



---

**FACULDADE DE TECNOLOGIA DE AMERICANA**  
**Curso Análise e Desenvolvimento de Sistemas**

Thiago Henrique Felix

**SIGE – SISTEMA DE GERENCIAMENTO ESCOLAR**

**Americana, SP**

**2017**



---

**FACULDADE DE TECNOLOGIA DE AMERICANA**

**Curso Análise e Desenvolvimento de Sistemas**

Thiago Henrique Felix

**SIGE – SISTEMA DE GERENCIAMENTO ESCOLAR**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Esp. Antonio Alfredo Lacerda.

Área de concentração: Engenharia de Software

**Americana, SP.**

**2017**

**FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS**  
**Dados Internacionais de Catalogação-na-fonte**

F36s      FELIX, Thiago Henrique

SIGE - Sistema de Gerenciamento Escolar. / Thiago Henrique Felix. – Americana, 2017.

100f.

Monografia (Curso de Tecnologia em Análise e Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Esp. Antonio Alfredo Lacerda

1 Sistemas de informação I. LACERDA, Antonio Alfredo II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.518

**SIGE – SISTEMA DE GERENCIAMENTO ESCOLAR**

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.  
Área de concentração: Engenharia de software.

Americana, 12 de dezembro de 2017.

**Banca Examinadora:**



---

Antonio Alfredo Lacerda  
Especialista  
Faculdade de Tecnologia de Americana



---

Diógenes de Oliveira  
Mestre  
Faculdade de Tecnologia de Americana



---

Eduardo Antonio Vicentini  
Mestre  
Faculdade de Tecnologia de Americana

## **AGRADECIMENTOS**

Agradeço a todos os professores por todo o conhecimento que me foi passado e graças a isso pude desenvolver o presente trabalho.

## DEDICATÓRIA

Aos meus pais, e todos os amigos que me acompanharam nesta caminhada de três anos rumo a graduação de tecnólogo.

## RESUMO

O objetivo deste trabalho é o desenvolvimento de um sistema escolar para o gerenciamento de informações básicas do dia a dia escolar, ou seja, o sistema faz o controle de professores, alunos, matérias, cursos, turmas entre outros que englobam as funções básicas de qualquer escola. A metodologia de engenharia reversa foi utilizada neste trabalho para a criação de seus diagramas, ou seja, primeiramente foi desenvolvido o software e logo após foi criado os diagramas de classe, diagrama de sequência, diagrama de caso de uso que foi parcialmente atualizado e logo após a documentação do sistema. Os principais objetivos esperados do desenvolvimento do software foram alcançados, sendo desenvolvido um software que atende todas as necessidades básicas para o gerenciamento escolar sem se aprofundar em uma regra de negócio específica.

**Palavras Chave:** Sistema; Gerenciamento; Diagrama;

## **ABSTRACT**

The objective of this work is the development of a school system for the management of basic information of the school day by day, that is, the system controls teachers, students, subjects, courses, classes and others that encompass the basic functions of any school. The reverse engineering methodology was used in this work to create their diagrams, that is, the software was first developed and soon after the class diagrams, sequence diagram, use case diagram was created that was partially updated and soon after the system documentation. The main objectives of software development were achieved, and software was developed that meets all the basic needs for school management without going deep into a specific business rule.

**Keywords:** System; Management; Diagram;



## SUMÁRIO

1	SISTEMAS DE INFORMAÇÃO .....	17
1.1	SISTEMAS DE INFORMAÇÃO GERENCIAL.....	18
1.2	SOFTWARE LIVRE .....	19
1.3	ENGENHARIA DE SOFTWARE.....	21
1.3.1	Processos de desenvolvimento de softwares .....	21
1.3.2	Cascata .....	22
1.3.3	Evolucionário .....	23
1.3.4	Incremental.....	24
1.3.5	Espiral.....	25
1.3.6	Extreme Programming .....	26
1.3.7	Scrum .....	26
1.4	FERRAMENTAS DE DESENVOLVIMENTO .....	27
1.4.1	Php .....	27
1.4.2	Firebird .....	28
1.4.3	Codeigniter .....	28
1.4.4	AdminLTE .....	28
2	LEVANTAMENTO DE REQUISITOS .....	29
2.1	REQUISITOS FUNCIONAIS .....	29
2.2	REQUISITOS NÃO FUNCIONAIS.....	30
2.2.1	Portabilidade.....	30
2.2.2	Segurança .....	30
2.2.3	Usabilidade.....	31
2.2.4	Documentação.....	31
3	FERRAMENTAS DE DOCUMENTAÇÃO DO SISTEMA .....	31
3.1	UML.....	31
3.1.1	Caso de uso .....	32
3.1.2	Classe.....	33
3.1.3	Sequência.....	33
3.2	MODELO ENTIDADE E RELACIONAMENTO .....	34
3.3	DIAGRAMA ENTIDADE E RELACIONAMENTO .....	35
4	ESTUDO DE CASO .....	36
4.1	CASO DE USO.....	36
4.1.1	Administrador.....	37
4.1.2	Professor .....	41

4.1.3	Aluno .....	42
4.2	DIAGRAMA DE CLASSES .....	42
4.2.1	Controlador cadastro .....	43
4.2.2	Controlador visualizar .....	45
4.2.3	Controlador menage .....	47
4.2.4	Controlador update .....	49
4.2.5	Controlador login .....	51
4.3	DIAGRAMA DE SEQUÊNCIA .....	52
4.3.1	Login .....	52
4.3.2	Recuperação de senha.....	53
4.3.3	Professor .....	54
4.3.4	Administrador.....	59
4.3.5	Aluno .....	60
4.3.6	Avisos .....	62
4.3.7	Cursos .....	63
4.3.8	Matéria.....	64
4.3.9	Turma .....	65
4.3.10	Frequência .....	66
4.3.11	Notas.....	67
4.4	MODELO ENTIDADE E RELACIONAMENTO (MER).....	67
4.5	DIAGRAMA ENTIDADE E RELACIONAMENTO (DER) .....	70
4.6	SOFTWARE .....	71
4.6.1	Login.....	71
4.6.2	Recuperação de senha.....	72
4.6.3	Tela inicial do Administrador.....	73
4.6.4	Gerenciamento de Alunos .....	74
4.6.5	Perfil do Aluno .....	75
4.6.6	Alteração de Aluno .....	76
4.6.7	Cadastro de Aluno .....	77
4.6.8	Ativação ou desativação de alunos .....	79
4.6.9	Gerenciamento de Professores .....	81
4.6.10	Cadastro de Professor.....	82
4.6.11	Gerenciamento de Administradores .....	83
4.6.12	Gerenciamento de Cursos.....	84
4.6.13	Cadastro de Curso .....	85
4.6.14	Informações do Curso .....	86

4.6.15	Gerenciamento de Matérias .....	87
4.6.16	Cadastro de Matéria .....	88
4.6.17	Informações da Matéria.....	89
4.6.18	Gerenciamento de Turmas.....	90
4.6.19	Cadastro de Turma .....	91
4.6.20	Gerenciamento de Frequência .....	92
4.6.21	Gerenciamento de Notas.....	93
4.6.22	Gerenciamento de Avisos .....	94
4.6.23	Matricula de Aluno em Turma .....	95
4.6.24	Configurações do Sistema .....	96
4.6.25	Log do Sistema .....	97
CONSIDERAÇÕES FINAIS .....		98
REFERÊNCIAS.....		100

## LISTA DE FIGURAS E DE TABELAS

Figura 1: Componentes de um sistema de informação .....	17
Figura 2: Modelo de processo de software cascata .....	23
Figura 3: Desenvolvimento evolucionário.....	24
Figura 4: Modelo de Desenvolvimento Incremental .....	25
Figura 5: Modelo de Desenvolvimento em Espiral .....	25
Figura 6: PHP logo .....	27
Figura 7: framework adminLTE .....	29
Figura 8: Exemplo do Diagrama de Caso de Uso .....	32
Figura 9: Exemplo de Diagrama de Classes .....	33
Figura 10: Exemplo do Diagrama de Sequência .....	34
Figura 11: Exemplo do Modelo de Entidade e Relacionamento.....	34
Figura 12: Exemplo do Diagrama Entidade e Relacionamento .....	35
Figura 13: Diagrama de caso de uso administrador 2.....	37
Figura 14: Diagrama de caso de uso administrador 1 .....	38
Figura 15: Diagrama de caso de uso administrador 3.....	39
Figura 16: Diagrama de caso de uso administrador 4.....	40
Figura 17: Diagrama de caso de uso professor.....	41
Figura 18: Diagrama de caso de uso aluno.....	42
Figura 19: Diagrama de classe cadastro .....	43
Figura 20: Diagrama de classe cadastro 2.....	44
Figura 21: Diagrama de classe visualizar.....	45
Figura 22: Diagrama de classe visualizar 2.....	46
Figura 23: Diagrama de classes manage .....	47
Figura 24: Diagrama de classe manage 2.....	48
Figura 25: Diagrama de classe update.....	49
Figura 26: Diagrama de classe update 2.....	50
Figura 27: Diagrama de classe login .....	51
Figura 28: Diagrama de sequência login.....	52
Figura 29: Diagrama de sequência recuperação de senha.....	53

Figura 30: Diagrama de seqüência gerenciamento de professor .....	54
Figura 31: Diagrama de seqüência cadastro de professor .....	55
Figura 32: Diagrama de seqüência alteração de professor .....	56
Figura 33: Diagrama de seqüência ativação de professor .....	57
Figura 34: Diagrama de seqüência desativação de professor .....	57
Figura 35: Diagrama de seqüência perfil do professor .....	58
Figura 36: Diagrama de seqüência gerenciamento de administradores .....	59
Figura 37: Diagrama de seqüência gerenciamento de alunos .....	60
Figura 38: Diagrama de seqüência matricula de aluno em turma .....	61
Figura 39: Diagrama de seqüência gerenciamento de avisos.....	62
Figura 40: Diagrama de seqüência gerenciamento de cursos .....	63
Figura 41: Diagrama de seqüência gerenciamento de matérias .....	64
Figura 42: Diagrama de seqüência gerenciamento de turmas .....	65
Figura 43: Diagrama de seqüência gerenciamento de frequências .....	66
Figura 44: Diagrama de seqüência gerenciamento de notas .....	67
Figura 45: Modelo entidade e relacionamento .....	68
Figura 46: Modelo entidade e relacionamento 2 .....	69
Figura 47: Diagrama de entidade e relacionamento.....	70
Figura 48: Tela de login.....	71
Figura 49: Tela recuperação de senha.....	72
Figura 50: Tela inicial do administrador.....	73
Figura 51: Tela de Gerenciamento de Aluno.....	74
Figura 52: Tela de Perfil do Aluno .....	75
Figura 53: Tela de alteração de aluno.....	76
Figura 54: Tela de perfil de aluno sucesso na alteração .....	76
Figura 55: Tela de cadastro de aluno.....	77
Figura 56: Tela de cadastro de aluno 2.....	78
Figura 57: Tela de gerenciamento de alunos ativação de alunos .....	79
Figura 58: Tela de gerenciamento de alunos desativação .....	80
Figura 59: Tela de gerenciamento de professores .....	81
Figura 60: Tela de cadastro de professor.....	82
Figura 61: Tela de gerenciamento de administrador .....	83
Figura 62: Tela de gerenciamento de cursos .....	84
Figura 63: Tela de cadastro de curso.....	85

Figura 64: Tela de cadastro de curso 2.....	85
Figura 65: Tela de informações do curso .....	86
Figura 66: Tela de gerenciamento de matérias .....	87
Figura 67: Tela de cadastro de matéria.....	88
Figura 68: Tela de cadastro de matéria 2.....	88
Figura 69: Tela de informações da Matéria.....	89
Figura 70: Tela de gerenciamento de turmas.....	90
Figura 71: Tela de cadastro de turma .....	91
Figura 72: Tela de cadastro de turma 2 .....	91
Figura 73: Tela de gerenciamento de frequência .....	92
Figura 74: Tela de gerenciamento de notas.....	93
Figura 75: Tela de Gerenciamento de avisos.....	94
Figura 76: Tela de matricula do aluno em turmas .....	95
Figura 77: Tela de configurações do sistema.....	96
Figura 78: Tela de log do sistema .....	97

## INTRODUÇÃO

A tecnologia ao longo das últimas décadas mudou o estilo de vida das pessoas, facilitado suas vidas, softwares de bancos possibilitam transações bancárias em poucos segundos através de aparelhos celulares, compras do mês realizadas em qualquer lugar através da internet, além da conexão de pessoas em diferentes lugares do mundo. Mas a evolução da tecnologia não traz apenas benefícios para as pessoas, mas também para empresas e o setor público com sistemas de informação capazes de facilitar o gerenciamento.

Mesmo com a tecnologia trazendo tantos avanços e possibilidades como os sistemas de informação computacionais, existem comércios, empresas, setores públicos, entre outros, que gerenciam seus negócios no papel.

É importante se atentar que sistemas de informação computacionais não são baratos, sua criação gera um custo e muitas vezes pequenos comércios tem receio em investir ou até mesmo há uma inexistência de recursos para o investimento, e é aí que entra o software livre, softwares criados por uma comunidade de diversos programadores ao redor do mundo e normalmente distribuídos gratuitamente.

Este modelo de software possibilita a entrada de pequenos negócios ou organizações de qualquer tipo que ainda possuem dúvidas sobre a melhoria que os sistemas de informação podem oferecer para seus negócios, já que normalmente possuem uma grande diferença para com os softwares proprietários que é em relação ao seu preço.

O objetivo deste trabalho é o desenvolvimento de um sistema para o gerenciamento escolar composto pelas principais funcionalidades do dia a dia escolar sem se aprofundar em uma regra de negócio específica, o *software* será distribuído de forma gratuita, podendo ser útil tanto no setor público quanto para o setor privado, o foco será na implementação de sistemas de informação em entidades escolares que ainda duvidam deste potencial.

Para criação desta plataforma será utilizado a linguagem de programação PHP juntamente com o banco de dados Firebird. O presente trabalho é estruturado em quatro capítulos, o primeiro capítulo tem como objetivo situar o leitor com os presentes

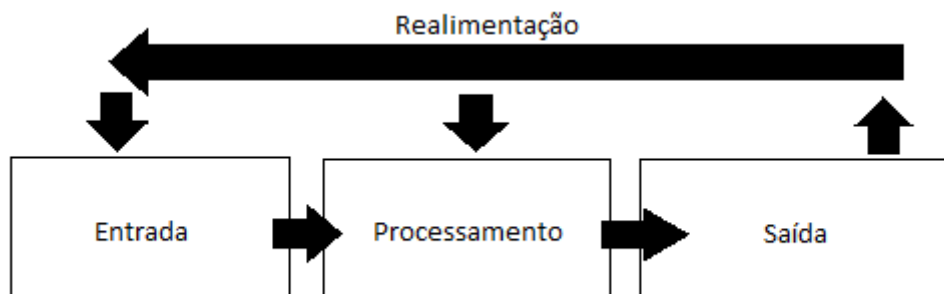
termos e conceitos aqui referenciados como a definição de sistemas de informação, sistemas de informação gerencial, *software* livre e por fim conhecer um pouco da linguagem utilizada no desenvolvimento. O segundo capítulo irá focar em todos os requisitos do sistema, ou seja, os principais objetivos para com este sistema sendo eles funcionais ou não funcionais. No terceiro capítulo será explicado e demonstrado toda a documentação do software sendo estes resultantes da engenharia de *software* e do banco de dados. E finalmente no ultimo capítulo será possível ver o sistema completo com seu manual de uso.



## 1 SISTEMAS DE INFORMAÇÃO

Stair e Reynolds (2009, p.7) definem: “Um sistema é um conjunto de elementos que interagem para realizar objetivos”. Um sistema de informação (SI) é um conjunto de componentes inter-relacionados para um determinado fim, estes componentes são: coleta de dados (entrada), manipulação de dados (processamento), armazenamento e retorno de dados (saída) e a realimentação de dados corretivo (mecanismo de realimentação), este mecanismo de realimentação é o elemento que leva as organizações a alcançarem seus objetivos, que com a análise dos dados de saída, realimenta o sistema com novos dados, como por exemplo um sistema de informação de uma loja de roupas, que consta que em uma determinada região um tipo de roupa vende em um número mais acelerado que em relação a outras regiões, analisando esta informação é possível concluir que este produto é essencial nesta região, sendo assim o estoque deve ser aumentado.

Figura 1: Componentes de um sistema de informação



Fonte: A autoria do Autor.

Stair e Reynolds (2009, p.10) definem: “A entrada é a atividade de captar e reunir os dados brutos”. Um exemplo seria um sistema de vendas, é necessário que se insira as informações do produto antes que fique disponível para o cliente, é na entrada onde estes dados serão inseridos ou em um sistema de rotas onde é necessário o local antes do processamento para a melhor rota.

Stair e Reynolds (2009, p.10) definem: “O processamento significa converter ou transformar dados em resultados úteis”. Para isso pode ser necessário a realização de cálculos, comparação de dados ou armazenamento o processamento é a parte

crucial de um sistema de informações. Neste componente é onde um sistema que determina o preço de um produto faz todos os cálculos necessários adicionando todo o custo de fabricação do produto, lucro além de impostos, ou um sistema de vendas online onde é feito cálculos dos produtos mais vendidos, mais procurados, menos procurados gerando um relatório para tomadas de decisão. Após todo o processamento a informação gerada pode ser armazenado para possíveis usos no futuro ou mesmo comparada com outras informações auxiliando para tomada de decisões.

Stair e Reynolds (2009, p.10) definem: “A saída envolve a produção de informações úteis, normalmente em forma de documentos e relatórios”. As saídas de um sistema pode ser a entrada de outros sistemas, sendo estas informações processadas novamente e gerando novas informações úteis.

Stair e Reynolds (2009, p.10) definem: “A realimentação é a informação originada no sistema, que é utilizada para fazer mudanças na entrada ou nas atividades de processamento”. Esta mudança seria uma atualização no sistema com a análise na informação de saída, são feitos novos parâmetros no sistema tanto na entrada de dados quanto no processamento de dados, um exemplo disto seria em um sistema de vendas que registra todo produto vendido e faz uma baixa no estoque, em sua saída no final do mês seria constado que a baixa no estoque de determinados produtos ficou abaixo de zero, o sistema não deveria realizar a baixa de estoque quanto o produto chega-se a zero com a análise da saída do sistema, é feita a realimentação atualizando a entrada de dados, não aceitando produtos com o estoque abaixo de um.

## 1.1 SISTEMAS DE INFORMAÇÃO GERENCIAL

Stair e Reynolds (2009, p.19) definem: “Um sistema de informação gerencial é um conjunto organizado de pessoas, procedimentos, softwares, banco de dados e equipamentos que fornecem informações rotineiras aos gerentes”. Este tipo de sistema de informação chamado de MIS (*managimnt information system*), tem foco em trazer relatórios periódicos aos gerentes e tomadores de decisão tendo foco na

eficiência organizacional englobando várias áreas organizacionais como marketing, produção, recursos humanos entre outras áreas.

Os MIS foram desenvolvidos nos anos de 1960 gerando relatórios periódicos para determinadas áreas organizacionais, com o passar do tempo os MIS proliferaram para todos os níveis organizacionais gerenciáveis.

## 1.2 SOFTWARE LIVRE

Ferreira (2009, p.32) diz: “Na década de 1940, surgiram os primeiros projetos de computadores digitais”. No surgimento dos primeiros projetos de computadores como ENIAC (*Eletronic Numerical Integratir and Computer*) ainda não havia o conceito de software, todo foco de pesquisa e desenvolvimento era no hardware, nas próximas décadas o hardware ganha espaço no cenário comercial ao mesmo tempo os pesquisadores começam a criar software para estes equipamentos com o intuito de torna-los úteis.

Nesta época os softwares eram criados por pesquisadores e o código era compartilhado entre universidades, como ainda não existia internet o código era enviado por correio e assim eram criados em comunidade, com o objetivo de aprender e melhorar o código.

Como o passar do tempo ficou evidente o potencial comercial do software com seus benefícios as grandes empresas, era um produto comercial incrível é aí que o cenário começa a mudar, um exemplo desta mudança foi o sistema operacional UNIX que inicialmente foi criado por pesquisadores, foi um marco importante na história da computação sendo vários softwares importantes baseados nele.

UNIX em sua criação tinha o código fonte divulgado entre universidades com o objetivo de se aprender e melhorar sendo uma importante descoberta científica, mas com o tempo os softwares começaram a ser desenvolvidos em larga escala com as empresas obrigando os programadores a assinarem acordos de não revelação, os programadores não poderiam divulgar o código fonte sendo vendido somente o código binário como um produto.

Ferreira (2009, p.32) diz: “Richard Matthew Stallman em meados dos anos 70 trabalhava no laboratório de IA (Inteligência Artificial) do MIT”. Nesta época a comunidade vivia seu auge, universidades compartilhavam entre si códigos fontes, o que na época era natural.

Stallman havia tido problemas com uma impressora nos laboratórios do MIT com atolamento de papeis, mas com uma mudança no código fonte da impressora foi possível avisar aos usuários que do problema ocorrido, a alteração não resolvia o problema, mas evitava que seus usuários ficassem esperando por um longo período de tempo.

O laboratório de IA do MIT certa vez ganhou uma impressora de presente da Xerox, Stallman procurou o código fonte da impressora como sempre fazia, mas com a pesquisa percebeu que somente era disponibilizado o código binário da impressora, ainda pesquisando Stallman encontrou um dos programadores do software da impressora e foi até ele, ao pedir o código fonte da impressora o programador o avisou que ele assinou um acordo de não divulgação não podendo divulgar o código fonte da impressora.

Foi aí que Stallman percebeu que o mundo no qual ele havia ajudado a crescer estava mudando o código fonte não era mais compartilhado, vários de seus amigos estavam assinando acordos de não revelação, ele percebeu que com este cenário crescente as empresas poderiam criar o monopólio do software já que mesmo pagando não se teria mais o direito de modificar o código. Então Stallman tomou a decisão, deixar o MIT para ajudar no desenvolvimento do software livre.

Ferreira (2009, p.36) diz: “Em 1984, Stallman iniciou o projeto o projeto GNU (GNU is not Unix) cujo o objetivo era de desenvolver um sistema operacional estilo UNIX que fosse software livre”. No ano seguinte em 1985 Stallman criou a Fundação de Software Livre (FSF – *Free Software Foundation*), uma organização sem fins lucrativos a partir disto foi criada a licença GPL (*General Public License*), esta licença garante que o software permaneça livre para sempre, sendo assim uma empresa não pode incluir o código com esta licença e vender como proprietário.

### 1.3 ENGENHARIA DE SOFTWARE

A engenharia de software é um ramo da engenharia cujo foco é o desenvolvimento dentro de custos adequados de sistemas de software de alta qualidade (SOMMERVILLE, 2007, p.3), o termo surgiu em 1968 em uma conferência na qual tinha-se o objetivo de discutir o que então era chamado de “Crise de software”, este termo surgiu após o lançamento de um novo hardware de computadores baseado em circuitos integrados, com isso veio a oportunidade de se criar software grandes que poderiam suprir as necessidades de grandes empresas, o problema era criar tais softwares sem métodos ou conceitos para garantir a qualidade do software dentro do prazo de desenvolvimento, e com isso surgiu a crise de software com projetos que superavam seus custos e prazos. Era necessário a criação de técnicas e conceitos para criação de grandes softwares, algumas das técnicas ainda existem até hoje enquanto novas são criadas a cada dia.

#### 1.3.1 Processos de desenvolvimento de softwares

Com o avanço rápido do hardware ao longo dos anos, veio a possibilidade de criação de grandes sistemas de informação, mas devido à grande complexidade em seu desenvolvimento foi necessário a criação de que na engenharia de software é chamado de processo de software. "Um processo de software é um conjunto de atividades que leva à produção de um produto de software" (SOMMERVILLE, 2007, p. 42), ou seja, é um conjunto de atividades necessária para o desenvolvimento seguro do software, as atividades fundamentais independentes do processo de software são:

Especificação de software: é a especificação de todas as funcionalidades do software e suas restrições, normalmente em softwares comerciais isso é definido juntamente com o cliente.

Projeto e implantação de software: Essas atividades tem como objetivo principal assegurar que as especificações do software sejam produzidas corretamente.

Validação do software: O software deve fazer o que foi especificado, nesta atividade essas especificações devem ser validadas no software.

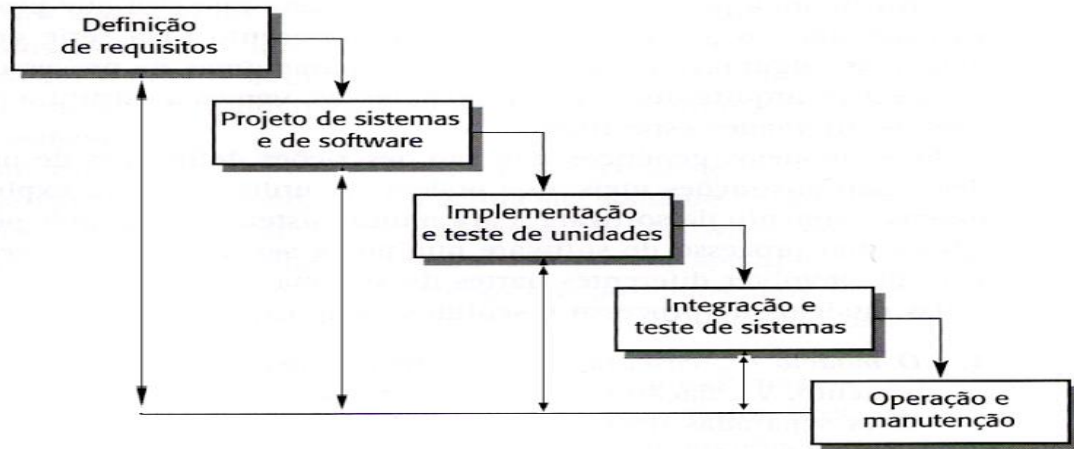
Evolução do software: As regras de negócios mudam com o tempo e os softwares devem acompanhar sendo assim necessário a evolução.

Não existe um processo de software correto para todos os projetos, o processo deve ser escolhido analisando o projeto, com um projeto complexo e com requisitos imutáveis o modelo de processo de *software* cascata é muito recomendado, mas para projetos com poucos requisitos e em constante mudança processos de desenvolvimento ágeis como scrum ou *extreaming programming* são os mais recomendados, a seguir será demonstrado os processos mais conhecidos sendo os já citados incluídos.

### 1.3.2 Cascata

Sommerville (2007, p.44) diz: “Devido ao encadeamento de uma fase com a outra, esse modelo é conhecido como modelo em cascata ou ciclo de vida de um software”. O modelo de processo de software cascata é um modelo tradicional, sendo utilizado somente em projetos específicos, nos quais cada atividade deve ser devidamente especificada para assim ir para a próxima. Na modelo cascata uma atividade somente pode ser iniciada se sua antecessora estiver terminada, devido a esta regra que este processo de software tem o nome de cascara demonstrada na Figura 2.

Figura 2: Modelo de processo de software cascata

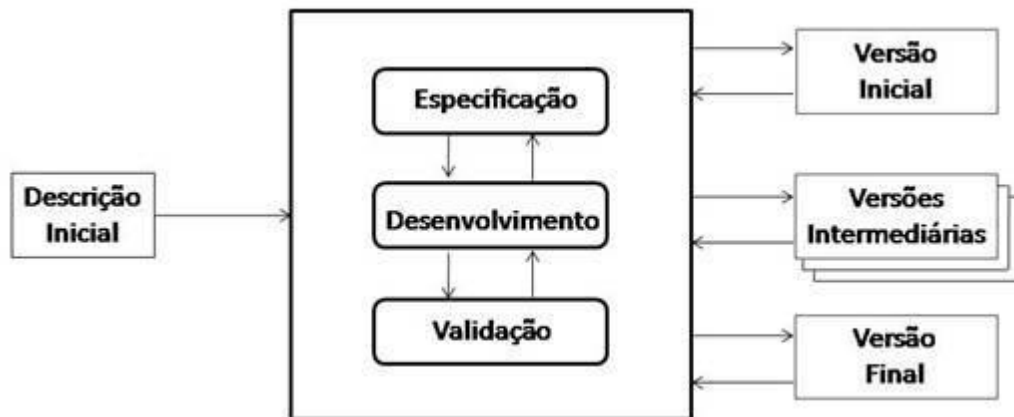


Fonte: <http://www.ebah.pt/content/ABAAfBgcAl/sistema-sanepar-na-linguagem-delph>

### 1.3.3 Evolucionário

Sommerville (2007, p.45) define: “O desenvolvimento evolucionário baseia-se na ideia de desenvolvimento de uma implantação inicial, expondo o resultado aos comentários do usuário e refinando por meio de várias versões [...]”. O modelo de processo de software evolucionário pode ser utilizado em projetos nos quais os requisitos são abstratos onde nem mesmo os clientes sabem ao certo o que querem, neste modelo primeiramente é desenvolvido uma versão inicial do sistema e baseado nos comentários do cliente, o sistema é modificado até sua versão final onde os objetivos do cliente são alcançados. Existem dois tipos de desenvolvimento evolucionário o primeiro é o desenvolvimento exploratório no qual o objetivo é trabalhar juntamente com o cliente explorando os requisitos até o desenvolvimento de um sistema final, o segundo chama-se prototipação *throwaway* (descartável) o objetivo deste é entender os requisitos focando-se na experimentação de requisitos mal compreendidos.

Figura 3: Desenvolvimento evolucionário



Fonte: Ian Sommerville, 2007, p.46.

A grande vantagem dessa versão é que sempre há especificação de requisitos sendo assim é muito mais fácil desenvolver um sistema ideal para o cliente, porém a documentação do sistema pode ser caótica.

#### 1.3.4 Incremental

O modelo de processo incremental (Figura 4) utiliza princípios do modelo de processo cascata definindo no início do projeto um conjunto de requisitos a serem desenvolvidos e também utiliza princípios do modelo evolucionário entregando o software em versões.

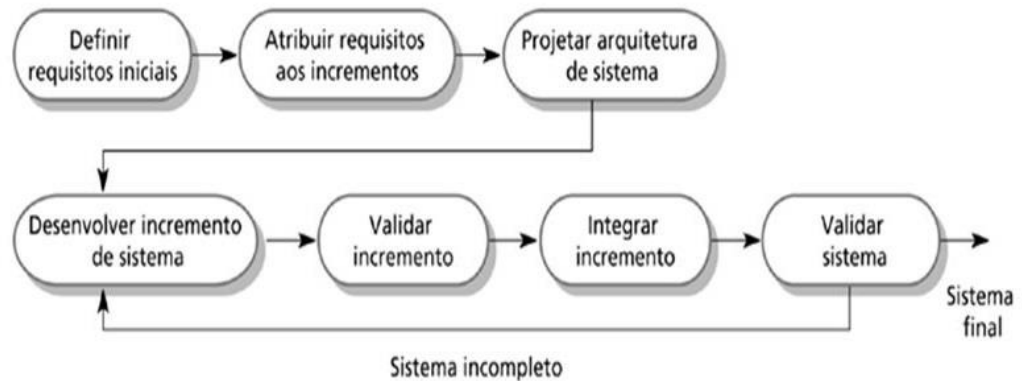
Em um processo de desenvolvimento incremental, o cliente identifica, em linhas gerais, os serviços a serem fornecidos pelo sistema (SOMMERVILLE, 2007, p.47).

Inicialmente o cliente define quais são os serviços mais importantes e os menos importantes, com isso é definido os incrementos, cada incremento irá implementar um conjunto de funcionalidade, sendo entregues de acordo com a prioridade dos serviços. O objetivo com tal abordagem é a rápida entrega de funcionalidades e recebendo assim o *feedback* do cliente.



**Figura 4: Modelo de Desenvolvimento Incremental**

**Figura 4.4**  
Entrega incremental.

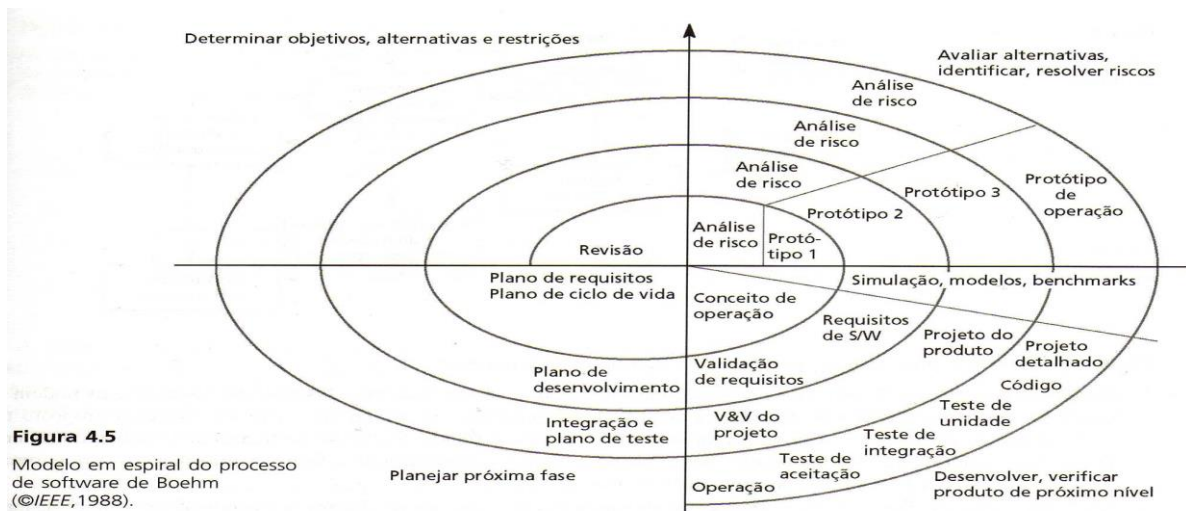


Fonte: Ian Sommerville, 2007, p.47

1.3.5 Espiral

No modelo em espiral (Figura 5) o processo de desenvolvimento é representado como uma espiral, sendo cada loop um processo de desenvolvimento. A espiral é dividida em quatro loops, o primeiro é a definição de objetivos do projeto, em seguida vem a avaliação e redução de riscos logo após vem o desenvolvimento e validação e por fim o planejamento, o projeto é revisto e se for decidido o prosseguimento os planos para a próxima fase do projeto são definidos.

**Figura 5: Modelo de Desenvolvimento em Espiral**



**Figura 4.5**  
Modelo em espiral do processo de software de Boehm (©IEEE, 1988).

Fonte: Ian Sommerville, 2007, p.49

### 1.3.6 Extreme Programming

O *extreme programming* é um modelo de desenvolvimento ágil para pequenas equipes com o objetivo de desenvolver softwares com requisitos vagos e que devido à falta de conhecimento dos requisitos do cliente estão em constantes mudanças.

No *extreme programming*, todos os requisitos são expressos como cenários (chamados histórias do usuário), que são implementados diretamente como uma série de tarefas (SOMMERVILLE, 2007, p.264).

Os programadores trabalham em pares e desenvolvem testes antes mesmo de criarem o código, quando o código é desenvolvido todas os testes devem ser executados com sucesso.

Em um processo XP, os clientes então intimamente envolvidos na especificação e priorização dos requisitos de sistema (SOMMERVILLE, 2007, p.264).

O cliente e os desenvolvedores desenvolvem um cartão de histórias que engloba todas as necessidades do cliente, após a definição do cartão de histórias a equipe o dividirá em tarefas, cada tarefa terá sua prioridade sendo o cliente o responsável pela escolha das tarefas que podem ser usadas imediatamente no negócio.

### 1.3.7 Scrum

Scrum é uma metodologia ágil assim como xp e é utilizada na gestão e planejamento de projetos de software.

Os projetos no Scrum são divididos em ciclos no qual são chamados de Sprints. Cada Sprint tem um conjunto de atividades que devem ser executadas. Todas as funcionalidades que serão implementadas no projeto ficam em uma lista chamada de *Product Backlog*. No início de cada *Sprint* o *Product Owner* prioriza todos os itens do *Product Backlog* e a equipe decide o que será implementado no dia.

## 1.4 FERRAMENTAS DE DESENVOLVIMENTO

As ferramentas de desenvolvimento são tecnologias utilizadas para o desenvolvimento do software e sua execução, todas as ferramentas utilizadas são obrigatoriamente gratuitas já que um dos objetivos centrais desta monografia é desenvolver um software livre.

### 1.4.1 Php

PHP é uma linguagem de programação criada em 1994 por Rasmus Lerdof, neste tempo a linguagem era apenas scripts para páginas dinâmicas que Rasmus utilizava para monitorar seu currículo online, mais tarde em 1995 foi lançado uma versão do PHP chamada de PHP/FI (*Personal Home Pages/Forms Interpreter*). Neste mesmo ano Rasmus liberou o código fonte do PHP para o público permitindo que qualquer um pode-se alterar os scripts. Com isso diversas pessoas tiveram a oportunidade de ajudar no desenvolvimento ajudando a achar bugs e os corrigi-los, ou seja, ajudar a linguagem a melhorar.

**Figura 6: PHP logo**



**Fonte:** <https://pt.wikipedia.org/wiki/PHP#/media/File:PHP-logo.svg>

Em 1997 PHP já era usado em uma grande parte da internet, sendo lançada no mesmo ano a segunda versão do PHP. Em uma pesquisa de Netcaft em 1998 mostrou que por volta de 60.000 domínios relataram ter uma parte em PHP.

Terceira versão do PHP foi escrita em 1997 quando dois jovens de uma Universidade de Israel chamados de Andi Gutmans e Zeev Suraski de Tel Aviv estavam desenvolvendo uma aplicação de eCommerce para um projeto da Universidade, mas a segunda versão do PHP não era suficiente foi aí então que os

começaram a reescrever o interpretador. Discutindo com Rasmus online os três decidiram criar uma nova linguagem na qual foi chamada simplesmente de 'PHP' tornando o significado um acrônimo recursivo - PHP: Hypertext Preprocessor. Deste então Andi Gutmans e Zeev Suraski vem trabalhando em novas versões do PHP chegando até a atual PHP 7.

#### 1.4.2 Firebird

Firebird é um gerenciador de banco de dados de código livre. O Firebird vem do Interbase 6.0 da Borland, a empresa abriu o código fonte sob a licença IPL (*Interbase Public License*). O Interbase era um gerenciador de banco de dados muito bem reconhecido na época e foi bem visto ao aos olhos da comunidade este feito.

O Firebird nasceu devido a iniciativa de alguns dos ex-funcionários da Borland que faziam parte do desenvolvimento do Interbase, e assim nasceu o Firebird.

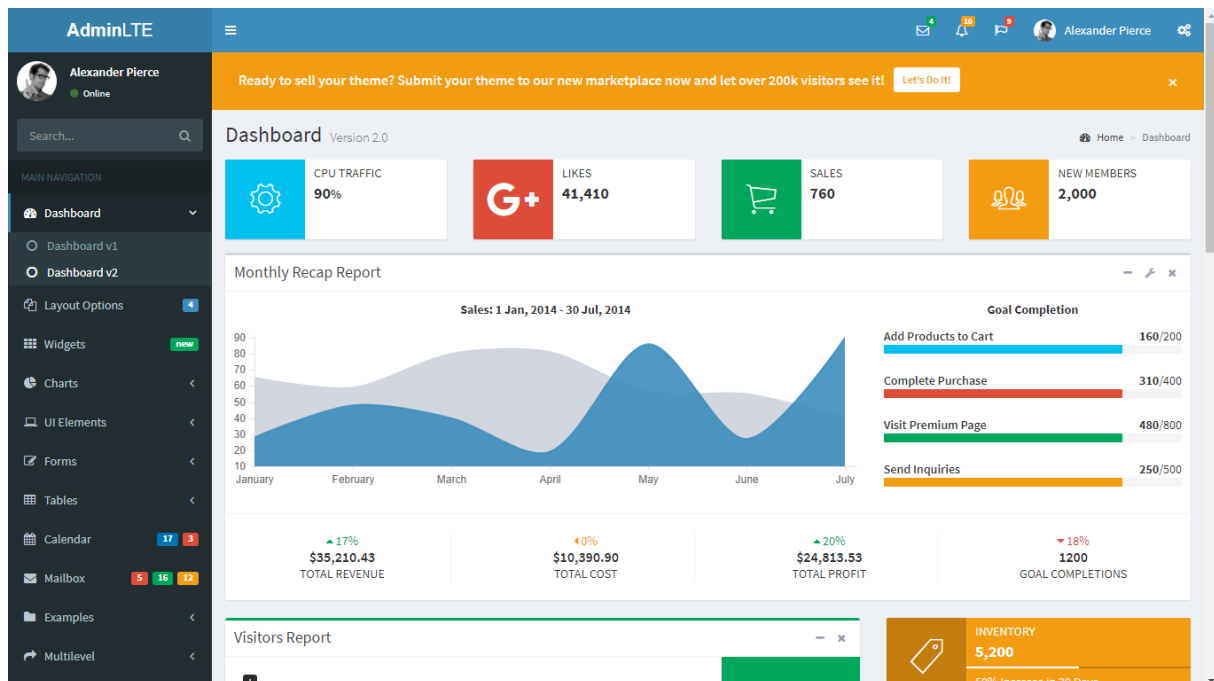
#### 1.4.3 Codeigniter

*Codeigniter* é um pacote de ferramentas para o desenvolvimento de sites em PHP, este pacote de ferramentas permite desenvolver sites com uma velocidade muito grande oferecendo funcionalidades já prontas e com um padrão pré-determinado, sendo assim o desenvolvedor se foca no projeto sem ter a necessidade de desenvolver pequenas funcionalidades que estão sempre presentes no desenvolvimento.

#### 1.4.4 AdminLTE

AdminLTE é um *framework front-end*, ou seja, é um pacote de ferramentas e processos para o desenvolvimento de interfaces para sites utilizando HTML5, CSS3 e JavaScript. AdminLTE utiliza o *bootstrap* e foi desenvolvido por Abdullah Almsaeed. Com o auxílio desta ferramenta o processo de desenvolvimento do site se torna muito mais rápido, além de possuir um padrão já conhecido por outros desenvolvedores tornando uma possível atualização no futuro muito mais amigável.

Figura 7: framework adminLTE



Fonte: <https://adminlte.io/themes/AdminLTE/index2.html>

## 2 LEVANTAMENTO DE REQUISITOS

Os requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007, p. 79), para o desenvolvimento de um sistema integrado de gerenciamento escolar, será documentado os requisitos funcionais e não funcionais.

### 2.1 REQUISITOS FUNCIONAIS

Requisitos funcionais são todas as funcionalidades que o software deverá ter:

- Somente o administrador do sistema poderá recuperar a senha
- O sistema deve ter um limite de tentativas de 5 vezes após este limite a conta deve ser bloqueada
- O cadastro no sistema somente pode ser feito pelo administrador
- O administrador poderá registrar, atualizar ou desativar outros administradores
- O administrador poderá registrar, atualizar ou desativar professores

- O administrador poderá registrar, atualizar ou desativar alunos
- O administrador poderá registrar, atualizar ou desativar matérias
- O administrador poderá registrar, atualizar ou desativar turmas
- O administrador poderá registrar, atualizar ou desativar cursos
- O administrador poderá gerenciar logs
- O administrador poderá enviar mensagens para outras pessoas no sistema
- O administrador poderá registrar, atualizar ou desativar atividades de matérias
- O administrador poderá registrar, atualizar ou desativar avisos no sistema
- O professor poderá registrar, atualizar ou desativar atividades de matérias
- O professor poderá registrar, atualizar ou deletar notas de alunos nas matérias
- O professor poderá enviar mensagens para outras pessoas no sistema
- O professor poderá inserir informações na matéria na qual está responsável
- O aluno poderá ver suas notas na matéria na qual está matriculado
- O aluno poderá ver sua grade de matérias referente ao seu curso
- O aluno poderá inserir e ver atividades da matéria no qual cursa

## 2.2 REQUISITOS NÃO FUNCIONAIS

Requisitos não funcionais são as qualidades e aspectos no qual o sistema deverá ter:

### 2.2.1 Portabilidade

O sistema será portátil, como é executado através de um navegador de web não importa qual é o sistema operacional, sendo assim um sistema que poderá ser acessado de diversas plataformas diferentes.

### 2.2.2 Segurança

Nenhuma informação poderá ser acessada sem autorização, para tal finalidade o sistema somente poderá ser acessado através de dados previamente cadastrados, ou seja, um sistema de verificação deve ser implementado.

### 2.2.3 Usabilidade

O sistema deve ser intuitivo, ícones de funcionalidades devem deixar ainda mais claro sua função, também deverá ser de rápido aprendizado sendo assim seguindo padrões de menu.

### 2.2.4 Documentação

O sistema será todo documentado para o aprendizado dos usuários, esta documentação estará disponível publicamente, além da documentação técnica do sistema.

## 3 FERRAMENTAS DE DOCUMENTAÇÃO DO SISTEMA

A documentação do sistema é de grande importância, pois as regras de negócio podem mudar e o sistema deve ser capaz de acompanhar tais mudanças, ou seja, o sistema sempre está em constantes atualizações, para que não fiquemos dependentes apenas de nossas memórias, já que facilmente podemos esquecer, é necessário documentar o sistema, para tal objetivo existem diversas ferramentas.

Neste projeto será utilizado UML para documentar a organização do sistema, os diagramas utilizados serão os mais recomendados sendo eles, diagrama de caso de uso, diagrama de sequência, diagrama de classes e diagrama de atividades através do software ASTAH.

### 3.1 UML

A UML – *Unified Modeling Language* ou Linguagem de Modelagem Unificada – é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos (GUEDES, 2009, p. 19).

Nesta linguagem existem diversos diagramas com o objetivo de esclarecer e simplificar o entendimento do sistema, mas dentre os mais importantes e que serão

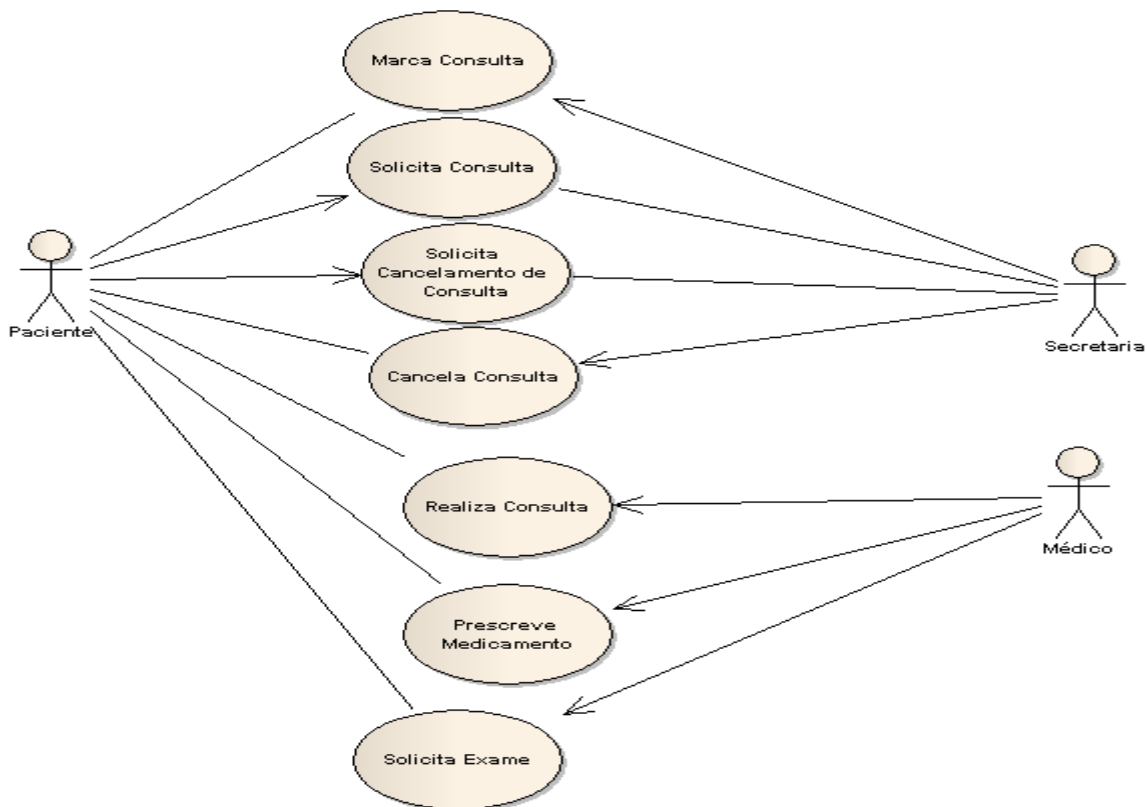
usados neste projeto estão, o diagrama de caso de uso, classes, atividade e diagrama de seqüência.

### 3.1.1 Caso de uso

Segundo Gilleanes T.A. Guedes (2009), “ O diagrama de casos de uso é o diagrama mais geral e informal da UML, utilizado normalmente nas fases levantamento e análise de requisitos do sistema [...]”.

O diagrama tem uma linguagem simples para que mesmo aqueles sem conhecimento técnico possam entender, o diagrama procura identificar os atores (usuários ou mesmo outros softwares) e as funcionalidades relacionadas com eles.

**Figura 8: Exemplo do Diagrama de Caso de Uso**



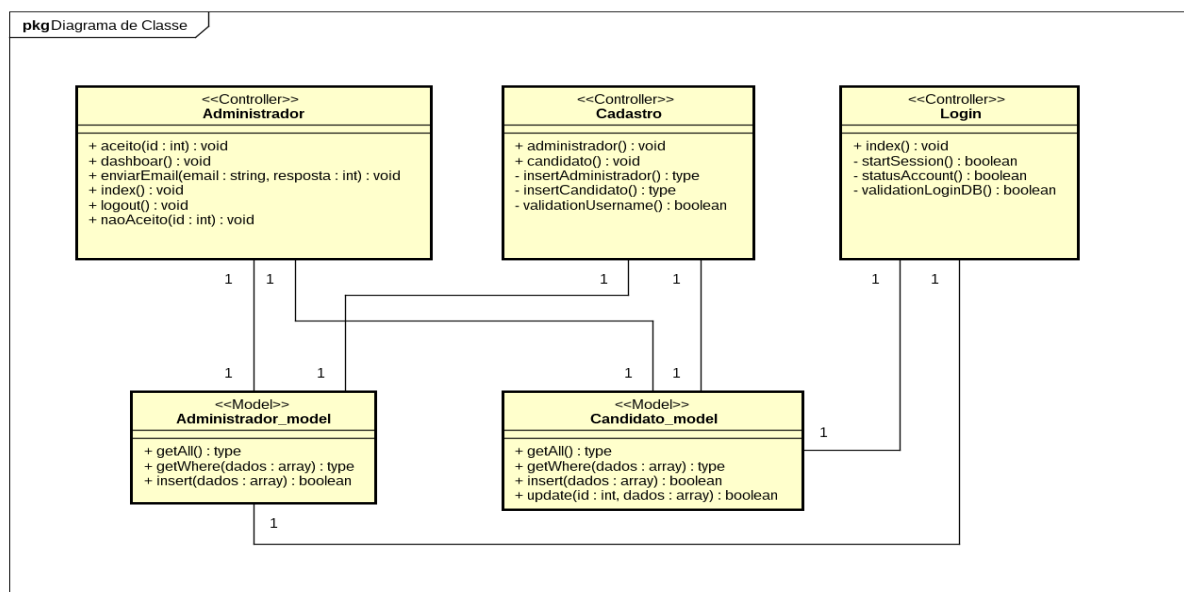
Fonte: <http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>



### 3.1.2 Classe

O diagrama de classes é sem dúvida o mais utilizado, ele define a estrutura das classes no sistema juntamente com seus métodos e atributos e demonstra como as classes trocam informações entre si.

**Figura 9: Exemplo de Diagrama de Classes**



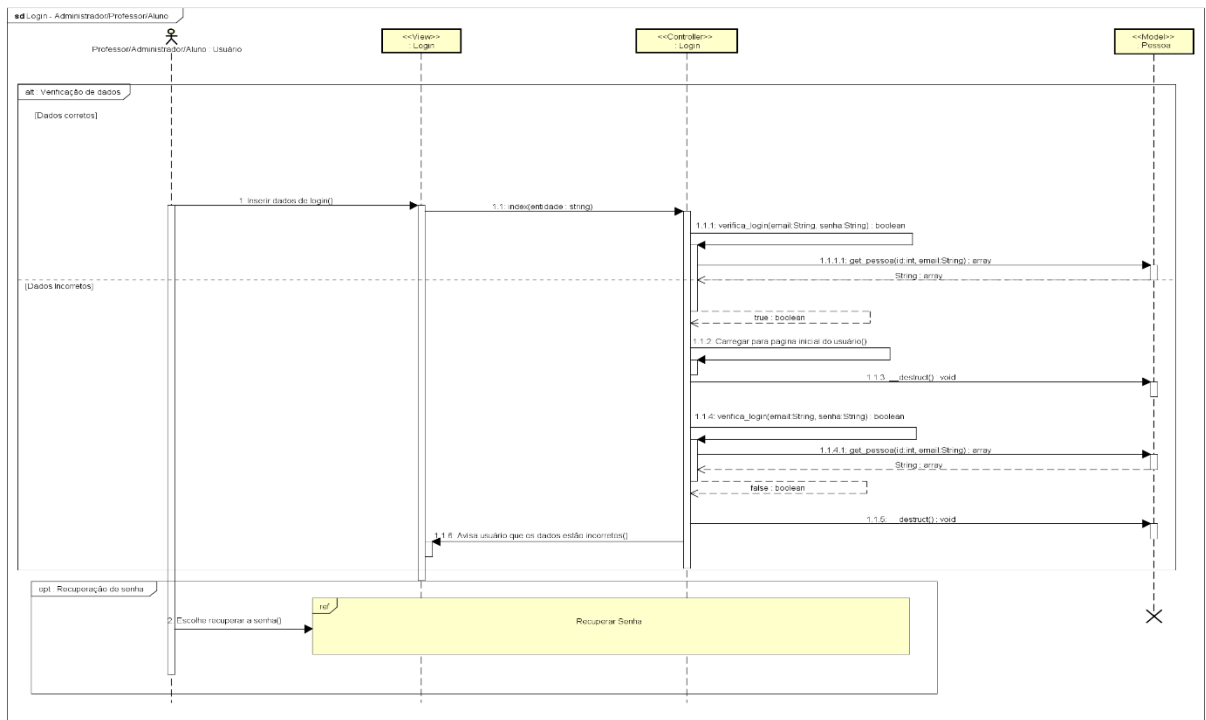
powered by Astah

**Fonte: A autoria do Autor.**

### 3.1.3 Sequência

O diagrama de sequência tem como objetivo detalhar como o sistema troca informações internamente entre objetos, este diagrama baseia-se no diagrama de caso de uso e os objetos nele demonstrados estão presentes no diagrama de classes.

Figura 10: Exemplo do Diagrama de Sequência

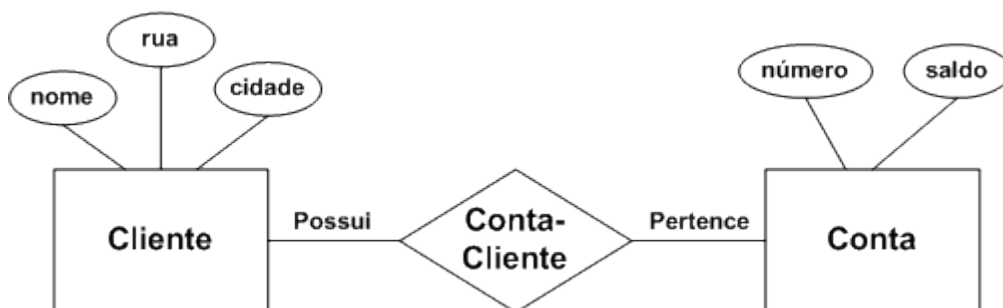


Fonte: A autoria do Autor.

### 3.2 MODELO ENTIDADE E RELACIONAMENTO

Como o próprio nome diz ele tem o objetivo de demonstrar as entidades do banco de dados e seus relacionamentos, este diagrama pode ser mostrado para pessoa com pouco conhecimento técnico, já que não tem informações técnicas.

Figura 11: Exemplo do Modelo de Entidade e Relacionamento

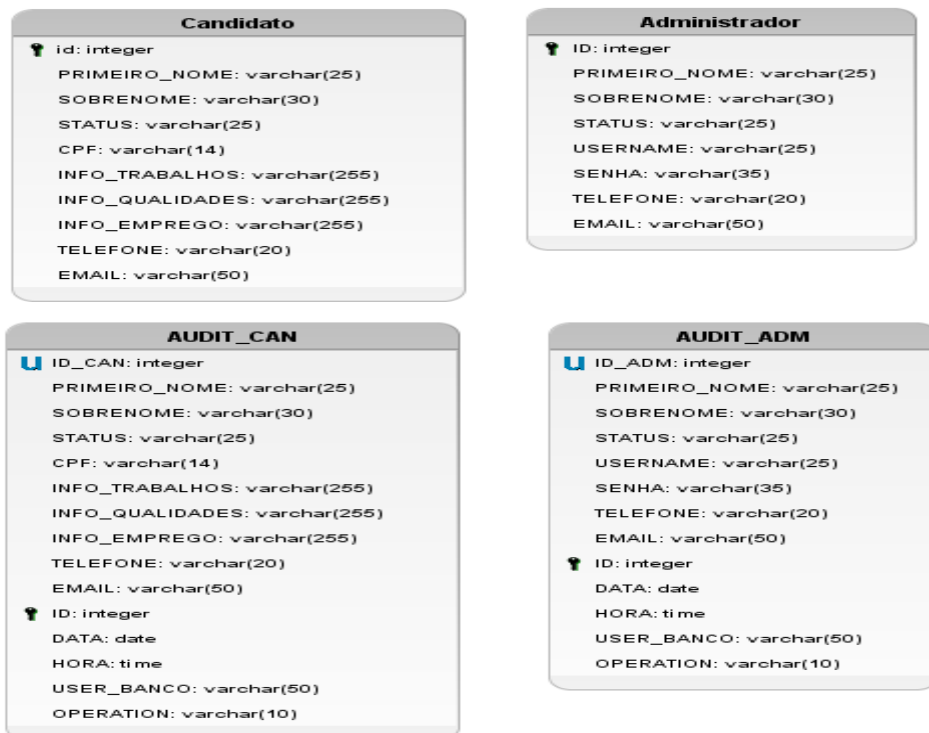


Fonte: <https://www.ime.usp.br/~andrrs/aulas/bd2005-1/aula6.html>

### 3.3 DIAGRAMA ENTIDADE E RELACIONAMENTO

Este diagrama é a base para o desenvolvimento do banco de dados tem como objetivo demonstrar todas as tabelas do banco de dados, seus atributos, tipos e seus relacionamentos.

Figura 12: Exemplo do Diagrama Entidade e Relacionamento



Fonte: A autoria do Autor.

## 4 ESTUDO DE CASO

Iremos desenvolver uma aplicação web para o auxiliar no desenvolvimento escolar, como cada escola tem suas próprias regras de negócio, nosso objetivo é desenvolver um software que seja genérico, sendo assim uma base para qualquer escola. As funções principais são o gerenciamento de administradores, professores, alunos, cursos, matérias e turmas além de um chat e relatórios, outras funções também serão projetadas e apresentadas na documentação, mas como se trata de um sistema integrado, ou seja, um conjunto de programas, não será possível desenvolver tudo em um espaço curto de tempo.

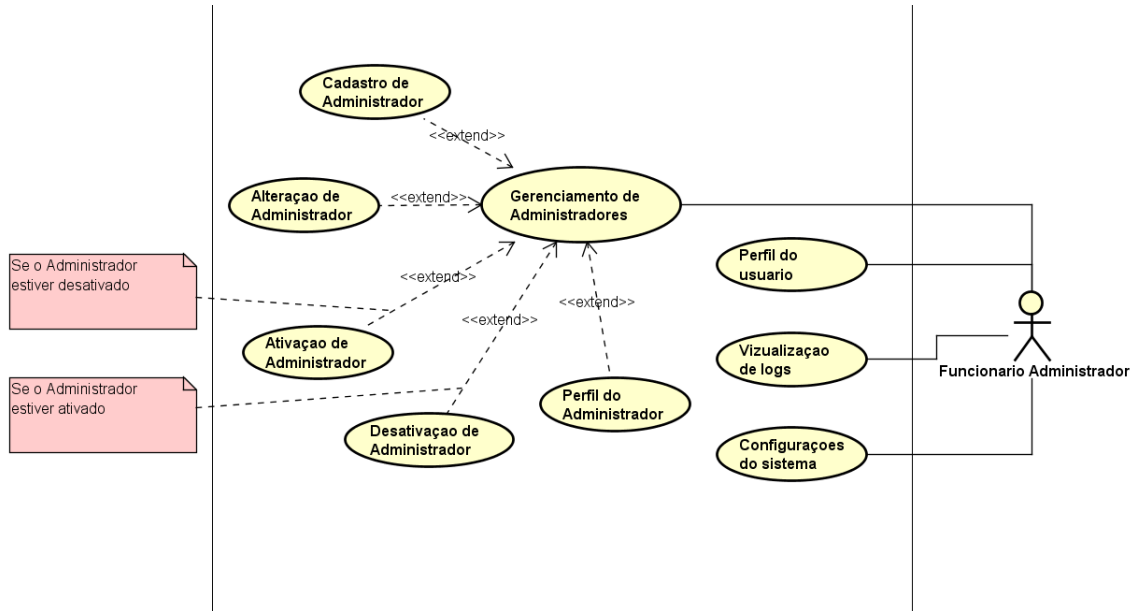
A organização do sistema será demonstrada com diagramas UML criados no software ASTAH em sua versão gratuita. A parte de banco de dados será com o modelo de entidade e relacionamento e o diagrama de entidade e relacionamento.

### 4.1 CASO DE USO

A seguir será apresentado o diagrama de caso, para uma melhor visualização do sistema o diagrama de caso de uso foi dividido em três partes nas quais são administrador, aluno e professor.

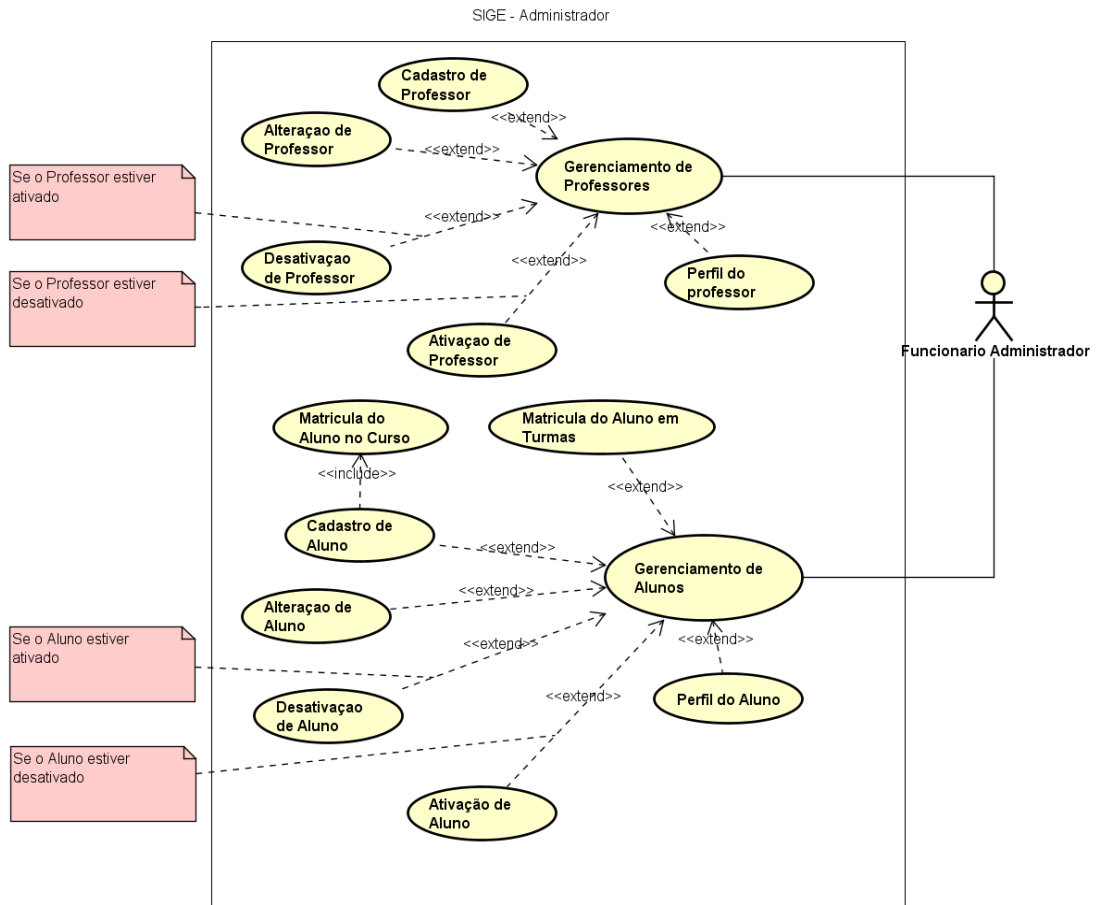
## 4.1.1 Administrador

Figura 13: Diagrama de caso de uso administrador 2



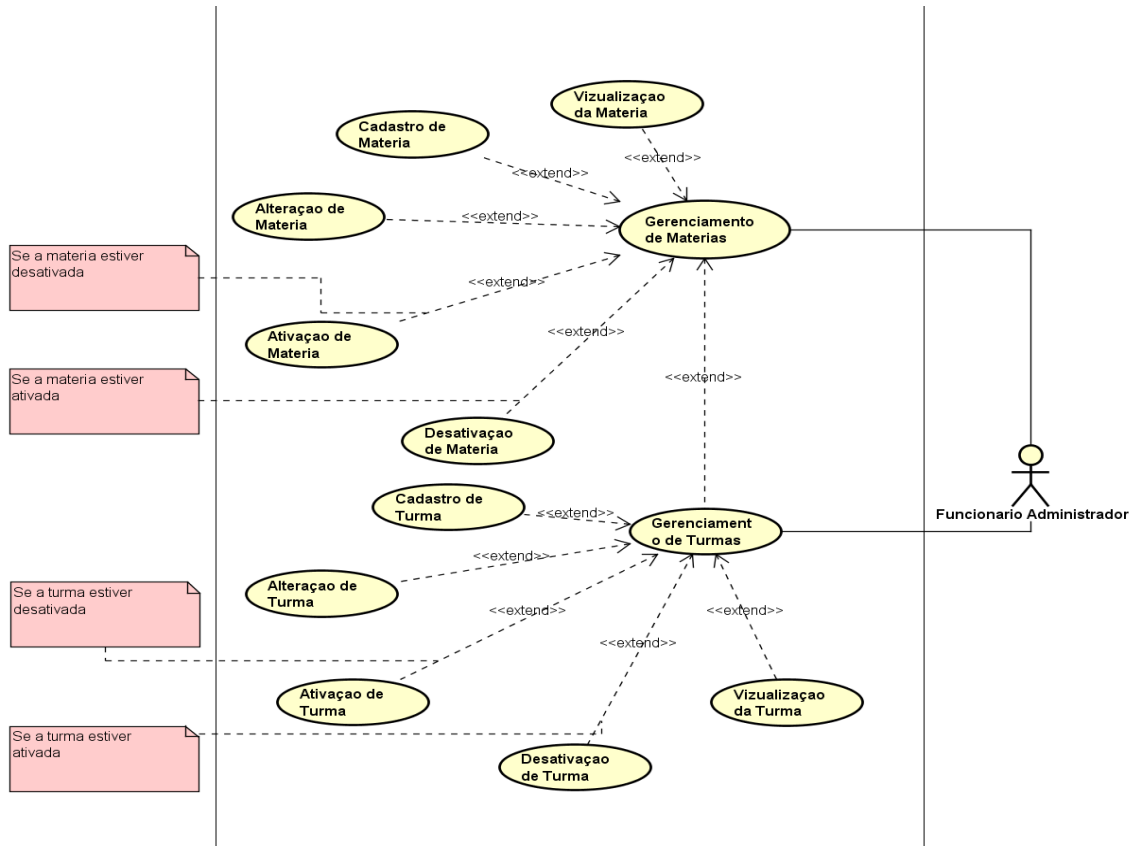
Fonte: Aatoria do Autor.

Figura 14: Diagrama de caso de uso administrador 1



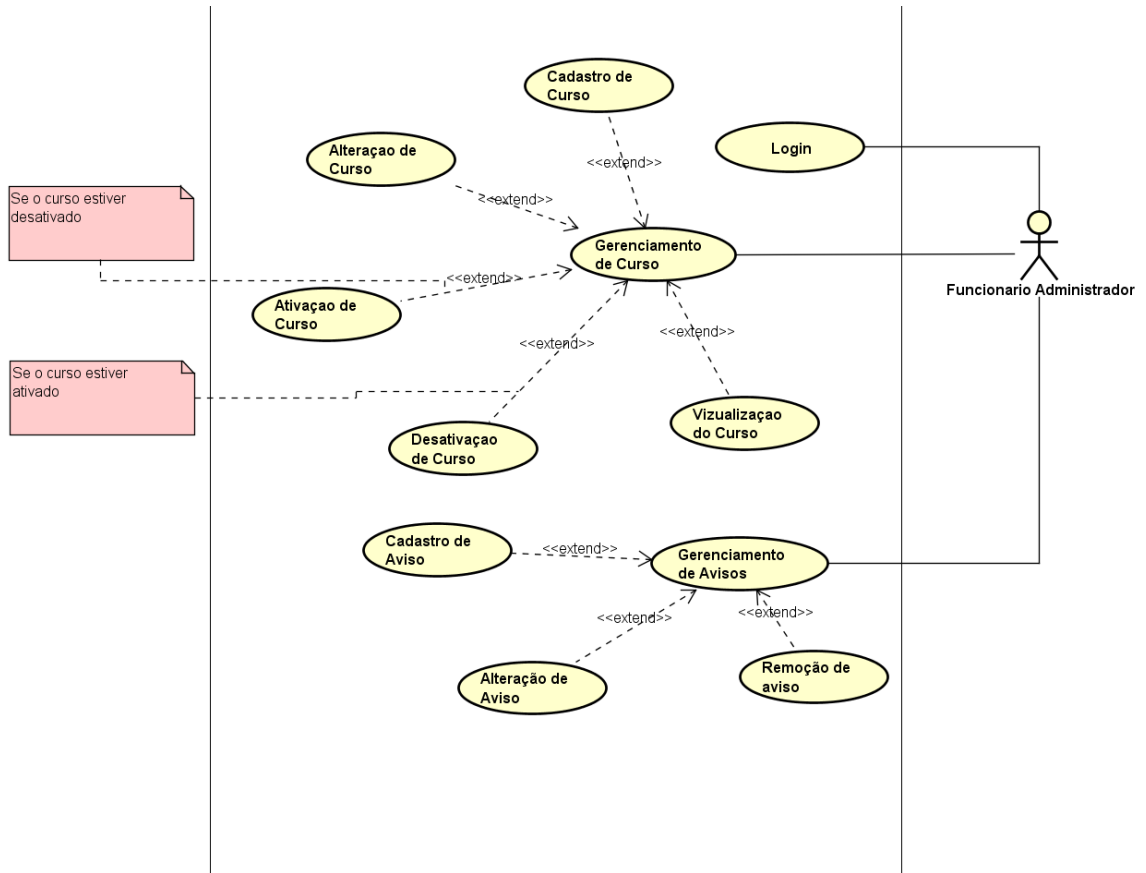
Fonte: A autoria do Autor.

Figura 15: Diagrama de caso de uso administrador 3



Fonte: Aatoria do Autor.

Figura 16: Diagrama de caso de uso administrador 4

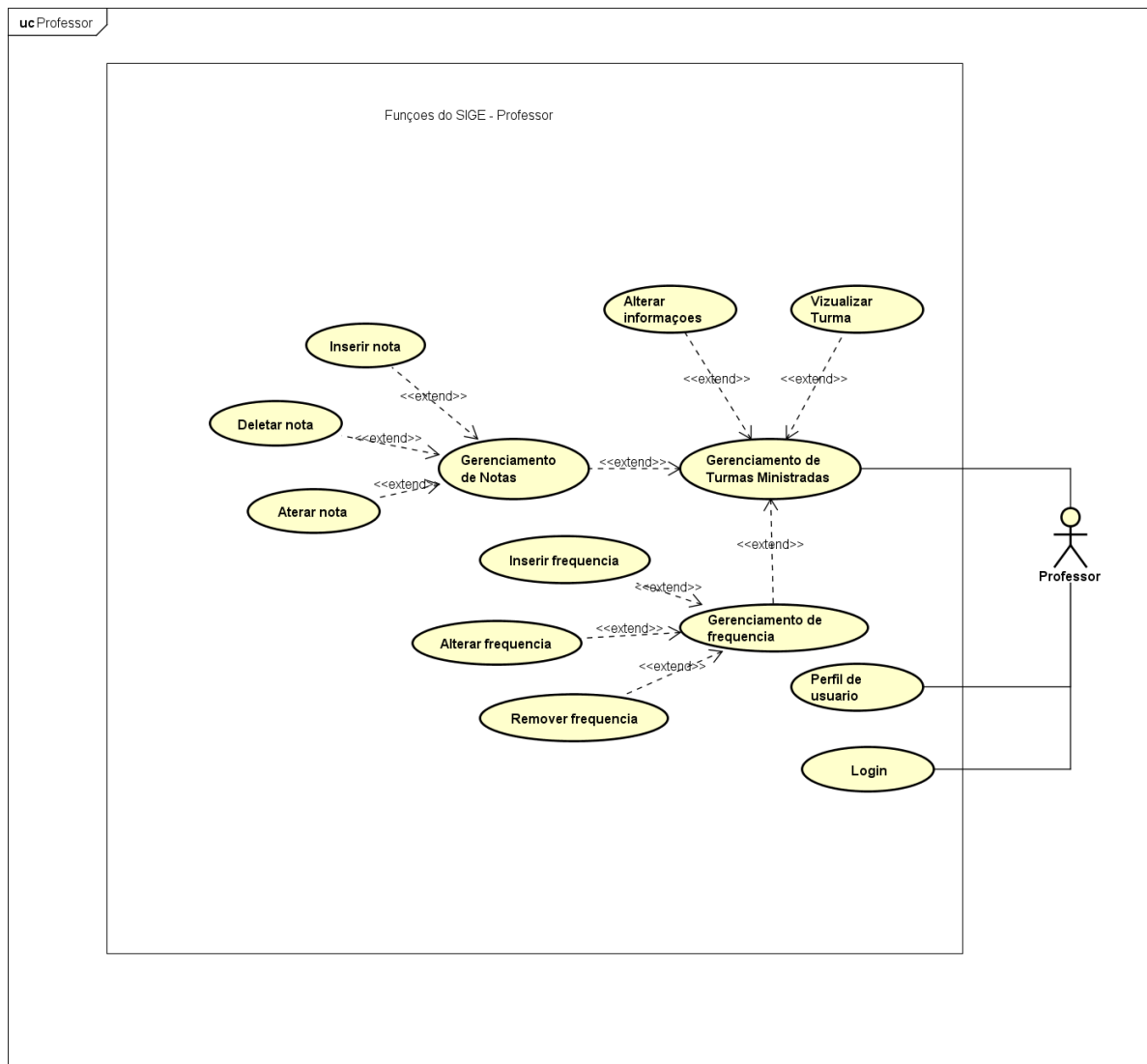


Fonte: Autoria do Autor.



## 4.1.2 Professor

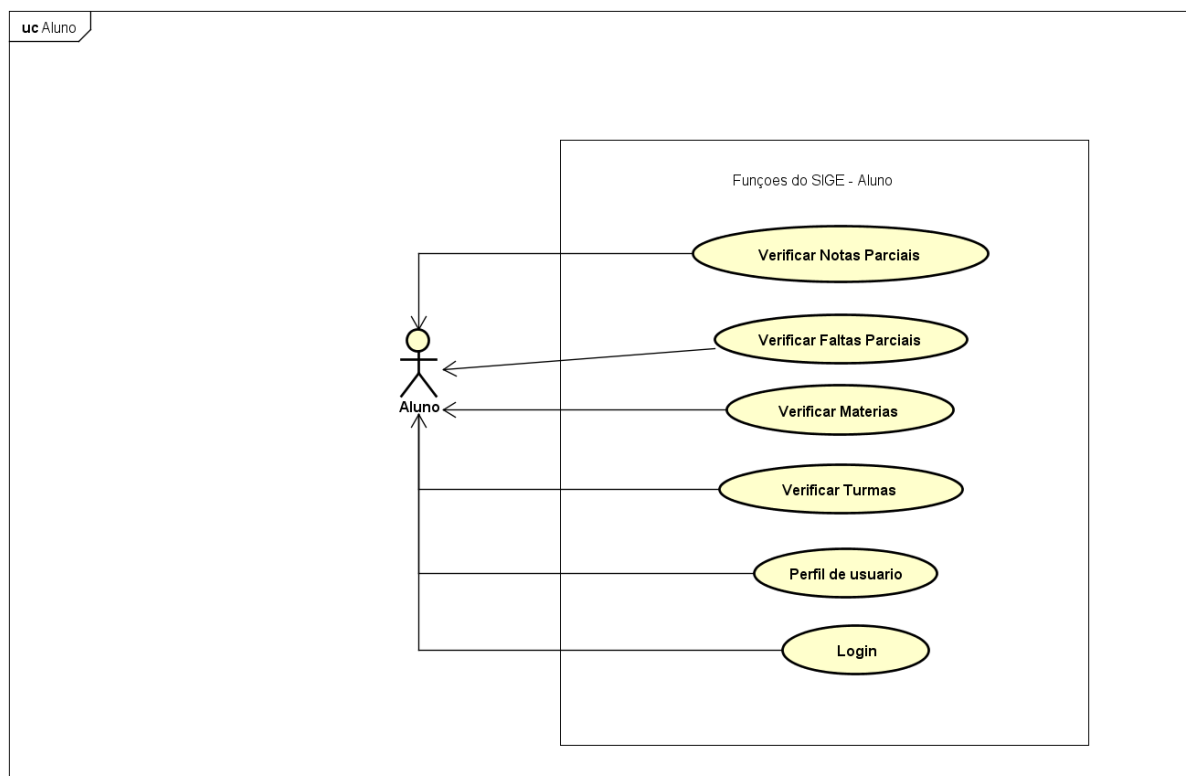
Figura 17: Diagrama de caso de uso professor



Fonte: Aatoria do Autor.

### 4.1.3 Aluno

Figura 18: Diagrama de caso de uso aluno



powered by Astah

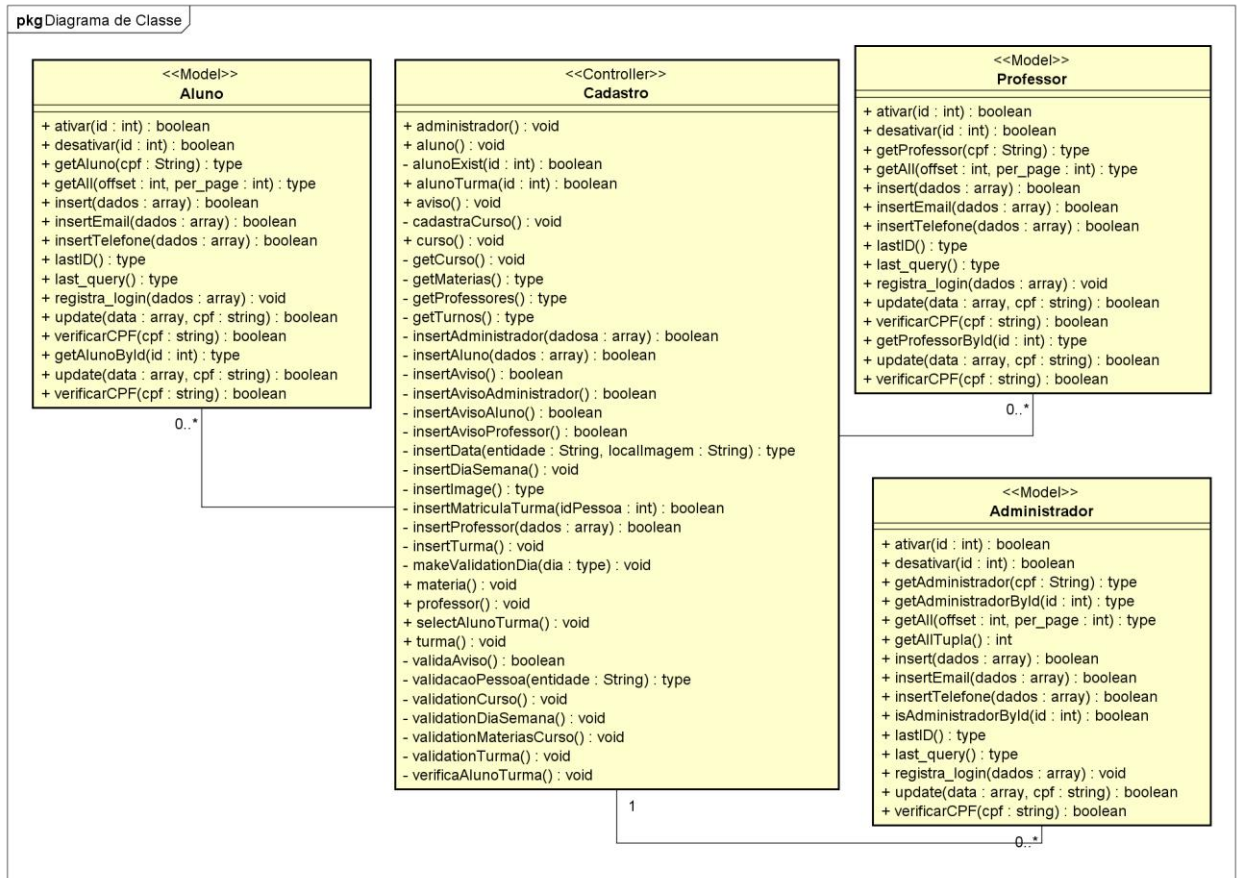
Fonte: Autoria do Autor.

## 4.2 DIAGRAMA DE CLASSES

Nessa seção será apresentado todos os diagramas de classes do sistema.

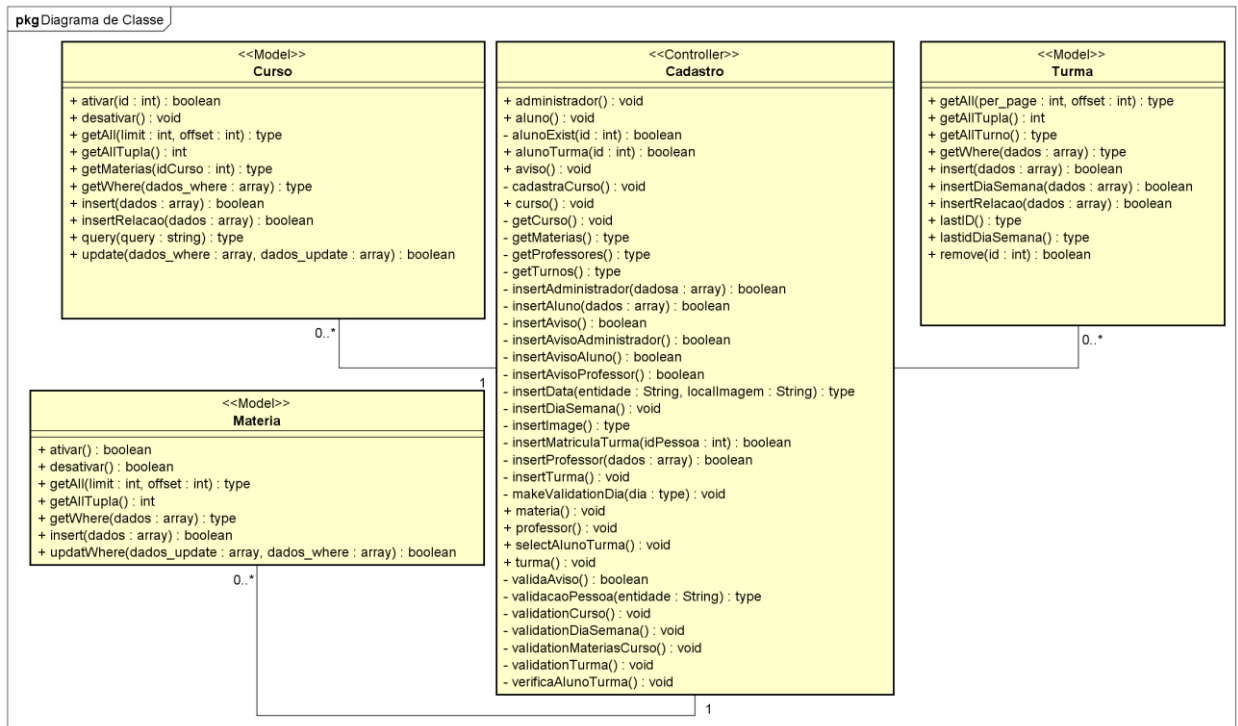
## 4.2.1 Controlador cadastro

Figura 19: Diagrama de classe cadastro



Fonte: Autoria do Autor.

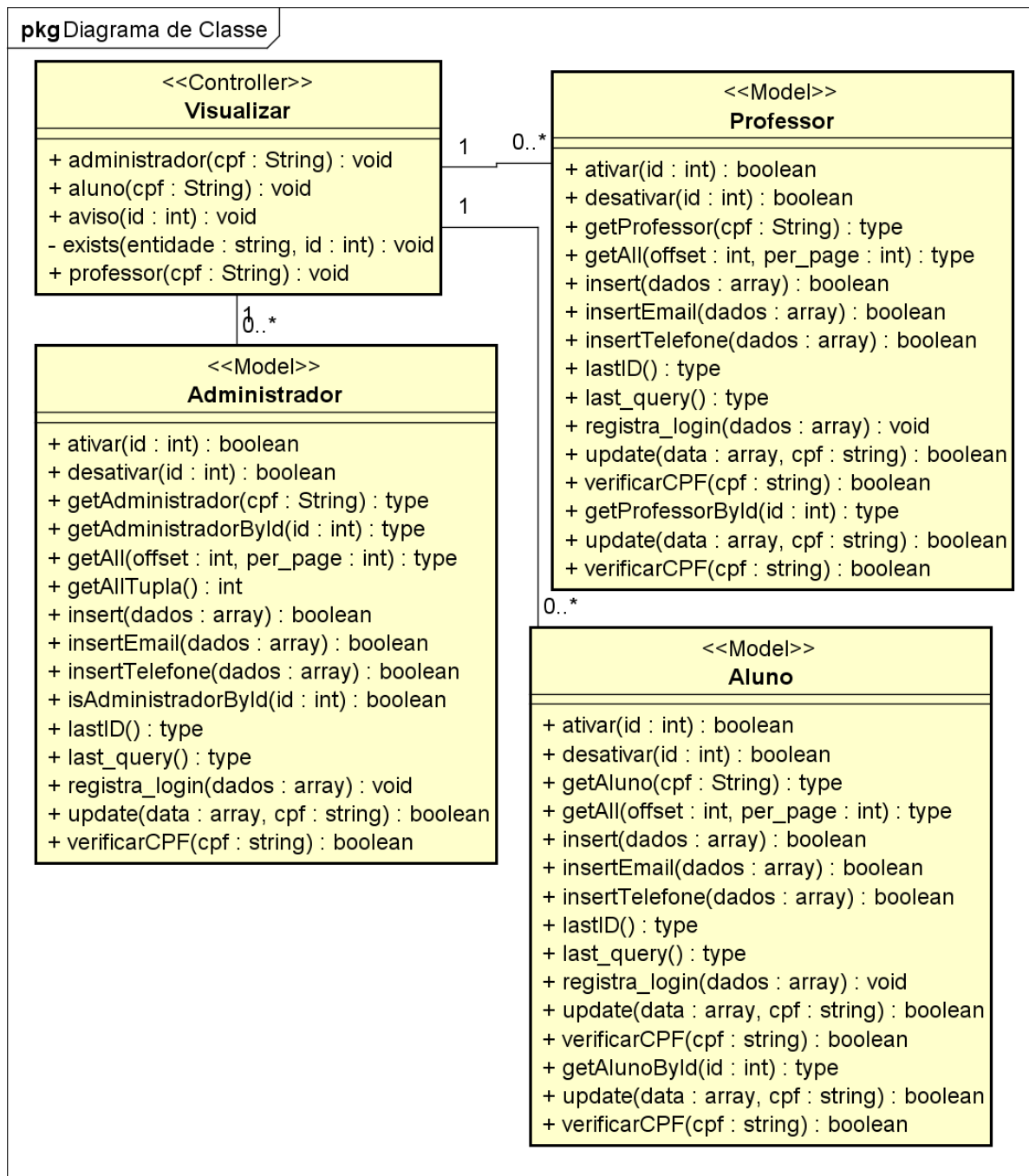
Figura 20: Diagrama de classe cadastro 2



Fonte: Aatoria do Autor.

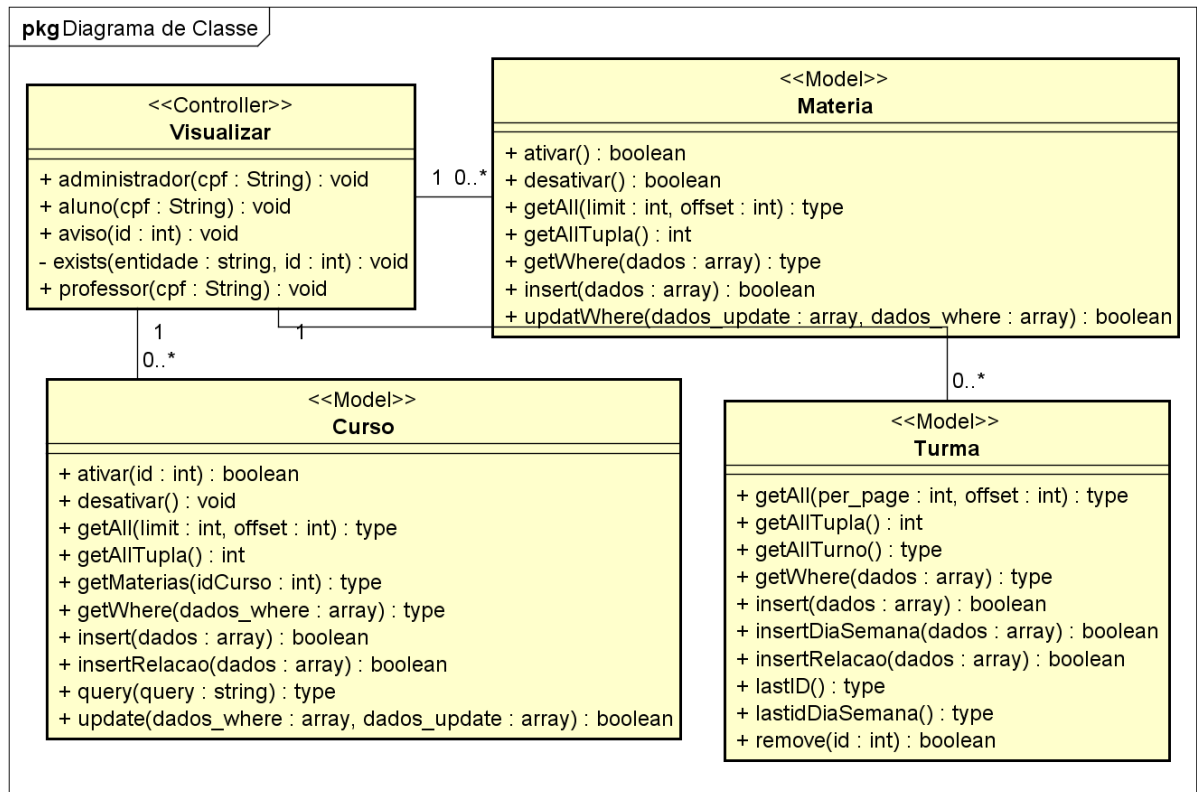
## 4.2.2 Controlador visualizar

Figura 21: Diagrama de classe visualizar



Fonte: Aatoria do Autor.

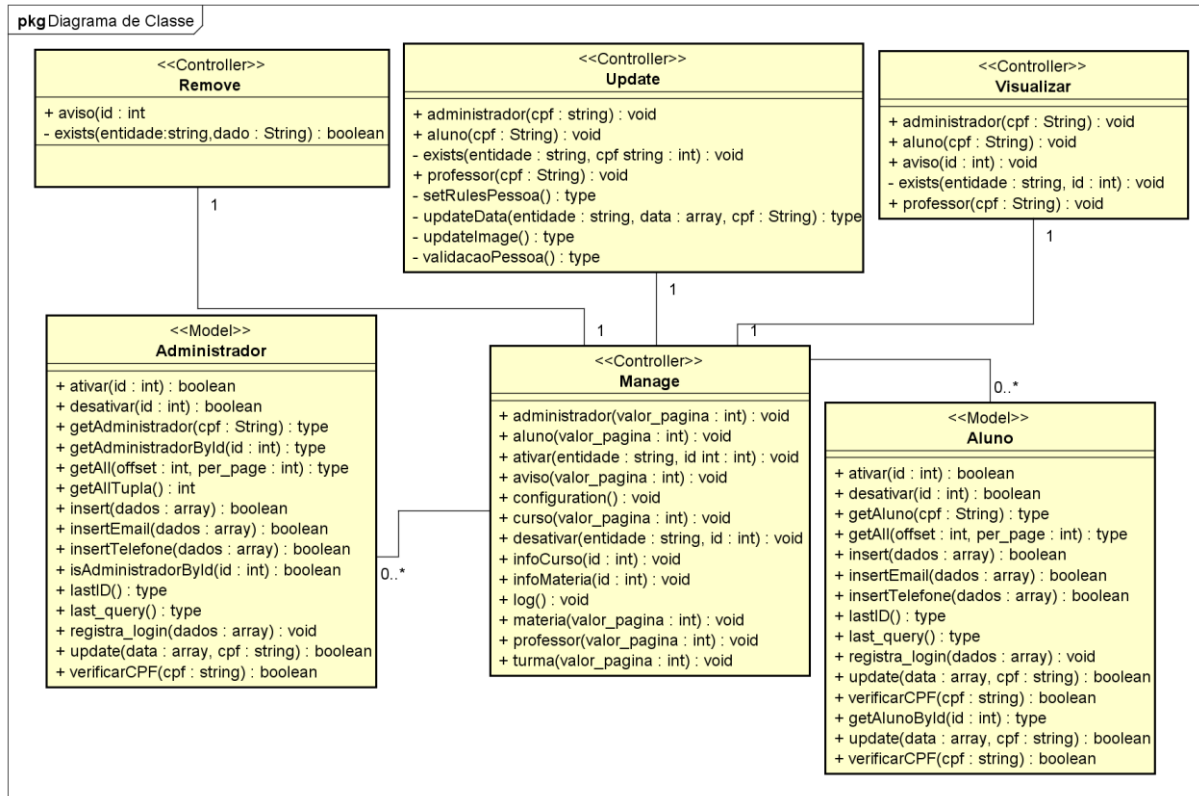
Figura 22: Diagrama de classe visualizar 2



Fonte: Aatoria do Autor.

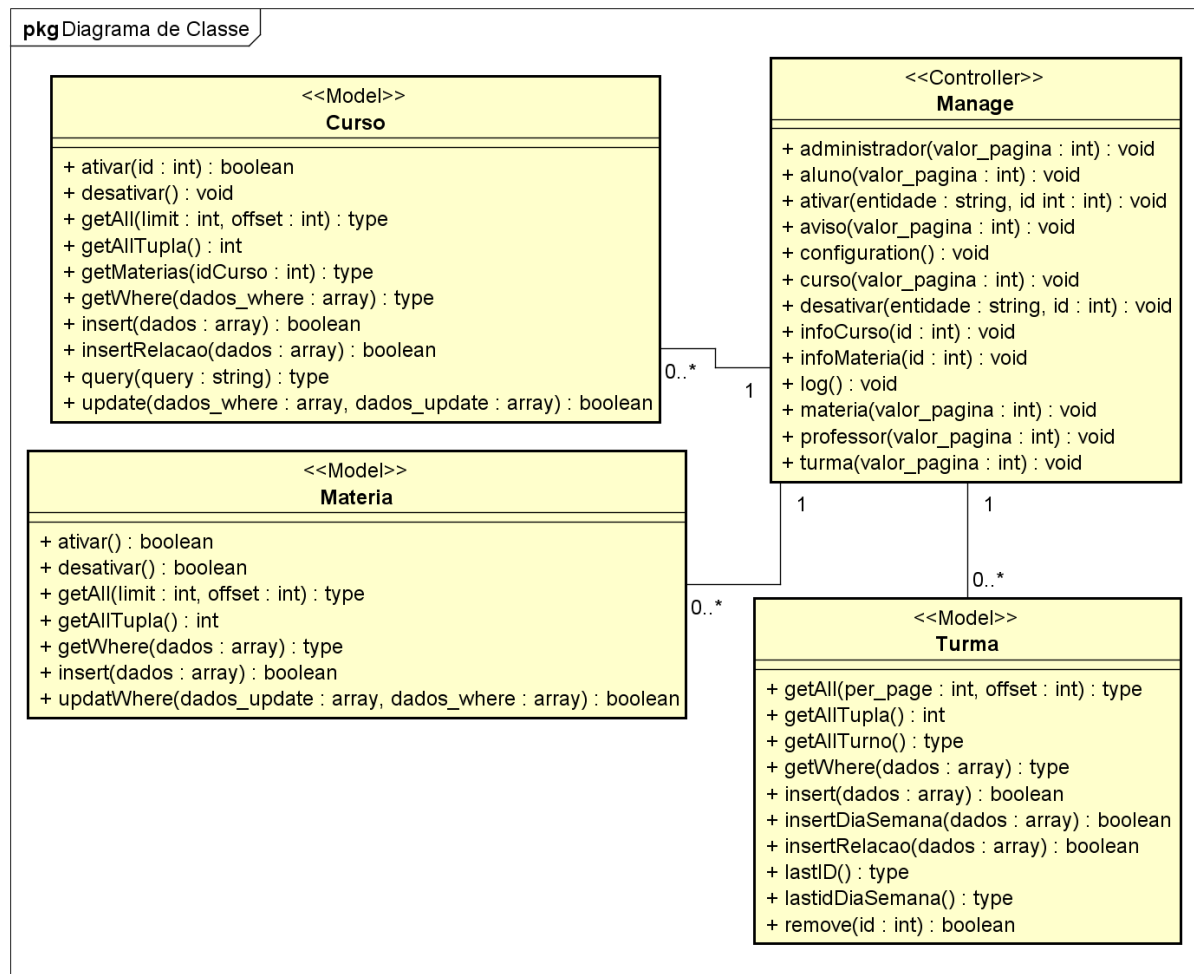
## 4.2.3 Controlador manage

Figura 23: Diagrama de classes manage



Fonte: Autoria do Autor.

Figura 24: Diagrama de classe manage 2

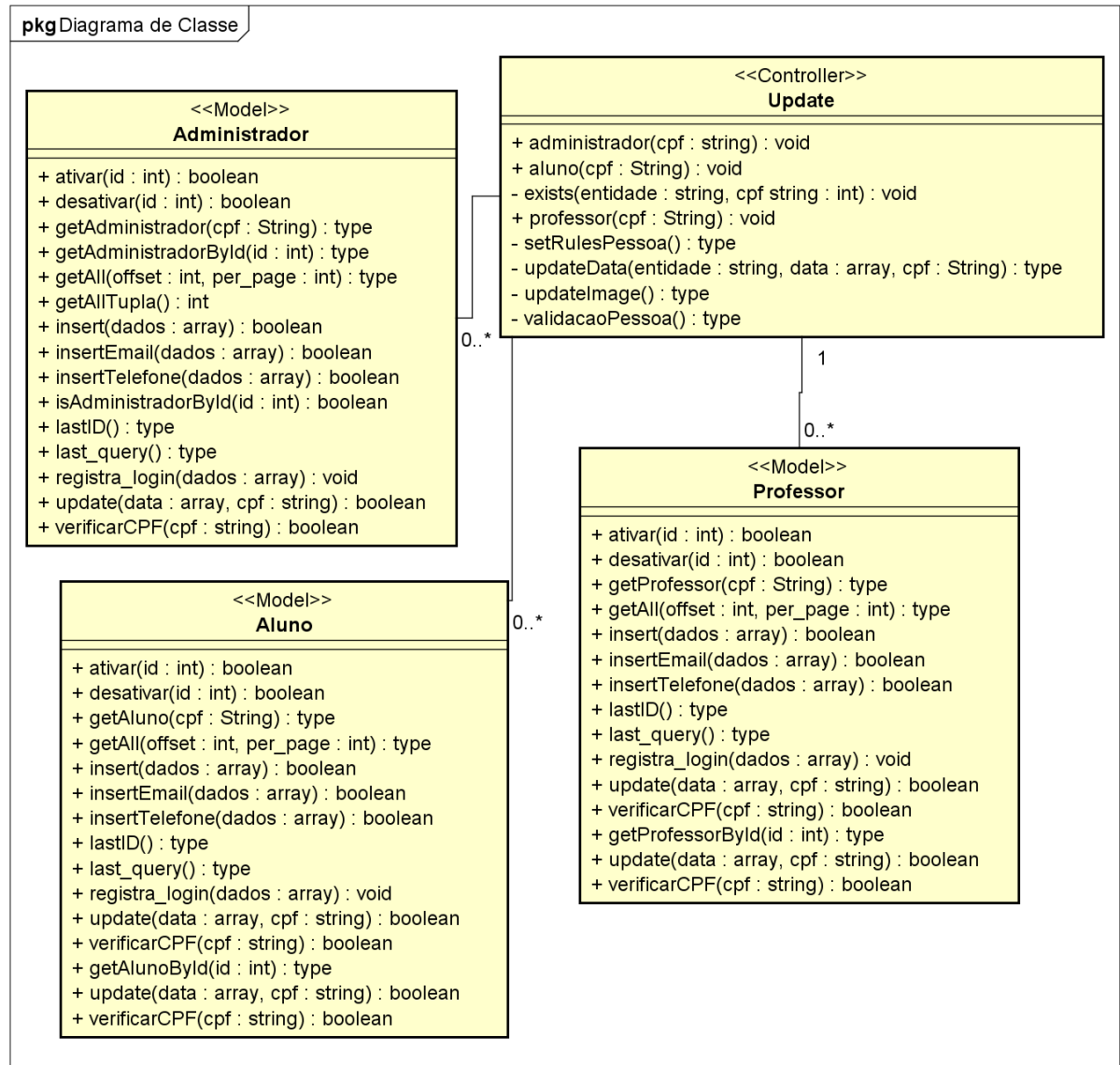


Fonte: Aatoria do Autor.



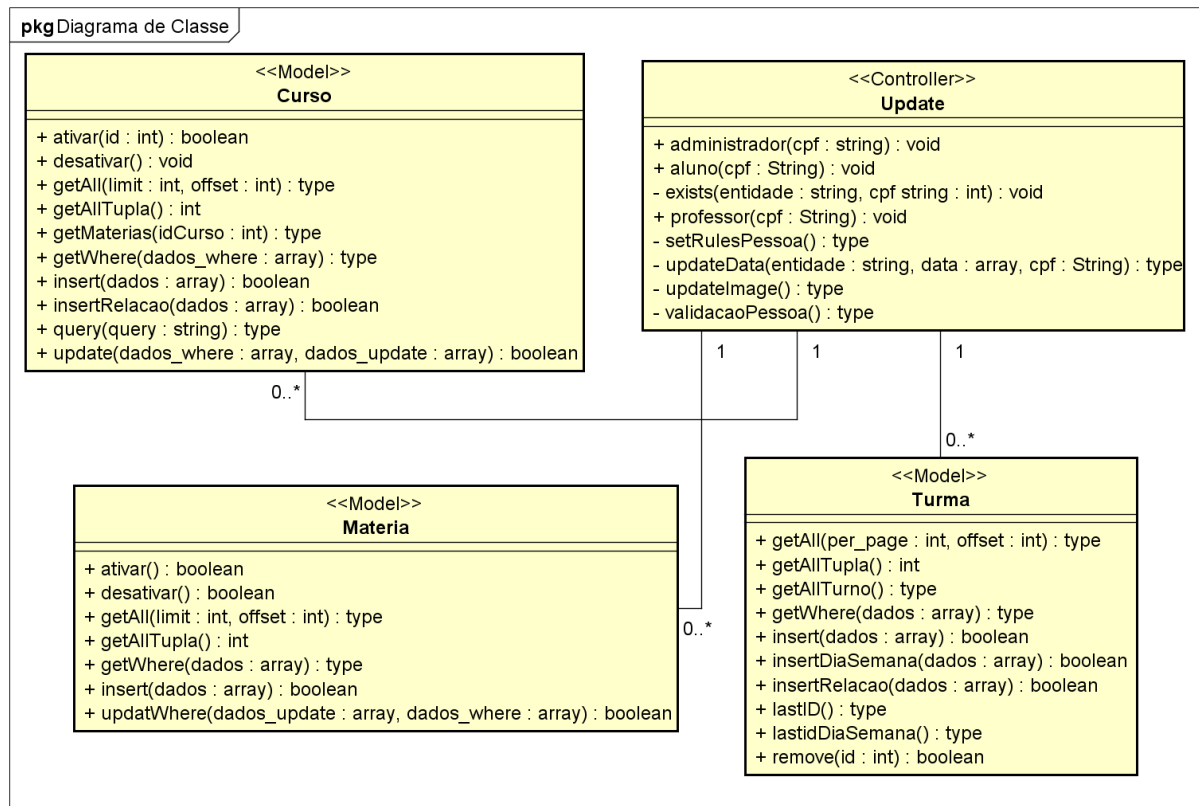
## 4.2.4 Controlador update

Figura 25: Diagrama de classe update



Fonte: Autoria do Autor.

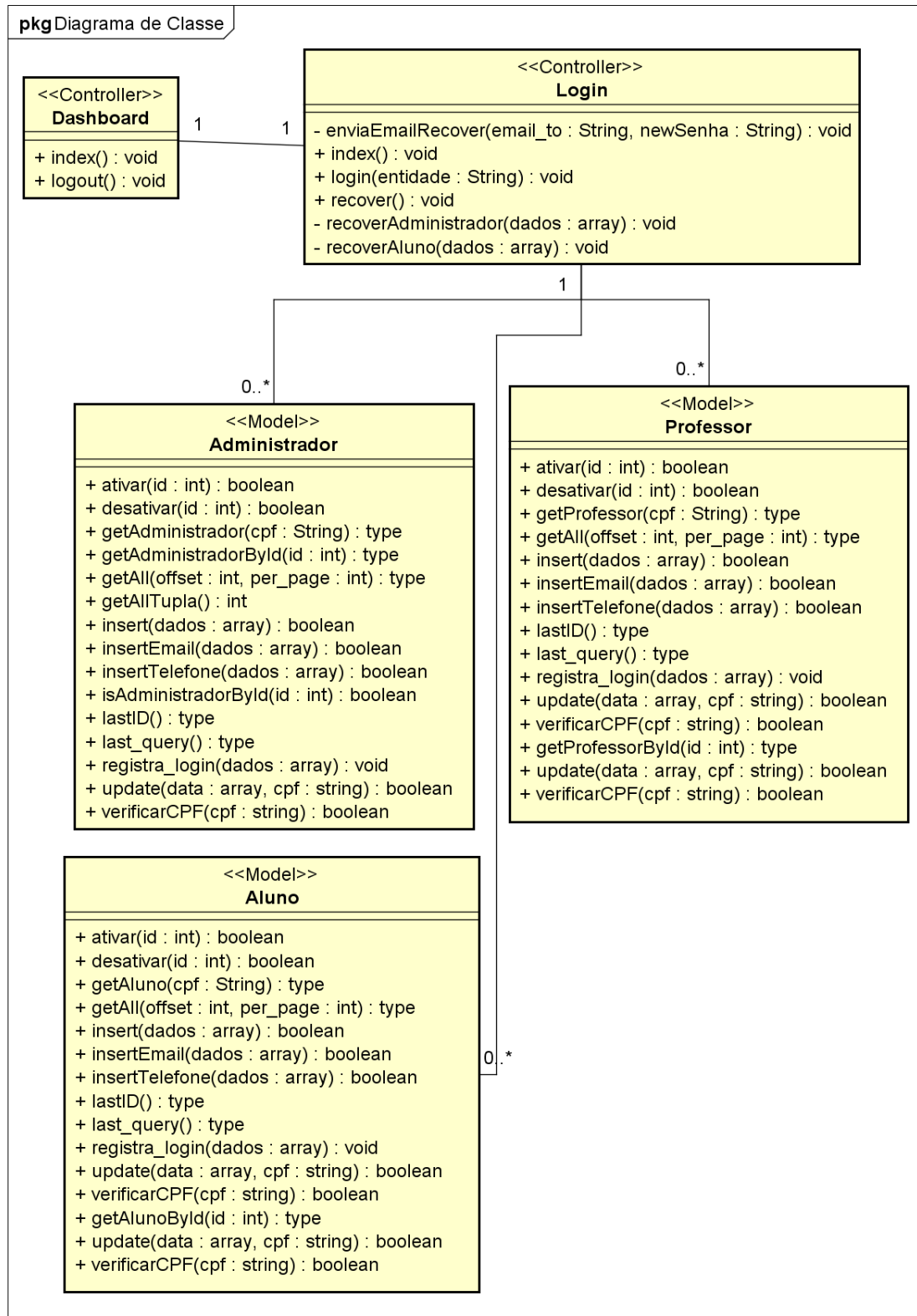
Figura 26: Diagrama de classe update 2



Fonte: Autoria do Autor.

## 4.2.5 Controlador login

Figura 27: Diagrama de classe login



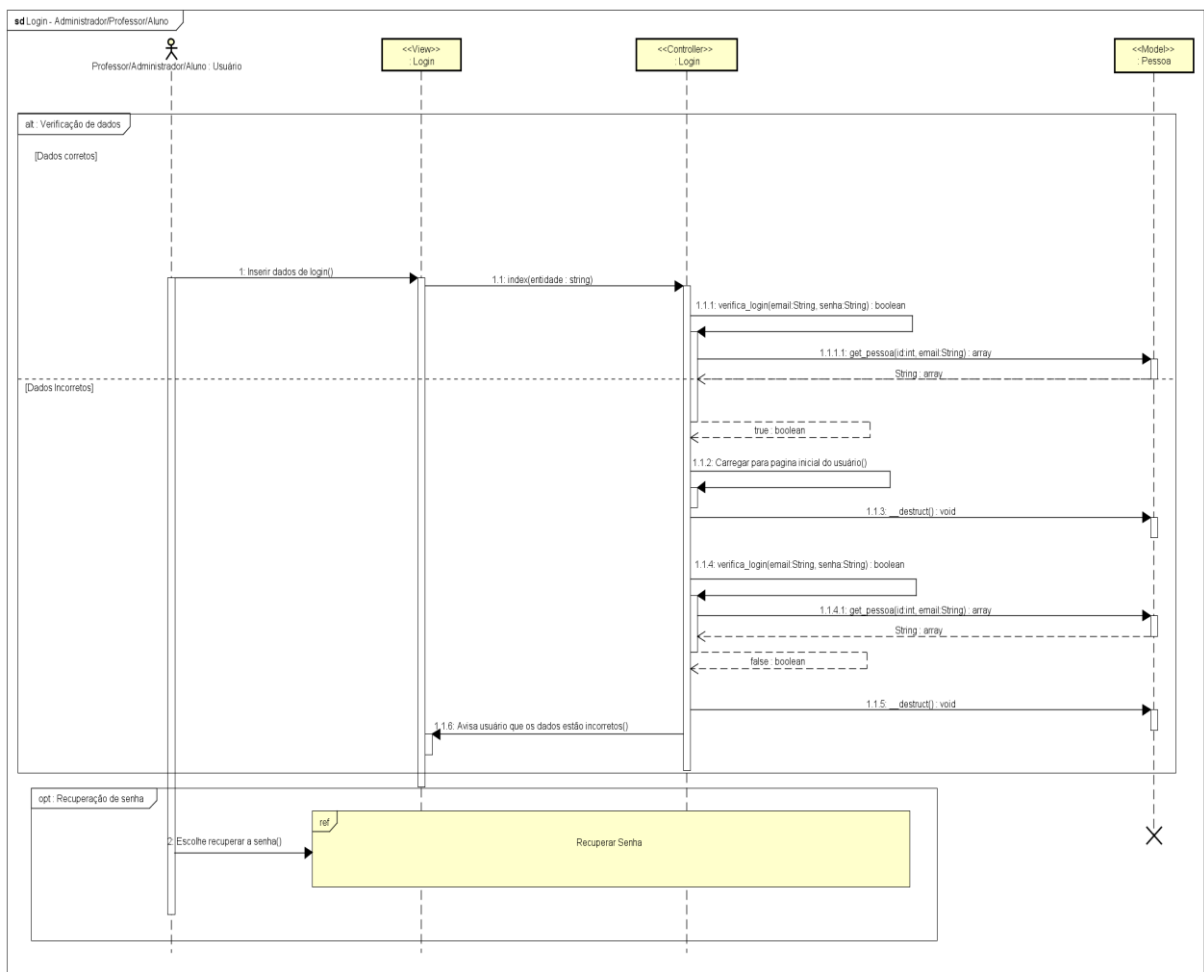
Fonte: Autoria do Autor.

### 4.3 DIAGRAMA DE SEQUÊNCIA

A seguir é apresentado os diagramas de sequência, tendo objetivo de explicar a arquitetura do sistema, porém como o sistema é muito grande será apresentado apenas as funcionalidades de maior relevância.

#### 4.3.1 Login

**Figura 28: Diagrama de sequência login**

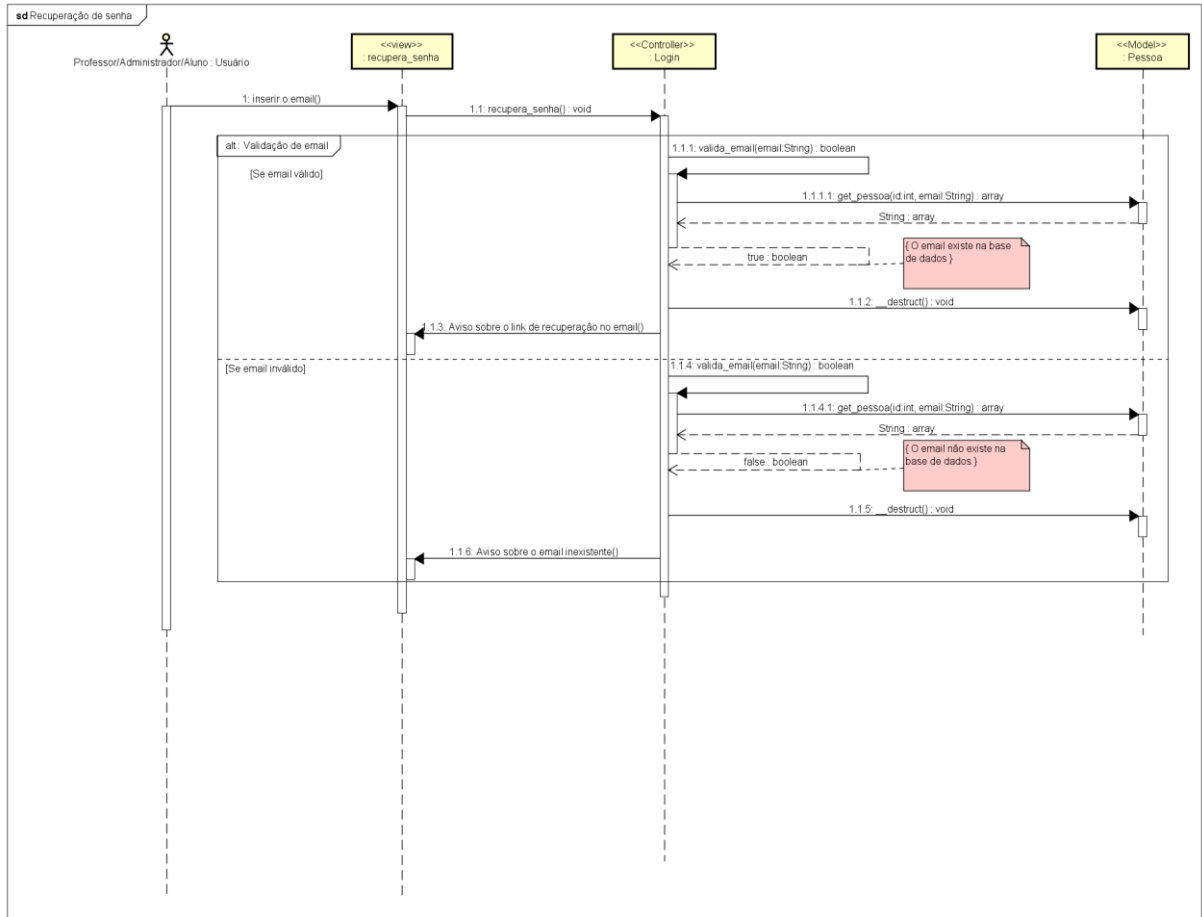


powered by Astah

Fonte: A autoria do Autor.

### 4.3.2 Recuperação de senha

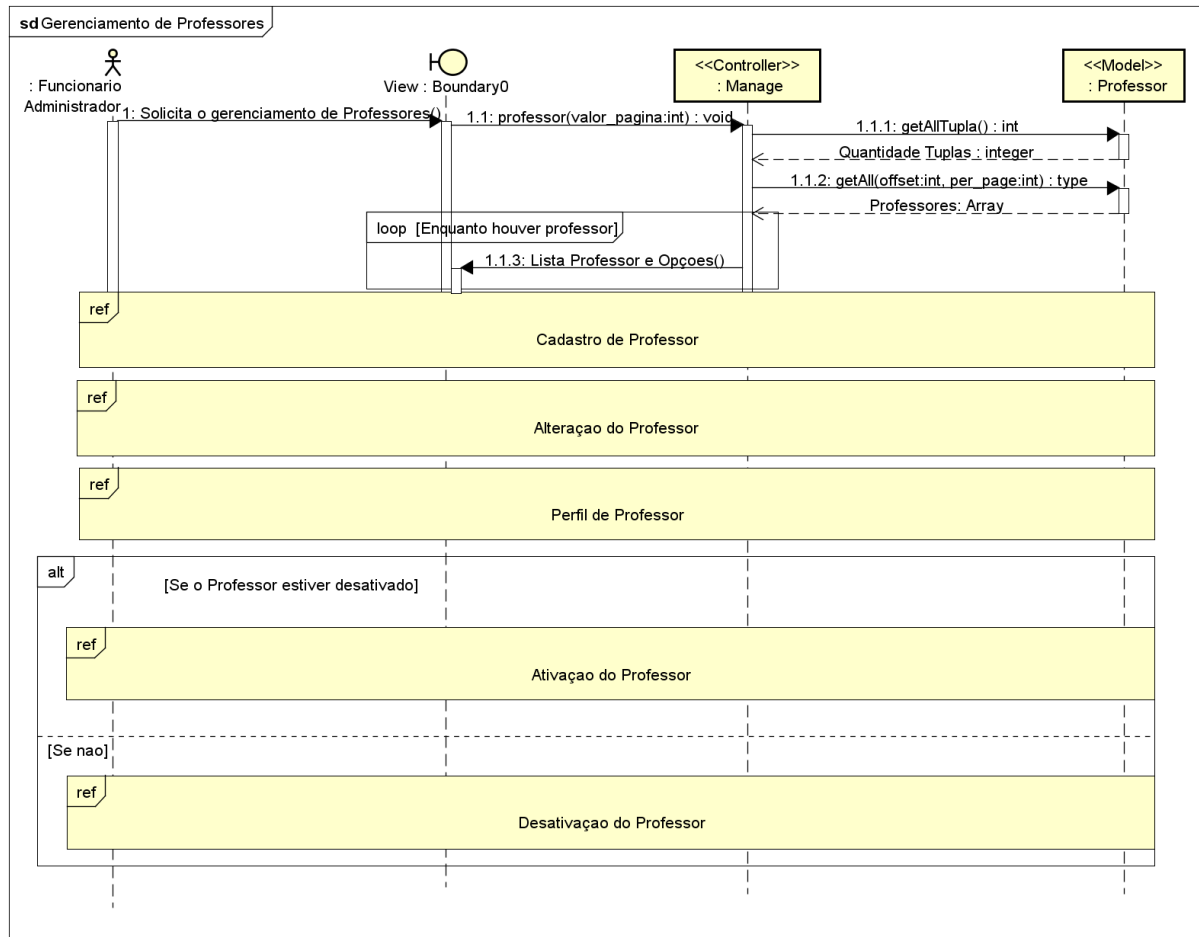
Figura 29: Diagrama de sequência recuperação de senha



Fonte: A autoria do Autor.

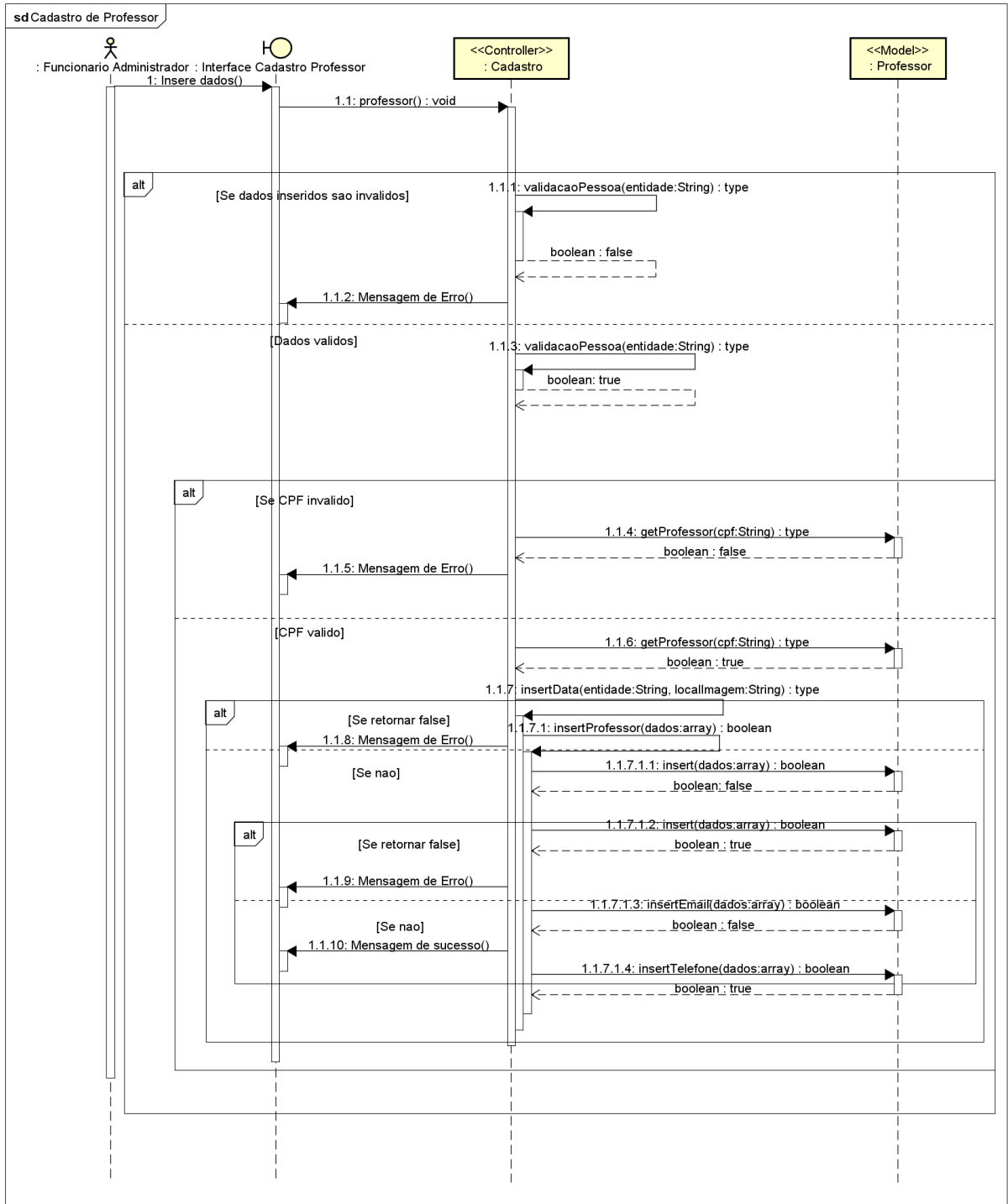
## 4.3.3 Professor

Figura 30: Diagrama de seqüência gerenciamento de professor



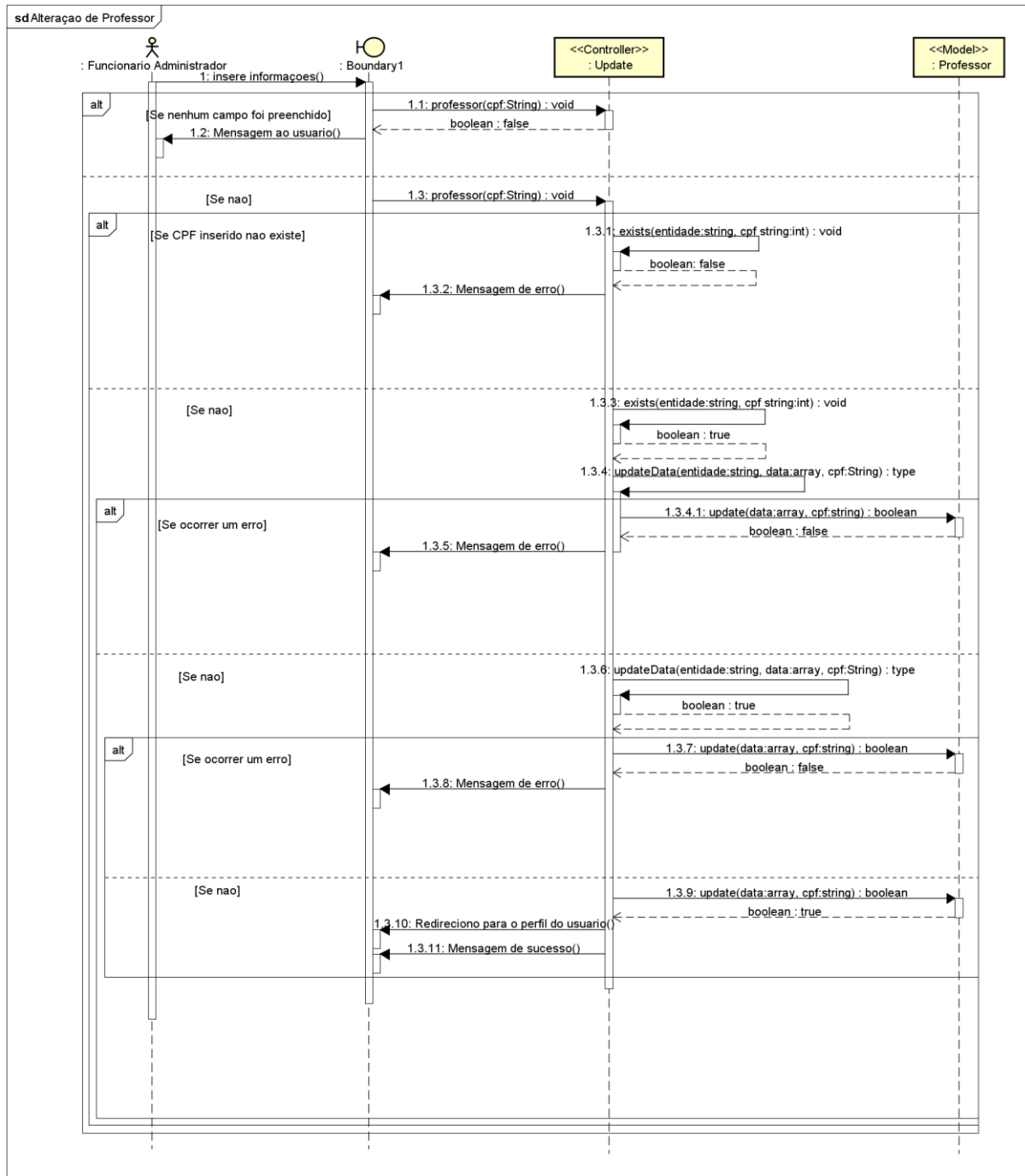
Fonte: Autoria do Autor.

Figura 31: Diagrama de sequência cadastro de professor



Fonte: Aatoria do Autor.

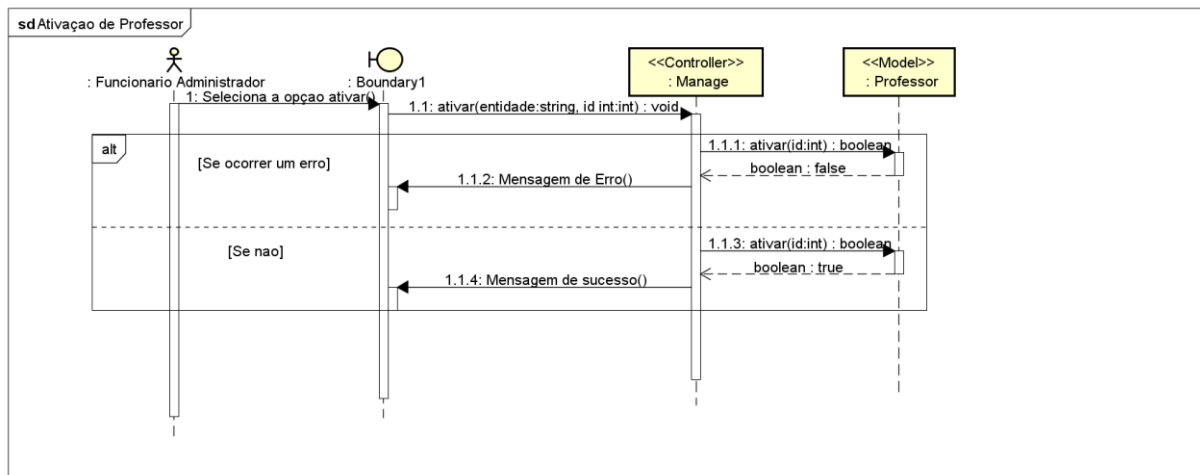
Figura 32: Diagrama de sequência alteração de professor



Fonte: A autoria do Autor.

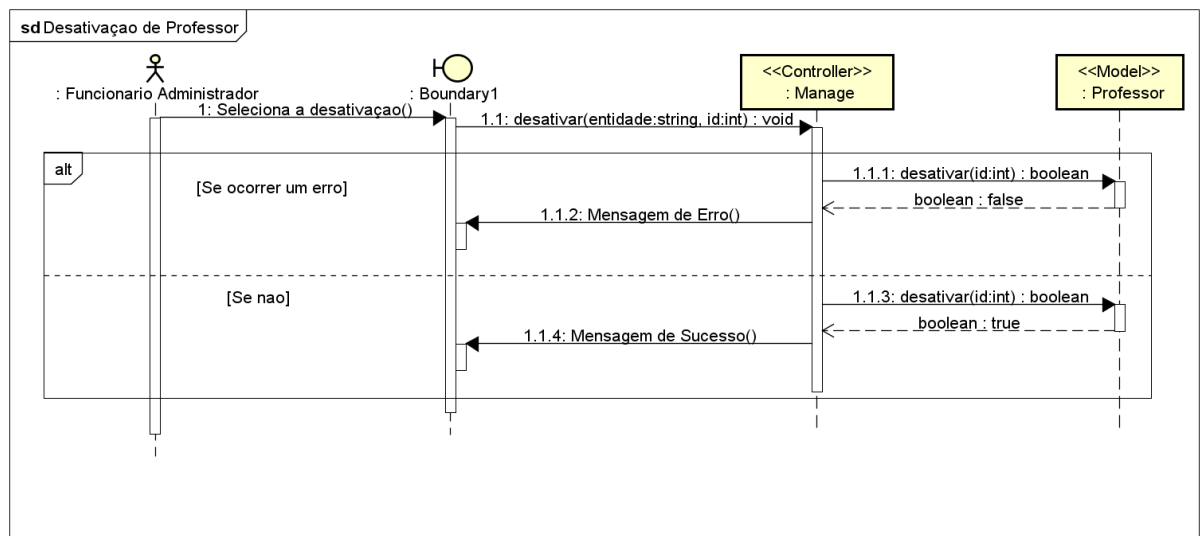


**Figura 33: Diagrama de seqüência ativação de professor**



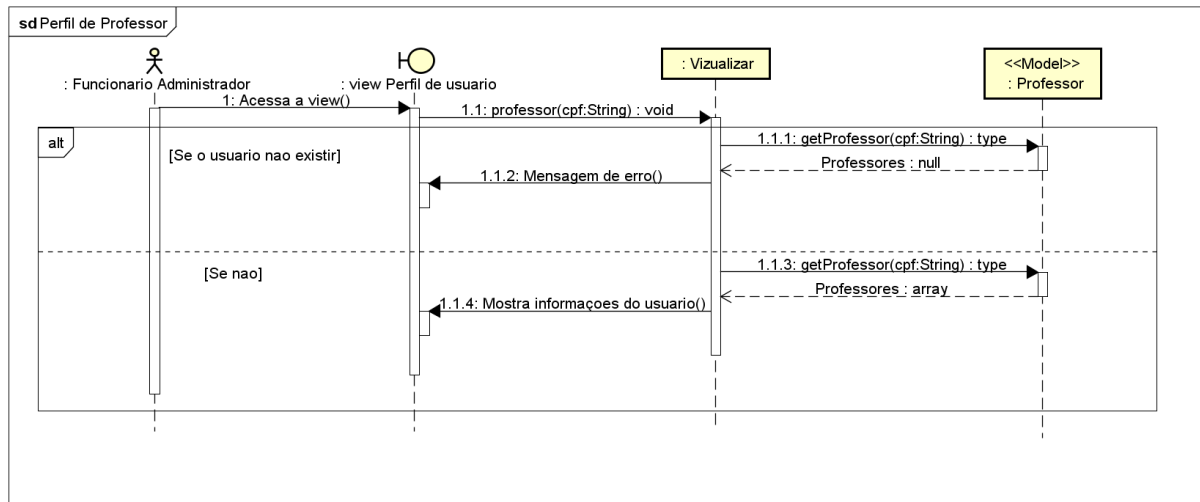
Fonte: Aatoria do Autor.

**Figura 34: Diagrama de seqüência desativação de professor**



Fonte: Aatoria do Autor.

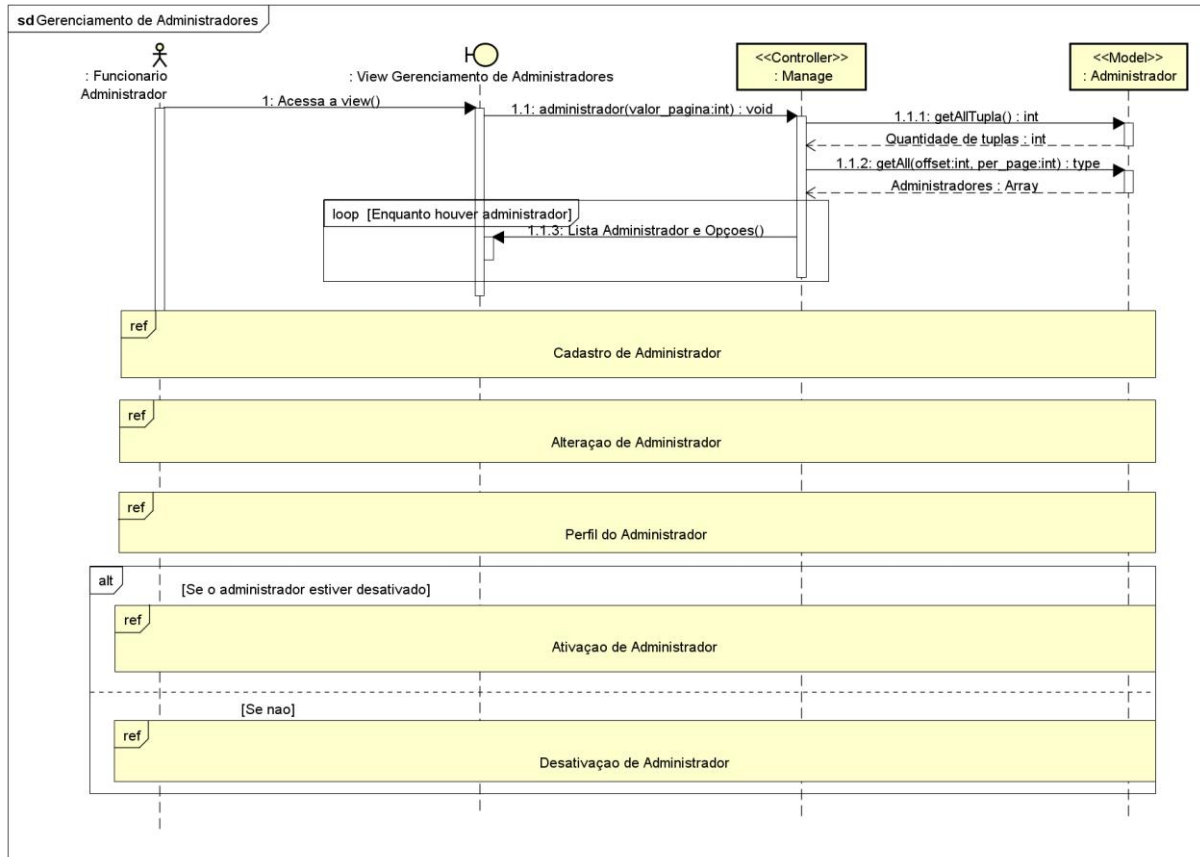
Figura 35: Diagrama de sequência perfil do professor



Fonte: A autoria do Autor.

## 4.3.4 Administrador

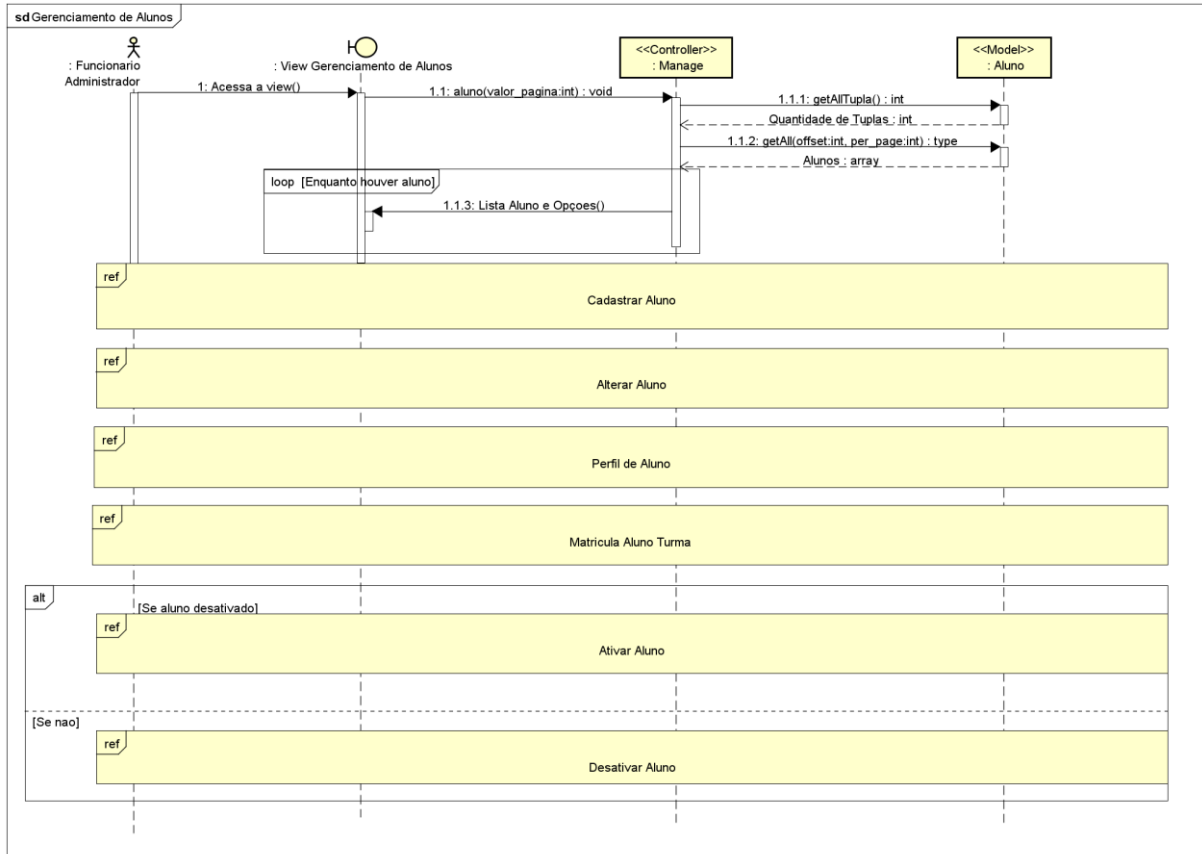
Figura 36: Diagrama de sequência gerenciamento de administradores



Fonte: A autoria do Autor.

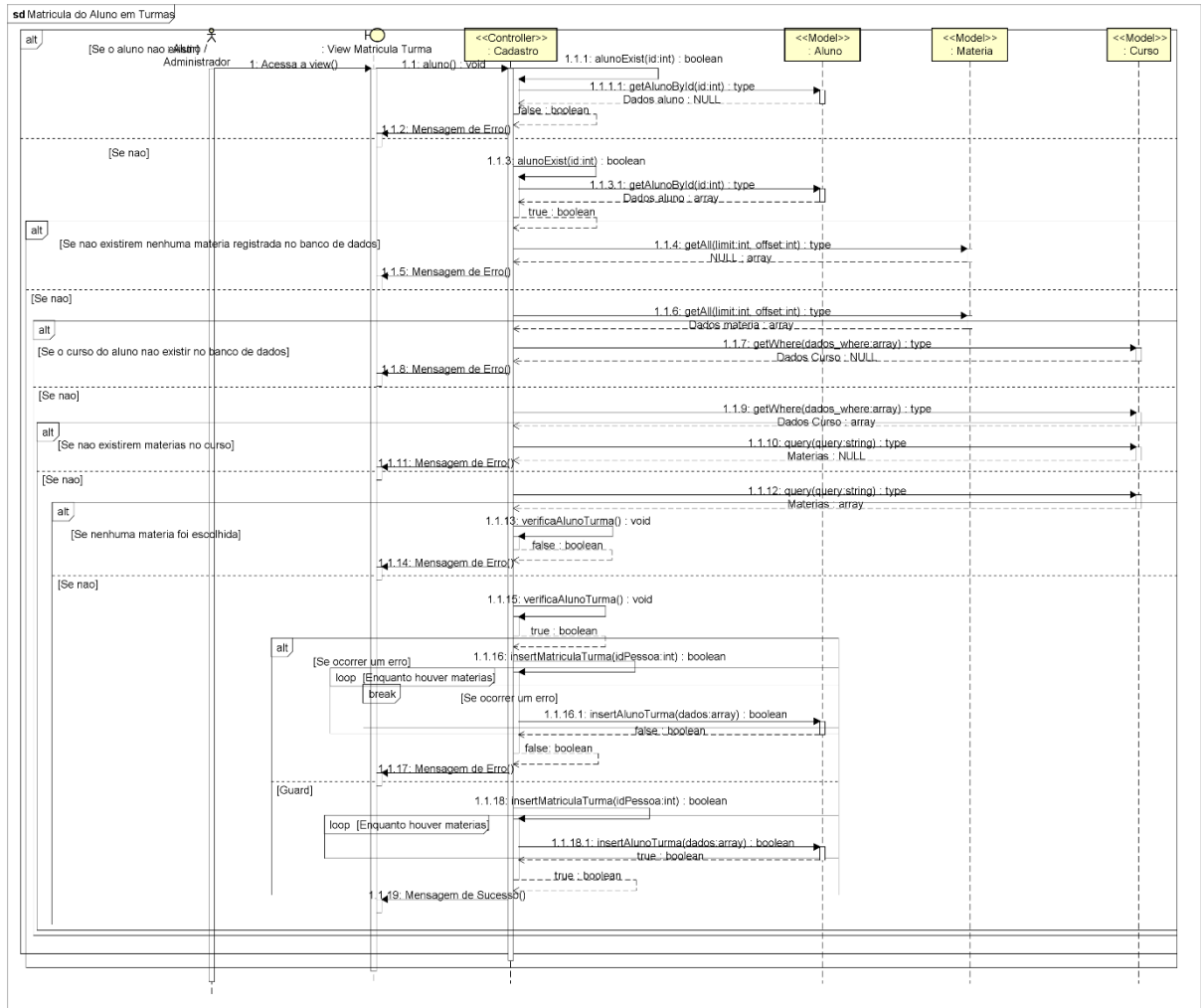
## 4.3.5 Aluno

Figura 37: Diagrama de seqüência gerenciamento de alunos



Fonte: Aatoria do Autor.

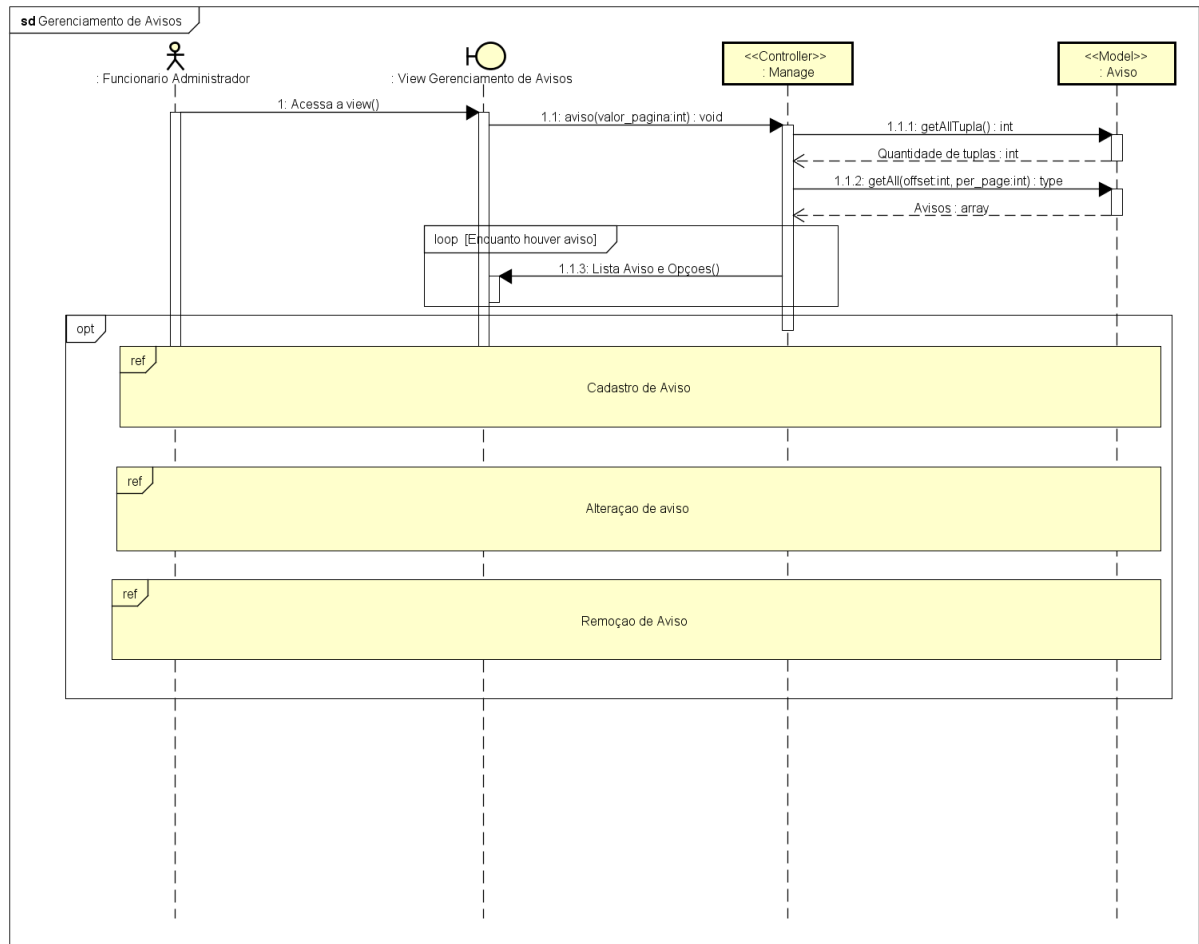
Figura 38: Diagrama de seqüência matricula de aluno em turma



Fonte: Autoria do Autor.

### 4.3.6 Avisos

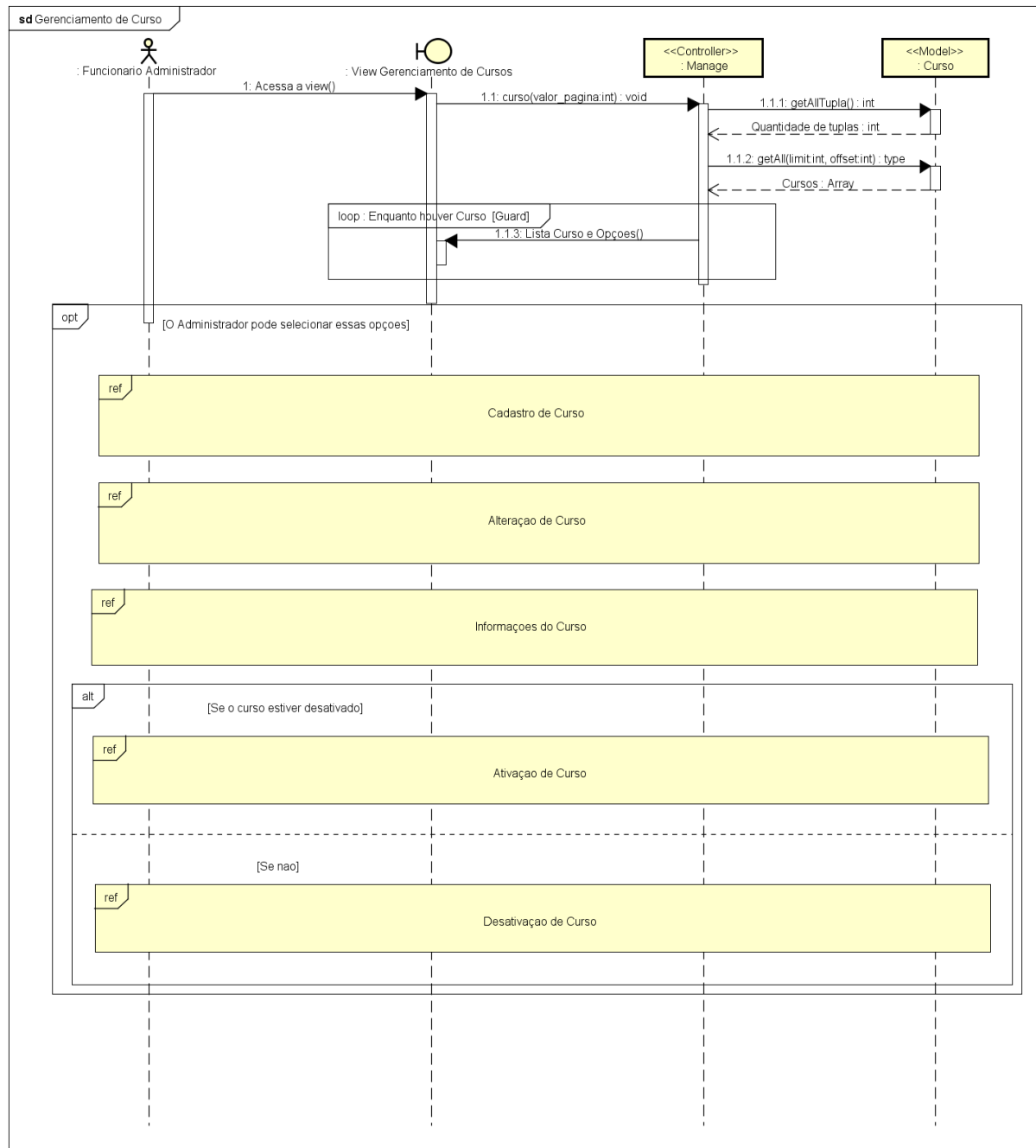
Figura 39: Diagrama de sequência gerenciamento de avisos



Fonte: Autoria do Autor.

## 4.3.7 Cursos

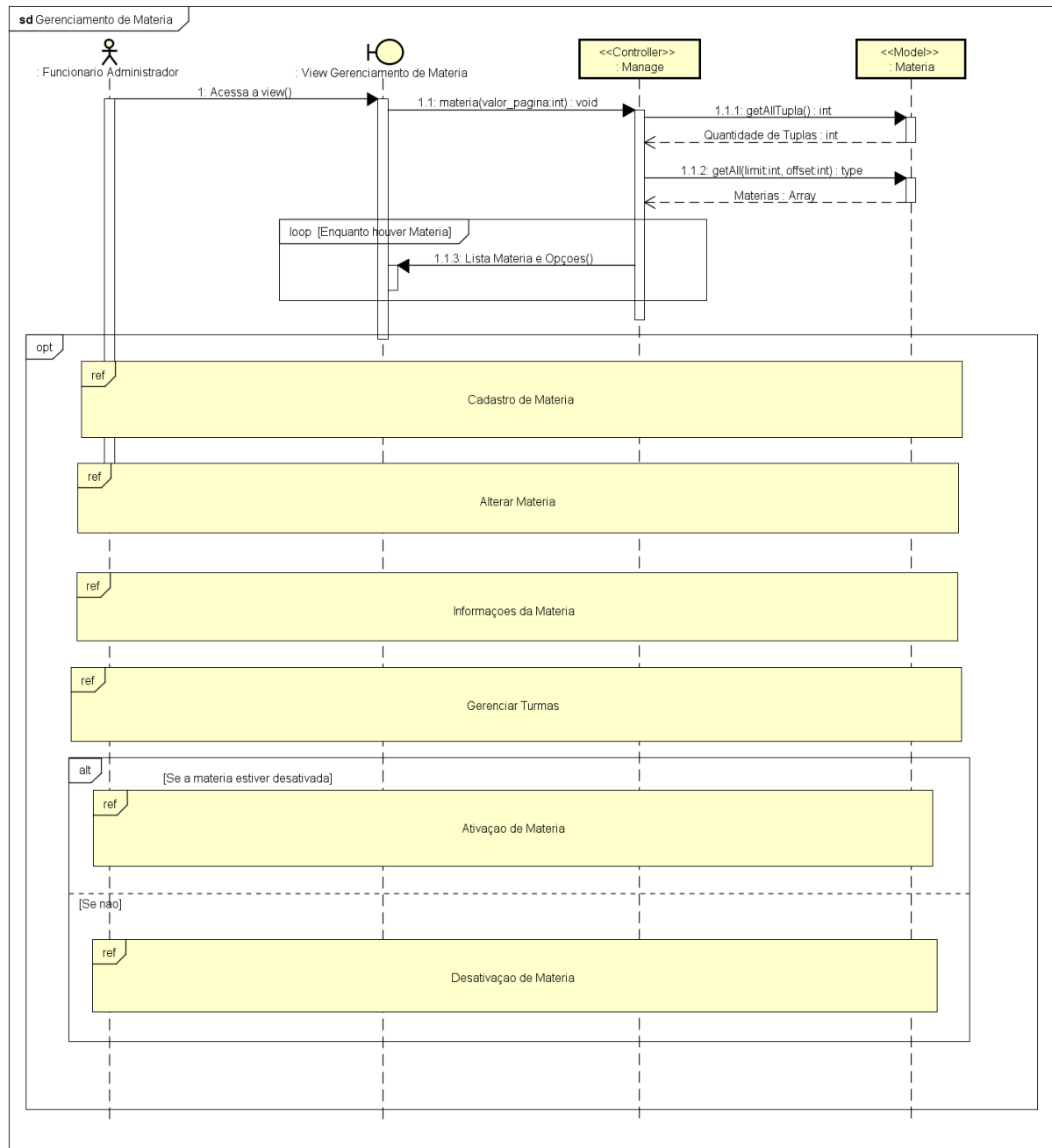
Figura 40: Diagrama de seqüência gerenciamento de cursos



Fonte: Autoria do Autor.

## 4.3.8 Matéria

Figura 41: Diagrama de seqüência gerenciamento de matérias

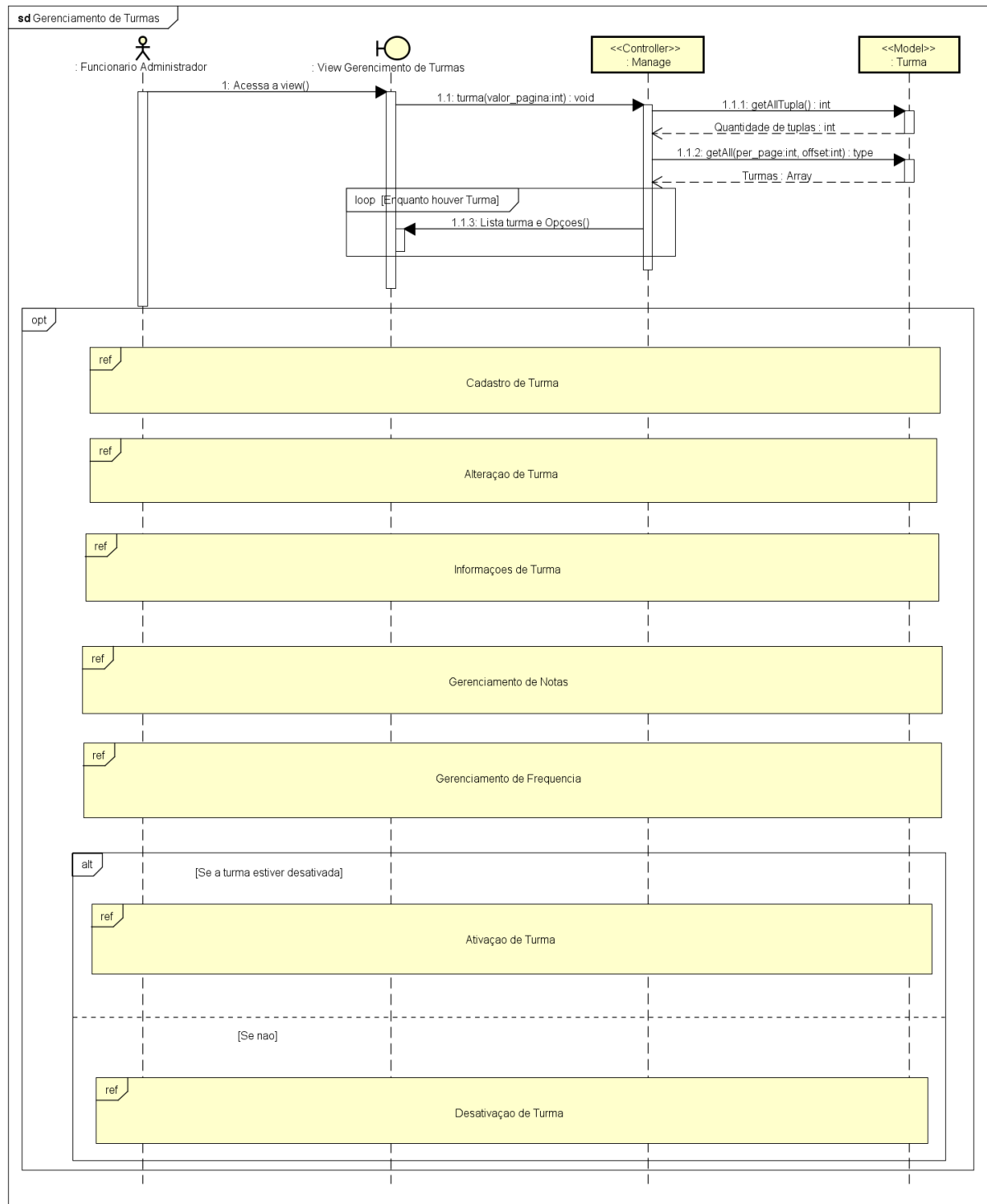


Fonte: Aatoria do Autor.



## 4.3.9 Turma

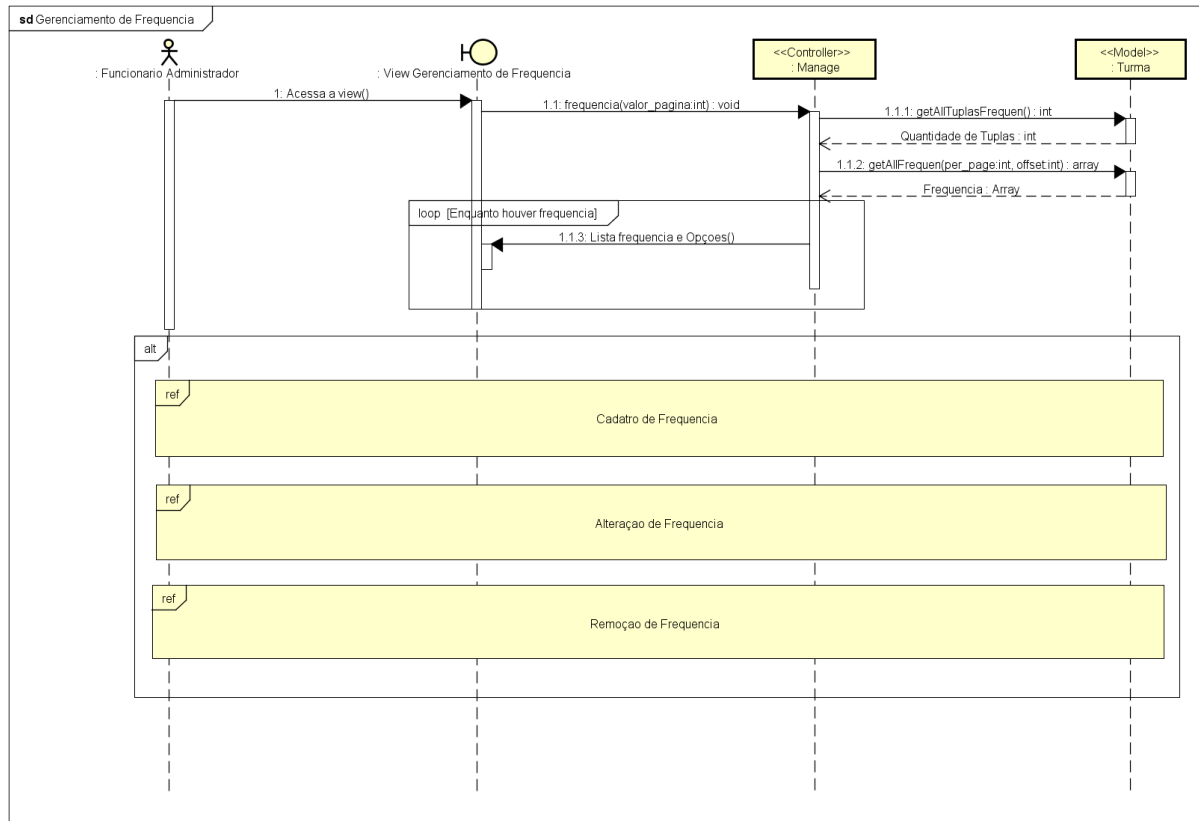
Figura 42: Diagrama de seqüência gerenciamento de turmas



Fonte: Autoria do Autor.

## 4.3.10 Frequência

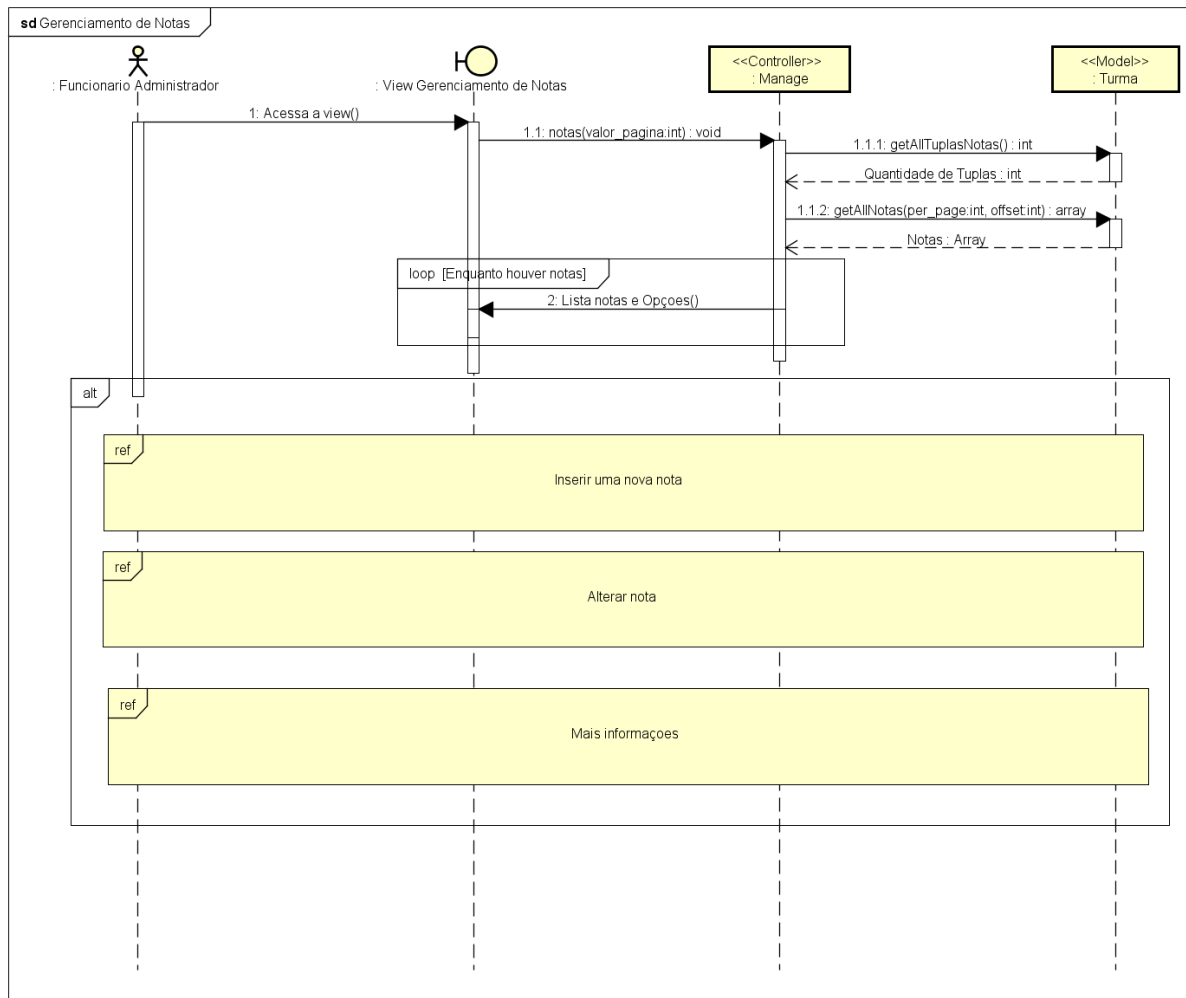
Figura 43: Diagrama de seqüência gerenciamento de frequências



Fonte: A autoria do Autor.

## 4.3.11 Notas

Figura 44: Diagrama de sequência gerenciamento de notas



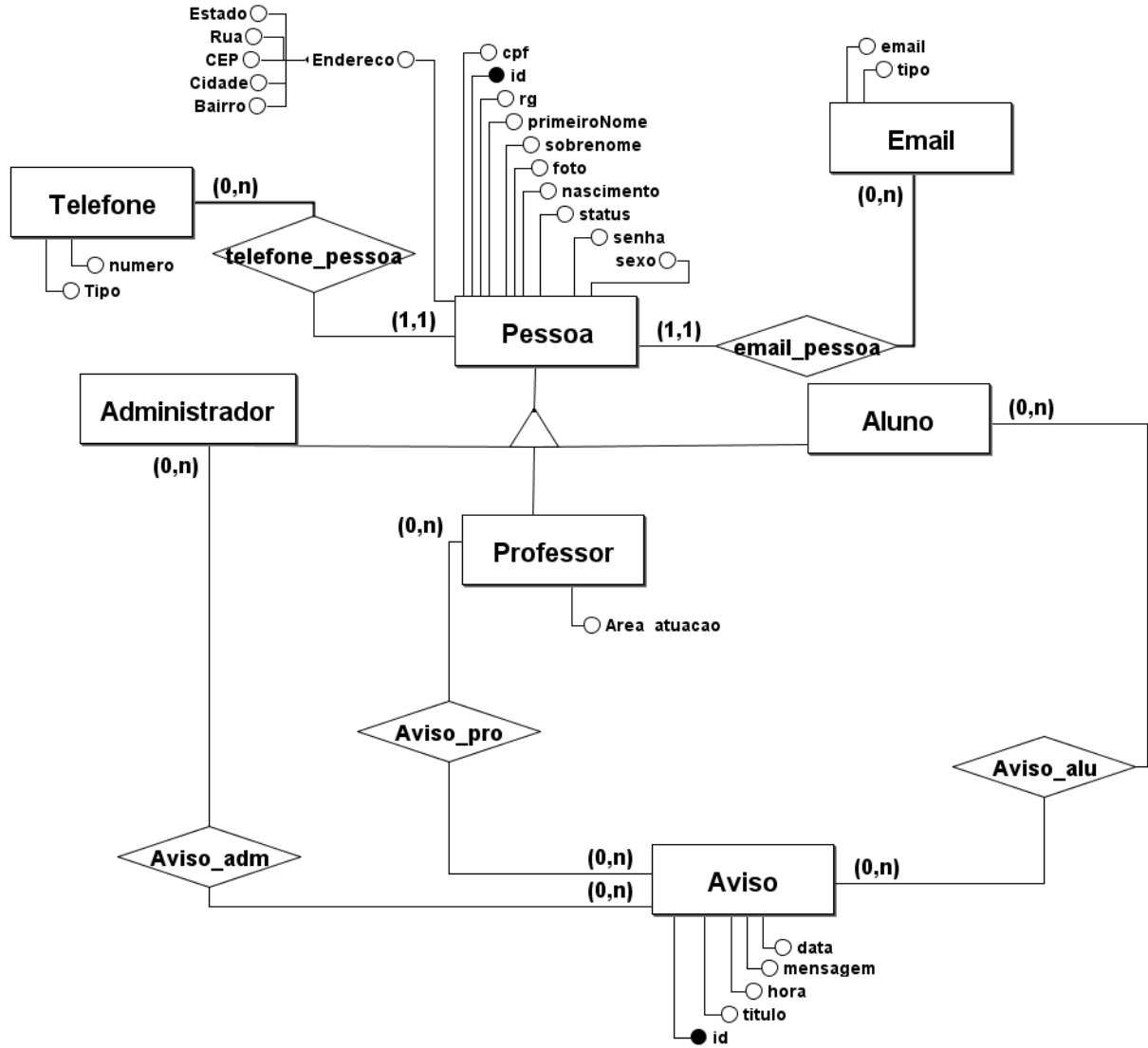
powered by Astah

Fonte: A autoria do Autor.

## 4.4 MODELO ENTIDADE E RELACIONAMENTO (MER)

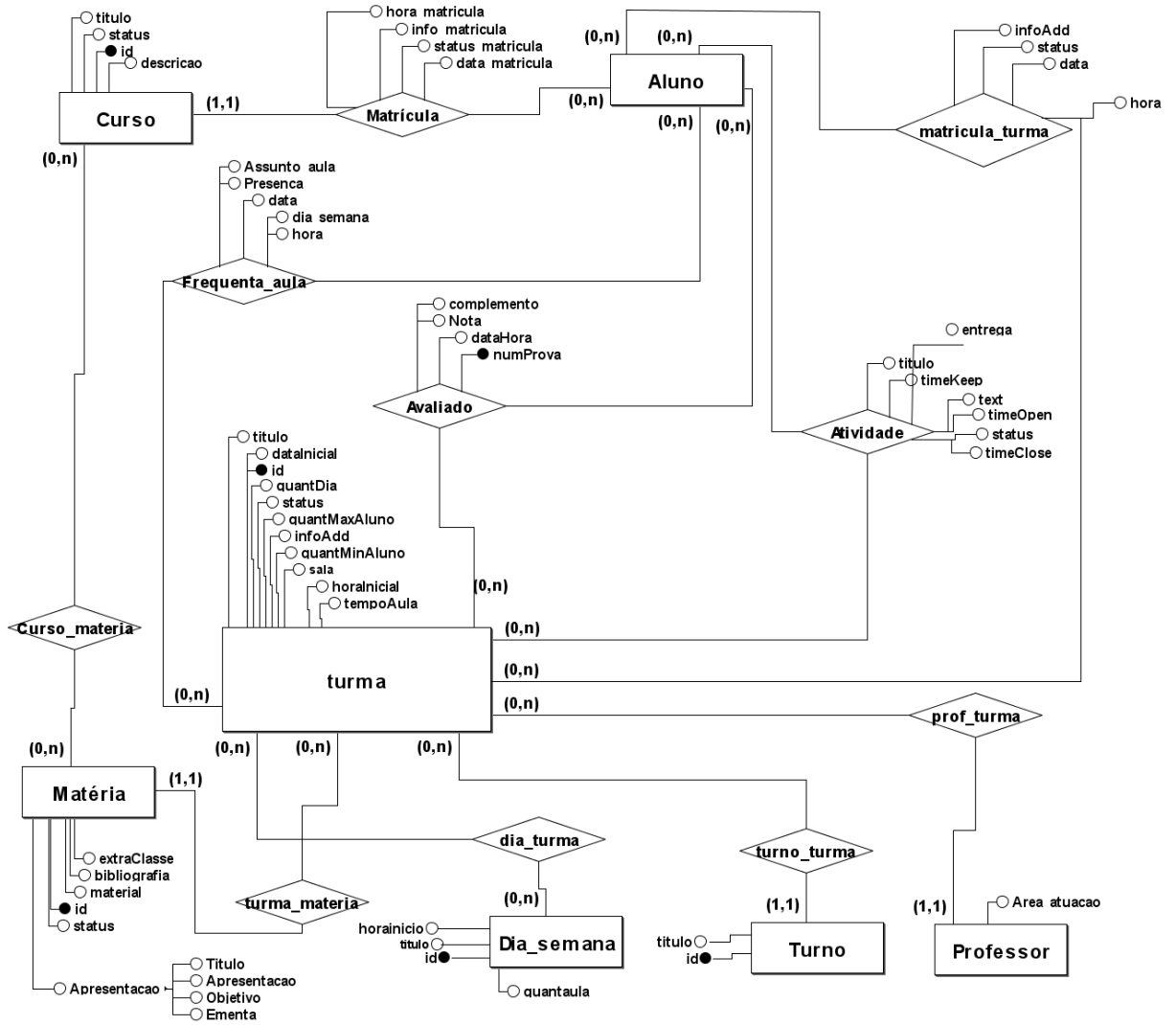
Aqui é apresentado o Modelo de entidade e relacionamento (MER), cujo o objetivo é apresentar a lógica do banco de dados sem atributos técnicos. Para uma melhor visualização o MER foi dividido em duas partes.

Figura 45: Modelo entidade e relacionamento



Fonte: A autoria do Autor.

Figura 46: Modelo entidade e relacionamento 2

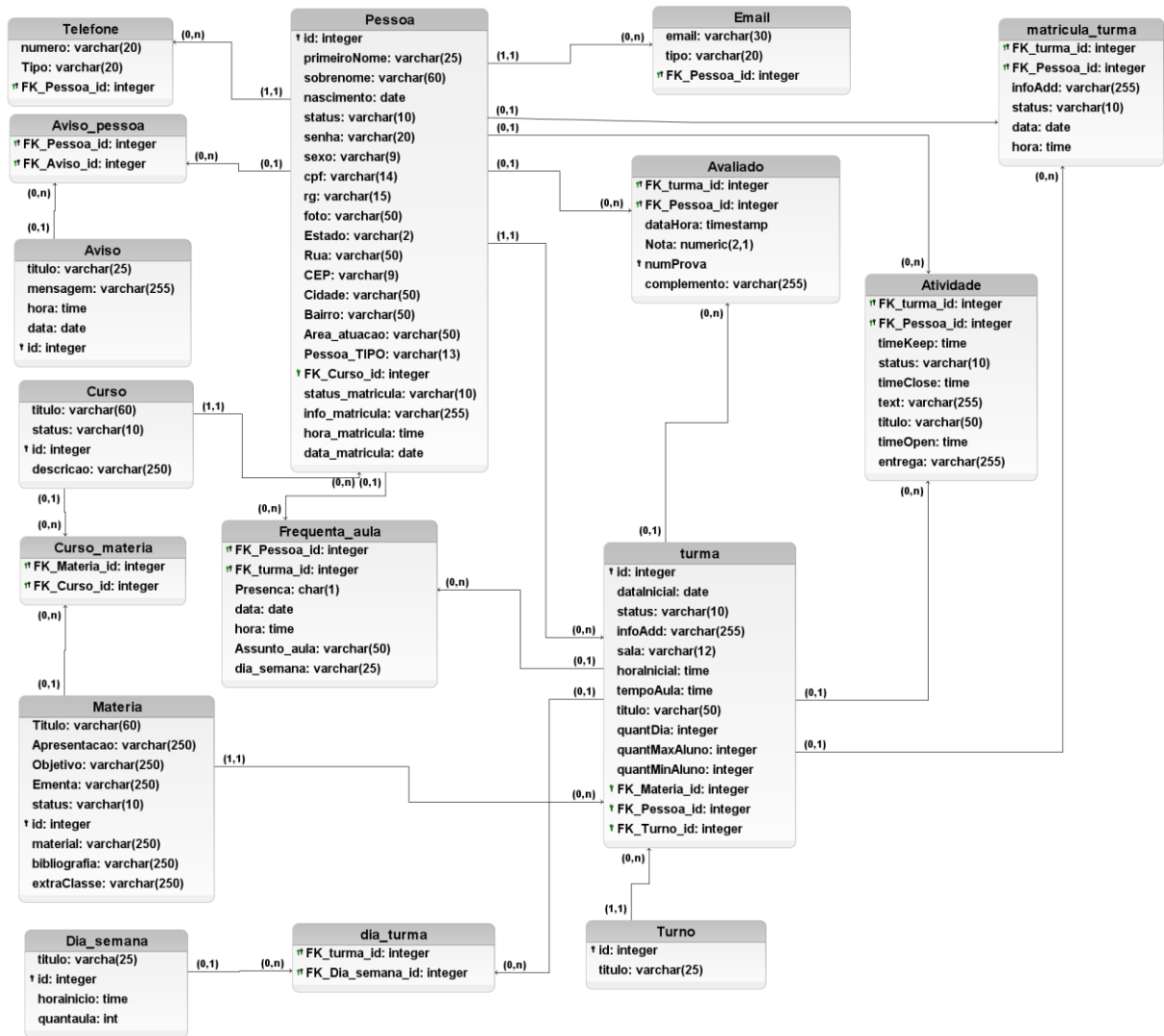


Fonte: Autoria do Autor.

#### 4.5 DIAGRAMA ENTIDADE E RELACIONAMENTO (DER)

A seguir é apresentado o Diagrama de entidade e relacionamento (DER), cujo objetivo é apresentar a lógica do banco de dados com atributos técnicos.

Figura 47: Diagrama de entidade e relacionamento



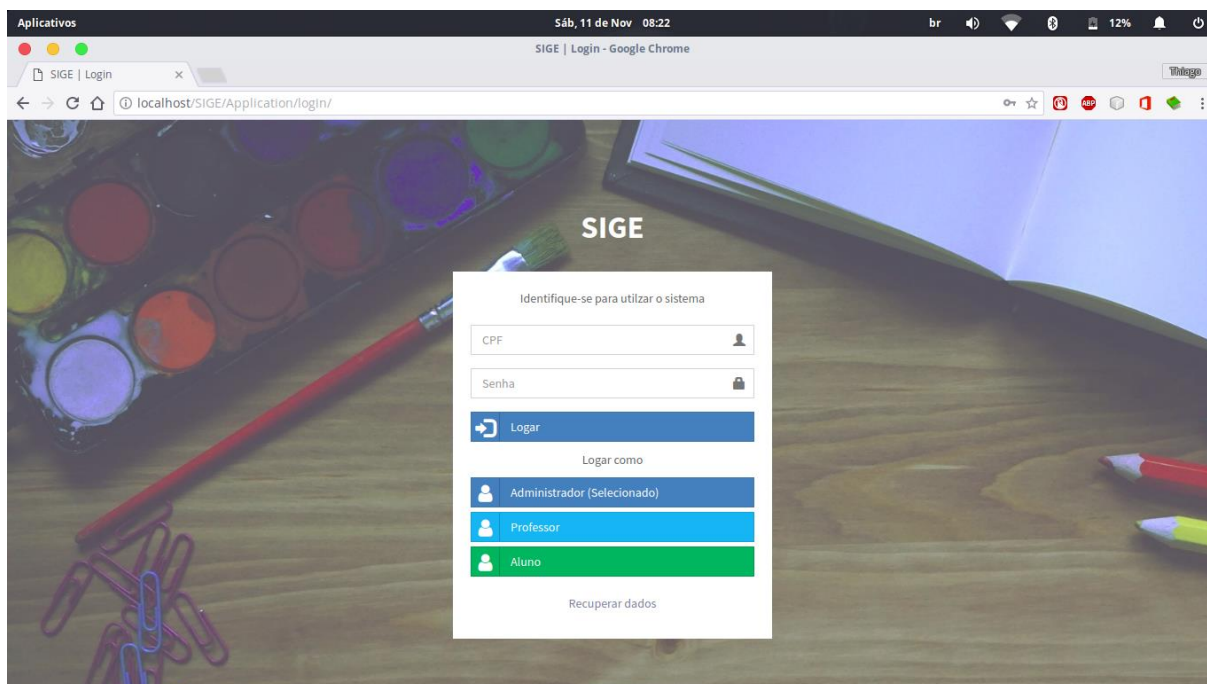
Fonte: Autoria do Autor.

## 4.6 SOFTWARE

Aqui será apresentado a documentação do software com suas principais funcionalidades que foram desenvolvidas, esta documentação é de vital importância para novos usuários. Nem todas as funcionalidades do sistema estão prontas até o presente momento dessa documentação do sistema devido a isso somente as funções prontas serão explicadas.

### 4.6.1 Login

**Figura 48: Tela de login**

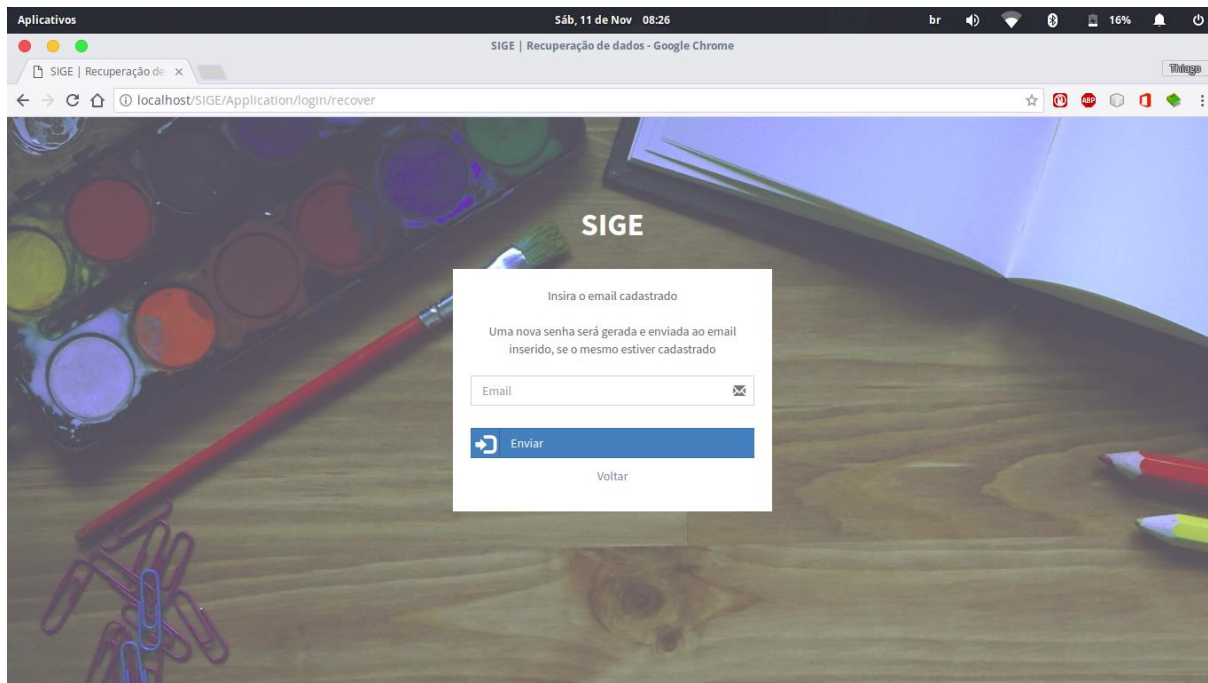


**Fonte: Autoria do autor.**

Está é a primeira tela do sistema, não há outra forma de entrar sem logar. Aqui é possível escolher três níveis de autorização sendo elas, administrador, professor ou aluno os dados necessários para o *login* são o e-mail e a senha, uma conta somente pode ser autenticada se estiver com o status ativo.

## 4.6.2 Recuperação de senha

**Figura 49: Tela recuperação de senha**



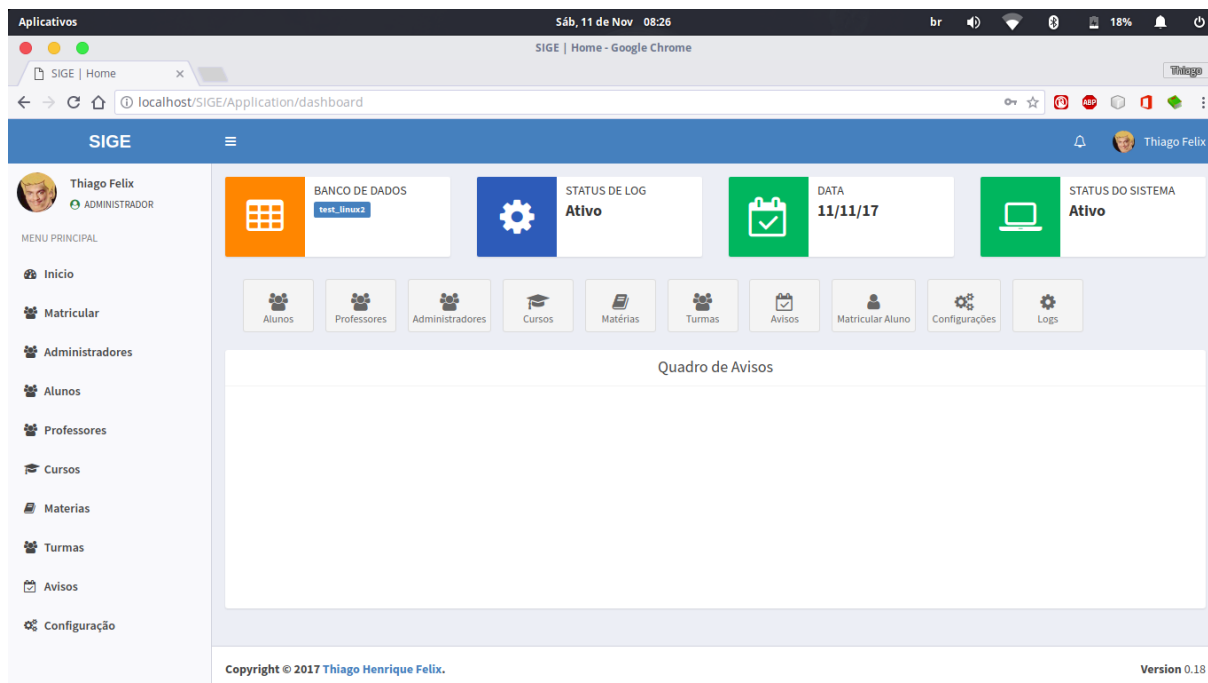
Fonte: A autoria do Autor.

Esta tela pode ser acessada pela tela inicial do sistema (item 4.6.1), aqui é possível recuperar a senha, ao inserir o *e-mail* o sistema irá verificar se existe alguma conta com este *e-mail*, se o sistema o encontrar será gerado uma nova senha e enviada para o *e-mail*.



### 4.6.3 Tela inicial do Administrador

Figura 50: Tela inicial do administrador



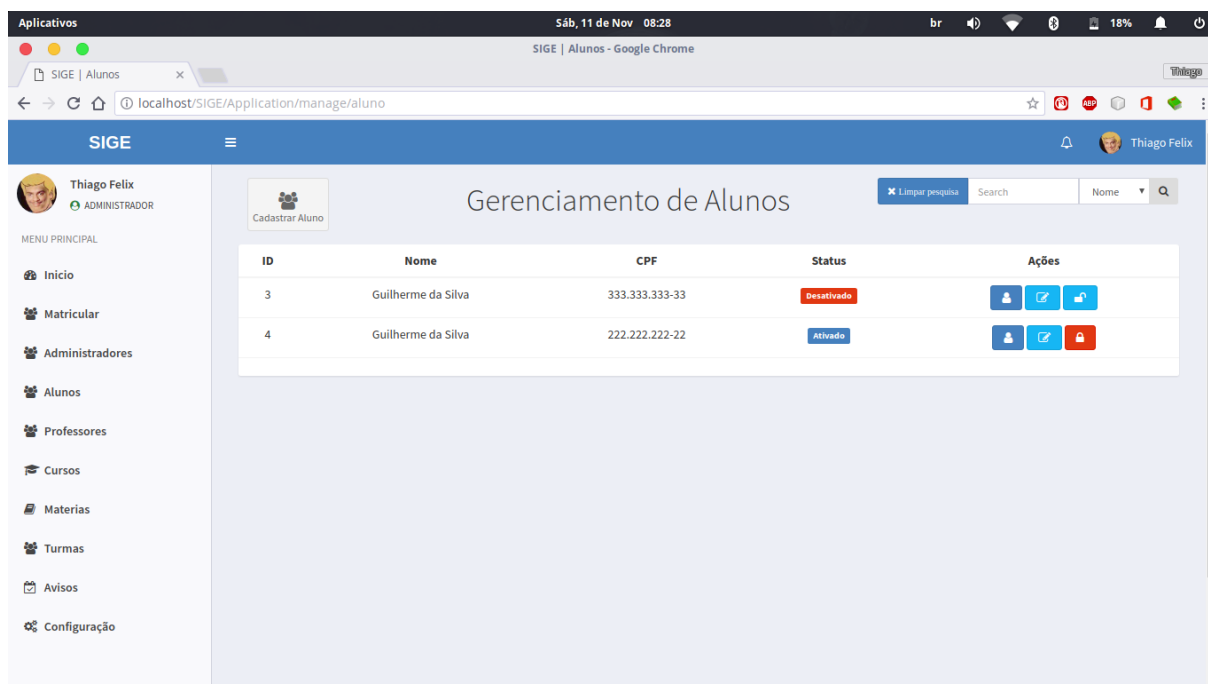
Fonte: Autoria do Autor.

Após o *login* do administrador (Item 4.6.1) o sistema irá redirecionar para esta página, a partir daqui é possível acessar todas as funcionalidades do sistema.

Os itens à cima da página indicam o status do sistema, data atual, status da geração de logs e o banco de dados utilizado no momento, logo abaixo estão os links rápidos que permitem um rápido acesso as principais funcionalidades do sistema, abaixo está o quadro de avisos este quadro lista todos os avisos registrados para os administradores, ou seja, avisos para alunos ou professores não são listados aqui. O menu lateral tem um rápido acesso as principais funcionalidades do sistema.

#### 4.6.4 Gerenciamento de Alunos

**Figura 51: Tela de Gerenciamento de Aluno**



**Fonte: Autoria do Autor.**

O gerenciamento de alunos pode ser acessado pelo menu lateral ou pela tela inicial do administrador (Item 4.6.3). Esta funcionalidade somente está disponível para administradores do sistema.

No gerenciamento de alunos é possível acessar o cadastro do aluno, a lista de todos os alunos do sistema e para cada aluno é possível visualizar o perfil, alterar, desativar e ativar.

A desativação do aluno o impedirá de entrar no sistema e sua ativação o permitirá.

#### 4.6.5 Perfil do Aluno

**Figura 52: Tela de Perfil do Aluno**



**Fonte: A autoria do Autor.**

O perfil do aluno pode ser acessado pelo gerenciamento de alunos (Item 4.6.3), esta funcionalidade somente pode ser acessada pelo administrador do sistema.

O perfil de aluno permite ver todas as informações da pessoa registradas no sistema.

#### 4.6.6 Alteração de Aluno

Figura 53: Tela de alteração de aluno

Aplicativos Sáb, 11 de Nov 08:32 br 23%

SIGE | Atualizar Aluno - Google Chrome

SIGE | Atualizar Aluno x

Localhost/SIGE/Application/update/Aluno/333.333.333-33

Thiago Felix ADMINISTRADOR

Gerenciar Aluno

### Atualizar Aluno

Insira apenas os campos que deseja atualizar

Escolher arquivo Nenhum arquivo selecionado

João

Sobrenome\*

dd/mm/aaaa

CEP

Selecione seu estado

Cidade

Bairro

Rua

Email\*

CPF\*

Telefone

Sexo\*

Senha\*

Confirme a senha\*

Cadastrar

MENU PRINCIPAL

- Início
- Matricular
- Administradores
- Alunos
- Professores
- Cursos
- Materias
- Turmas
- Avisos
- Configuração

Fonte: Autoria do Autor.

Figura 54: Tela de perfil de aluno sucesso na alteração

Aplicativos Sáb, 11 de Nov 08:32 br 23%

SIGE - Perfil de Usuário - Google Chrome

SIGE - Perfil de Usuário x

Localhost/SIGE/Application/visualizar/aluno/333.333.333-33

SIGE

Thiago Felix ADMINISTRADOR

Dados de aluno atualizados com sucesso

Gerenciar Aluno

### Perfil do Usuário

João da Silva  
Aluno

CPF 333.333.333-33

RG 12.312.313-2

Sexo Masculino

Editar

Excluir

Endereço

- ESTADO: MA
- CIDADE: Americana
- BAIRRO: São Paulista
- CEP: 51651-561
- RUA: Dos Jardins

Dados Pessoais

- CEF: 333.333.333-33
- RG: 12.312.313-2
- TELEFONE:
- REGISTRO: 3

MENU PRINCIPAL

- Início
- Matricular
- Administradores
- Alunos
- Professores
- Cursos
- Materias
- Turmas
- Avisos
- Configuração

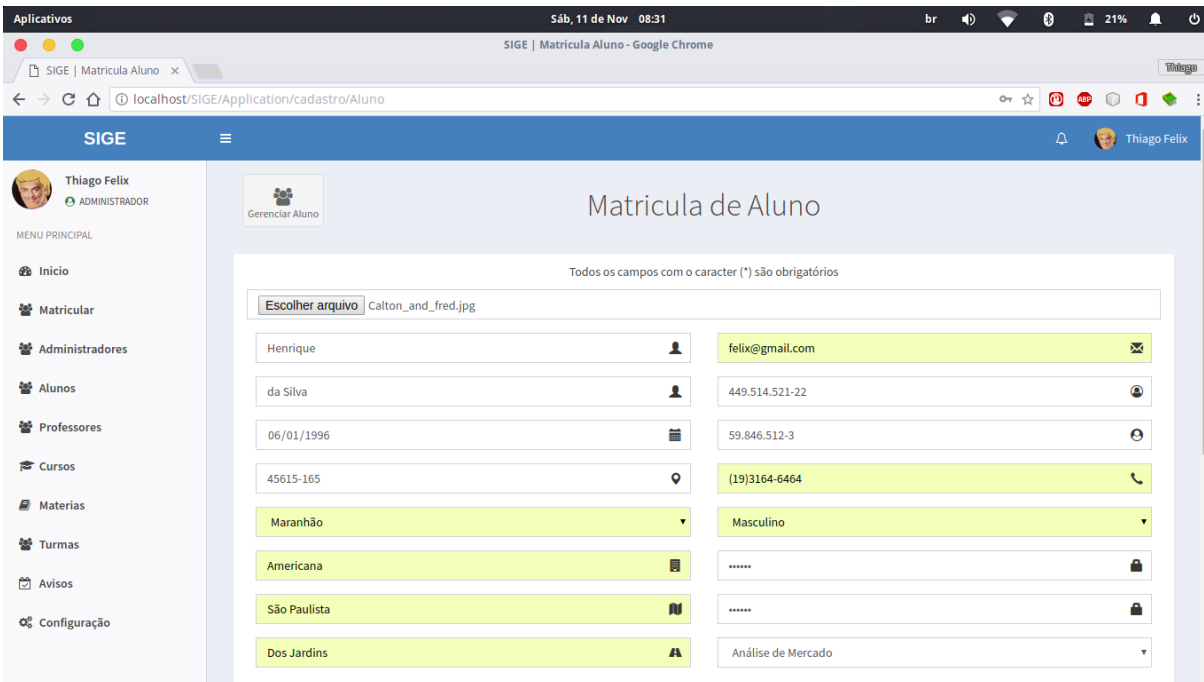
Fonte: Autoria do Autor.

A tela de alteração de aluno pode ser acessada pelo gerenciamento de alunos (Item 4.6.3), esta funcionalidade somente pode ser acessada pelo administrador do sistema.

A tela de alteração de aluno permite alterar informações do aluno, para alterar basta inserir a informação e continuar se os dados forem validados o sistema irá redirecionar para a tela de perfil de aluno e indicará o sucesso da alteração.

#### 4.6.7 Cadastro de Aluno

**Figura 55: Tela de cadastro de aluno**



The screenshot shows a web browser window displaying the 'SIGE | Matricula Aluno' page. The browser's address bar shows 'localhost/SIGE/Application/cadastro/Aluno'. The page has a blue header with the 'SIGE' logo and a user profile for 'Thiago Felix' (ADMINISTRADOR). A left sidebar menu lists various system functions like 'Início', 'Matricular', 'Administradores', 'Alunos', 'Professores', 'Cursos', 'Materias', 'Turmas', 'Avisos', and 'Configuração'. The main content area is titled 'Matricula de Aluno' and contains a form with the following fields:

- File upload: 'Escolher arquivo' with 'Calton\_and\_fred.jpg' selected.
- First Name: 'Henrique'.
- Last Name: 'da Silva'.
- Birth Date: '06/01/1996'.
- CPF: '45615-165'.
- Address: 'Maranhão', 'Americana', 'São Paulista', 'Dos Jardins' (with a dropdown arrow).
- Email: 'felix@gmail.com'.
- Phone: '449.514.521-22'.
- Mobile Phone: '59.846.512-3'.
- WhatsApp: '(19)3164-6464'.
- Gender: 'Masculino'.
- Two password fields (masked with '.....').
- City: 'Análise de Mercado' (with a dropdown arrow).

A note above the form states: 'Todos os campos com o caracter (\*) são obrigatórios'.

Fonte: A autoria do Autor.

**Figura 56: Tela de cadastro de aluno 2**

The screenshot shows a web browser window with the URL `localhost/SIGE/Application/cadastro/Aluno`. The page features a sidebar menu on the left with options: Professores, Cursos, Materias, Turmas, Avisos, and Configuração. The main form area contains several input fields and dropdown menus. The fields are: a date field with '06/01/1996', a text field with '45615-165', a dropdown menu with 'Maranhão', a dropdown menu with 'Americana', a dropdown menu with 'São Paulista', and a dropdown menu with 'Dos Jardins'. On the right side, there are fields for '59.846.512-3', '(19)3164-6464', 'Masculino', two masked password fields, and a dropdown menu with 'Análise de Mercado'. Below these fields is a text area for 'Informações adicionais da Matricula:' with a rich text editor toolbar. At the bottom of the form is a blue button labeled 'Cadastrar'.

**Fonte: Autoria do Autor.**

A tela de cadastro de aluno pode ser acessada pelo gerenciamento de alunos (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

No cadastro de aluno todos os campos com o símbolo (\*) são obrigatórios, se estes campos não forem preenchidos o sistema não irá cadastrar.

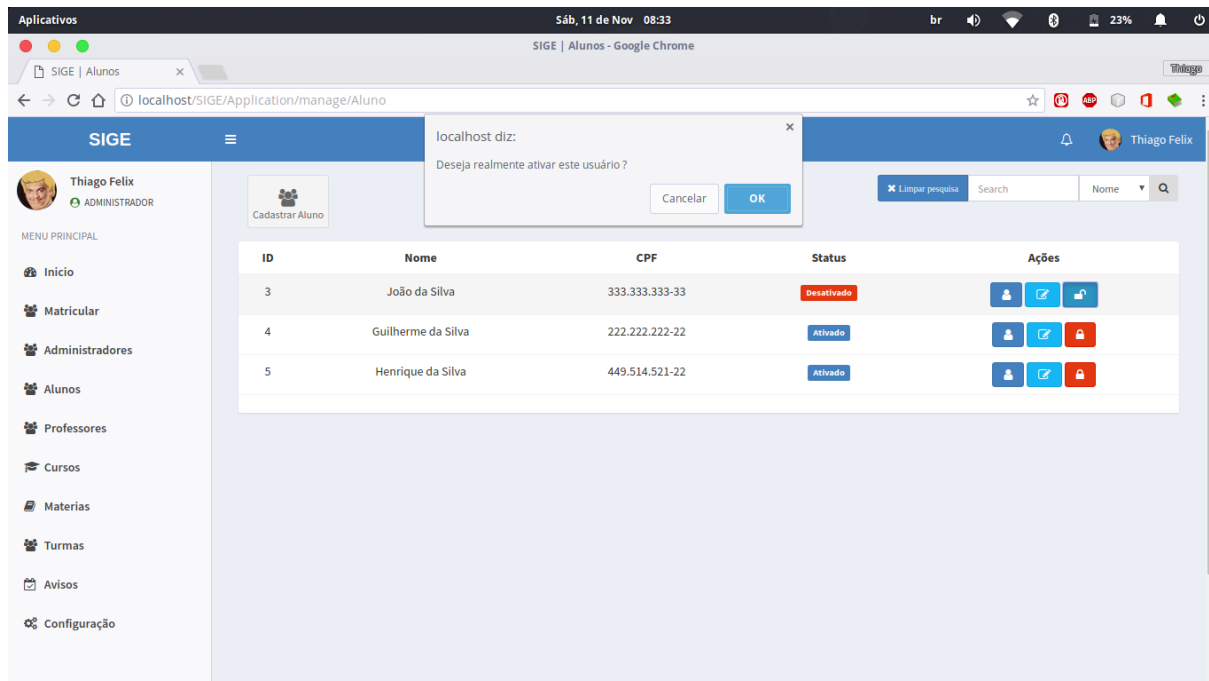
Para cadastrar um aluno é necessário ter ao menos um curso cadastrado no sistema caso contrário o sistema irá persistir para cadastrar um curso primeiro.

Não é possível cadastrar um novo aluno com um CPF de outro aluno, ou seja, não é possível cadastrar duas vezes o mesmo CPF em um aluno.

Após o cadastro se os dados forem validados o sistema irá redirecionar para o gerenciamento de alunos e listará o novo usuário juntamente com os outros.

#### 4.6.8 Ativação ou desativação de alunos

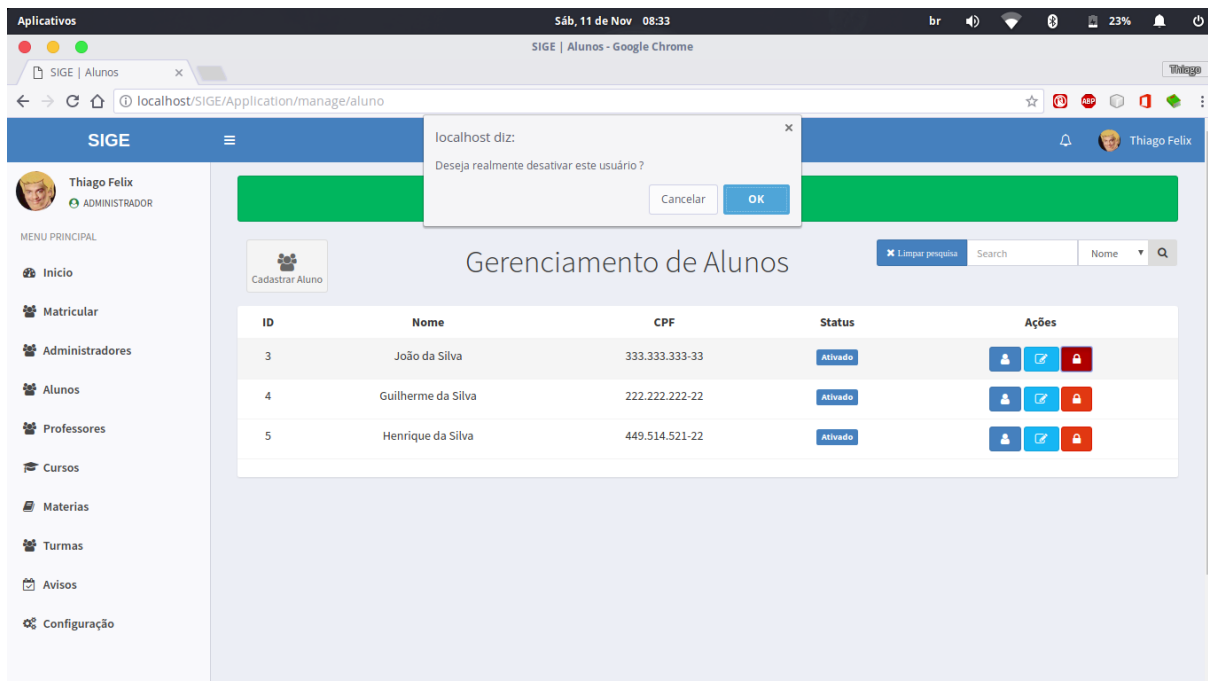
Figura 57: Tela de gerenciamento de alunos ativação de alunos



Fonte: Aatoria do Autor.

A tela de gerenciamento de alunos pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

Para ativar o aluno acesse o botão com um ícone de cadeado aberto e confirme logo após o sistema irá informar que o usuário está ativado, com isso o aluno agora pode acessar o sistema.

**Figura 58: Tela de gerenciamento de alunos desativação**

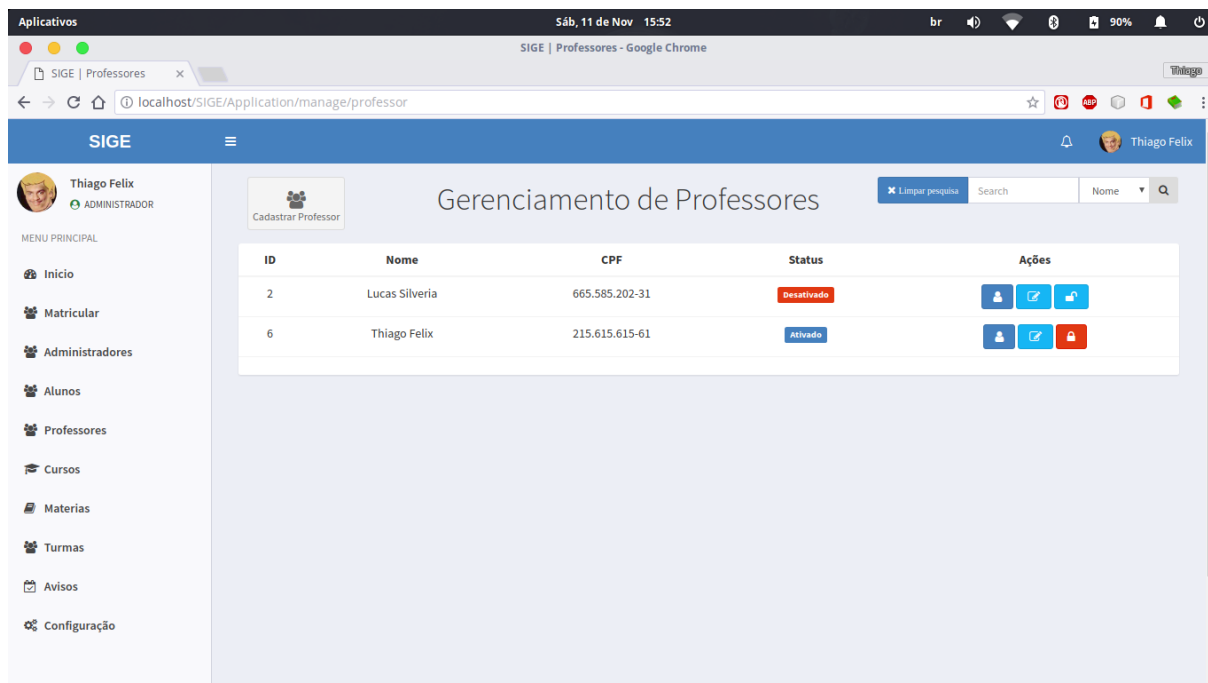
**Fonte: Autoria do Autor.**

A desativação funciona do mesmo jeito da ativação do aluno, esta opção somente está disponível quando o aluno está ativado. Para desativar basta acessar o botão vermelho e confirmar o sistema irá informar se ocorreu um erro ou se foi desativado com sucesso.



## 4.6.9 Gerenciamento de Professores

**Figura 59: Tela de gerenciamento de professores**



**Fonte: Autoria do Autor.**

A tela de gerenciamento de professores pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

No gerenciamento de professores é possível acessar o cadastro de professor, a lista de todos os professores do sistema e para cada professor é possível visualizar o perfil, alterar, desativar e ativar.

A desativação do professor o impedirá de entrar no sistema e sua ativação o permitirá.

## 4.6.10 Cadastro de Professor

**Figura 60:** Tela de cadastro de professor

Aplicativos Sáb, 11 de Nov 15:53 SIGE | Professores - Google Chrome

localhost/SIGE/Application/manage/professor

Thiago Felix ADMINISTRADOR

Gerenciar Professor

### Cadastro de Professor

Todos os campos com o caracter (\*) são obrigatórios

Escolher arquivo Nenhum arquivo selecionado

Bianca da Silva

06/05/1996

34551-561

Maranhão

Americana

São Paulista

Dos Jardins

thiagoacd12@gmail.com

448.554.865-88

56.165.165-1

(19)3464-6787

Masculino

.....

.....

Cadastrar

Aguardando localhost...

**Fonte:** Autoria do Autor.

A tela de cadastro de professor pode ser acessada pelo gerenciamento de professores (Item 4.6.9) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

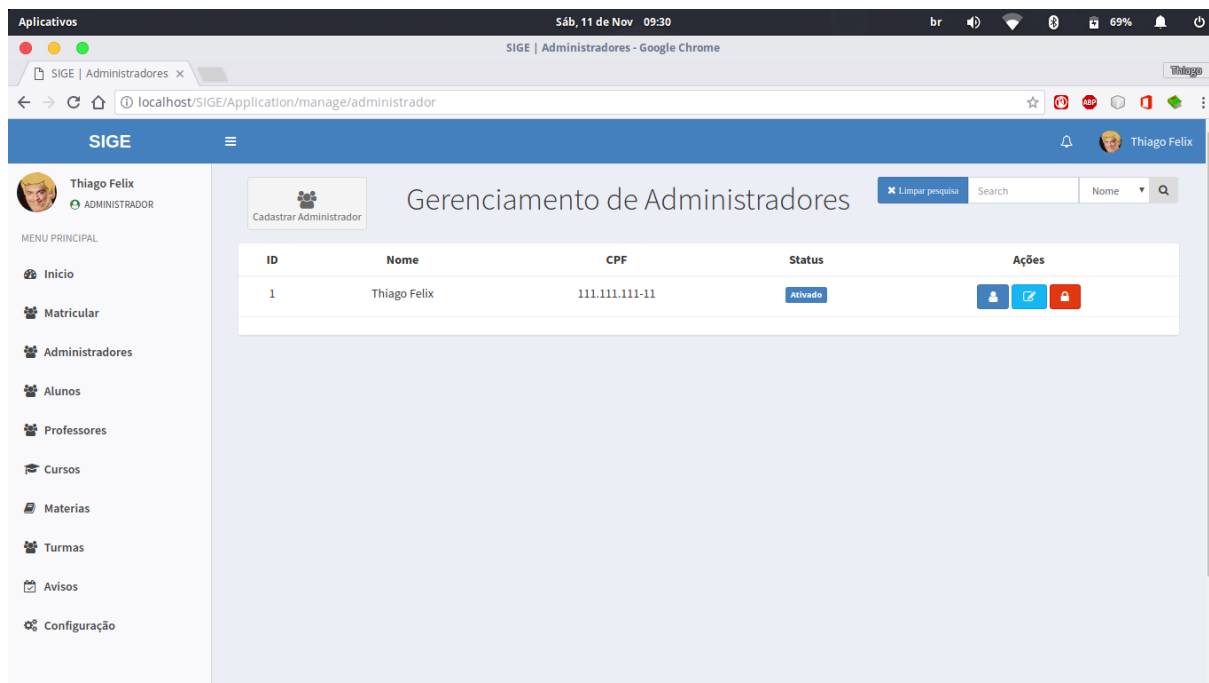
No cadastro de professor todos os campos com o símbolo (\*) são obrigatórios, se estes campos não forem preenchidos o sistema não irá cadastrar.

Não é possível cadastrar um novo professor com um CPF de outro professor, ou seja, não é possível cadastrar duas vezes o mesmo CPF em um professor.

Após o cadastro se os dados forem validados o sistema irá redirecionar para o gerenciamento de professores e listará o novo professor juntamente com os outros.

#### 4.6.11 Gerenciamento de Administradores

**Figura 61: Tela de gerenciamento de administrador**



**Fonte: A autoria do Autor.**

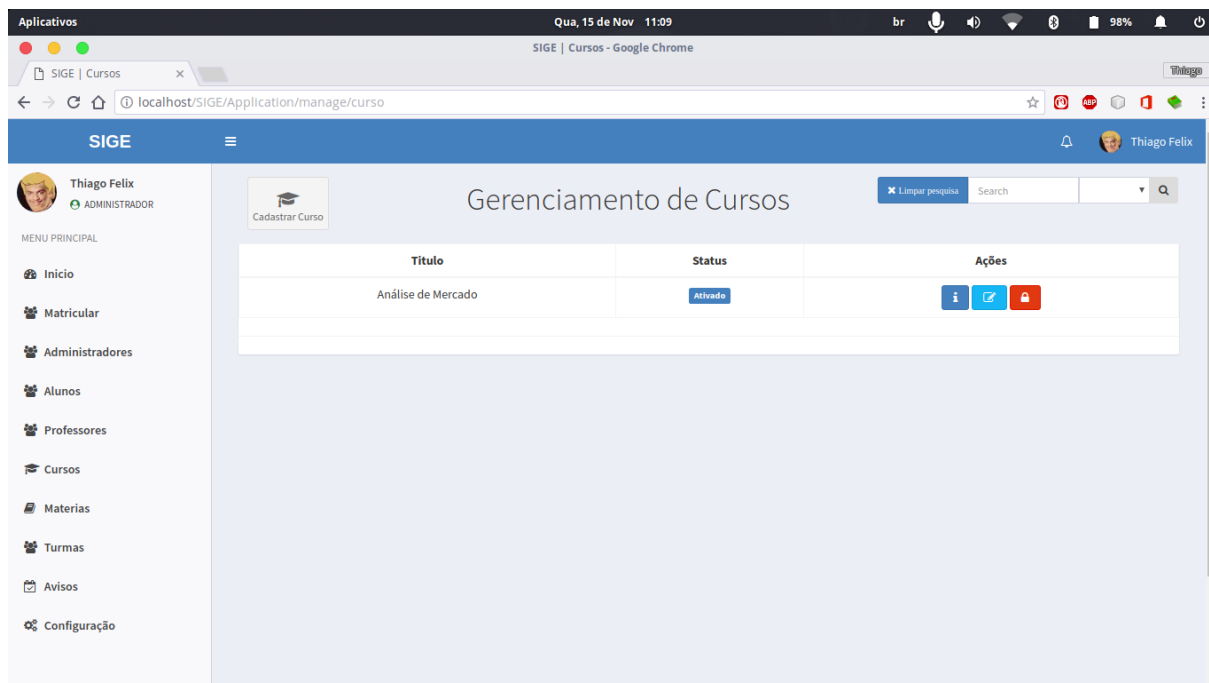
A tela de gerenciamento de administradores pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

No gerenciamento de administradores é possível acessar o cadastro de administrador, a lista de todos os administradores do sistema e para cada administrador é possível visualizar o perfil, alterar, desativar e ativar.

A desativação do administrador o impedirá de entrar no sistema e sua ativação o permitirá.

## 4.6.12 Gerenciamento de Cursos

**Figura 62:** Tela de gerenciamento de cursos



**Fonte:** A autoria do Autor.

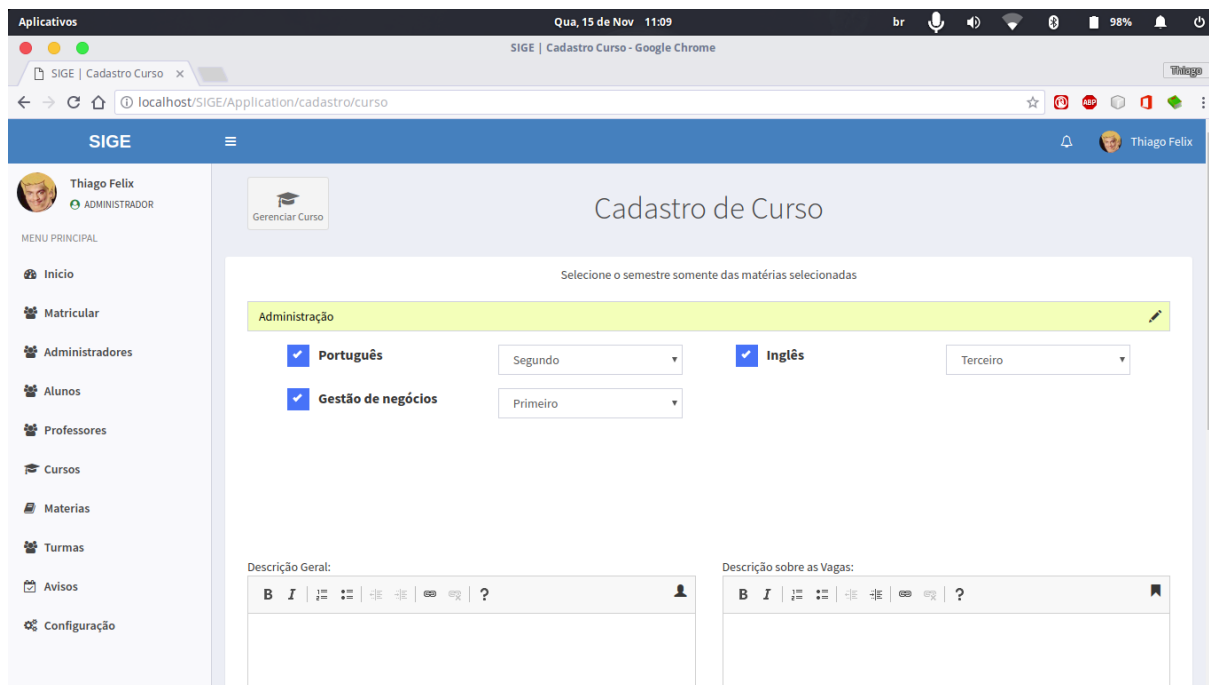
A tela de gerenciamento de cursos pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

No gerenciamento de cursos é possível acessar o cadastro de curso, a lista de todos os cursos do sistema e para cada curso é possível visualizar suas informações, alterar, desativar e ativar.

A desativação do curso o impedirá qualquer aluno de se matricular neste curso e cancelará todas as turmas deste curso que estiverem ativos no momento.

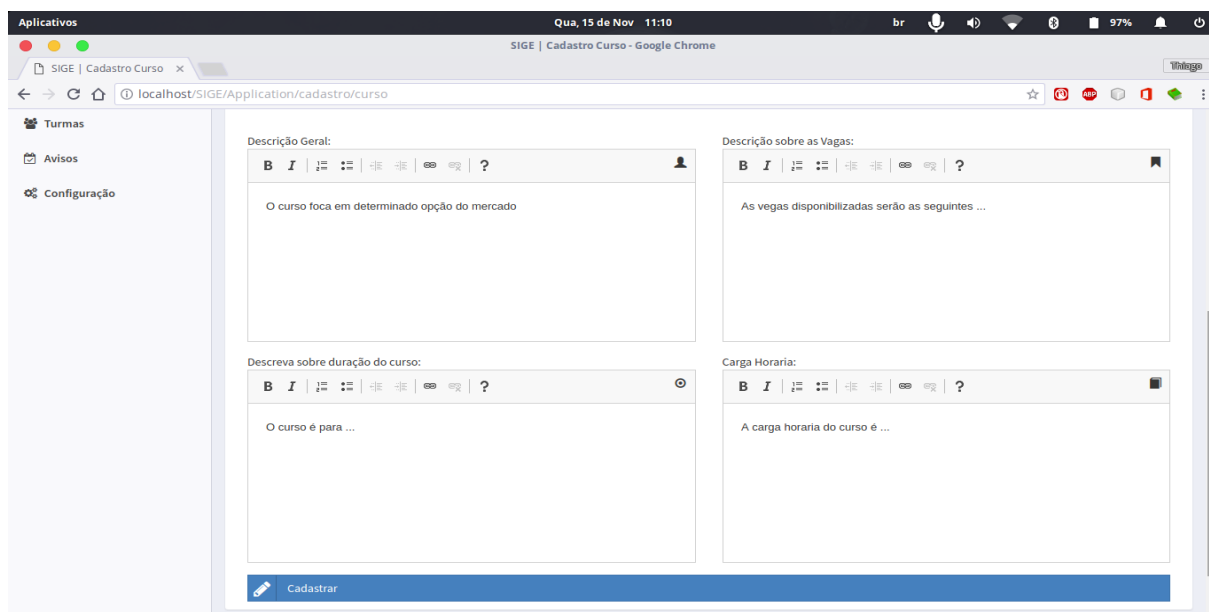
## 4.6.13 Cadastro de Curso

Figura 63: Tela de cadastro de curso



Fonte: A autoria do Autor.

Figura 64: Tela de cadastro de curso 2



Fonte: A autoria do Autor.

No canto superior esquerdo do gerenciamento de cursos é possível acessar o cadastro de curso.

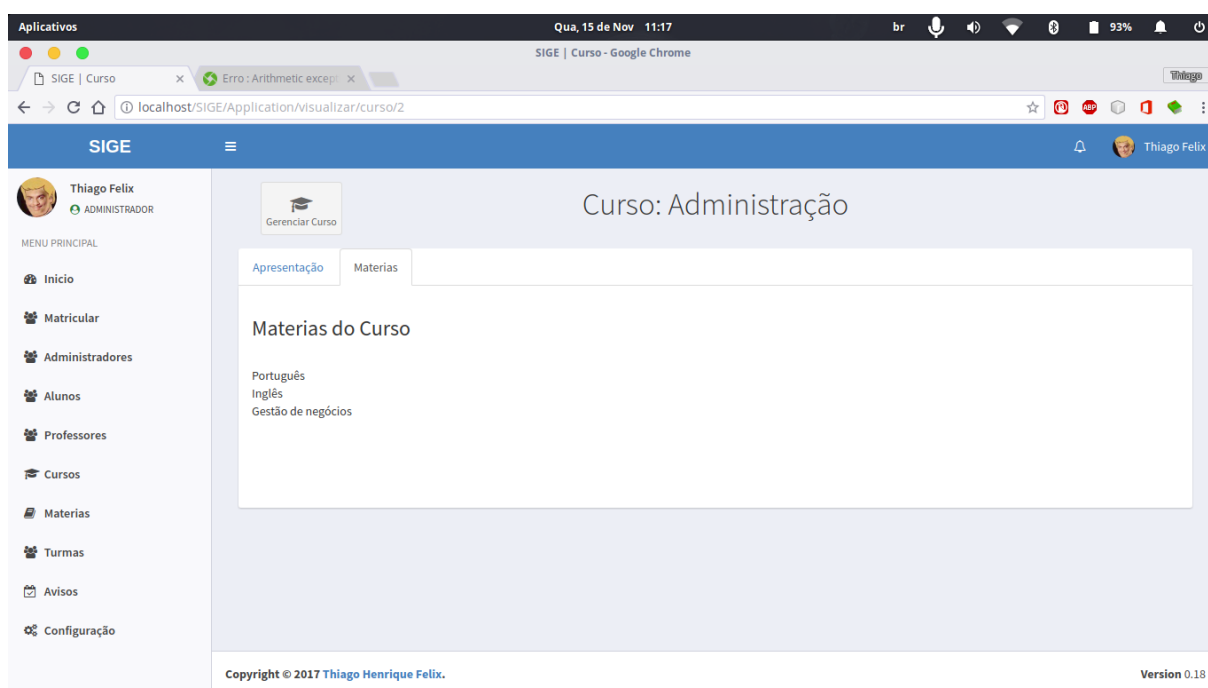
Todos os campos com o símbolo (\*) são obrigatórios e se não preenchidos o sistema irá informar o usuário que um erro ocorreu.

Para cadastrar um novo curso é necessário selecionar as matérias do curso se caso não haver nenhuma matéria cadastrada no sistema, o sistema irá informar que não é possível prosseguir sem o cadastro de no mínimo uma matéria no sistema. Ao selecionar uma matéria é necessário informar qual o semestre no qual aquela matéria será exigida.

Após o cadastro do curso o usuário será redirecionado para o gerenciamento de cursos e o sistema irá informar que a matéria foi cadastrada com sucesso ou não.

#### 4.6.14 Informações do Curso

**Figura 65: Tela de informações do curso**

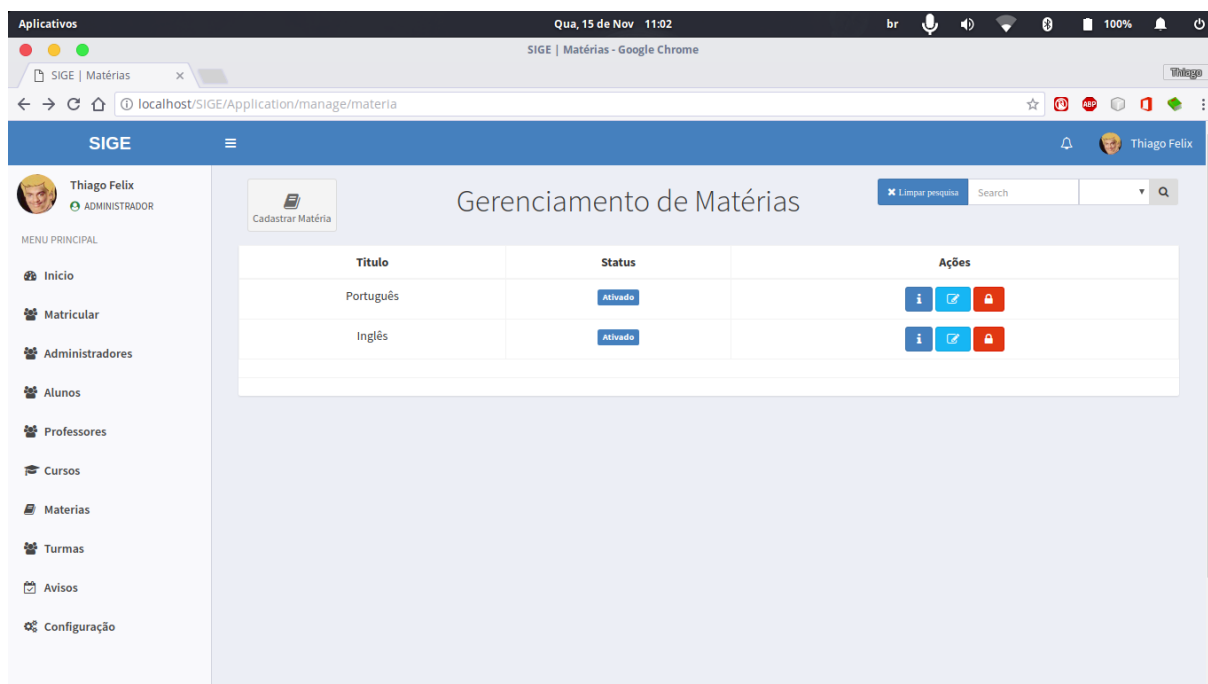


**Fonte: A autoria do Autor.**

As informações do curso são acessíveis na primeira opção do item “ação” da lista de cursos na tela de gerenciamento de cursos redirecionando para a tela de informações do curso.

#### 4.6.15 Gerenciamento de Matérias

**Figura 66: Tela de gerenciamento de matérias**



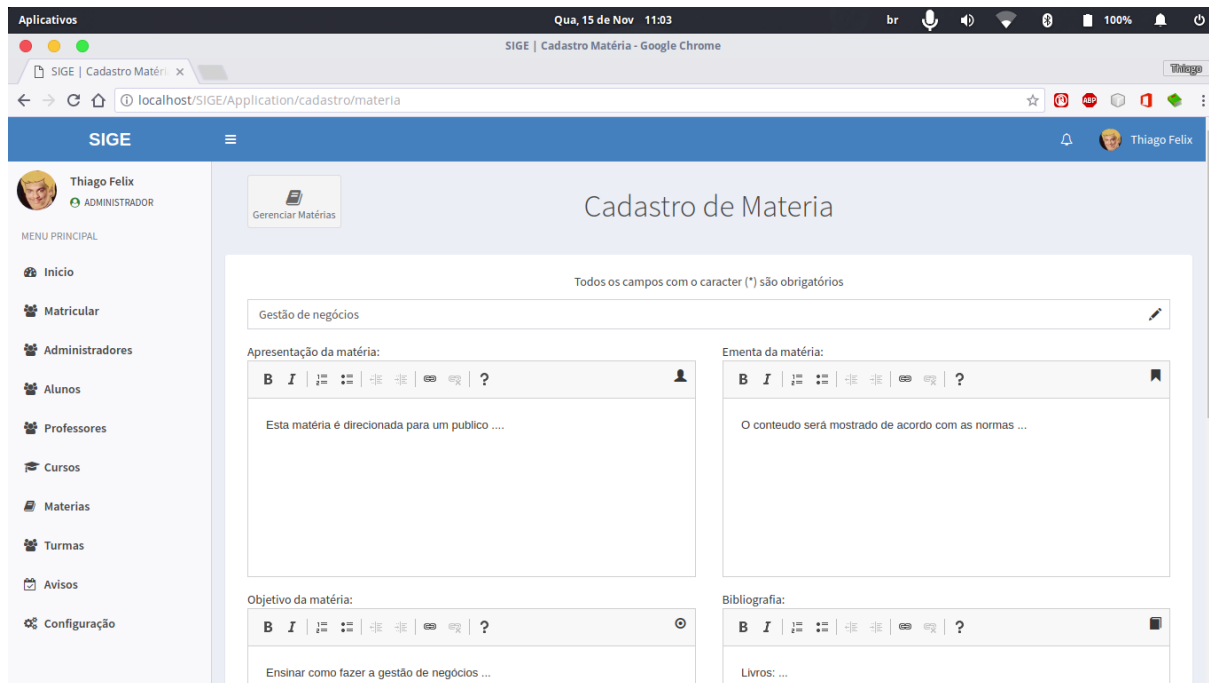
**Fonte: A autoria do Autor.**

A tela de gerenciamento de matérias pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

No gerenciamento de matérias é possível acessar o cadastro de matéria, a lista de todas as matérias do sistema e para cada matéria é possível visualizar suas informações, alterar, desativar e ativar.

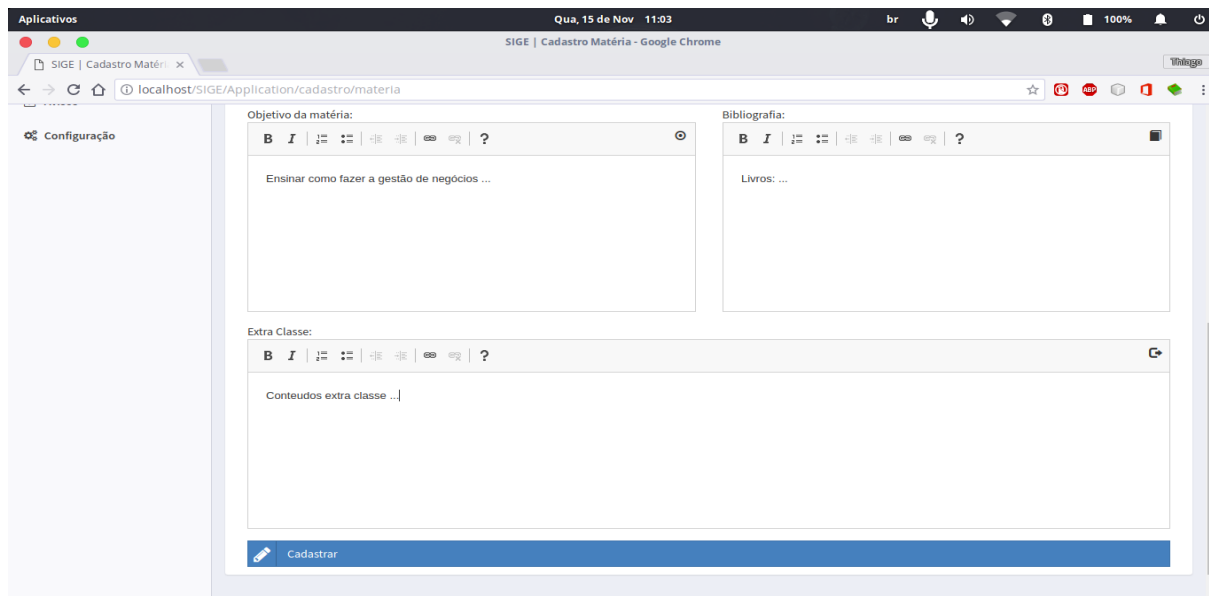
## 4.6.16 Cadastro de Matéria

**Figura 67: Tela de cadastro de matéria**



Fonte: A autoria do Autor.

**Figura 68: Tela de cadastro de matéria 2**



Fonte: A autoria do Autor.

Nesta sessão será demonstrado como realizar o cadastro de uma nova matéria no sistema.

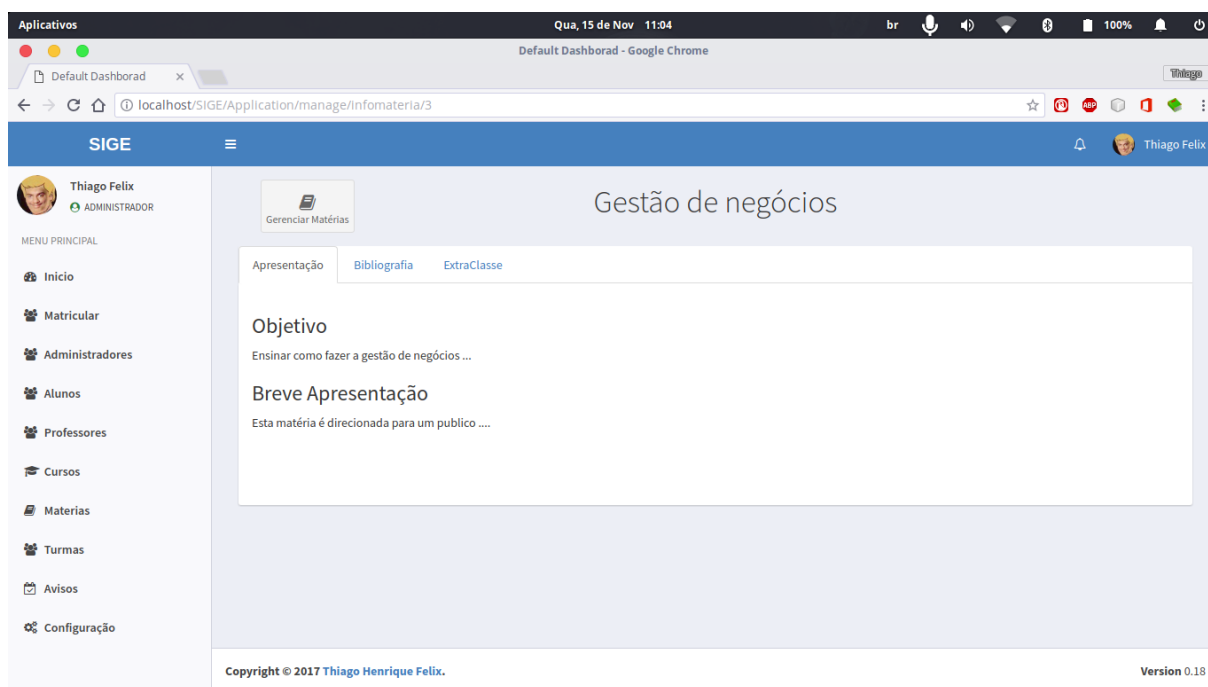


Para acessar o cadastro de matéria é necessário primeiro acessar o gerenciamento de matérias.

O cadastro de matéria somente exige que seja inserido o título da matéria os outros campos são opcionais, porém mesmo com o sistema permitindo a escolha de não preenchimento dos campos é recomendado inserir essas informações já que as mesmas são importantes para entender o objetivo da matéria.

#### 4.6.17 Informações da Matéria

**Figura 69: Tela de informações da Matéria**

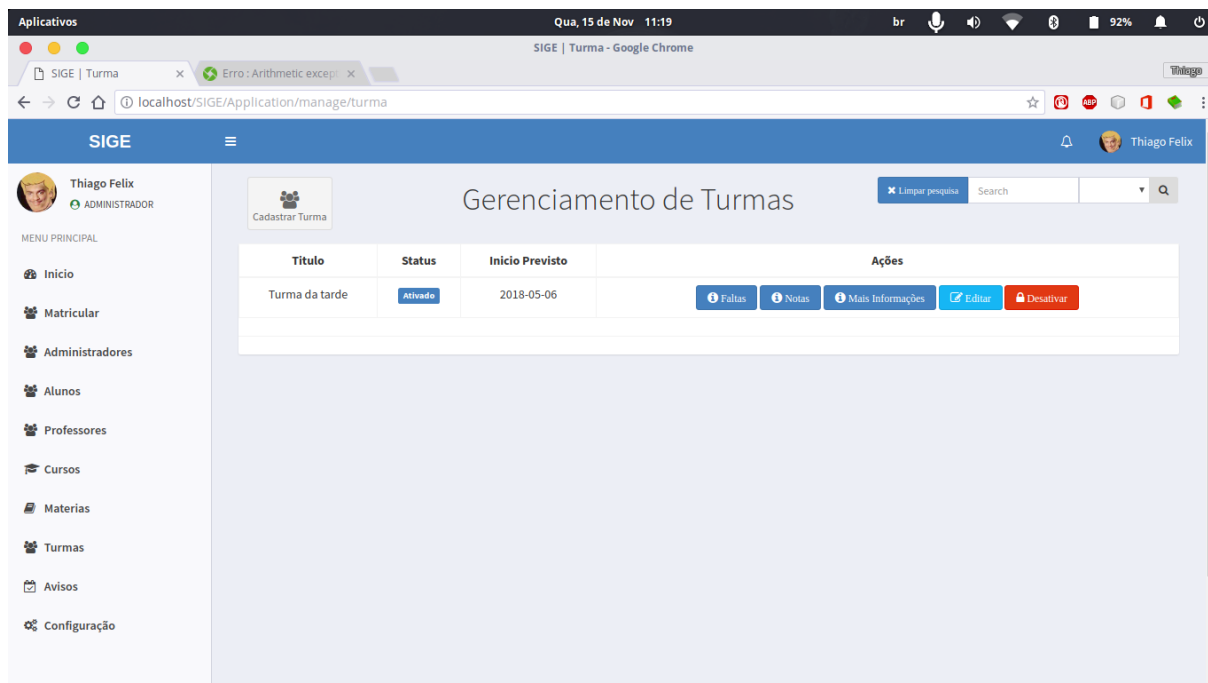


**Fonte: Autoria do Autor.**

As informações da matéria são acessíveis na primeira opção do item “ação” da lista de matérias na tela de gerenciamento de matérias redirecionando para a tela de informações do curso.

## 4.6.18 Gerenciamento de Turmas

**Figura 70: Tela de gerenciamento de turmas**



**Fonte: Aatoria do Autor.**

A tela de gerenciamento de turmas pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

Nesta tela é possível acessar todas as funcionalidades referentes as turmas como o cadastro de turma, alteração de turmas, informações da turma, gerenciamento de notas da turma, gerenciamento de frequências da turma, ativação da turma e desativação da turma.

## 4.6.19 Cadastro de Turma

Figura 71: Tela de cadastro de turma

Fonte: Aatoria do Autor.

Figura 72: Tela de cadastro de turma 2

Fonte: Aatoria do Autor.

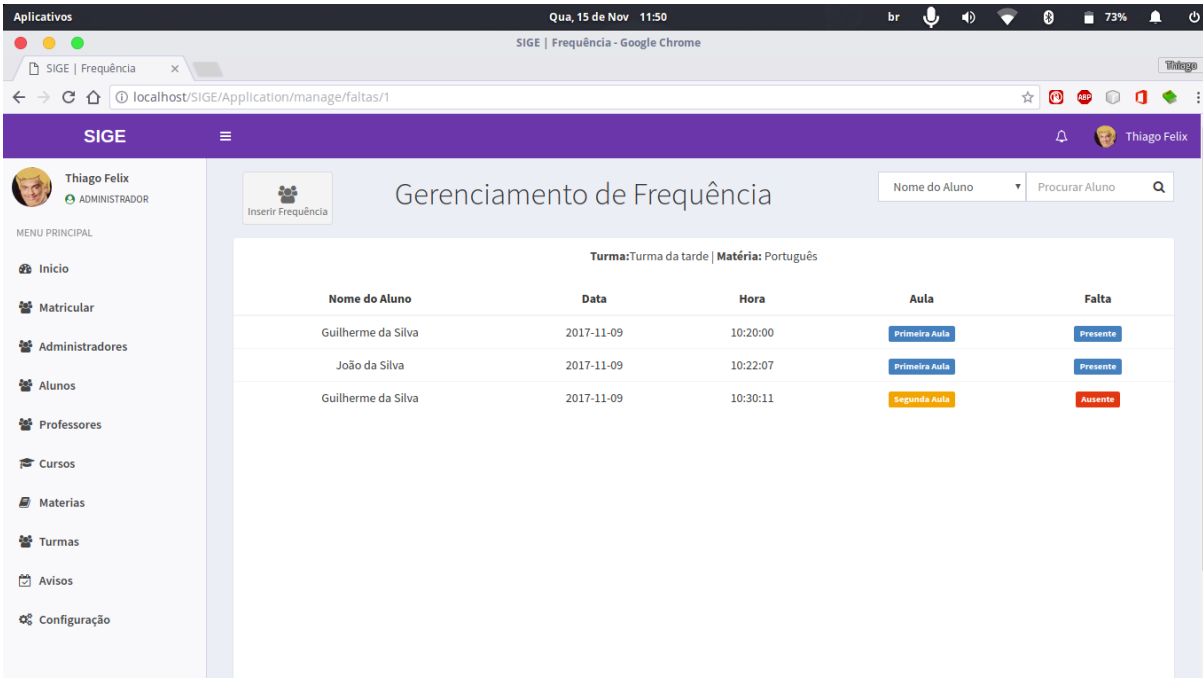
O cadastro de turma está disponível no gerenciamento de turmas e somente pode ser acessado pelo administrador do sistema.

Para cadastrar uma nova turma é necessário ter ao menos uma matéria cadastrada no sistema item (4.6.9) e um professor caso contrário o sistema irá informar que é necessário cadastra-los primeiro.

Todos os campos do com o símbolo (\*) são obrigatórios e se não forem cadastrados o sistema irá retornar um erro.

#### 4.6.20 Gerenciamento de Frequência

**Figura 73: Tela de gerenciamento de frequência**



The screenshot shows the 'Gerenciamento de Frequência' page in the SIGE system. The page title is 'Gerenciamento de Frequência' and the sub-header is 'Turma: Turma da tarde | Matéria: Português'. There is a search bar for 'Nome do Aluno' and a search button 'Procurar Aluno'. Below the search bar is a table with the following data:

Nome do Aluno	Data	Hora	Aula	Falta
Guilherme da Silva	2017-11-09	10:20:00	Primeira Aula	Presente
João da Silva	2017-11-09	10:22:07	Primeira Aula	Presente
Guilherme da Silva	2017-11-09	10:30:11	Segunda Aula	Ausente

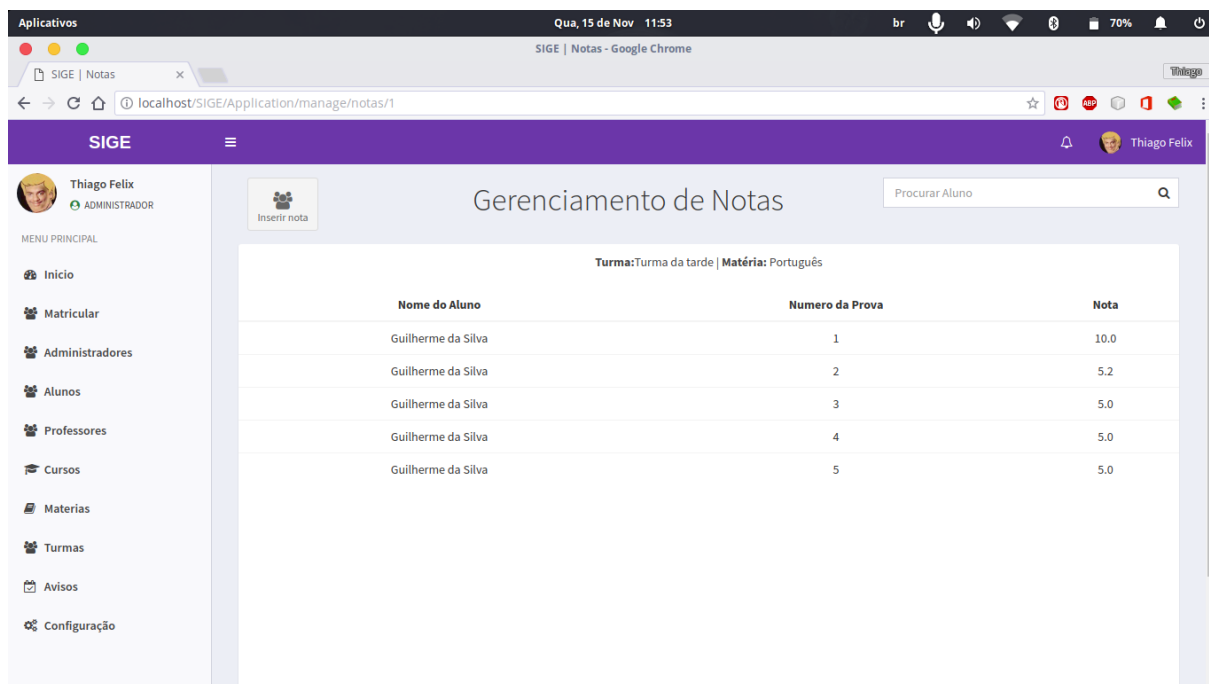
**Fonte: A autoria do Autor.**

O gerenciamento de frequência pode ser acessado no gerenciamento de turmas item (4.6.10).

Nesta tela é possível acessar o cadastro de frequência e visualizar a lista de frequência dos alunos.

## 4.6.21 Gerenciamento de Notas

**Figura 74: Tela de gerenciamento de notas**



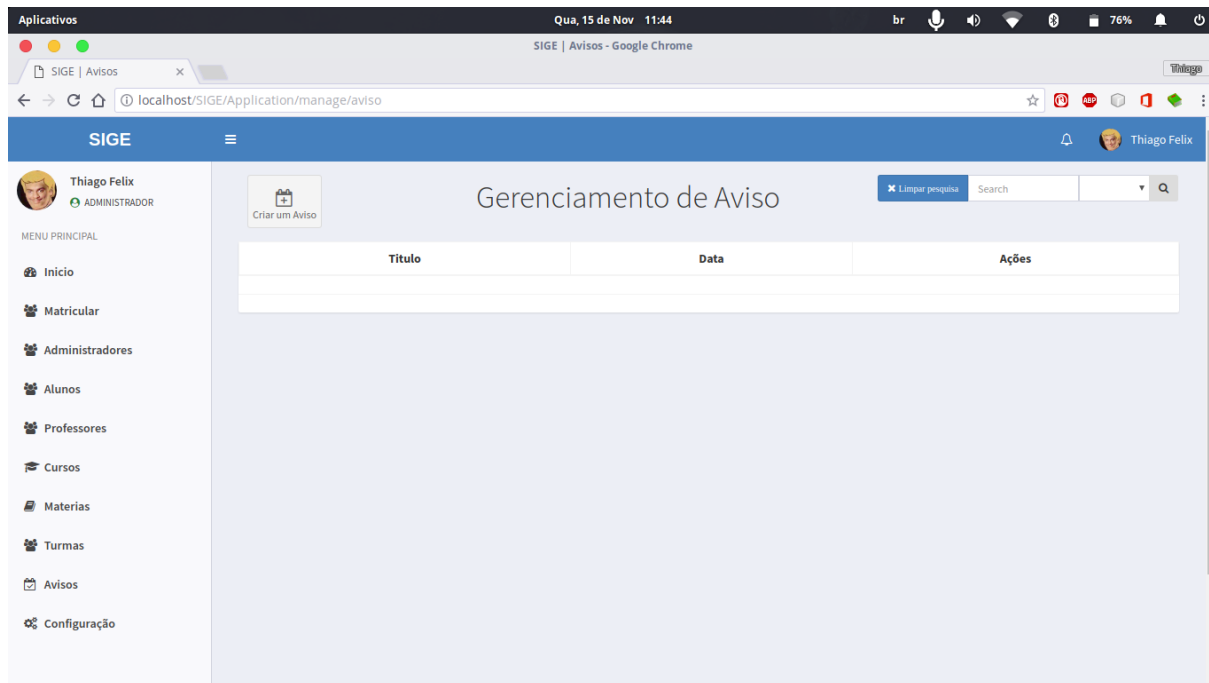
**Fonte: Autoria do Autor.**

O gerenciamento de notas pode ser acessado no gerenciamento de turmas item (4.6.10).

Nesta tela é possível acessar o cadastro de notas e visualizar a lista de nota dos alunos.

## 4.6.22 Gerenciamento de Avisos

**Figura 75: Tela de Gerenciamento de avisos**



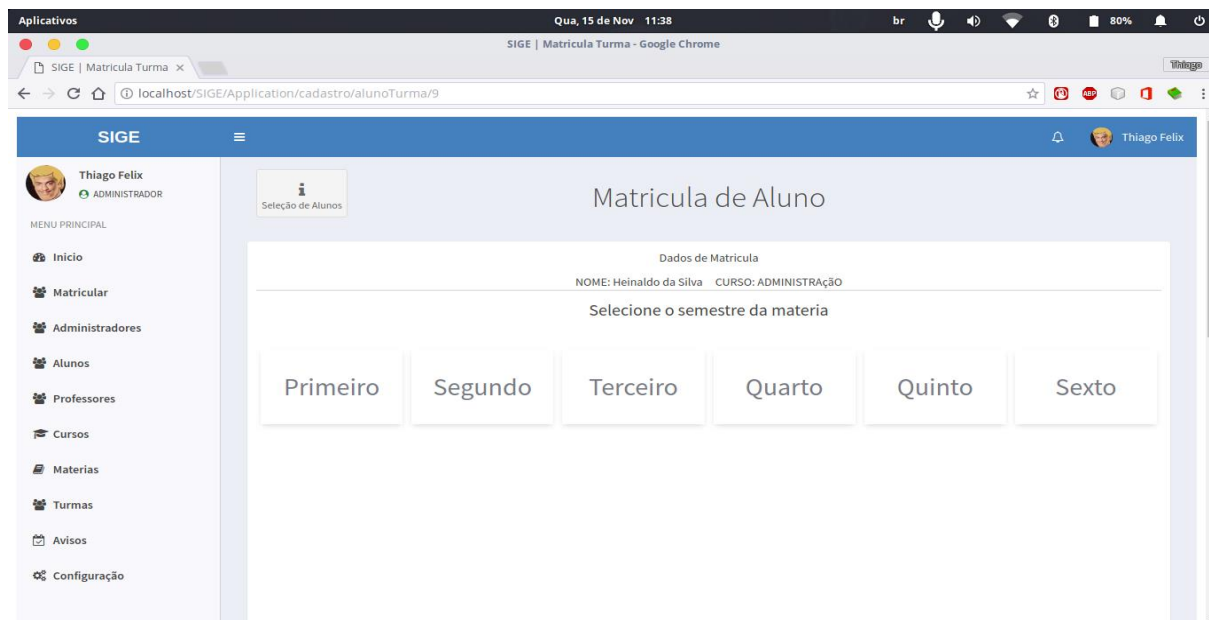
**Fonte: A autoria do Autor.**

A tela de gerenciamento de avisos pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

No gerenciamento de avisos é possível acessar o cadastro de aviso, a lista de todos os avisos do sistema e para cada aviso é possível visualizar informações e deletar.

#### 4.6.23 Matricula de Aluno em Turma

**Figura 76: Tela de matricula do aluno em turmas**



**Fonte: Aatoria do Autor.**

A tela de matricula de aluno pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

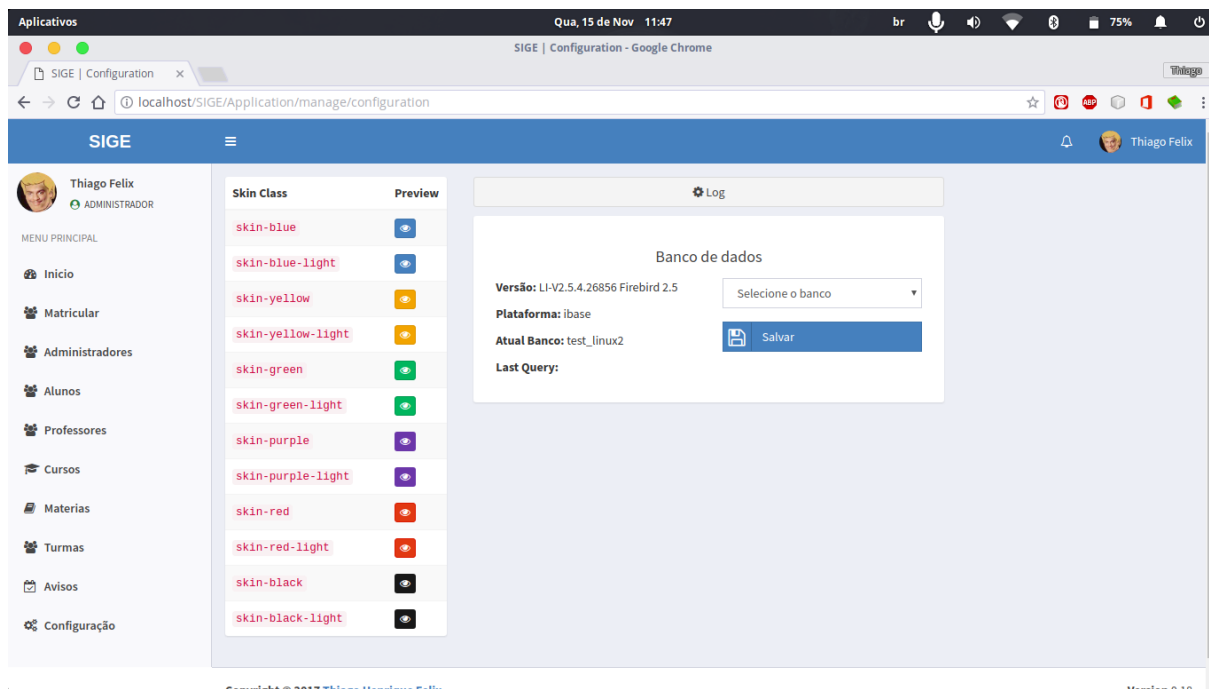
Nesta tela é onde é realizada a matricula do aluno em uma turma, uma turma é a representação de uma matéria e podem existir turmas de diversos horários.

O aluno não pode ser matriculado em mais de uma turma por matéria, somente é permitido a matricula em turmas de matérias do curso do aluno.

Para cadastrar basta selecionar o semestre da turma, em seguida o sistema irá listar todas as matérias desse semestre e para cada irá disponibilizar as turmas. Ao escolher a turma basta selecionar e por fim confirmar.

## 4.6.24 Configurações do Sistema

Figura 77: Tela de configurações do sistema



Fonte: A autoria do Autor.

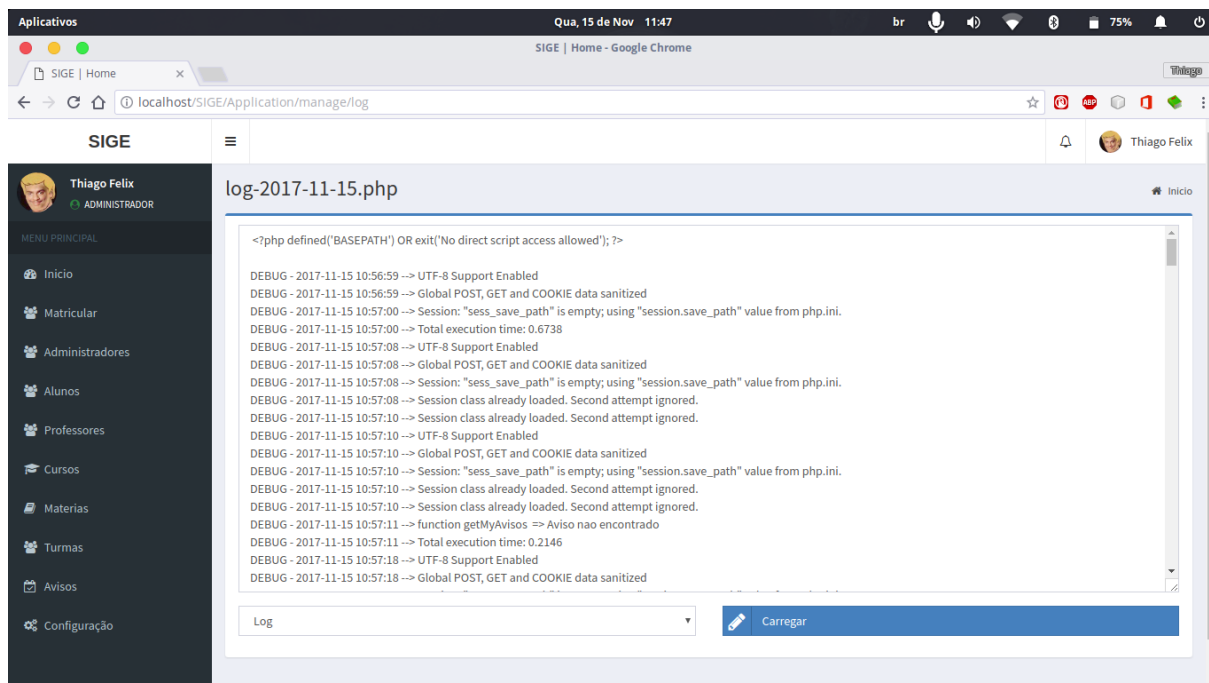
A tela de configurações pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade pode ser acessada por todos, mas algumas funcionalidades somente estão disponíveis para o administrador do sistema.

Nesta tela é possível mudar o tema do sistema, ver informações do banco de dados, mudar o banco de dados e acessar os logs do sistema.



## 4.6.25 Log do Sistema

Figura 78: Tela de log do sistema



Fonte: A autoria do Autor.

A tela de log pode ser acessada pela tela inicial do administrador (Item 4.6.3) ou pelo menu lateral, esta funcionalidade somente pode ser acessada pelo administrador do sistema.

Nesta tela é possível acessar todos os logs do sistema através de sua data.

## CONSIDERAÇÕES FINAIS

O principal objetivo deste trabalho foi desenvolver um sistema de gerenciamento escolar com a finalidade de auxiliar as funções básicas do dia a dia escolar. A principal motivação é devido a muitas escolas mesmo nos dias de hoje ainda não utilizarem o computador como ferramenta para otimizar as tarefas a nível gerencial do dia a dia escolar. A ferramenta desenvolvida faz o gerenciamento de professores, administradores, alunos, matérias, cursos, avisos e turmas completando assim o gerenciamento básico de uma escola.

O modelo de processo de *software* utilizado foi o incremental, ou seja, eram definidos requisitos destes requisitos se criavam os serviços e a cada incremento se desenvolvia uma determinada quantia de serviços. Este modelo foi escolhido devido a inexperiência no desenvolvimento de *software* tendo criado anteriormente apenas *softwares* com o objetivo de se aprofundar nos estudos. O processo de análise de requisitos acabou que se tornando trabalhoso devido a inclusão ou revisão de novos requisitos a cada implementação havendo a necessidade em rever os requisitos anteriores. Devido à falta de conhecimento de algumas ferramentas como o *framework backend* Codeigniter, *framework frontend* AdminLTE e o modelo MVC o desenvolvimento da plataforma foi demorado, mas com ganho de conhecimento, sendo aprimorado a cada fase do projeto. A maioria das funções do sistema foi desenvolvida com sucesso, porém algumas não puderam ser desenvolvidas devido à falta de tempo como por exemplo a geração de notas finais do aluno e a liberação do aluno para o cadastro de um novo curso, também não pode ser desenvolvido a lista de turmas do professor, bem como outros módulos, sendo que o modulo principal, “administrador”, está disponível. O principal objetivo deste trabalho foi alcançado sendo desenvolvido uma plataforma com objetivo de auxiliar as funções básicas de uma escola como o gerenciamento de alunos, curso, matérias, etc. No desenvolvimento deste trabalho obtive uma grande experiência no desenvolvimento de *software*, uma parte dessa experiência foi na fase de requisitos que mesmo utilizando um método ágil cujo o objetivo é o desenvolvimento rápido, os requisitos devem ser bem definidos e o objetivo principal do projeto deve ser estabelecido desde o início, pois no desenvolvimento do projeto novas ideias foram surgindo, porém o

tempo não permitia o desenvolvimento e assim funções antes estabelecidas acabaram por ser atrasadas.

Quanto ao futuro pretendo dar continuidade ao projeto de forma a atender plenamente podendo talvez em algum momento servir de referência para minha carreira profissional.

## REFERÊNCIAS

**Começando com CodeIgniter (Getting Started With CodeIgniter).** Disponível em: [https://www.codeigniter.com/user\\_guide/overview/getting\\_started.html](https://www.codeigniter.com/user_guide/overview/getting_started.html). Acessado em: 12 ago. 2017, às 9h32min.

**Firebird: Poderoso, leve, gratuito e Open-Source.** Disponível em: <https://www.devmedia.com.br/firebird-poderoso-leve-gratuito-e-open-source-revista-clubedelphi-135-parte-1/22887>. Acessado em: 17 nov. 2017, às 11h02min.

GUEDES, Gilleanes T.A. **UML2: uma abordagem prática.** São Paulo/SP: Novatec, 2009.

**História do PHP.** Disponível em: [https://secure.php.net/manual/pt\\_BR/history.php.php](https://secure.php.net/manual/pt_BR/history.php.php). Acessado em: 20 set. 2017, às 18h28min.

OGLIO, Pablo Dall'. **PHP: programando com orientação a objetos.** São Paulo/SP: Novatec, 2007.

OLIVA, Alexandre et al (Org.). **A revolução do Software Livre.** Manaus: Comunidade Sol, 2009. 1-58 p.

SOMMERVILLE, Ian. **Engenharia de Software.** Trad. Selma Shin Shimizu Melnikoff. 8ªed. São Paulo/SP: Pearson, 2007.

STAIR, Ralph M.; REYNOLDS, George W. **Princípios de Sistemas de Informação.** Trad. Harue Avritscher. 9ª. ed. São Paulo/SP: Cengage Learning, 2011. p. 1-20.

TELES, Vinicius Manhães. **Scrum.** 2015. Disponível em: <http://www.desenvolvimentoagil.com.br/scrum>. Acessado em: 15 out. 2017, às 13h05min.