

Desenvolvimento de Sistema Gerenciador de Aprendizagem para Escola de Informática

Rafael Marcondes Teixeira¹, Regiane Orlovski², Sergio Ribeiro³

^{1 2 3}Tecnologia em Análise e Desenvolvimento de Sistemas – Faculdade Guairacá

CEP 85010-000 – Guarapuava – PR – Brasil

³Universidade Estadual de Ponta Grossa, Departamento de Informática, Mestrado em Computação Aplicada em Agricultura

CEP 84030-900 – Ponta Grossa – PR – Brasil

rafael@vital.net.br¹, regianeorlovski@hotmail.com², professor@sergioribeiro.com.br³

Abstract. *This article aims to show the development of software called Learning Management System for School of Informatics. Deployed in laboratories shows that control of information obtained and reviewed with the system can replace paper thus avoiding loss of files. This control oriented education in the laboratory was developed and brought functions that help the teacher and student in consultation exercises and sending notes among other features. Based upon standard on MVC projects with the PHP object-oriented language, the system is based on the life cycle prototyping methods for evolutionary development of their construction are included for clear and easy software. The implementation of the system resulted in efficient aid in the classroom to teachers of the institution, giving the user control works, exercises and notes of his students room, making it no longer necessary to use paper for these functions in the computer labs. Implementing this software has considerably reduced the time required for execution of tasks performed by the teacher and his students.*

Resumo. *Este artigo tem como objetivo mostrar o desenvolvimento do software intitulado Sistema Gerenciador de Aprendizagem para Escola de Informática. Implantado em laboratórios mostra que o controle de informações obtidas e repassadas com o sistema podem substituir o papel assim evitando perdas de arquivos. Esse controle voltado a educação em laboratório foi elaborado e trouxe funções que ajudam o professor e aluno no envio de exercícios e consulta de notas entre outras características. Embasado em padrão de projetos MVC com a linguagem orientada a objetos PHP, o sistema tem como base o ciclo de vida prototipação evolutiva com métodos para desenvolvimento de software claros e fáceis de serem compreendidos para sua construção. A implementação do sistema resultou no auxílio eficiente em sala de aula aos professores da instituição, dando facilidade no controle de trabalhos, exercícios e notas de seus alunos, tornando não mais necessário o uso de papel para essas funções nos laboratórios de informática. A implementação desse software diminuiu consideravelmente o tempo necessário para execução de tarefas realizadas pelo professor e seus alunos.*

Introdução

A procura por capacitação na área da Tecnologia da Informação se torna cada vez maior e escolas de informática são requisitadas para dar auxílio e ensino. Aulas práticas e teóricas são fornecidas, e sistemas para controle das mesmas são necessários, pois estes buscam tornar as tarefas mais ágeis para que os alunos estejam satisfeitos no momento de realizar essas aulas. Nesse aspecto desenvolveu-se um *software* intitulado Sistema Gerenciador de Aprendizagem para Escola em Informática (SGAEI) o qual surgiu da necessidade de controle das notas que os alunos da instituição possuem depois da entrega de seus exercícios, trabalhos e provas, pois os professores tinham dificuldades em administrar essas informações. A partir desse pressuposto foi formulada a ideia de construir um *software* que desse auxílio nessa área.

Antes da utilização desse sistema a realidade era outra, o professor usava o tradicional padrão de chamadas e exercícios em papel, mas o aumento expressivo da informática, redução de preços dos computadores e as novas tecnologias de informação, substituíram essa realidade. Se tratando de uma escola de informática que disponibiliza para cada aluno um computador, a falta de um Sistema Gerenciador de Aprendizagem é uma desvantagem na instituição.

As ferramentas utilizadas para construção do sistema foram, o IDE *NetBeans* utilizado como ambiente de desenvolvimento para programação *web* com HTML, CSS e PHP. O *MySQL Workbench* foi usado para construção e modelagem do Banco de Dados, que tem como função guardar informações do sistema para que possam ser acessadas posteriormente. Por último o *Astah*Community* para elaboração dos diagramas de caso de Uso, Classe e Sequência nos padrões UML. Este trabalho também apresenta a elaboração do ciclo de desenvolvimento e os métodos utilizados para a construção do sistema. O ciclo de vida abordado foi o de Prototipação Evolutiva, seguindo suas etapas de desenvolvimento que são concepção do Software, Projeto e implementação do protótipo inicial, Refinamento do Protótipo até sua aceitação, Conclusão e Entrega do protótipo.

O objetivo deste trabalho é apresentar o desenvolvimento de um sistema para gerenciamento de ensino em cursos de informática, o qual auxilia no controle das aulas lecionadas pelos professores e tem a finalidade de ajudar o aluno durante a realização da aula.

Fundamentação Teórica

A gestão de aprendizagem visa disponibilizar recursos e funcionalidades, as quais devem armazenar, gerenciar e distribuir conteúdo para o aprendiz. Esta é uma forma interativa de treinamento e avaliação a qual se faz com o uso de computadores dentro de salas de aulas, levando a percepção da tecnologia nos dias de hoje em relação a educação.

Segundo Ribeiro Neto (1999), a competição com redução de custo e ganho de produção faz com que organizações procurem inovar na parte tecnológica, para permitir vantagens competitivas. Isso faz as pessoas que almejam por vagas em certas organizações, procurarem se especializar para conseguirem ficar ou entrar no mercado de trabalho, e não perder oportunidades e nem vagas em novas empresas ou na mesma

que está. A solução encontrada para isso é buscar local qualificado que forneça cursos em várias áreas da informática.

As escolas de informática investem nesse setor, buscando sistemas de gestão de aprendizagem *Learning Management System* (LMS) ou *Course Management System* (CMS). Conforme Renato (2007), LMS é um aplicativo que visa ajudar educadores a desenvolver cursos *on-line*, ou suporte *on-line* para cursos presenciais, com uma melhor qualidade e várias ferramentas disponíveis devido aos recursos do sistema.

O desenvolvimento do Sistema Gerenciador de Aprendizagem Para Escola de Informática surgiu da necessidade do professor de informática ter ao seu dispor um software que o auxiliasse. Lecionar aulas sem nenhum sistema para ajuda, faz tempo ser perdido e controle de exercícios, trabalhos e notas serem mal executados. Com a adoção de um sistema gerenciador de aprendizagem o controle de arquivos fica mais agradável e com maior qualidade.

Para Coulouris et al. (2007) as áreas da educação são beneficiadas sempre que as informações são acessadas com recursos de qualidade acessíveis e com menos restrições. Nessa área um ambiente virtual para educação vai fornecer um local favorável ao ensino. Para isso é necessário simplicidade, que irá auxiliar no processo de controle dessa aprendizagem, ao contrário o professor terá que recorrer a outras técnicas para auxiliá-lo, como no caso de anotações e planilhas, tornando complexo e desmotivante a aula para os alunos.

A implementação de um sistema gerenciador de aprendizagem tornou a instituição de ensino mais eficiente para seus alunos e professores, pois isso fez com que atividades em papel fossem substituídas por arquivos *on-line*, facilitando o ensino e deixando o mesmo ágil e com mais satisfação para ambos os lados. Conforme Macdonald (2006) um sistema de gestão de aprendizagem é uma ferramenta imprescindível no processo de ensino e orientação de alunos, onde acompanhamento e conhecimento é tomado e levado em consideração. Segundo Valente (2002), ambientes usando a informática para auxiliar o processo da aprendizagem, mudam o foco de educação centrada apenas no professor, para novos conhecimentos construídos na informática.

Para ter uma visão mais ampla da implementação do sistema SGAEI é interessante contextualizar Tecnologia da Informação (TI), segundo Valente (2002) os computadores estão juntos no processo ensino-aprendizagem desde o instante que foram inventados, já foram utilizados como máquina de ensinar, e agora são tidos com um auxiliar de aprendizagem e construção dos conhecimentos que o aluno obtêm. Oliveira (1996), descreve a TI como um conjunto que inclui computadores, redes de comunicação, subprodutos dessas tecnologias e a comunicação por meio da computação. Contudo a implementação de TI em uma empresa pode ser de grande ajuda ou pode resultar em prejuízo. Para Beraldi e Escrivão Filho (2000) o baixo preço a cada dia maior dos computadores e a integração de software incentiva muitos empresários a investirem pesado na área de TI para melhorar o desempenho da empresa e competir com as demais.

Segundo Beraldi e Escrivão Filho (2000), um investimento desse tipo pode ser desastroso se os funcionários não forem capacitados para tal, sendo aconselhável avaliações antes de futuras implementações e posteriormente treinamentos. Isso leva a percepção de que a TI afeta diretamente pessoas que devem ou querem se adaptar ao

mercado de trabalho. Para Ribeiro Neto (1999), a utilização dessas tecnologias é vital para o dia a dia de uma organização o que provoca mudanças profundas na empresa, mas que acaba por contribuir para que a mesma obtenha sucesso.

Quando a TI é implementada em uma empresa, isso é feito visando benefícios que serão trazidos para a organização, buscando uma melhor qualidade ao trabalho para que no fim possa trazer lucros ao negócio. Porém existem vários fatores que levam ao uso de Tecnologia em uma organização, segundo Gonçalves (1994) as razões para implementação de TI nas empresas variam bastante, mas que na maioria dos casos são em razão aquisitiva, existindo forte influência na competitividade e desempenho com a concorrência. Fatores como clientela também são levados em conta, porém objetivos internos tem um destaque muito importante, como a redução de custos, ganho de recursos e o próprio controle da empresa.

Muitas vantagens são proporcionadas para a empresa com a implementação da TI, porém é necessário antes disso, analisar e saber quais os benefícios para a organização. Caso possua algum benefício, os encarregados pela corporação deverão aprofundar-se no assunto para obter maior aproveitamento do mesmo e analisar qual a melhor arquitetura da informação para ser usada por essa empresa. Segundo Turban, McLean e Wetherbe (2002), um plano de alto nível com base em necessidades envolvendo informação de uma empresa, sendo que a mesma assegura que a TI na instituição possa resolver deficiências estratégicas na organização, são definidas como arquitetura da informação.

Para Beraldi e Escrivão Filho (2000), depois de levantada a necessidade de usar sistemas de informação, a organização deve avaliar o que será necessário para poder escolher aquilo que seja mais eficaz para posterior implementação. Isso envolve pesquisas em empresas similares que implementarão sistemas informatizados, é importante também o contrato ou preparo de pessoal qualificado para que os benefícios sejam alcançados. Também deve haver uma análise do atual estado da empresa, identificar os processos informatizados e possíveis melhorias e também definir metas para implementação.

Tendo em vista a importância da TI na empresa em questão, e levantado os requisitos necessários para desenvolvimento de um sistema que auxilie professores e alunos, a escolha de um ciclo de vida a ser aplicado se faz útil. Para a criação do SGAEI o ciclo de vida abordado foi o de Prototipação, segundo Rezende (2005) é uma metodologia de construção de software experimental com uma forma mais rápida e baixos custos para demonstrações, onde as avaliações e modificações são feitas de acordo com os pedidos dos clientes. Seguindo mais afundo a prototipagem, vários modelos da mesma podem ser estudados para que posteriormente um deles seja aplicado no desenvolvimento do software que empregará a prototipação. Dentre os modelos de prototipação estão o incremental Iterativo, Prototipação Rápida Descartável, Prototipação Evolutiva e Espiral.

O modelo de prototipação escolhido para a construção do SGAEI foi prototipação evolutiva. Para Cordeiro (2003), este tipo basicamente inicia com a concepção do sistema empregando diversos ciclos de re-projeto, re-implementação, re-avaliação. Este meio de prototipação pode ser aplicado em contextos onde os requisitos são instáveis ou desconhecidos, e ainda quando existe necessidade de testes de algoritmos antes de sua implementação. As vantagens proporcionadas por este modelo

são uma menor taxa de defeitos devido a uma melhor definição das especificações do sistema, buscando promover maior participação e visibilidade dos clientes no processo de desenvolvimento. Porém isso pode trazer algumas desvantagens, como é o caso da pouca manutenibilidade devido á rapidez que o sistema é gerado, também constantes mudanças em sua estrutura farão parte, já que o cliente acompanha com mais ênfase seu desenvolvimento. Definido o ciclo de vida para a construção do sistema, é possível apresentá-lo ao usuário de uma forma mais contundente e concreta, podendo ser feito com apresentações ou até mesmo de uma forma já abstrata em um computador. Nesse momento a psicologia das cores se faz útil, já podendo ser empregada no protótipo, para Farina (2000) a cor azul apresenta intelectualidade, meditação e confiança, cores estimulam e predispõe à simpatia, oferecendo uma sensação de paz para produtos e serviços que precisam demonstrar sua segurança e estabilidade.

Concluída a escolha do ciclo de vida é necessário definir qual a forma de programação, as linguagens usadas e os softwares de desenvolvimento. Na área da programação existem duas categorias, a estruturada e a orientada a objetos. Dall'Oglio (2009) descreve programação estruturada como um modo de construção de sistemas, formado por procedimentos e variáveis que não são agrupadas no contexto do mundo real. Orientação a objetos segundo Lobo (2009) é um padrão de desenvolvimento de *software* baseado em classes, onde essas são formas de criação de objetos, os quais serão utilizados no sistema. Nessa forma de programação os objetos trocam mensagens e realizam ações na memória do computador, enquanto as classes definem as características e ações dos objetos. Quando um *software* é projetado com orientação a objetos, modelos são criados para representar o sistema de forma objetiva, isto faz os detalhes serem visualizados antes da codificação por meio de diagramas, onde é mostrado como o *software* será implementado.

Esses diagramas são criados por meio da *Unified Modeling Language* – Linguagem de Modelagem Unificada (UML). Segundo Dennis e Wixom (2011), a UML consiste em oferecer terminologias comuns para os conceitos baseados em objetos e métodos de diagramação, para que sejam utilizados em modelagens de qualquer projeto de desenvolvimento de *software*, desde a fase de análise até a implementação do sistema.

Para Silva e Videira (2001) no principio de propor várias notações de produção para diferentes tipos de modelos razoavelmente independentes com algum nível de integração a UML é como na engenharia civil, onde um prédio é representado por modelos complementares. Silva e Videira (2001) ainda descrevem UML como linguagem diagramática, utilizável como especificação, visualização e documentação de sistemas. Dentre os padrões da UML os que serão utilizados no desenvolvimento do sistema são diagramas de caso de Uso, diagramas de Classe e diagramas de Sequência.

Diagramas de Caso de Uso são usados para descrever funções completas de um sistema, aplicações ou produtos, assim esses modelos devem ser compreensíveis por qualquer pessoa que as visualize. Para Dennis e Wixom (2011) esses diagramas devem comunicar em alto nível as operações que o sistema deve realizar, pois os diagramas seguintes são baseados em cima deles. Silva e Videira (2001) descrevem Casos de Uso como a relação entre atores e casos de utilização de um dado sistema.

O diagrama de Classe descreve segundo Silva e Videira (2001), uma visão estática, o escopo da aplicação por meio de classes e seus relacionamentos. Conforme

Dennis e Wixom (2011), o diagrama de Classes auxilia na visualização do sistema em desenvolvimento, destacando as classes e os relacionamentos no programa.

O diagrama de Sequência apresenta iterações entre objetos em um período de tempo, e também fornece ajuda na identificação de métodos para classes do sistema. Segundo Silva e Videira (2001) os diagramas de Sequência representam ilustrações entre os objetos e um determinado período de tempo, os objetos são representados por suas linhas de vida e interagem trocando mensagens em um determinado ciclo de tempo.

Após a modelagem dos diagramas, a linguagem de programação para formação do código-fonte é escolhida, a qual ajudará o programador a construir as funções e tarefas do *software*. A categoria de programação definida para o desenvolvimento do sistema é o padrão de orientação a objetos, e a linguagem de programação o *Personal Home Pages* (PHP). Segundo Dall'Oglio (2009) PHP possui uma grande extensibilidade, pois com ele existe a possibilidade de conexão com as várias extensões de bancos de dados, para isso o mesmo usa o *PHP Data Objects* (PDO) a qual possui compatibilidade com os diversos *drivers*, PHP também possui sintaxe consistente e orientação a objetos, a partir da versão 5.3 do PHP é usado *namespace* o qual segundo Denoncourt (2011) resolve o problema de colisão de nomes das classes com este recurso.

Também é necessário a implementação da linguagem de marcação *Hyper Text Mark-up Language* (HTML) que funciona como o esqueleto de um *website*. Para Freeman e Freeman (2009) a linguagem de marcação de hipertexto oferece maneiras de criar textos com *Tags* para que os navegadores de internet possam saber qual a estrutura de uma página, onde as etiquetas ou rótulos são usados para informar ao navegador como deve ser apresentado o *website*.

Como o HTML é responsável por organizar a estrutura da página, para dar mais elegância a mesma, o *Cascading Style Sheet* (CSS) é empregado no sistema. Segundo Freeman e Freeman (2008) com a utilização de CSS podem ser criados estilos para personalização das páginas e também possuir total controle sobre os mesmos. Para deixar o sistema mais interativo com o usuário, como no caso de validações de formulários, é empregado o *javascript*, que segundo Morrison (2008) transforma uma página *web* deixando-a interativa, fazendo com que possa saber as necessidades do usuário e posteriormente processar as mesmas e atender aos desejos de quem está navegando em determinada página. Para facilitar o desenvolvimento do sistema e deixá-lo intuitivo é empregado o *framework* de *front end Bootstrap*, segundo Magno (2012) é uma ferramenta poderosa com melhorias em HTML, CSS e *javascript*.

As linguagens de texto, programação e consequentemente a aplicação em um todo tem uma melhor organização de sua estrutura com o uso do padrão de projetos *Model View Controller* (MVC). Dall'Oglio (2009) o descreve como o mais conhecido dos *design pattern's*, e uma aplicação dividida em três camadas, sendo cada letra que compõem o nome deste *pattern* a representação de seus aspectos.

Ainda segundo Dall'Oglio (2009), *model* de modelo, é um objeto representando informações do domínio de negócios da aplicação. *View* de Visualização traz as definições da interface com o usuário e a organização da tela, é responsável somente pela visualização, não exercendo nenhum tipo de controle de fluxo de execução. E *controller* de controle traz a manipulação das entradas dos usuários, sua interpretação e

execução das tarefas correspondentes a eles, essa camada é formada pelos objetos que recebem informações das *views* para que possam atualizar o modelo de dados de acordo com o as ações dos usuários.

Para Dall'Oglio (2009) separando a aplicação nesses três aspectos várias vantagens são dadas ao desenvolvedor, um exemplo é a separação que acontece entre *view* e *model*, onde é possível utilizar um mesmo objeto do modelo em várias visualizações distintas.

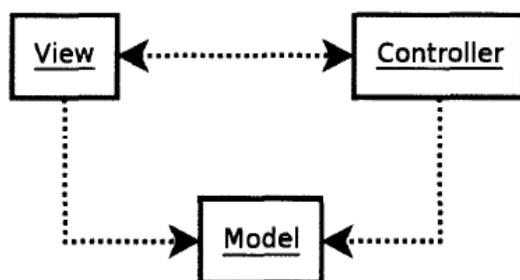


Figura 1. Modelo Mvc.

Fonte: Dall'Oglio(2009)

Para a criação do código fonte, organização do padrão de projeto MVC e desenvolvimento de *interface*, um *software* também conhecido como *Integrated Development Enviroment (IDE)* ou Ambiente Integrado de Desenvolvimento deve ser usado na criação do sistema, pois fornece ferramentas que suportam métodos de desenvolvimento de *softwares*. Para Sebesta (2002), este ambiente de programação é um conjunto de instrumentos usados no desenvolvimento de um *software*. O mesmo pode ser constituído de um editor de texto, um sistema de arquivos e um compilador, contendo várias ferramentas, onde essas são acessadas pela *interface* do IDE. Para a construção do SGAEI foi determinado o uso do IDE Netbeans 7.3, este *software* além de ser gratuito e de código aberto, pode ser executado em diversas plataformas, contendo suporte para sistemas em *Desktop* e *Web* para desenvolvimento do código fonte e organização do padrão de projetos MVC.

Outro fator importante na estrutura de um sistema é o Banco de Dados, responsável por armazenar informações e manter a integridade das mesmas. Segundo Beighley (2010), os dados inseridos no sistema ficam armazenados em tabelas, e o banco de dados é um contêiner que guarda todas as tabelas e outras estruturas *Structured Query Language* – Linguagem de Consulta Estruturada (SQL) relacionadas a elas. Para o acesso e manipulação de informações no Banco de Dados o aplicativo usado é nomeado *Database Management System (DBMS)* ou (SGBD). Para Elmasri e Navathe (2011), um Sistema Gerenciador de Banco de Dados é um grupo de programas que proporciona aos usuários a criação e a sustentação de um Banco de Dados. O SGBD simplifica a definição, construção, manipulação e compartilhamento do Banco de Dados com diversas aplicações.

Para auxílio com o Banco de Dados durante o desenvolvimento do SGAEI, o MySQL 5.5.16 foi definido como SGBD, segundo Welling e Thomson (2005) o MySQL é um Sistema Gerenciador de Banco de Dados relacional, poderoso e muito rápido. Por

ser livre e possibilitar a visualização de dados decorrentes de operações durante a fase de construção do sistema. Isso é possível graças ao *phpMyAdmin* 3.4.5 por ser um *software* com ambiente gráfico e de fácil entendimento compatível com vários navegadores, fazendo com que seja simples a exclusão, alteração e visualização de tabelas. O acesso a parte gráfica do *phpMyAdmin* é feito com o auxílio de um servidor *web*, mesmo que seja localmente. O Apache 2.2.21 foi utilizado para isso, pois é *OpenSource* e também compatível com os demais *softwares* utilizados no desenvolvimento do sistema. As operações pelo *phpMyAdmin* são executadas graças ao Xampp 2.5 que faz com que seja possível rodar o MySQL.

Etapas do Desenvolvimento do Trabalho

O desenvolvimento do SGAEI surgiu a partir da ideia de construir um *software* que desse auxílio na área de educação, com isso requisitos para sua construção foram levantados pela instituição e entregues ao desenvolvedor, como controle de usuários, cadastro de cursos, cadastro de matérias e exercícios, cadastro de aluno e notas, assim surgindo o nome Sistema Gerenciador de Aprendizagem para Escola de Informática ou SGAEI. Com a decisão de realmente construir este *software* o primeiro passo foi levantar seus requisitos e definir a escolha de um ciclo de vida para seu desenvolvimento. Dentre os requisitos levantados o sistema *web* visa disponibilidade em qualquer lugar, remoção de arquivos em papel, esses virando conteúdos digitais, controle de atividades e comunicação professor com aluno.

A partir de decisões tomadas nas reuniões com a instituição, a prototipação evolutiva foi o ciclo de vida escolhido para o desenvolvimento do sistema, abordada devido ao *software* proposto exigir melhorias de acordo com sua fase de testes e posterior implementação, pois o modelo proporciona o re-projeto, re-implementação, re-avaliação, até a aceitação final do *software*. De acordo com o ciclo de Prototipação Evolutiva, a primeira etapa é a concepção do *software*, nesta etapa foi definido com o cliente que o propósito do sistema seria atender as necessidades dos professores e alunos. As características identificadas junto a funcionalidades para o *software* foram documentadas, e também as necessidades que o mesmo supre. Os requisitos nessa fase foram levantados pelo analista com o método observação, de acordo com as necessidades observadas nas aulas lecionadas pelos professores da instituição de informática nos vários cursos que a mesma fornece, tanto em aulas com um único aluno, como em aulas de turmas.

Para melhor eficiência do ciclo de vida são criados diagramas nos padrões estabelecidos pela UML, com base na facilidade que fornecem para modelagens de casos de Uso, diagramação de Classes e criação de diagramas de Sequência. Os diagramas referentes encontram-se nos Apêndices A, B, C, D, E, F, G, H e I. Para realizar estas modelagens foi utilizado o *software astah*Community* 6.7.0. O objetivo dos diagramas de Caso de Uso é compreensão de comportamentos do sistema por qualquer pessoa, onde suas formas se parecem com as atividades a serem realizadas pelo sistema. Nesses diagramas são identificados os tipos de usuário que vão interagir com o sistema, seus papéis e funções.

Os Casos de Uso representam tarefas, serviços ou funções, e quase sempre relacionadas a atores que são representações de humanos ou máquinas, aqui mostrando os atores Alunos, Professores e Administradores do sistema. Diagramas de Classe são

referencias aos atributos, métodos e relações entre as classes que o software deve possuir, servindo de base para diagramas de sequência, as principais classes do sistema são *login*, *aluno*, *curso*, *matéria*, *adm*, *professor*, *endereço*, *pessoa*, *email*, *fone*, *exercício* e *nota*. O Diagrama de Sequência representa a troca de mensagens entre os objetos do sistema, ou uma troca de processos do *software* em questão, mostrando como eles contribuem para o cenário de cada caso de uso que o sistema possui.

Os professores foram entrevistados para definir as tarefas que o *software* possui, assim recebendo as telas para cadastro de usuário, curso, aula, nota e aluno. Antes disso a falta de controle com as notas, trabalhos, e provas era considerável, o sistema proporcionou ao aluno visualizar as aulas dos respectivos cursos como as notas obtidas nos mesmos, também foi visto que o tempo perdido com a distribuição de exercícios em papel em uma aula de 2 horas era grande e cansativa para alunos e professor, isso também sendo contornado com a adoção do *software* SGAEI.

A segunda etapa para a Construção do Sistema de acordo com o Ciclo de vida adotado é “Projeto e implementação do protótipo inicial”. Nesta etapa os aspectos para construção são definidos e mostrados ao cliente, como as regras de negócio, a ordem e apresentação de dados preenchidos pelos usuários. Essas informações se tornam claras com a construção inicial do sistema onde dificuldades observadas no estágio anterior do ciclo de vida mostram que o tempo necessário para realizar determinada tarefa deve ser menor com a implementação do sistema proposto. Alguns aspectos verificados com professores da instituição, como o uso de papel para determinados exercícios, perda dos mesmos e de notas de alunos foram levantados, com isso a primeira interface foi desenvolvida e apresentada na instituição, contendo uma simples tela com poucas cores, mas com funções como cadastro de alunos e cadastro de exercícios, onde o papel que era usado para isso foi retirado de circulação com base nos testes feitos com os alunos e professores em aula.

O protótipo foi hospedado no próprio servidor da instituição não tendo valores de domínio para sua instalação, a recepção inicial foi aceita com afinidade pelos usuários, onde os mesmos constataram que alguns objetos da tela ficaram fora do lugar, e campos que deveriam ser preenchidos com uma grande quantidade de texto não permitiam tal ato. Dificuldade na escrita também foi observada, pois as caixas de texto não possuíam campos para editá-los como é o caso de processadores de texto, isso tudo foi consertado nessa primeira etapa, onde também foram trocadas as cores do sistema para uma melhor visualização do mesmo.

Nesta etapa foi definido a arquitetura que o sistema deve possuir, o mesmo foi desenvolvido com estrutura *web*, pois assim poderia ser acessado de qualquer lugar com sinal de internet, sua linguagem de programação foi o PHP, e sua categoria à orientada a objetos. O IDE *Netbeans* 7.3 para escrita do código e o SGBD *MySQL* 5.5.16 para controle do Banco de Dados também foram definidos nesta etapa do ciclo de vida, assim os *layouts* dos primeiros protótipos foram construídos com HTML, CSS e PHP e ligados ao Banco de Dados. A modelagem lógica do Banco de Dados do sistema foi feita a princípio das demais etapas de desenvolvimento, pois ela tem influência total na construção de um *software*, no caso do SGAEI as tabelas construídas para armazenamento dos dados podem ser vistas no apêndice J, são elas *adm*, *professor* e *aluno* onde as mesmas tem nomes representando os respectivos atores, ligadas as tabelas *pessoa*, *endereço*, *fone* e *e-mail* para obtenção dessas informações.

As tabelas *curso*, *matéria*, *exercício* e *nota* constituem o conjunto para informações referentes as áreas dos cursos, sobrando assim a tabela *login* com os usuários e senhas do sistema e *tipousuario* onde são encontrados os níveis 1 para administradores, 2 para professores e 3 para alunos do sistema. A base para organização do *software* vem da modelagem do Banco de Dados, pois possibilita a leitura de informações, restrições e relações do sistema. Para modelagem do Banco de Dados o *software* empregado foi o MySQL *Workbench* 5.2.47, este permite de forma prática criar a estrutura do Banco de Dados, como as tabelas, chaves primárias, estrangeiras, atributos, e cardinalidade. Após criadas as tabelas, chaves, relacionamentos do modelo lógico, é possível realizar a conexão com o Banco de Dados MySQL, transformando estas tabelas em linhas de código para que o SGBD trabalhe com a linguagem padrão de Banco de Dados o SQL.

A terceira etapa do ciclo de Prototipação Evolutiva é o Refinamento do Protótipo até sua aceitação, aqui o sistema começa a ser implementado, de acordo com os resultados obtidos nas etapas anteriores. Nesse caso um *layout* azul e branco foi desenvolvido para chamar a atenção dos alunos e prender sua atenção, pois a faixa etária dos mesmos varia bastante na instituição, fornecendo cursos para crianças, até idosos. A programação orientada a objetos foi escolhida para desenvolver o sistema, pois com ela existe um controle melhor de informações, usando funções, métodos e atributos. Como a prototipação evolutiva proporciona re-implementação, e no caso de refinamento do sistema, novas funções, ferramentas e futuras práticas podem se tornar realidade graças à orientação a objetos, usada em conjunto com o MVC.

A linguagem de programação escolhida foi o PHP, devido ao mesmo proporcionar uma base *on-line* junto ao HTML, assim fornecendo acesso ao sistema em qualquer lugar com sinal a internet. Finalizada a etapa de seleção de todas as ferramentas, é iniciada a construção do código fonte, baseada nos dados e resultados obtidos em fases anteriores. Com o HTML é construído a estrutura básica de uma página *web*, onde se faz necessário o uso de *tags*, as quais colocadas no lugar correto formam o *layout* da página, como exemplo disso existe a estrutura básica de uma página *web* formada pelas *tags* que indica o início e fim do documento e também onde todas as *tags* devem ficar, *head* ou cabeçalho é o lugar onde todas as *tags* cujas informações que não são mostradas na tela devem ficar, e *body* ou corpo onde tudo que é exibido pelo navegador deve estar.

O CSS faz a página ficar com melhor aspecto em seu *layout*, usado em conjunto com as *tags divs* do HTML no atributo *id* para identificação, é possível construir blocos que formam o *layout* da página. Com o *layout* definido é necessário deixa-lo interativo ao usuário, para isso é usado o *javascript*. Exemplo dessa interatividade esta no formulário HTML, eles são formados pelas *tags forms* e seus vários *input types*, os quais geram entradas de textos, caixas de textos, botões entre outros. Nas linhas de códigos dessas entradas é possível chamar por meio do manipulador *onclick* eventos e funções, com clique de mouse dos usuários geralmente sobre o elemento ao qual é aplicado, ressaltando que o *Bootstrap* auxiliou nessas funções e contribuiu com o visual do sistema.

Com PHP a conexão do sistema com o Banco de Dados é possível, especificamente graças ao PDO presente na linguagem, o qual se adapta aos diversos *drivers*, para isso a variável deve receber a linha de código *mysql:host=servidor;*

dbname=database, e depois ser chamada por *new PDO("...")*. Além do PDO no PHP 5.3 existe *namespace*, o qual deixa as classes serem nomeadas da mesma forma facilitando na orientação a objetos, o sistema SGAEI foi mapeado de acordo com a estrutura de pastas para facilitar o *autoloading* das classes graças ao *namespace*.

Telas para administrador, aluno e professor estão presentes no sistema, cada uma delas possui determinadas funções para cada caso. Durante essa fase o sistema é testado pelo cliente e observações são anotadas, como exemplo a forma que o usuário realiza operações e se as necessidades são atendidas pelo sistema, nesse caso o administrador cadastra cursos e usuários, professores cadastram alunos, exercícios, trabalhos e notas, e os alunos consultam cursos, trabalhos, exercícios e notas. Logo após essa parte é realizado o refinamento de requisitos para correção de falhas, funcionalidades que devem ser implementadas e melhorias na interface.

E a última etapa do ciclo a ser abordada é a conclusão e entrega do protótipo, onde é confeccionado uma versão final do sistema, as várias funções já devem estar presentes no protótipo, tanto para administrador quanto para professores e alunos. Tendo todos os requisitos levantados e aprovação da instituição nas fases de avaliação, o sistema pode ser finalizado contendo as seguintes telas com suas funções, cadastro de Usuários, cadastro de Cursos para o Administrador, cadastro de Alunos, trabalhos e exercícios para Professor, e consulta de Trabalhos, exercícios e notas para Alunos, ressaltando que as telas de administrador e professores contem campos de pesquisa e edição de dados.

Estas telas que são mostradas ao usuário, estão na pasta *view*, pois a programação do sistema foi no padrão de projetos MVC. Na pasta *model* estão os códigos referentes as classes que o software possui, como exemplo *adm*, *aluno*, *curso*, *professor* entre outras. Na pasta *controller* é encontrado códigos que manipulam os dados do sistema com exclusão e inserção, ressaltando que o mesmo foi programado na orientação a objetos.

Vale lembrar que a prototipação evolutiva é um método que torna protótipos modificáveis até a obtenção de um *software* final, fazendo com que essa primeira versão não seja exatamente a última, pois o cliente pode achar novas funcionalidades e posteriormente fazer um pedido para inserção em seu sistema.

Resultados

O desenvolvimento do sistema proporcionou o entendimento sobre re-projeto, re-implementação, re-avaliação, as quais são etapas do ciclo de vida prototipação evolutiva, mostrando como cada uma delas é importante para a construção de um bom *software*.

Com os testes e entrevistas realizados antes da implementação do sistema, foi possível notar o tempo perdido com a distribuição e organização de trabalhos ou exercícios em sala de aula, onde o aluno entrevistado além de ter que esperar tal papel chegar a ele, relatou que não era agradável trabalhar com folhas já que os laboratórios possuem computadores. O sistema SGAEI comprovou com eficácia, que seria uma grande vantagem para a instituição, o qual logo depois de implementado se tornou de grande ajuda para o professor e facilitou no ensino e controle dos alunos.

O resultado obtido pelo *software* foi imediato gerando tranquilidade ao aluno, que poderia consultar suas notas e exercícios enviados, nas telas do sistema. Para a instituição e professor o tempo economizado foi vital, pois com o ele as aulas deram maior ênfase no conhecimento e ensino do que na organização do material.

Com a construção do *software*, as linguagens usadas obtiveram novas abordagens para sua utilização, conseqüentemente forçando a atualização dos conhecimentos adquiridos para construção do sistema. Um exemplo disso é a visão de como o padrão de projetos é importante na construção e organização de softwares na atualidade, fornecendo grande auxílio na Orientação a Objetos. Usando o padrão MVC o *software* pode ser atualizado com grande facilidade tanto na parte gráfica quanto na parte funcional, dando a instituição uma grande vantagem caso mudem alguma forma de trabalho e isso reflita no SGAEI.

Um melhor conhecimento das ferramentas usadas no desenvolvimento do sistema foi adquirido, graças ao levantamento de informações dos mesmos descobriu-se que o trabalho poderia ser feito com maior facilidade, para isso foram realizados estudos para conseguir o máximo da eficiência de cada *software* como *Netbeans* e *Astah*, com eles os resultados são satisfatórios em relação ao tempo para desenvolvimento do código fonte e organização de arquivos necessários na estrutura do sistema. Os *softwares* livres usados no desenvolvimento do trabalho acabaram gerando benefícios, pois além de serem livres e não necessitarem de valores finais acabam facilitando o desenvolvido do sistema. A compreensão sobre a importância da tecnologia da informação e a visão de seus clientes no perfil da empresa foi adquirida graças as pesquisas feitas para o desenvolvimento do *software*, visto que várias necessidades para o crescimento de muitas empresas depende da TI e não seria diferente em uma escola de informática.

Os resultados obtidos pelo *software* são de grande utilidade na instituição, nele é encontrado o conteúdo das aulas, os alunos que participam das mesmas e também facilita o trabalho durante os exercícios, pois o tempo que antes era gasto buscando materiais e distribuindo os mesmos para cada aluno não existe mais. Quanto à aula com um único aluno o cronograma disponibilizado no sistema auxilia o professor no preparo correto do conteúdo abordado em aula. Tanto os resultados para a instituição e também ao desenvolvedor foram aceitos, pois do lado da empresa a percepção a novas tecnologias para auxílio foi repensada e para o desenvolvedor, o trabalho junto a empresa mostrou o cuidado que o mesmo deve ter na hora de construir o *software* da forma que o cliente necessita.

Considerações Finais

Finalizado o desenvolvimento do sistema, sua implementação ocorreu seguindo o ciclo de vida adotado, que proporciona re-implementação caso for necessária e requisitada pelos usuários. Isso pode acontecer em qualquer momento, pois o sistema será utilizado diariamente nas aulas e no decorrer desse tempo necessidades podem ser percebidas e novas funcionalidades acrescentadas em seu *layout*.

Avaliações posteriores serão efetuadas determinando junto aos clientes, se mais funções são necessárias e se o sistema atende as necessidades relatadas no início do ciclo de vida, tais como a satisfação por parte dos professores e alunos com o uso do *software* nas aulas da instituição.

Também é importante destacar os componentes utilizados para construção de uma boa arquitetura da informação e que foram o diferencial em relação ao desenvolvimento, pois esses são de total importância para o *software*, como é o caso do Banco de Dados que guarda informações necessárias para o sistema. O ciclo de vida escolhido que mostra as etapas a serem seguidas para a construção do *software*. Os diagramas UML que facilitam a visualização de classes e métodos, a Orientação a Objetos que ajudam na organização de componentes do sistema.

Com a construção do sistema percebeu-se que a Tecnologia da Informação auxilia e facilita as atividades no ambiente educacional, em especial no ambiente em questão que o sistema foi desenvolvido e implementado. Porém, não bastando somente isso, deve-se ter cuidado em relação a professores qualificados, efetuar pesquisas de satisfação dos clientes, no caso em questão por parte dos alunos e estudar a viabilidade da implantação da tecnologia.

Referências

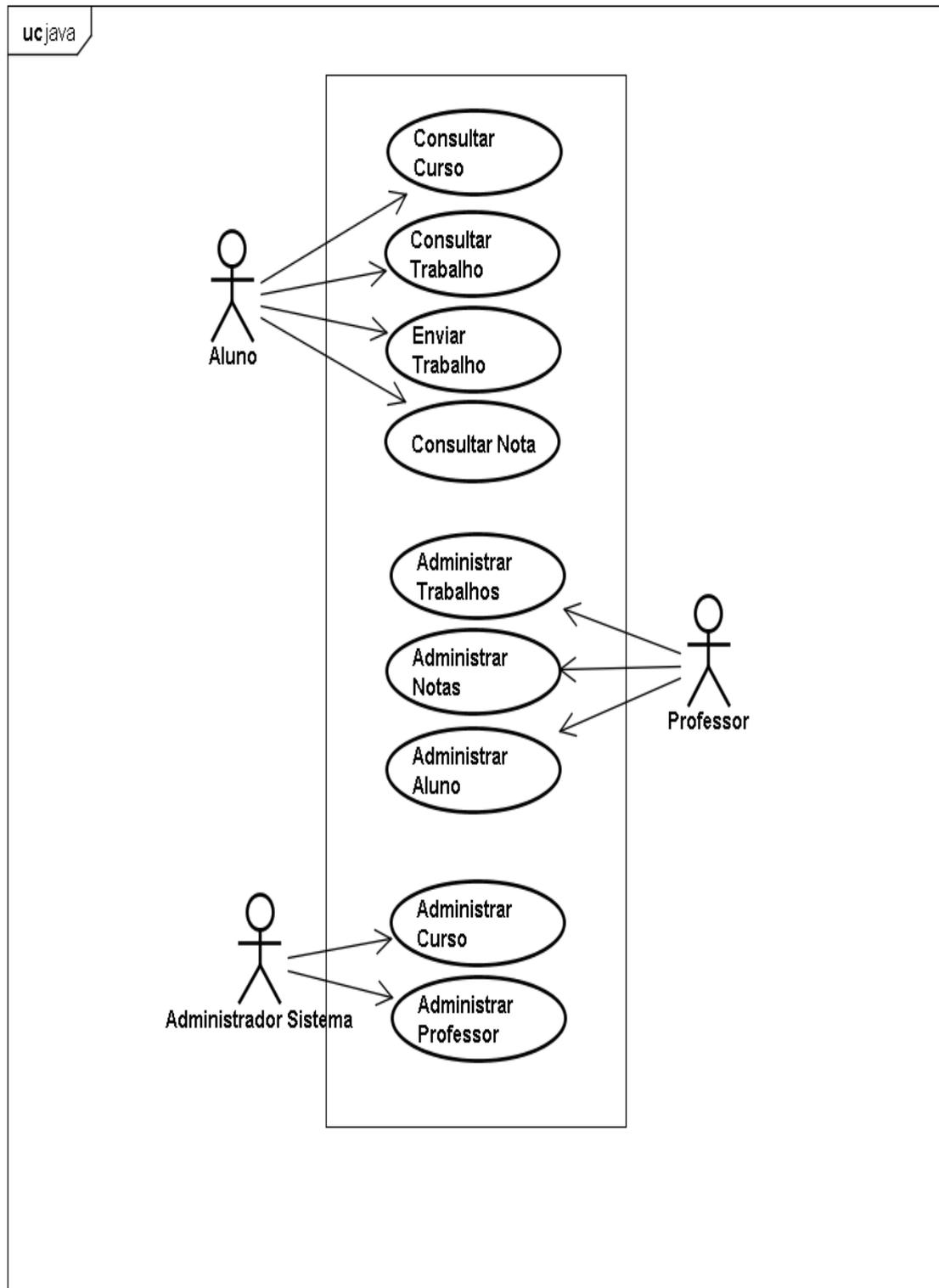
- Beighley, L & Morrison, M. Head First PHP & MySQL. O' Reilly Media, Inc. Estados Unidos, 2009.
- Beighley, L. Use a Cabeça! (Head First) SQL. Alta Books Editora. Rio de Janeiro – RJ, 2010.
- Beraldi, L. C. & Escrivão Filho, E. Impacto da Tecnologia de Informação na Gestão de Pequenas Empresas. Ci. Inf., Brasília, v.29, n1, p. 46-50, Jan./Abr, 2000.
- Cordeiro, S, E. Introdução a Ciclo de Vida de Software. 2003.
- Coulouris, G. Dollimore, J e Kindberg, T (2007). Distributed Systems: Concepts and Design. Addison Wesley.
- Dall'Oglio, P. PHP Programando com Orientação a Objetos. Editora Novatec Ltda., 2 ed., São Paulo – SP, 2009.
- Dennis, A. & Wixom, B. H. Análise e Projeto de Sistemas. Tradução: Michele Geinhart, 2 ed., Livros Técnicos e Científicos Editora Ltda., Rio de Janeiro – RJ, 2011.
- Elmars, R. & Navathe, S. B. Sistemas de banco de dados. [*Fundamentals of database systems*]. Tradução de Daniel Vieira. 6 ed. São Paulo: Pearson Addison-Wesley, 2011.
- Farina, Modesto, Psicodinâmica das cores em comunicação. Edgard Blusher., 5 ed. São Paulo -- SP, 2000.
- Freeman, E. & Freeman, E. Use a cabeça! HTML com CSS e XHTML. Editora Alta Books., 2 ed., Rio de Janeiro – RJ, 2008.
- Gonçalves, J. E. L. Os impactos das novas tecnologias nas empresas prestadoras de serviços. Revista de Administração de Empresas, São Paulo, v.1, n.34, p. 63-68, Jan/Fev, 1994.
- Denoncourt, D. (2011) “developerWorks” , “Aproveitando Namespaces do PHP V5.3 para um Código Legível Passível de Manutenção”, <https://www.ibm.com/developerworks/br/library/os-php-5.3namespaces>, Outubro.
- Lobo, E. J. R. Guia Prático de Engenharia de Software. Universo dos livros Editora Ltda., São Paulo – SP, 2009.
- Oliveira, A. C. M. da C. Tecnologia de Informação: Competitividade e políticas públicas. Revista de Administração de Empresas, São Paulo, v.36, n.2, p.34-43, Abr/Maio/Jun, 1996.
- Macdonald, J (2006). Blended Learning and Online Tutoring. Gower Editora, 2006.

- Magno, A. (2012) “WebPoint” , “Globo Bootstrap” , blog.alexandremagno.net/2012/08/globo-bootstrap, Outubro.
- Mokarzel, F. C. & Soma, N. Y. Introdução à Ciência da Computação. Elsevier Editora Ltda., Rio de Janeiro – RJ, 2008.
- Morrison, M. Use a cabeça! Javascript. Editora Alta Books., 2 ed., Rio de Janeiro – RJ, 2008.
- Renato, S. (2007) “Intitudo EduMed” , “Ambiente de Ensino e Aprendizagem via Internet A plataforma Moodle” , [http://www.ead.edumed.org.br/file.php/1/PlataformaMoodle .pdf](http://www.ead.edumed.org.br/file.php/1/PlataformaMoodle.pdf) , Outubro.
- Rezende, D. A. Engenharia de Software e Sistemas de Informação. Brasport Livros e Multimidia Ltda., 3 ed., Rio de Janeiro – RJ, 2005.
- Ribeiro Neto, L. G. Os impactos da tecnologia de informações nas organizações: uma visão política. R. Un. Alfenas, Alfenas, 5:95-101, 1999.
- Silva, R. M. A. & Videira, E. A. C. UML, Metodologias e Ferramentas CASE. Centro Atlântico Lda., 1 ed., Porto – Lisboa – Portugal, 2001.
- Sebesta, R. W. Conceitos de Linguagem de programação [*Concepts of programming language*]. Tradução: Addison Wesley Higher Education, Artmed Editora S.A., 5 ed., Porto Alegre – RS, 2002.
- Turban, E., McLean E. & Wetherbe J. Tecnologia da Informação para Gestão: Transformando os negócios na economia digital [*Information techonology for management: transforming business in the digital economy*]. Tradução: John Wiley & Sons, Inc., Artmed Editora S.A., 3 ed. Porto Alegre – RS, 2002.
- Valente, J.A. A Espiral da Aprendizagem e as Tecnologias da Informação e Comunicação: repensando Conceitos. Em M. C. R. A Joly (org.). *A Tecnologia no Ensino: implicações para a aprendizagem.* (p. 41-62). São Paulo: Casa do Psicólogo – SP, 2002.
- Welling, L. Thomson, L. PHP e MySQL desenvolvimento Web. Tradução: Edson Furmankiewicz e Adriana Kramer. Elsevier. 6° Reimpressão. Rio de Janeiro – RJ, 2005.

Apêndices

Diagramas UML referentes ao Sistema Gerenciador de Aprendizagem para Escola de Informática.

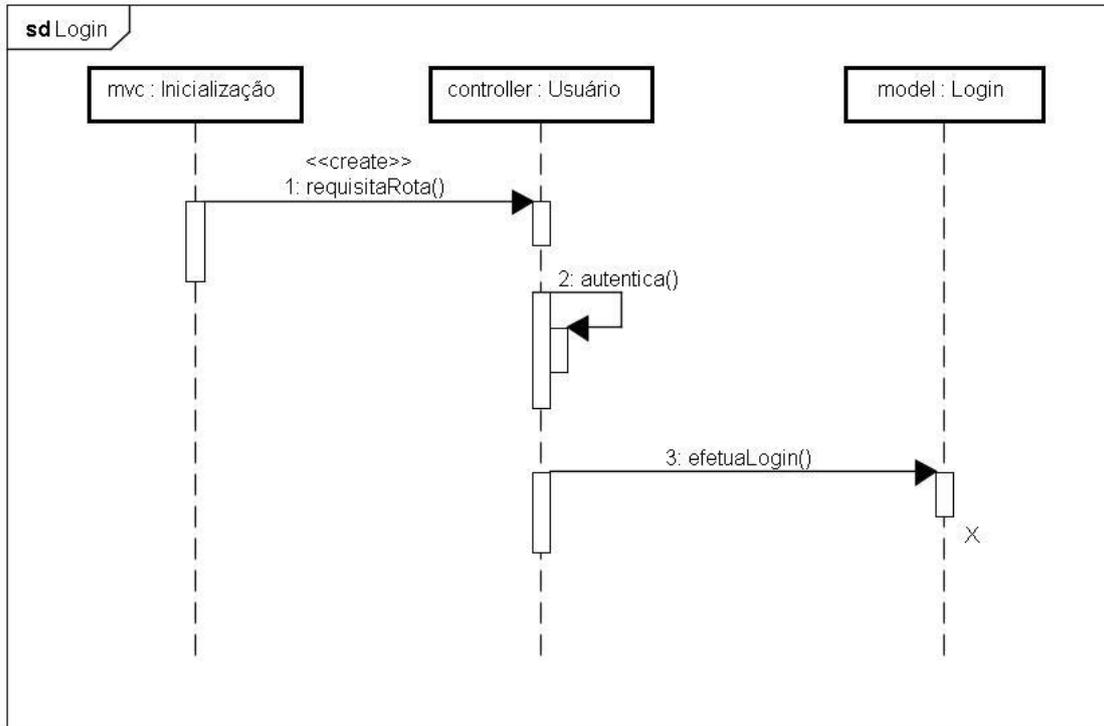
Apêndice A: Diagrama de Caso de Uso.



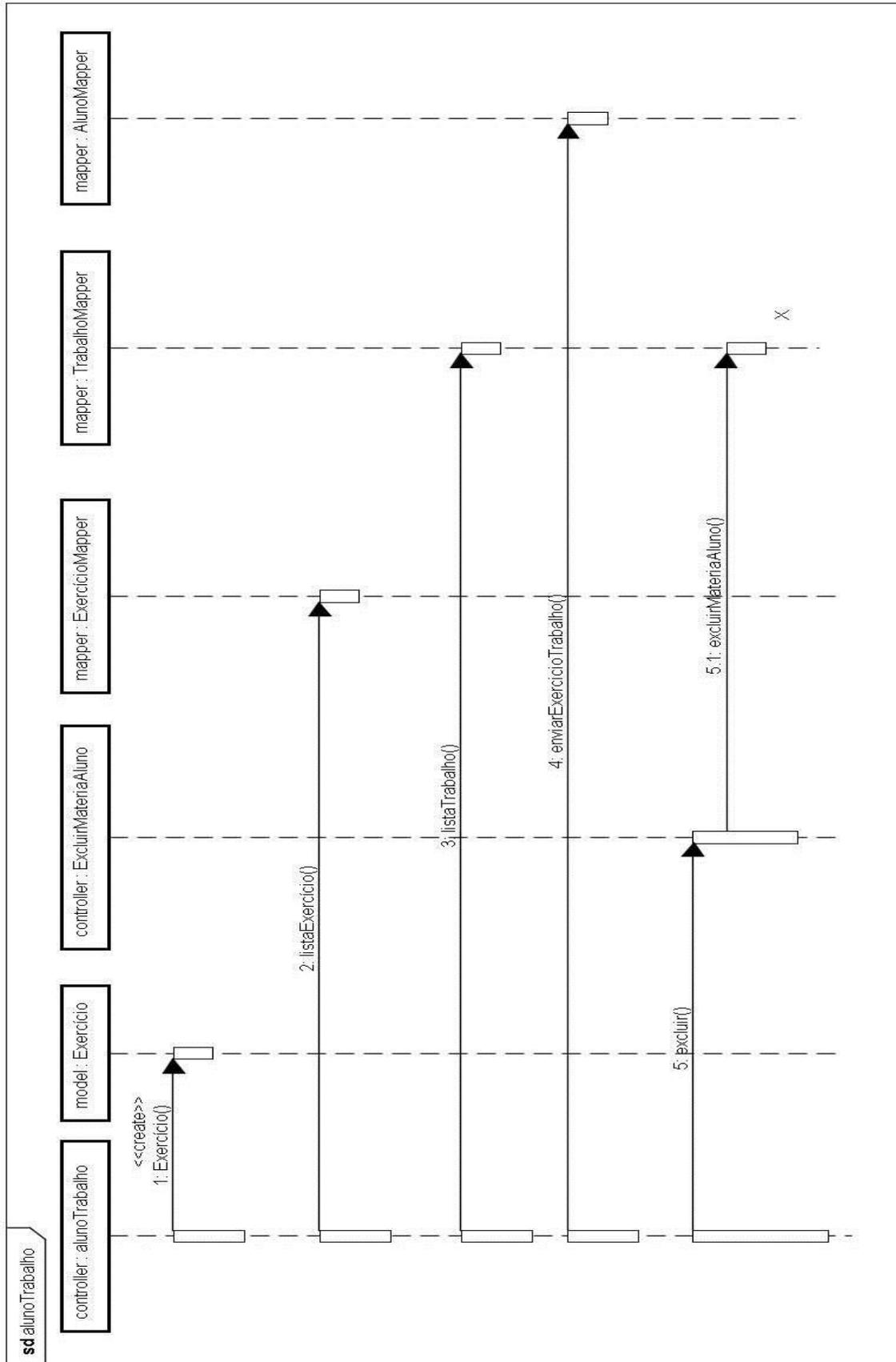
Apêndice B: Diagrama de Classes.



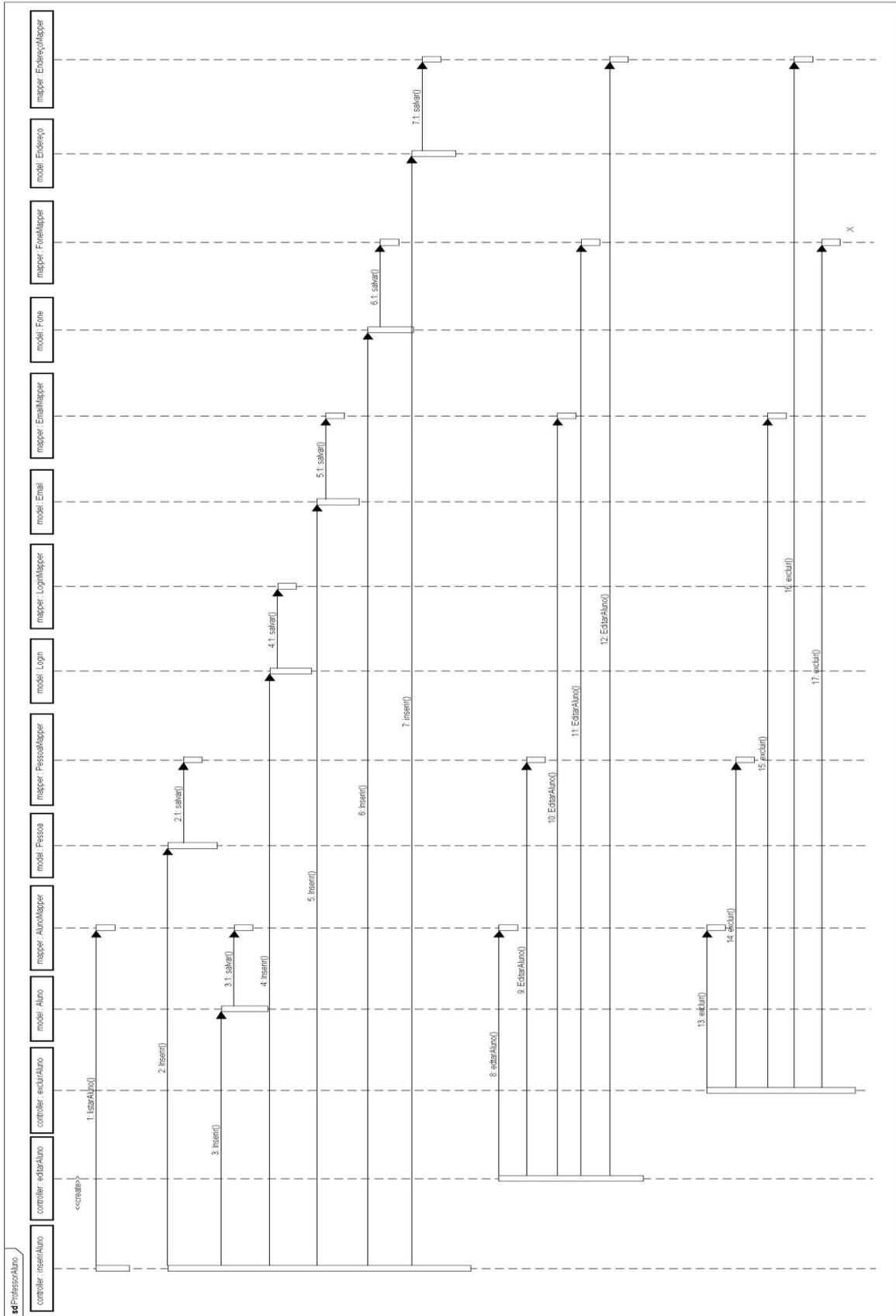
Apêndice C: Diagrama de Sequência Login.



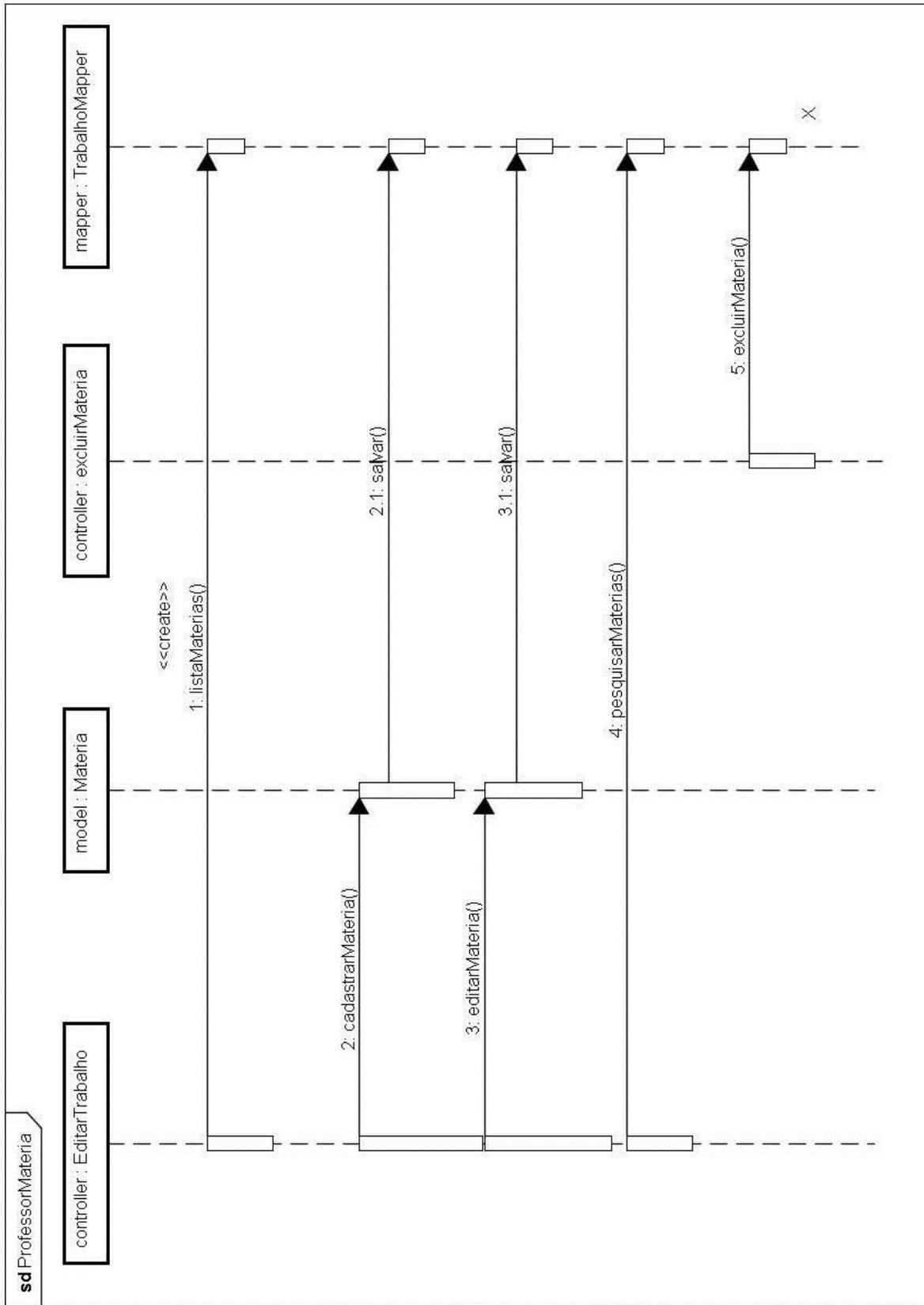
Apêndice D: Diagrama de Sequência Aluno.



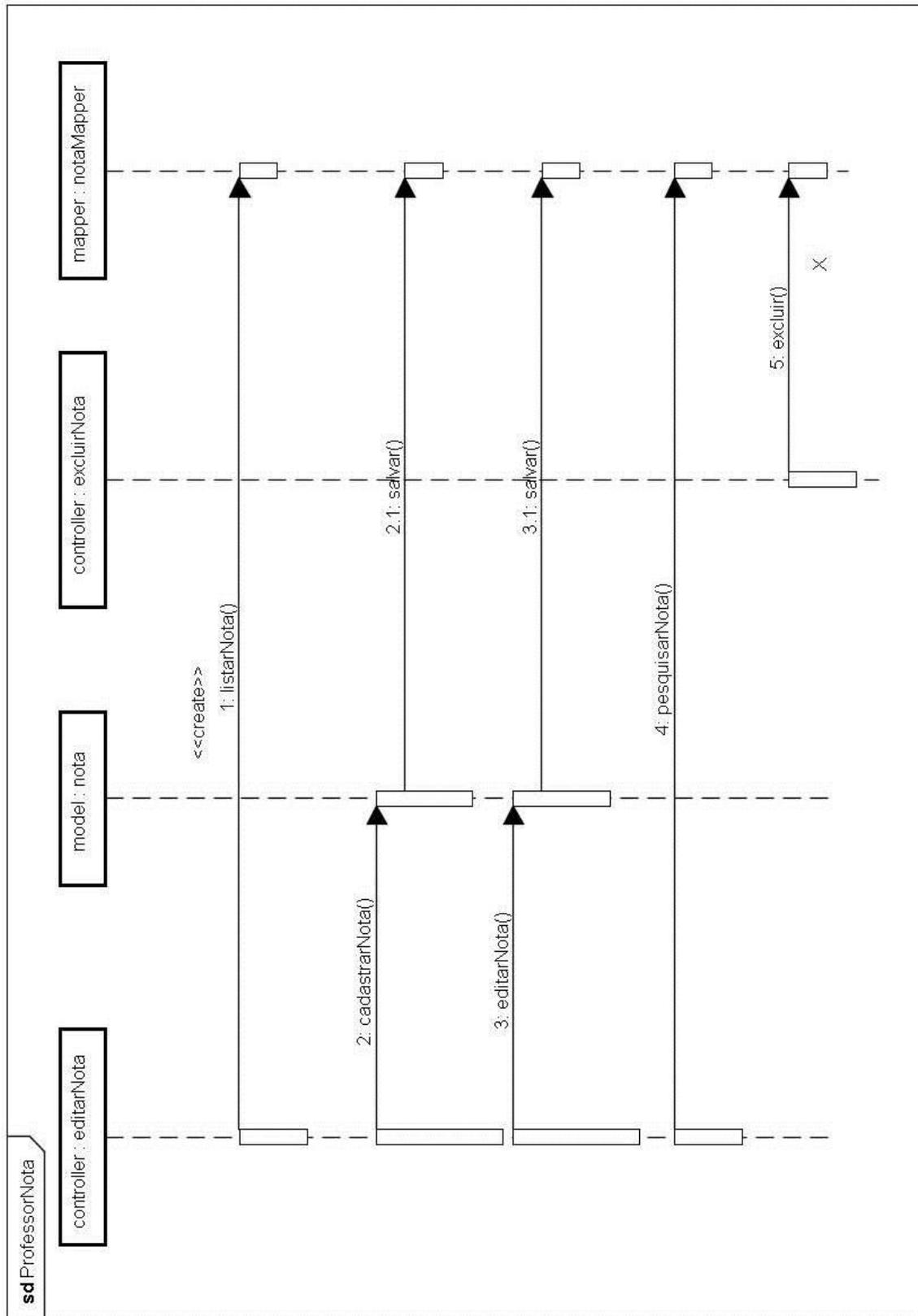
Apêndice E: Diagrama de Sequência Professor insere Aluno.



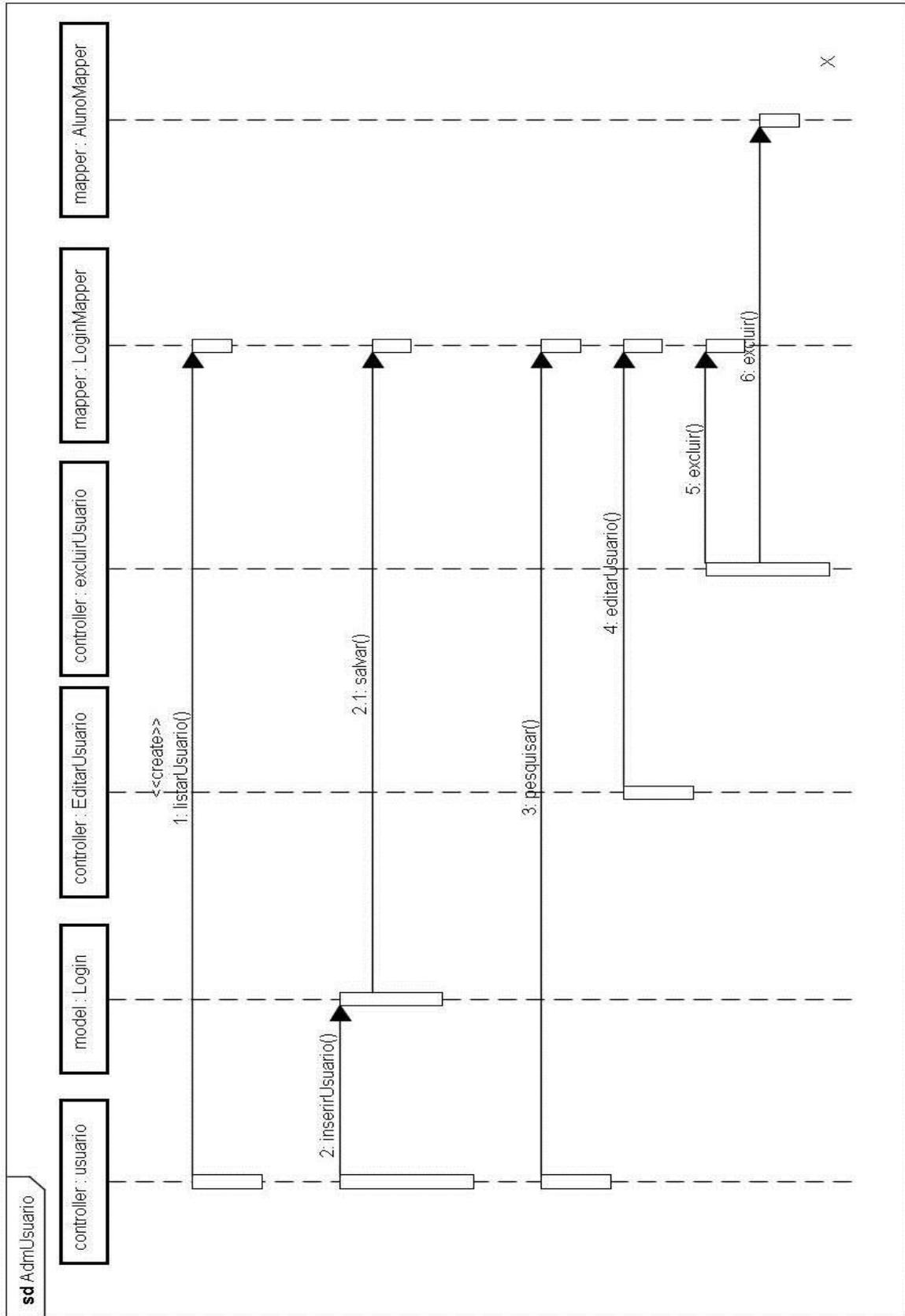
Apêndice F: Diagrama de Sequência Professor insere Matéria.



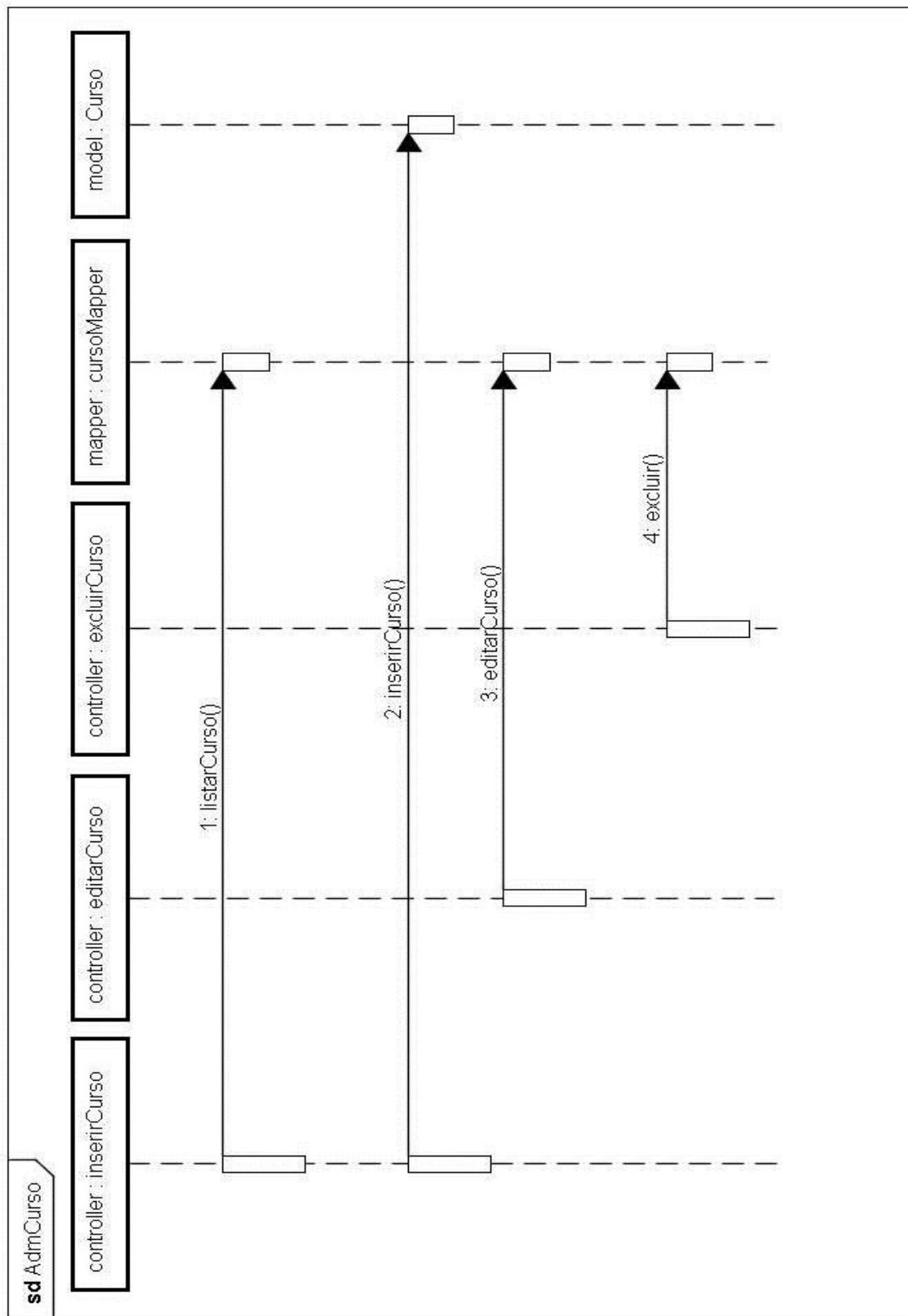
Apêndice G: Diagrama de Sequência Professor insere Nota.



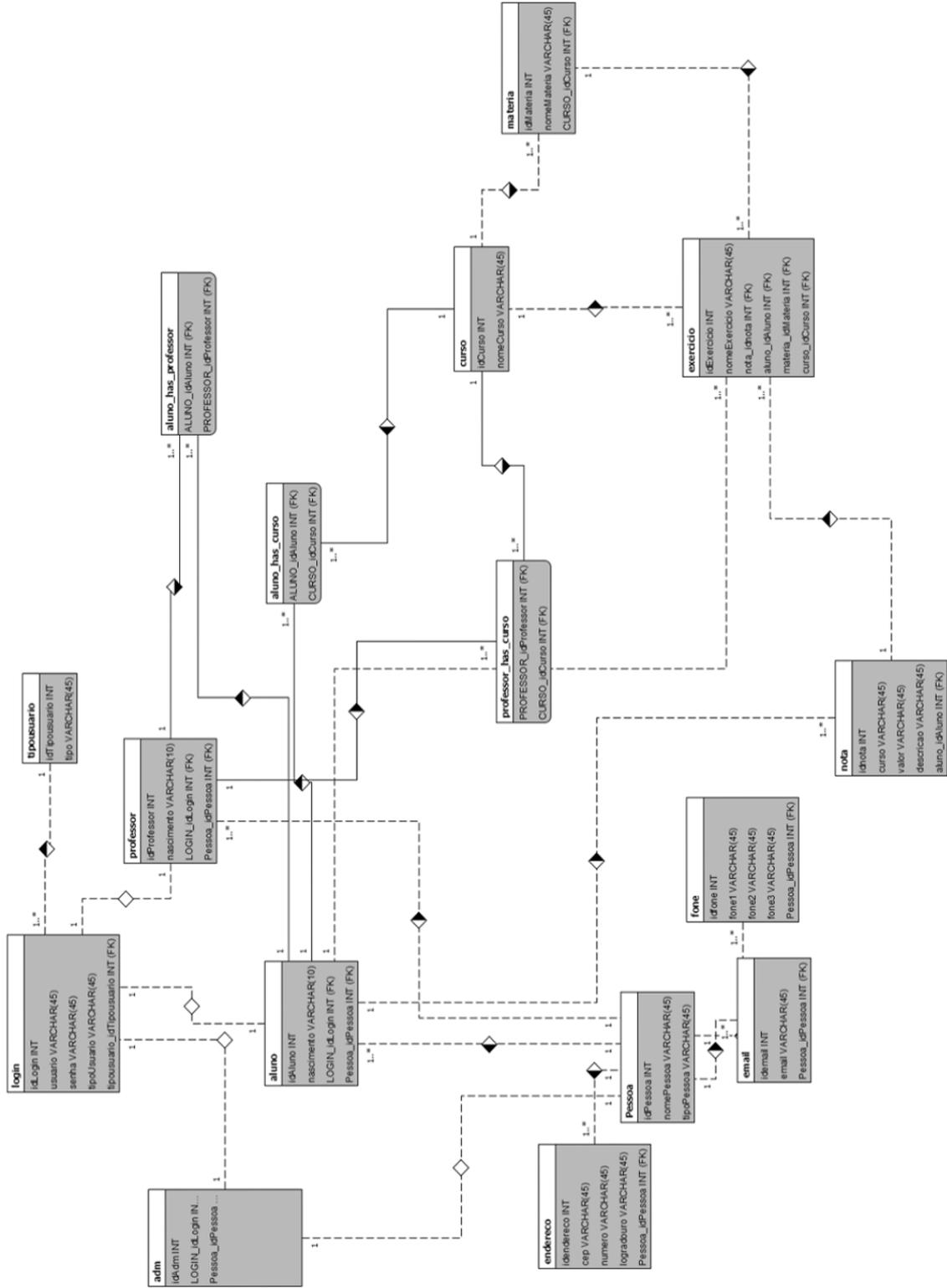
Apêndice H: Diagrama de Sequência Adm insere Usuário.



Apêndice I: Diagrama de Sequência Adm insere Curso.



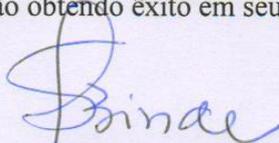
Apêndice J: Modelagem do Banco de Dados.



Apêndice K: Declaração de Implementação do Software.

DECLARAÇÃO DE IMPLEMENTAÇÃO DE SOFTWARE

Declaramos que o Srº **Rafael Marcondes Teixeira** aluno na Faculdade **Guairacá** do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, no ano letivo de 2013 implementou o **Sistema Gerenciador de Aprendizagem para Escola em Informática** em nossa instituição obtendo êxito em seu trabalho.



Vital Net Consultoria em Informática



Rafael Marcondes Teixeira

Guarapuava. PR

Novembro de 2013

05 485 153/0001-89

VITALNET CONSULTORIA EM
INFORMÁTICA LTDA

RUA SALDANHA MARINHO, 1226
SALA 03 - CENTRO

CEP 85010-290 GUARAPUAVA - PR