como objetivo suprir todas as necessidades encontradas na fase de concepção do projeto e atingir o objetivo principal do mesmo.

Como projeto futuro, pretende-se melhorar a organização do menu principal, criando menus separados por operações, como cadastrar e listar. Também serão desenvolvidos outros tipos de relatórios e um aplicativo *mobile* que permitirá aos usuários acessarem o sistema por meio dos seus *smartphones*. Com isso, espera-se tornar o Sistema de Controle Acadêmico mais robusto e flexível.

REFERÊNCIAS

LARMAN, Craig; BRAGA, Rosana Vaccare (Trad.). *Utilizando UML e Padrões*. 3. ed. Porto Alegre: Bookman, 2007.

LUCKOW, Décio Heinzemann; MELO, Alexandre Altair de. *Programação Java para a Web*. São Paulo: Novatec, 2010.

SOMMERVILLE, Ian. *Engenharia de software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011.