

UNIVERSIDADE FEDERAL FLUMINENSE
DANIEL BARBOSA PEREIRA

**O DESENVOLVIMENTO DE UM SISTEMA ONLINE DE GERENCIA-
MENTO ESCOLAR**

Niterói
2019

DANIEL BARBOSA PEREIRA

**O DESENVOLVIMENTO DE UM SISTEMA ONLINE DE GERENCIA-
MENTO ESCOLAR**

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Orientador:

Jean Zahn

NITERÓI

2019

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

P436d Pereira, Daniel Barbosa
 O DESENVOLVIMENTO DE UM SISTEMA ONLINE DE GERENCIAMENTO
 ESCOLAR / Daniel Barbosa Pereira ; Jean Zahn, orientador.
 Niterói, 2019.
 90 p. : il.

 Trabalho de Conclusão de Curso (Graduação em Tecnologia
 de Sistemas de Computação)-Universidade Federal Fluminense,
 Instituto de Computação, Niterói, 2019.

 1. Aplicação web. 2. Produção intelectual.I. Zahn, Jean,
 orientador. II. Universidade Federal Fluminense. Instituto de
 Computação. III. Título.

CDD -

DANIEL BARBOSA PEREIRA

**O DESENVOLVIMENTO DE UM SISTEMA ONLINE DE GERENCIA-
MENTO ESCOLAR**

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Niterói, 02 de julho de 2019.

Banca Examinadora:

Prof. Jean de Oliveira Zahn, MSc. – Orientador
UFF – Universidade Federal Fluminense

Prof. Douglas Paulo de Mattos, MSc. – Orientador
UFF – Universidade Federal Fluminense

“Dedico este trabalho primeiramente a Deus,
por ser essencial em minha vida, meu guia,
ao meu pai Marcelo, minha mãe Iara e aos
meus irmãos.”

AGRADECIMENTOS

A Deus, por ter me dado saúde e força para superar as dificuldades.

A meu Orientador Jean Zahn pelo suporte, correções e incentivo que me concedeu durante o curso.

Agradeço aos meus pais, pela confiança e investimentos empregados nos meus estudos, e por estarem sempre compartilhando e aconselhando nos meus momentos de alegrias e tristezas.

E a todos que direta ou indiretamente fizeram parte de minha formação, o meu muito obrigado

“Aposto nas pessoas ousadas, cultas e que são criativas, porque elas conseguem inspirar outras para gerenciar novos trabalhos, buscar novas alternativas ou abrir novos empreendimentos”.

Helgir Girodo

RESUMO

A supremacia do papel está com seus dias contados quando o assunto é melhorar as tarefas diárias de escolas. A Internet e suas ferramentas estão revolucionando o mundo. Porém, a verdade é que, mesmo diante desse cenário promissor, muitas escolas não aderiram a essas tecnologias digitais. Nesse cenário é comum que a secretaria seja o foco de alguns problemas. Com fichários, atas, pastas, prontuários formando pilhas na mesa do gestor, além de um arquivo morto ocupando espaço, faz com que o espaço fique desorganizado, os funcionários fiquem sobrecarregados, documentos são perdidos e o andamento da rotina escolar sofre atrasos. Dessa forma, esse trabalho propõe implementar um Sistema de Gerenciamento Escolar a fim de reduzir fastidioso trabalho de suas secretarias além de fornecer informações relevantes a professores e alunos. Para isto, este trabalho seguiu as etapas propostas pela Engenharia de Software do modelo incremental de ciclo de vida, em que primeiramente foi feita a análise e o levantamento de requisitos através da técnica de entrevistas. Em seguida o sistema foi modelado usando as definições da linguagem UML. E através dessas definições, foram elaborados diagramas padronizados que descrevem funcionalidades do sistema, finalizando-se então, em sua implementação.

Palavras-chaves: Sistema Escolar, Aplicação WEB, Modelo Incremental, MVC.

LISTA DE ILUSTRAÇÕES

Figura 1: Modelo incremental	22
Figura 2: Diagramas UML	24
Figura 3: Representação da comunicação do modelo MVC	25
Figura 4: Diagrama de caso de uso do Administrador	33
Figura 5: Diagrama de caso de uso do Professor	34
Figura 6: Diagrama de caso de uso do Aluno	34
Figura 7: Arquitetura e fluxo de dados do CodeIgniter	48
Figura 8: XAMPP	50
Figura 9: Template AdminLTE	51
Figura 10: Interface PhpStorm	52
Figura 11: Diagrama conceitual de classes	53
Figura 12: Diagrama de classes de controle	55
Figura 13: Diagrama de Classes do Administrador_Controller	56
Figura 14: Diagrama de classes da camada Modelo	58
Figura 15: Modelo do Banco de Dados	60
Figura 16: Interface de Autenticação	61
Figura 17: Telas de Recuperação de Senha	62
Figura 18: Tela Inicial da Administração	63
Figura 19: Tela de listagem de Administradores	64
Figura 20: Tela de listagem de Turmas	65
Figura 21: Tela de listagem de Disciplinas	65
Figura 22: Tela de listagem de Estudantes	66
Figura 23: Tela de listagem de Exames	67
Figura 24: Tela de cadastro de Administradores	68
Figura 25: Tela de cadastro de Turmas	69
Figura 26: Tela de cadastro de Disciplinas	70
Figura 27: Tela de cadastro de Estudantes	71
Figura 28: Tela de cadastro de Professores	72

Figura 29: Tela de cadastro de Exames.....	73
Figura 30: Tela resultado Acadêmico	74
Figura 31: Tela de gerenciamento de Notas	75
Figura 32: Tela principal de Frequência	76
Figura 33: Tela de visualização Frequência escolar	77
Figura 34: Tela de Cadastro da Frequência Escolar	77
Figura 35: Página inicial do Estudante	78
Figura 36: Tela Meus Dados	79
Figura 37: Tela Minhas Disciplinas.....	80
Figura 38: Tela Exames Agendados	80
Figura 39: Tela Avaliação Acadêmica	81
Figura 40: Tela Minha Frequência.....	82

LISTA DE TABELAS

Tabela 1: Caso de uso - Autenticar	36
Tabela 2: Caso de uso - Recuperar Senha	37
Tabela 3: Caso de uso – Gerenciamento de Administradores	38
Tabela 4: Caso de uso - Gerenciamento de Professores.....	39
Tabela 5: Caso de uso - Gerenciamento de Alunos.....	40
Tabela 6: Caso de uso - Gerenciamento de Turmas.....	40
Tabela 7: Caso de uso - Gerenciamento de Disciplinas.....	41
Tabela 8: Caso de uso - Gerenciamento de Exames.....	42
Tabela 9: Caso de uso - Atribuir as Notas dos Exames	43
Tabela 10: Caso de uso - Controlar a Frequência dos Alunos	44

LISTA DE GRÁFICOS

Gráfico 1: Estatísticas de frameworks PHP utilizados em projetos	47
--	----

LISTA DE ABREVIATURAS E SIGLAS

BD – Banco de dados

RF – Requisito fundamental

RN – Regra de negócio

RNF – Requisito não fundamental

UC – Caso de uso

UML – Linguagem de Modelagem Unificada

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVO GERAL	16
1.2	OBJETIVOS ESPECÍFICOS	17
1.3	JUSTIFICATIVA	17
1.4	ESTRUTURA DO TRABALHO	18
2	FUNDAMENTAÇÃO TEÓRICA.....	19
2.1	ENGENHARIA DE SOFTWARE.....	19
2.1.1	Modelo de Ciclo de Vida (Modelo de Processo).....	20
2.1.2	Modelo incremental evolutivo	20
2.2	MODELAGEM E UML	23
2.3	ARQUITETURA MVC.....	24
3	ANÁLISE E ESPECIFICAÇÃO DO PROJETO	27
3.1	LEVANTAMENTO DE REQUISITOS	27
3.1.1	Requisitos funcionais.....	28
3.1.2	Requisitos não funcionais.....	30
3.1.3	Regras de negócio	31
3.2	CASOS DE USO	32
3.2.1	Diagramas de Casos de Uso.....	33
3.2.2	Listagem de Casos de Uso	35
3.2.3	Descrição dos Casos de Uso	36
4	PROJETO E IMPLEMENTAÇÃO DO SISTEMA.....	46
4.1	FERRAMENTAS UTILIZADAS.....	46
4.1.1	Framework CodeIgniter	46
4.1.2	MySQL e MySQL Workbench.....	49
4.1.3	XAAMP (PHP, MySQL e Apache)	49
4.1.4	Bootstrap	50

4.1.5	JetBrains PhpStorm.....	51
4.2	DIAGRAMA DE CLASSES CONCEITUAL	52
4.3	DIAGRAMA DE CLASSES DE PROJETO	53
4.4	MODELO LÓGICO DE DADOS.....	59
4.5	INTERFACE DO SISTEMA	61
4.5.1	Tela de Login e Recuperação de Senha	61
4.5.2	Tela Principal da Administração	63
4.5.3	Interface de Listagem Genérica	63
4.5.4	Tela de cadastro de Administradores	67
4.5.5	Tela de cadastro de Turmas.....	68
4.5.6	Tela de cadastro de Disciplinas.....	69
4.5.7	Tela de cadastro de Estudantes	70
4.5.8	Tela de cadastro de Professores.....	71
4.5.9	Tela de cadastro de Estudantes	72
4.5.10	Interface de Resultado Acadêmico.....	73
4.5.11	Interface de Frequência.....	75
4.5.12	Interfaces do módulo Estudante	78
5	CONCLUSÕES E TRABALHOS FUTUROS.....	83
	REFERÊNCIAS BIBLIOGRÁFICAS	85
	APÊNDICE A – CÓDIGO DE CRIAÇÃO DAS TABELAS DO BD	87

1 INTRODUÇÃO

A informatização da Secretaria escolar é de vital importância nos dias de hoje, sendo este o setor da instituição responsável pela manutenção de registros, os arquivos dos registros de alunos e funcionários, controle de notas, frequências, além de comunicados e expedições de forma a apoiar o processo de desenvolvimento escolar. Ou seja, é a Secretaria a responsável por todos os eventos burocráticos e legais de funcionamento da instituição.

No caso de instituições de ensino de grande, médio ou mesmo as de pequeno porte, com séries variadas, a informatização da secretaria é uma necessidade. Além de facilitar a vida da comunidade acadêmica, o uso de softwares específicos acaba por estreitar a relação com os alunos e suas famílias.

Porém, apesar de sua importância, a verdade é que, muitas vezes a ela é negligenciada. Mas a partir do momento que estudantes precisam buscar por documentos ou marcar uma reunião, a secretaria escolar exerce um papel fundamental no gerenciamento da escola de uma forma geral.

Ambientado nesse cenário, o presente trabalho propõe o desenvolvimento de um sistema de gestão para informatizar a secretaria de modo a trazer agilidade, integração e uma maior fidelidade aos conteúdos produzidos.

1.1 OBJETIVO GERAL

O principal objetivo deste trabalho é analisar, modelar e implementar um sistema informatizado de gestão escolar, a ser operado em ambiente web e que apoie e facilite a vida de alunos, professores e funcionários, além de fornecer informações relevantes sobre a vida acadêmica dos estudantes.

1.2 OBJETIVOS ESPECÍFICOS

Para chegar ao objetivo geral, é necessário estabelecer alguns objetivos específicos, destacando-se dentre eles:

- Definir as atividades que serão gerenciadas pelo sistema a ser feito;
- Realizar a análise de requisitos;
- Realizar a análise da documentação de requisitos, identificando as características essenciais para desenvolver a estrutura básica do software de gerenciamento escolar;
- Modelar os diagramas do sistema;
- Descrever os diagramas;
- Realizar testes num ambiente controlado (servidor local).

1.3 JUSTIFICATIVA

A informatização da secretaria abre um leque de opções para a instituição pois o meio digital pode facilitar o acesso do histórico escolar do aluno, seu desempenho em cada disciplina o registro das notas, além de poder analisar qualquer dificuldade que possa ter longo de sua trajetória escolar dentro da instituição.

O mesmo pode ocorrer caso os professores queiram alguma informação com professores e/ou funcionários. Se essa pesquisa fosse realizada no modelo tradicional, por meio da busca física em inúmeras pastas e armários, horas serão desperdiçadas. Com a utilização de um software adequado, o usuário necessita realizar o login no sistema e consultar as informações instantaneamente.

Além disso, ele permite contribuir com a preservação do meio ambiente. Com a desburocratização de processos e as informações sendo registradas de maneira integrada e digitalmente, isso diminuiria a quantidade de papel utilizados e armazenados pelas escolas.

1.4 ESTRUTURA DO TRABALHO

No Capítulo 2, será apresentado o referencial teórico de alguns conceitos e técnicas que foram utilizadas no desenvolvimento deste sistema. Inicialmente damos uma breve descrição sobre o que é Engenharia de Software, adentrando pela definição de ciclos de vida e descrevendo o modelo utilizado neste projeto. Logo depois, mostramos a importância da modelagem utilizando a linguagem UML e a arquitetura MVC.

O Capítulo 3 é dedicado à análise e especificação do projeto. Nele são apresentados os atores do sistema, os requisitos funcionais e não funcionais, as regras de negócio, além dos diagramas de casos de uso com suas respectivas descrições.

O Capítulo 4 apresentará os detalhes da modelagem e a implementação do sistema. Em que primeiramente descrevemos as principais ferramentas utilizadas, passando para os diagramas de classe (conceituais e lógicos), pela modelagem de dados e finalizando na implementação da interface gráfica do sistema.

E finalmente, no Capítulo 5, são apresentadas as conclusões, possíveis melhorias e indicações de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O propósito desta seção é mostrar os conceitos básicos e as características que contextualizam tecnologicamente o projeto, dos documentos de especificação do software, além de discutir sobre as tecnologias utilizadas, de modo que o leitor tenha uma visão introdutória sobre os conceitos de cada um dos elementos fundamentais do projeto.

2.1 ENGENHARIA DE SOFTWARE

Segundo a **IEEE**, “Engenharia de software é a aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção do software; isto é, a aplicação de engenharia ao software”. [1]

E segundo Ariadne Maria:

É uma disciplina dentro da engenharia que reúne metodologias, métodos e ferramentas a ser utilizados, desde a percepção do problema até o momento em que o sistema desenvolvido deixa de ser operacional, visando resolver problemas inerentes ao processo de desenvolvimento e ao produto de software. [2]

Engenharia de Software utiliza o **Processo de Desenvolvimento**, que consiste na criação de documentos, artefatos e marcos, capazes de representar o contexto do software, levando em consideração recursos, ferramentas, prazos, restrições, e outros aspectos que envolvem o desenvolvimento de um produto de software, para no final produzir software de qualidade [3]. Neste contexto podemos definir:

- Um **método** é uma abordagem técnica que contem um passo a passo para realizar uma ou mais tarefas indicadas por um modelo de vida, visando otimizar a execução das atividades que foram especificadas.

- Um **processo de software** é um conjunto de passos ordenados, constituídos por atividades e resultados que levam a produção de um produto de software.
- Uma **ferramenta** automatizam e auxiliam os processos e os métodos.

2.1.1 Modelo de Ciclo de Vida (Modelo de Processo)

O ciclo de vida de um software indica todas as etapas do desenvolvimento de um software, ela contém processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema, desde a definição de seus requisitos até sua descontinuidade. [4]

A definição de um modelo de ciclo de vida (ou modelo de Processo) é o primeiro passo a ser feito no processo de desenvolvimento de um software. Os principais modelos podem ser agrupados em três categorias principais: **modelos sequenciais, modelos incrementais e modelos evolutivos**. Normalmente os modelos contemplam as seguintes fases:

- Análise e especificação de requisitos;
- Projeto de Software;
- Implementação;
- Testes;
- Entrega e Implantação.

2.1.2 Modelo incremental evolutivo

O modelo incremental pode ser visto como uma filosofia básica que comporta diversas variações. O seu princípio primordial é que o sistema é desenvolvido

para satisfazer um subconjunto dos requisitos especificados e, em versões posteriores, o programa é aumentado com novas funcionalidades que satisfazem mais requisitos. Sendo que a cada ciclo é entregue uma versão intermediária do produto para uso e avaliação detalhada do usuário. Para tal, requisitos têm de ser minimamente levantados e há de se constatar que o sistema é modular, de modo que se possa planejar o desenvolvimento em incrementos. [5]

Dessa forma, assume-se, de início, que o software agregará novas funcionalidades, e a que a cada nova funcionalidade ou conjunto de novas funcionalidades será um incremento. O primeiro incremento normalmente contém apenas as funcionalidades centrais, ou seja, apenas são atendidos os requisitos básicos para que o software entre em operação. Outras características são tratadas em ciclos subsequentes, como mostra a Figura 1.

Normalmente esse modelo é mais utilizado em projetos longos, em que novas funcionalidades são acrescentadas ao longo do tempo e o prazo de entrega é curto.

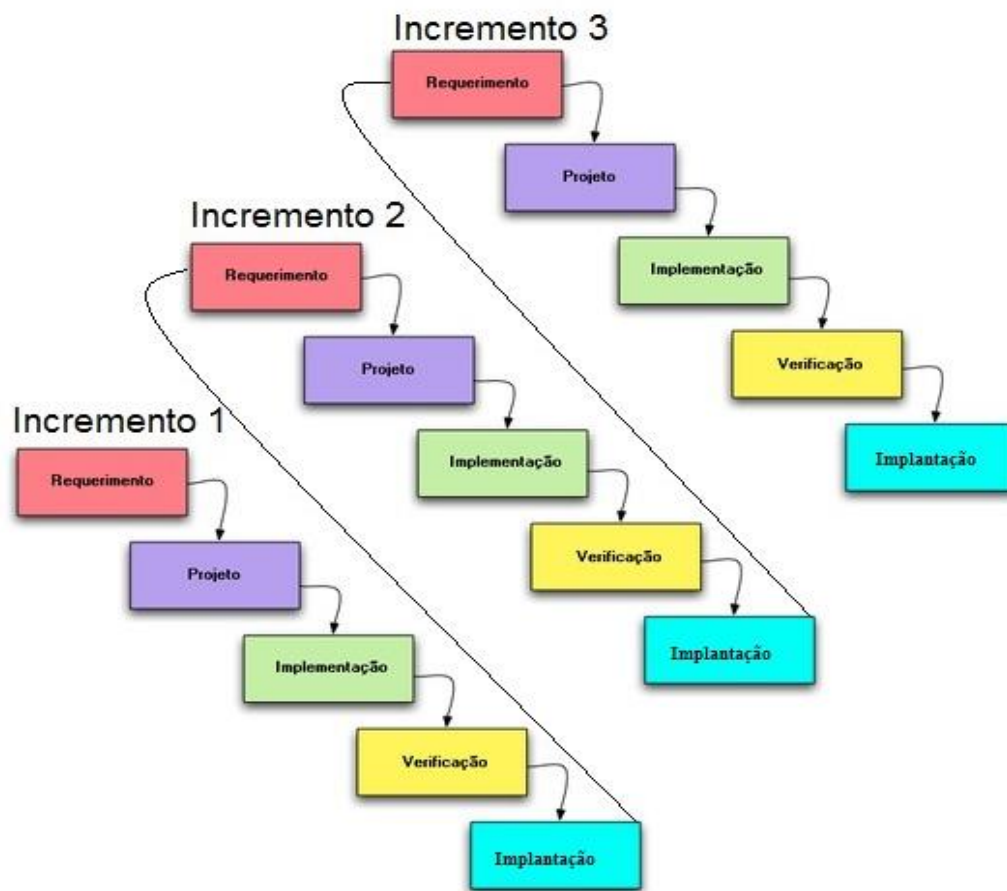


Figura 1: Modelo incremental

Existem três tipos incrementos:

- **Evolutivos:** Produtos de cada etapa de desenvolvimento são aproveitados em cada nova passagem pela etapa.
- **Descartáveis:** Produtos das etapas de desenvolvimento são descartados a cada novo protótipo é construído do início.
- **Operacional:** Requisitos são elucidados através de protótipos e o produto final é construído paralelamente a construção dos protótipos.

Neste trabalho será utilizado o incremento do tipo evolutivo.

2.2 MODELAGEM E UML

É importante ressaltar a importância da modelagem de sistemas. Um modelo é uma simplificação da realidade. Segundo Sommerville (2007), "os modelos são construídos para que haja uma melhor compreensão do sistema que está sendo desenvolvido por meio de alguns elementos de maneira que cada modelo apresenta uma visão ou perspectiva do sistema" [6].

Além disso, de acordo com Booch [7], os objetivos principais para criar modelos são:

- Eles ajudam a visualizar o sistema como ele é ou como desejamos que ele seja;
- Eles permitem especificar a estrutura ou o comportamento de um sistema;
- Eles proporcionam um guia para a construção do sistema;
- Eles documentam as decisões tomadas no projeto.

Desse modo, para alcançar esses objetivos e para facilitar a comunicação entre os participantes de um projeto, uma linguagem de modelagem padronizada se faz necessária. Sendo que hoje em dia quase todas as representações gráficas são baseadas em UML.

Neste contexto, a UML é uma linguagem gráfica de modelagem para visualização, especificação, construção e documentação de artefatos de sistema de software. Ela foi criada com o objetivo de criar modelos abstratos de processos, sendo que estes modelos são apresentados através de diagramas. [8]

O diagrama tem como finalidade ajudar na compreensão do sistema a ser desenvolvido e cada um oferece uma visão diferente acerca dos elementos que constituem esse sistema. Conforme a Figura 2, o UML 2.2 possui 14 diagramas divididos em duas categorias:

- Os **Diagramas Estruturais** que tratar dos aspectos estáticos de um sistema.

- Os **Diagramas Comportamentais** que contribuem para visualizar as partes dinâmicas de um sistema sendo conhecidos como diagramas comportamentais.

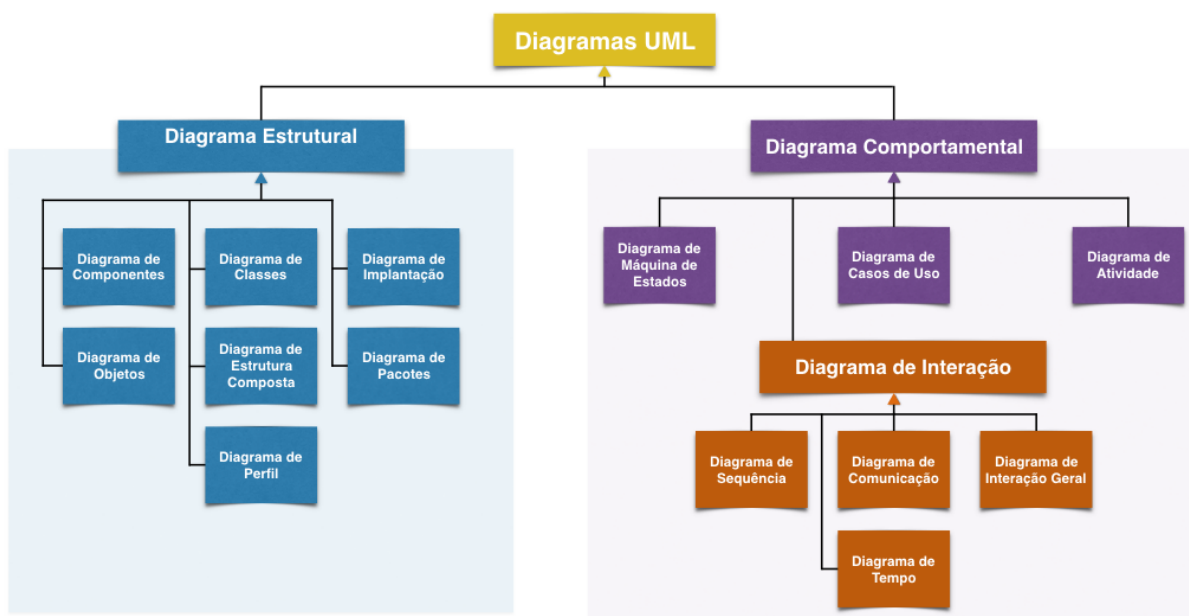


Figura 2: Diagramas UML

Para modelagem deste sistema foram utilizados os respectivos diagramas: diagrama de casos de uso, de classes, de objetos, de atividades e de sequência.

2.3 ARQUITETURA MVC

A arquitetura MVC (Modelo – Visão – Controle) é um padrão de arquitetura de software criado nos anos 80 por Trygve Reenskaug.

Atualmente essa arquitetura é muito utilizada em projetos de pequena e grande escala, como por exemplo os frameworks, devida a sua arquitetura, pois cada uma, das suas três camadas, é muito bem definida, sendo que cada uma executa somente sua função.

O grande objetivo da arquitetura MVC é separar as regras de negócios da lógica de apresentação, a interface com o usuário, permitindo dessa forma a separação física de componentes em camadas.

Esta arquitetura auxilia os desenvolvedores a construir aplicações separando seus principais componentes, a manipulação e armazenamento dos dados, as funções que irão trabalhar com as entradas dos dados e a visualização do usuário. A arquitetura MVC especifica onde cada tipo de lógica deve estar localizada na aplicação [9]. Na Figura 3 temos a interação desses três componentes.

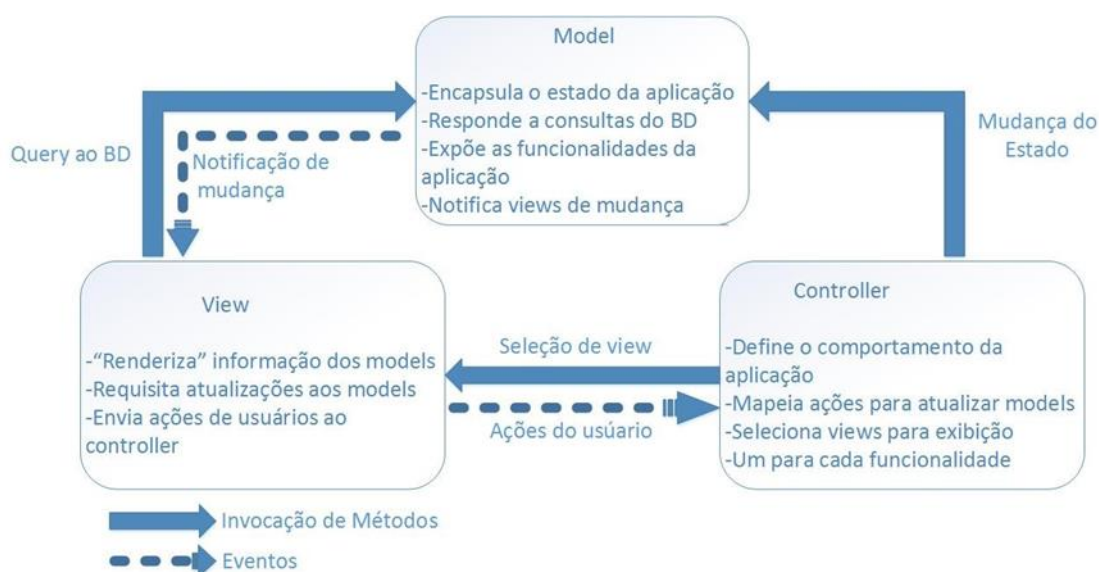


Figura 3: Representação da comunicação do modelo MVC. Adaptado de JAVABLUETPRINTS, MVC. Disponível em <http://www.oracle.com/technetwork/java/mvc-detailed-136062.html>

A camada Modelo (Model) representa os dados do sistema, sendo ela a responsável pela comunicação e o tratamento dos dados que estão armazenados no banco de dados. É nesta camada que estão contidas as operações que são realizadas num banco de dados, como por exemplo de criar, deletar, ler e atualizar (CRUD).

Na camada Visualização (View) fica tudo que é necessário para exibir dados, ou seja, é a camada que contém apresentação ou a interface do sistema, sendo ela que será apresentada por usuário, permitindo o encaminhamento de ações do usuário para o controlador, além da requisição de dados do modelo passando pelo controle.

A camada Controle (Controller) tem como responsabilidade, a administração de todo o fluxo da aplicação, é o que move a aplicação. Ele captura as ações do usuário mapeando-as para tarefas que o modelo irá realizar. Ou seja, ele trabalha com os dados da View e escolhe que operação utilizará da camada Modelo. Com base nessa interação, o controlador seleciona uma visualização que será exibida com os possíveis dados solicitados pelo usuário.

3 ANÁLISE E ESPECIFICAÇÃO DO PROJETO

O sistema proposto tem como propósito permitir que alunos, administradores, professores, funcionários tenham acesso a dados relevantes de forma integrada, além de permitir funções próprias de cada ator, por exemplo, um professor fazer lista de chamada ou um administrador de inserir novos alunos.

Os incrementos do sistema serão divididos por atores, sendo cada ator será considerado um módulo, dessa forma, cada módulo implementado será considerado um incremento, conforme explicado no Capítulo 2.

O primeiro incremento é responsável pela implementação do núcleo do sistema, módulo **NÚCLEO**, com a maioria das regras de negócio associados às entidades, além do usuário e perfil de administrador.

O segundo incremento é responsável pela implementação do perfil e usuário Aluno, denominado módulo **USUÁRIO**. O terceiro incremento pelo perfil e usuário Professor, módulo **PROFESSOR**.

A seguir serão apresentadas as regras de negócio, os requisitos funcionais e não funcionais do sistema, os diagramas de casos de uso e suas especificações.

3.1 LEVANTAMENTO DE REQUISITOS

De maneira geral os requisitos são funcionalidades do sistema, ou seja, são instruções que o usuário seguir para atingir os objetivos do sistema. Segundo Sommerville (2007), “os requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais.” [6]. E de acordo com ARAUJO (2012) “um requisito é basicamente uma condição ou capacidade que o sistema deve atender para satisfazer”. Isso demonstra que os requisitos refletem as necessidades dos usuários de um sistema que ajuda a resolver alguma situação ou problema. Podemos citar como exemplo a consulta de informações ou a venda de um produto.

Desse modo, podemos definir levantamento de requisitos como sendo a atividade de descobrir quais são as funções que o sistema deve realizar e quais são as restrições que existem sobre essas funções.

Neste projeto nós utilizamos a técnica de entrevistas para fazer o levantamento dos requisitos e das regras de negócio.

A técnica de entrevistas é uma das mais simples de se utilizar além de possuir uma excelente eficácia, sendo deste modo utilizado em boa parte dos processos de engenharia de requisitos.

3.1.1 Requisitos funcionais

Os requisitos funcionais especificam ações que um sistema deve fazer, sem levar em consideração restrições físicas.

Então no contexto do sistema a ser desenvolvido, os seguintes requisitos foram levantados.

NÚCLEO

RF01. O software deve identificar e validar todos os usuários que desejarem acessá-lo, identificando seu perfil;

RF02. O sistema deve disponibilizar ao usuário identificado as funcionalidades associadas ao seu perfil e ao seu papel no sistema;

RF03. O sistema deve permitir que o usuário recupere a senha;

RF04. O sistema deve permitir que se associe disciplinas às turmas;

RF05. O sistema deve permitir a associação um estudante a uma turma;

RF06. O sistema deve permitir a atribuição das notas dos exames por disciplina;

RF07. O sistema deve permitir a alteração das notas dos exames;

RF08. O sistema deve permitir a atribuição da frequência escolar por aluno;

RF09. O sistema deve permitir a atribuição da frequência escolar por turma;

RF10. O sistema deve permitir a visualização da frequência escolar por turma;

RF11. O sistema deve permitir a visualização da frequência por aluno;

RF12. O sistema deve permitir o administrador criar, alterar, visualizar, consultar e excluir um aluno;

RF13. O sistema deve permitir o administrador criar, alterar, visualizar, consultar e excluir um professor;

RF14. O sistema deve permitir o administrador criar, alterar, visualizar, consultar e excluir uma disciplina;

RF15. O sistema deve permitir o administrador criar, alterar, excluir, consultar um exame;

RF16. O sistema deve permitir agendar um exame;

RF17. O sistema deve permitir a visualização de exames agendados por turma;

RF18. O sistema deve permitir a visualização de exames por aluno;

ALUNO

RF19. O sistema deve permitir que o aluno veja seus dados;

RF20. O sistema deve permitir que o aluno veja sua frequência;

RF21. O sistema deve permitir que o aluno veja suas notas;

RF22. O sistema deve permitir que o aluno edite seus dados;

RF23. O sistema deve permitir que o aluno veja seus exames agendados;

PROFESSOR

RF24. O sistema deve permitir o professor editar seus dados;

RF25. O sistema deve permitir o professor visualizar e consultar um aluno;

RF26. O sistema deve permitir o professor visualizar e consultar outro professor;

RF27. O sistema deve permitir o professor visualizar e consultar uma disciplina;

RF28. O sistema deve permitir o professor visualizar e consultar um exame;

RF29. O sistema deve permitir o professor adicionar ou alterar as notas dos exames de uma disciplina;

RF30. O sistema deve permitir o professor realizar ou alterar a chamada de uma turma;

RF31. O sistema deve permitir que o professor veja a frequência de um aluno;

RF32. O sistema deve permitir o professor criar ou alterar um exame;

RF33. O sistema deve permitir o professor agendar um exame;

3.1.2 Requisitos não funcionais

Os requisitos não funcionais são aqueles que definem as características e as qualidades globais do sistema e que não estejam diretamente relacionadas as funções do sistema. Esses requisitos podem ser em relação a confiabilidade, usabilidade, portabilidade, segurança, de desempenho ou arquitetura por exemplo.

Assim sendo, os seguintes requisitos foram levantados de maneira geral para todos os módulos:

RNF1. O sistema deve ter uma interface amigável;

RNF2. O sistema deve garantir a integridade dos dados;

RNF3. As senhas dos usuários devem ter alguma camada de proteção;

RNF4. A persistência das informações deve ser implementada em um Sistema Gerenciador de Bancos de Dados Relacional;

RNF5. Os componentes e módulos do sistema podem ser reutilizados;

RNF6. A linguagem de programação deve ser PHP;

RNF7. O layout da página deve ser baseado no framework Bootstrap;

RNF8. O sistema deve seguir a arquitetura MVC.

3.1.3 Regras de negócio

Regra de negócio é o que define a forma de fazer o negócio, refletindo a política interna, o processo definido e/ou as regras básicas de conduta. Ou seja, é um conjunto de instruções que os usuários já seguem e que o sistema a ser desenvolvido deve contemplar. Restrições, validações, condições e exceções do processo são exemplos clássicos de regras de negócio [10].

Segundo Alenquer (2002):

Sob o ponto de vista de sistema de informações, regra de negócio é uma sentença que define ou restringe algum aspecto do negócio, sua intenção é manter a estrutura do , ou controlar ou influenciar algum aspecto do negócio [11].

Dessa forma, as seguintes regras foram levantadas, de maneira geral, para esse sistema:

- RN1.** Cada usuário terá um código único que não pode ser alterado;
- RN2.** Somente usuários autenticados podem acessar o sistema;
- RN3.** Cada usuário terá um e-mail único;
- RN4.** Cada usuário terá uma área reservada e distinta no sistema e somente ele pode acessa-la;
- RN5.** O sistema terá perfil de acesso distintos por usuários;
- RN6.** O sistema deve validar todos os campos nos formulários e todos os campos obrigatórios devem ser preenchidos;
- RN7.** Cada turma tem somente um professor responsável;
- RN9.** Turmas com alunos associados não devem ser excluídas;
- RN10.** Cada disciplina deve esta estar associada a somente uma turma e a um único professor;
- RN11.** O professor pode lecionar várias disciplinas;
- RN12.** O professor só pode criar e alterar exames das disciplinas que leciona;
- RN13.** O professor só pode agendar provas das disciplinas que leciona.

RN14. O professor só pode ver a frequência dos alunos da turma que se encontra associado;

RN15. O professor só poderá gerenciar as notas das disciplinas que leciona;

RN16. O professor só poderá atribuir/alterar a frequência para turma que se encontra associado;

RN17. Cada prova deve estar associada a somente um período de aplicação;

RN18. Existem somente 4 períodos (4 bimestres) de aplicação para as provas;

RN19. O aluno só pode estar associado a uma única turma;

RN20. O aluno só pode ver os dados do seu perfil;

RN21. O aluno só pode ver sua frequência;

RN22. O aluno só pode ver as disciplinas da turma que se encontra inscrito;

RN23. O aluno só pode ver os exames agendados para turma em que se encontra inscrito;

RN24. Somente o administrador pode adicionar ou remover usuários;

RN25. Somente administradores podem gerenciar e adicionar novas turmas e disciplinas;

RN26. Somente usuários com nível de acesso acima de aluno pode visualizar e consultar dados de Alunos, Professores, Turmas, Disciplinas e Exames agendados;

RN27. O aluno só pode ver os dados do seu perfil;

RN25. O aluno só pode ver os dados do seu perfil;

RN26. O aluno só pode ver os dados do seu perfil;

3.2 CASOS DE USO

Nesta Seção serão apresentados os diagramas de caso de uso com suas respectivas descrições.

Um caso de uso indica uma funcionalidade que o sistema deve ter, dessa forma, um diagrama de casos de uso apresenta uma visão geral das funcionalidades do sistema, definindo o que ele faz e não se preocupando em como o sistema faz.

No sistema projetado, nós trabalhamos com 3 tipos de usuários (atores): Administrador, Professor e Aluno. Sendo que cada um deles possui um diagrama de uso de maneira a facilitar a visualização. Além disso, o caso de uso chamado de “Gerenciar” do ator Administrador, contém as funcionalidades de adição, remoção, alteração e consulta com poucas variações de um caso de uso para o outro.

Além disso, tanto para efeitos práticos como também para evitar a redundância de informações, na Subseção 3.2.3 serão apresentadas somente as tabelas com as descrições dos casos de uso do Administrador, visto que, esses casos aborda toda a funcionalidade e privilégios do sistema sendo os casos de outros atores meramente uma redução destes.

3.2.1 Diagramas de Casos de Uso

Na Figura 4 temos o diagrama do ator Administrador:

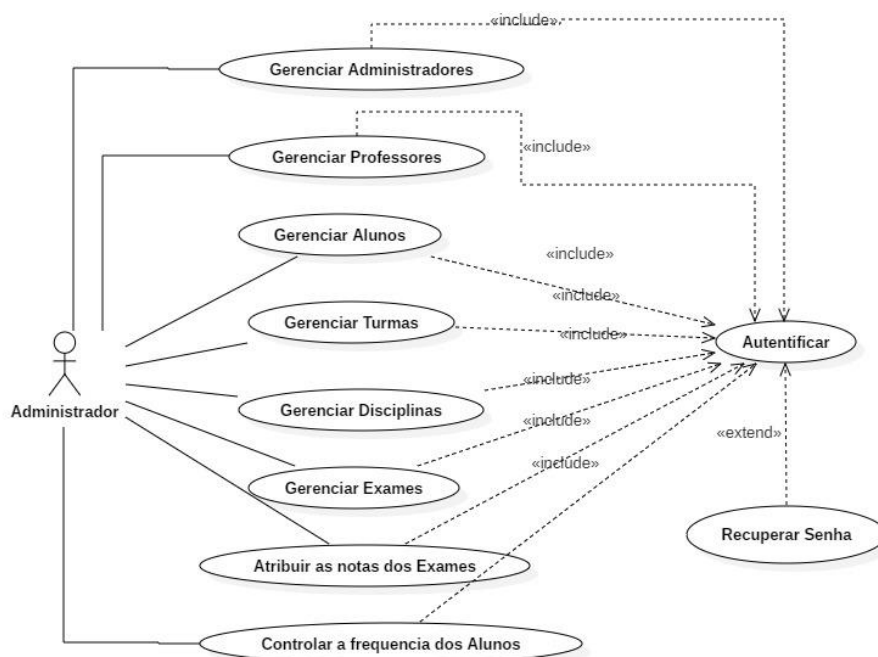


Figura 4: Diagrama de caso de uso do Administrador

Na Figura 5 temos o diagrama do ator Professor:

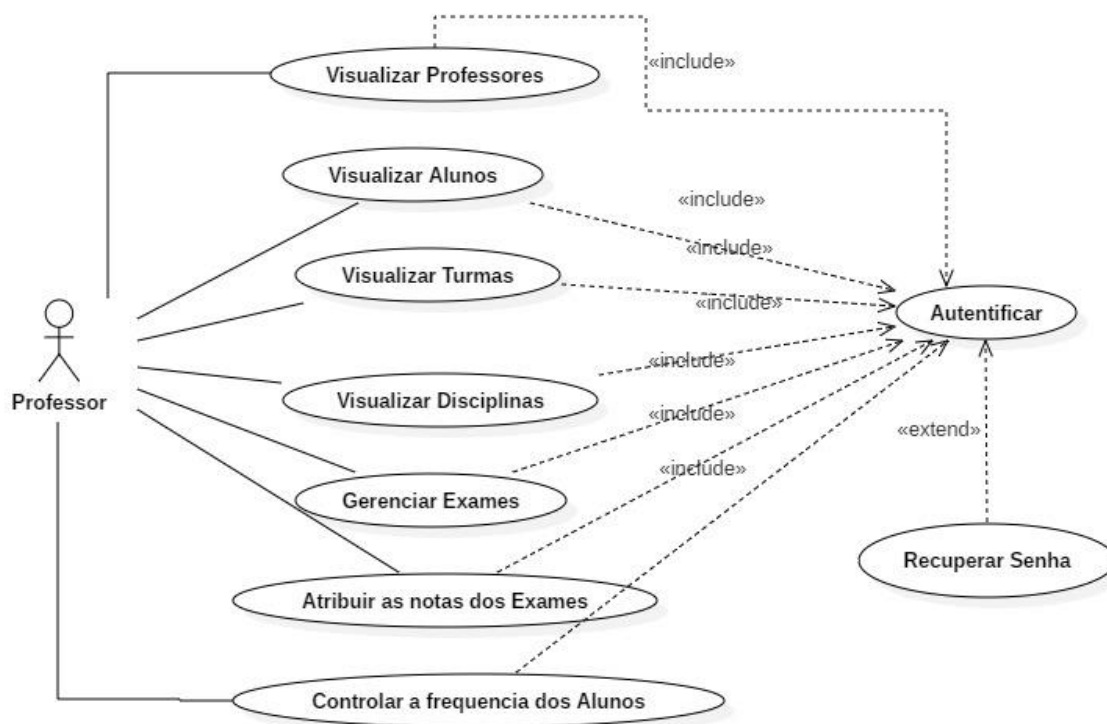


Figura 5: Diagrama de caso de uso do Professor

E por último, na Figura 6 temos o diagrama do ator Aluno:

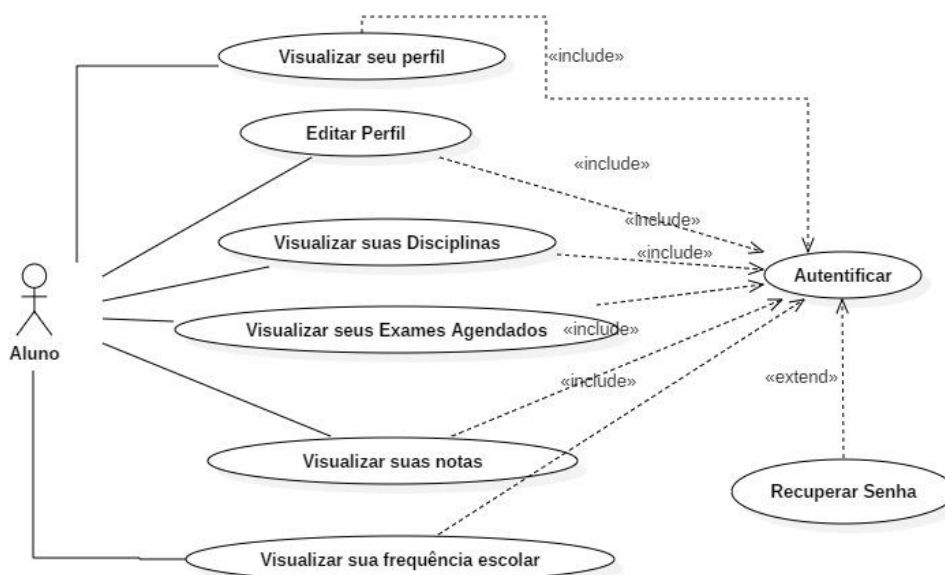


Figura 6: Diagrama de caso de uso do Aluno

3.2.2 Listagem de Casos de Uso

Nesta Subseção nós apresentaremos uma lista de Casos de Uso separadas por módulos.

ADMINISTRADOR

- UC1.** Autenticar (válido para todos os usuários);
- UC2.** Recuperar senha;
- UC3.** Gerenciar Administradores;
- UC4.** Gerenciar Professores;
- UC5.** Gerenciar Alunos;
- UC6.** Gerenciar Turmas;
- UC7.** Gerenciar Disciplinas;
- UC8.** Gerenciar Exames;
- UC9.** Atribuir as Notas dos Exames;
- UC10.** Controlar a Frequência dos Alunos;

PROFESSOR

- UC1.** Autenticar (válido para todos os usuários);
- UC2.** Recuperar senha (válido para todos os usuários);
- UC11.** Visualizar Professores;
- UC12.** Visualizar Alunos;
- UC13.** Visualizar Turmas;
- UC14.** Visualizar Disciplinas;
- UC15.** Gerenciar Exames;
- UC16.** Atribuir as Notas dos Exames;
- UC17.** Controlar a Frequência dos Alunos;

ALUNO

- UC1.** Autenticar (válido para todos os usuários);
- UC2.** Recuperar senha (válido para todos os usuários);
- UC18.** Visualizar seu perfil;

UC19. Editar seu perfil;

UC20. Visualizar suas disciplinas;

UC21. Visualizar seus Exames agendados;

UC22. Visualizar suas Notas;

UC23. Visualizar sua Frequência escolar;

3.2.3 Descrição dos Casos de Uso

Nas tabelas abaixo, são descritos os principais casos de uso do sistema, ou seja, do módulo Administrador com as seguintes informações:

- Descrição Geral – Contém uma breve descrição do que faz o caso de uso e de seu objetivo;
- Atores – Os atores vinculados ao caso de uso;
- Pré-condições – Condições que devem ser atendidas para executar o caso de uso com sucesso;
- Pós-condições – O estado do sistema após ser executado o caso de uso;
- Fluxo Principal – É o cenário principal do caso de uso;
- Fluxo Alternativo – É um cenário alternativo tomado a partir do fluxo principal;
- Fluxo Exceção – É um cenário para tratar as exceções do sistema. Não é um cenário escolhido, mas sim, o resultado da validação de dados.

UC1 - Autenticar:

Tabela 1: Caso de uso - Autenticar

Caso de Uso 1 - Autenticar (Login)	
Descrição Geral:	Permite que os usuários tenham acesso as funcionalidades do sistema.

Atores:	Administrador, Professor e Aluno
Pré-condições:	O usuário deve ter cadastro no sistema
Pós-Condições:	O usuário estará logado no sistema e terá acesso as funcionalidades
Fluxo Principal:	1. O sistema encaminha o ator para página de login 2. O ator preenche seus dados no formulário de login 3. O ator clica no botão entrar para enviar seus dados 4. O sistema verifica os dados digitados 5. O sistema envia o ator para sua página principal
Fluxo Exceção:	1e. No passo 4 o ator preencheu um e-mail que não existe, retorna para o passo 2 sinalizando com uma mensagem de erro. 2e. No passo 4 o ator preencheu a senha incorreta, retorna para o passo 2 sinalizando com uma mensagem de erro.

UC2 – Recuperar Senha:

Tabela 2: Caso de uso - Recuperar Senha

Caso de Uso 2 - Recuperar Senha	
Descrição Geral:	Permite que os usuários recuperem suas senhas
Atores:	Administrador, Professor e Aluno
Pré-condições:	O usuário deve ter cadastro no sistema
Pós-Condições:	O usuário terá uma nova senha vinculado ao seu cadastro
Fluxo Principal:	1. O sistema encaminha o ator para página de login 2. O ator seleciona a opção para recuperar a senha 3. O sistema mostra o formulário para recuperar senha 4. O ator preenche seu e-mail e seu perfil no formulário 5. O sistema verifica se existe o e-mail para o tipo de ator escolhido 6. O sistema confirma que as instruções para recuperar a senha foi enviado para o e-mail cadastrado 7. O usuário acessa o e-mail e clica no link enviado pelo o sistema 8. O sistema valida a chave enviada no link do sistema 9. O sistema abre o formulário de redefinição de senha

	10. O ator preenche a nova senha e clica em trocar
	11. Sistema confirma a alteração da senha.
Fluxo Exceção:	1e. No passo 4 o ator preencheu um e-mail que não existe, retorna para o passo 3 e envia uma mensagem de alerta;

UC3 – Gerenciamento de Administradores:

Tabela 3: Caso de uso – Gerenciamento de Administradores

Caso de Uso 3 - Gerenciamento de Administradores	
Descrição Geral:	Esse caso permite o cadastro, atualização, remoção e a consulta de administradores do sistema
Atores:	Administrador
Pré-condições:	O ator deve ter completado o UC1 com sucesso
Pós-Condições:	
Fluxo Principal:	
	1. O ator seleciona a opção gerenciar administradores
	2. O sistema exibe a listagem dos administradores
	3. O ator seleciona adicionar um novo administrador
	4. O sistema exibe um formulário de cadastro
	5. O ator preenche os dados
	6. O sistema verifica os dados
	7. O sistema cadastra o administrador com sucesso
Fluxo Alternativo A:	
	1a. No passo 3 o ator clicar em Editar
	2a. O sistema mostra um formulário com os dados do administrador
	3a. O ator modifica os dados
	4a. O sistema valida os dados
	5a. O sistema atualiza com sucesso.
Fluxo Alternativo B:	1b. No passo 3 o ator clica em Detalhar
	2b. O sistema mostra os dados do usuário selecionado.
Fluxo Alternativo C:	1c. No passo 3 o ator clica em Deletar
	2c. O sistema deleta o usuário selecionado

UC4 – Gerenciamento de Professores

Tabela 4: Caso de uso - Gerenciamento de Professores

Caso de Uso 4 - Gerenciamento de Professores	
Descrição Geral:	Esse caso permite o cadastro, atualização, remoção e a consulta de Professores do sistema
Atores:	Administrador
Pré-condições:	O ator deve ter completado o UC1 com sucesso
Pós-Condições:	
Fluxo Principal:	
	1. O ator seleciona a opção gerenciar professores
	2. O sistema exibe a listagem dos professores
	3. O ator seleciona adicionar um novo professor
	4. O sistema exibe um formulário de cadastro
	5. O ator preenche os dados
	6. O sistema verifica os dados
	7. O sistema cadastra o professor com sucesso
Fluxo Alternativo A:	
	1a. No passo 3 o ator clicar em Editar
	2a. O sistema mostra um formulário com os dados do professor
	3a. O ator modifica os dados do professor
	4a. O sistema valida os dados
	5a. O sistema atualiza com sucesso.
Fluxo Alternativo B:	1b. No passo 3 o ator clica em Detalhar
	2b. O sistema mostra os dados do usuário selecionado.
Fluxo Alternativo C:	1c. No passo 3 o ator clica em Deletar
	2c. O sistema verifica se o professor está vinculado a uma turma
	2c. O sistema deleta o professor selecionado

UC5 – Gerenciamento de Alunos

Tabela 5: Caso de uso - Gerenciamento de Alunos

Caso de Uso 5 - Gerenciamento de Alunos	
Descrição Geral:	Esse caso permite o cadastro, atualização, remoção e a consulta de Alunos do sistema
Atores:	Administrador
Pré-condições:	O ator deve ter completado o UC1 com sucesso O sistema deve ter pelo menos uma turma cadastrada
Pós-Condições:	
Fluxo Principal:	
	1. O ator seleciona a opção gerenciar alunos
	2. O sistema exibe a listagem dos alunos
	3. O ator seleciona adicionar um novo aluno
	4. O sistema exibe um formulário de cadastro
	4. O sistema carrega a lista de todas as turmas numa caixa no formulário
	5. O ator preenche os dados e vincular o aluno a uma turma
	6. O sistema verifica os dados
	7. O sistema cadastra o aluno com sucesso
Fluxo Alternativo A:	
	1a. No passo 3 o ator clicar em Editar
	2a. O sistema mostra um formulário com os dados do aluno
	3a. O ator modifica os dados do aluno
	4a. O sistema valida os dados
	5a. O sistema atualiza o cadastro com sucesso.
Fluxo Alternativo B:	1b. No passo 3 o ator clica em Detalhar
	2b. O sistema mostra os dados do usuário selecionado.
Fluxo Alternativo C:	1c. No passo 3 o ator clica em Deletar
	2c. O sistema deleta o aluno selecionado
	3c. O sistema deleta a frequência do aluno
	4c. O sistema deleta as notas do aluno

UC6 – Gerenciamento de Turmas:

Tabela 6: Caso de uso - Gerenciamento de Turmas

Caso de Uso 6 - Gerenciamento de Turmas

Descrição Geral:	Esse caso permite o cadastro, atualização, remoção e a consulta de Turmas do sistema
Atores:	Administrador
Pré-condições:	O ator deve ter completado o UC1 com sucesso
	O sistema deve ter pelo menos um professor cadastrado
Pós-Condições:	
Fluxo Principal:	
	1. O ator seleciona a opção Turmas
	2. O sistema exibe todas as turmas com os respectivos professores responsáveis
	3. O ator seleciona adicionar uma nova turma
	4. O sistema exibe um formulário de cadastro
	4. O sistema carrega a lista de todos os professores numa caixa no formulário
	5. O ator preenche os dados e vincular a turma a um professor
	6. O sistema verifica os dados
	7. O sistema cadastra a turma com sucesso
Fluxo Alternativo A:	
	1a. No passo 3 o ator clicar em Editar
	2a. O sistema mostra um formulário com os dados da turma
	3a. O ator modifica os dados da turma
	4a. O sistema valida os dados
	5a. O sistema atualiza a turma com sucesso.
Fluxo Alternativo B:	1b. No passo 3 o ator clica em Deletar
	2b. O sistema verifica se tem algum aluno vinculado a turma
	3b. O sistema deleta a turma

UC7 – Gerenciamento de Disciplinas:

Tabela 7: Caso de uso - Gerenciamento de Disciplinas

Caso de Uso 7 - Gerenciamento de Disciplinas	
Descrição Geral:	Esse caso permite o cadastro, atualização, remoção e a consulta de Disciplinas do sistema
Atores:	Administrador
Pré-condições:	O ator deve ter completado o UC1 com sucesso
	O sistema deve ter pelo menos um professor cadastrado

	O sistema deve ter pelo menos uma turma cadastrada
Pós-Condições:	
Fluxo Principal:	
	1. O ator seleciona a opção gerenciar Disciplinas
	2. O sistema exibe uma listagem de todas as disciplinas com os respectivos professores e turmas associadas
	3. O ator seleciona adicionar uma nova disciplina
	4. O sistema exibe um formulário de cadastro
	5. O sistema carrega a lista de todos os professores numa caixa no formulário
	6. O sistema carrega a lista de todas as turmas numa caixa no formulário
	7. O ator preenche os dados e vincular a disciplina a turma e ao professor
	8. O sistema verifica os dados
	9. O sistema cadastra a nova disciplina com sucesso
Fluxo Alternativo A:	
	1a. No passo 3 o ator clicar em Editar
	2a. O sistema mostra um formulário com os dados da disciplina
	3a. O ator modifica os dados da turma
	4a. O sistema valida os dados
	5a. O sistema atualiza a turma com sucesso.
Fluxo Alternativo B:	1b. No passo 3 o ator clica em Deletar
	2b. O sistema deleta a disciplina

UC8 – Gerenciamento de Exames

Tabela 8: Caso de uso - Gerenciamento de Exames

Caso de Uso 8 - Gerenciamento de Exames	
Descrição Geral:	Esse caso permite o cadastro, atualização, remoção e a consulta de Exames do sistema
Atores:	Administrador
Pré-condições:	O ator deve ter completado o UC1 com sucesso
	O sistema deve ter pelo menos uma disciplina cadastrada
	O sistema deve ter pelo menos uma turma cadastrada
Pós-Condições:	

Fluxo Principal:	
	1. O ator seleciona a opção gerenciar Exames
	2. O sistema exibe uma listagem de todos exames com os respectivos professores e turmas associadas
	3. O ator seleciona adicionar um novo exame
	4. O sistema exibe um formulário de cadastro
	5. O sistema carrega a lista de todas as turmas numa caixa no formulário
	6. Baseado na turma escolhida no passo 5, o sistema carrega todas as disciplinas vinculadas a esta turma
	7. O sistema carrega a lista dos períodos de avaliação numa caixa de formulário
	8. O ator preenche os dados e vincular o período, a turma e a disciplina ao exame
	9. O sistema verifica os dados
	10. O sistema cadastra um novo exame com sucesso
Fluxo Alternativo A:	
	1a. No passo 3 o ator clicar em Editar
	2a. O sistema mostra um formulário com os dados do exame
	3a. O ator modifica os dados do exame
	4a. O sistema valida os dados
	5a. O sistema atualiza o exame com sucesso.
Fluxo Alternativo B:	1b. No passo 3 o ator clica em Deletar
	2b. O sistema deleta o exame
	2b. O sistema deleta todas as notas associadas a este exame

UC9 – Atribuir as Notas dos Exames

Tabela 9: Caso de uso - Atribuir as Notas dos Exames

Caso de Uso 9 - Atribuir as Notas dos Exames	
Descrição Geral:	Esse caso permite registrar e alterar as notas dos exames dos alunos
Atores:	Administrador
Pré-condições:	O ator deve ter completado o UC1 com sucesso
	O sistema deve ter pelo menos um exame cadastrado
	O sistema deve ter pelo menos uma disciplina cadastrada
	O sistema deve ter pelo menos um aluno cadastrado

	O sistema deve ter pelo menos uma turma cadastrada
Pós-Condições:	
Fluxo Principal:	
	1. O ator seleciona a opção Atribuir Notas.
	2. O sistema exibe um formulário com 3 Select-box: Período, Turma e Exame (Disciplina)
	3. O ator seleciona o Período e a Turma
	4. O sistema carrega no terceiro Select-box os exames pesquisados com base no Período e da Turma
	5. O ator seleciona o Exame
	6. O ator seleciona a opção pesquisar
	7. O sistema carrega a lista de alunos que fizeram o exame escolhido com suas respectivas notas
	8. O ator atribui ou altera as notas de cada estudante
	9. O ator seleciona a opção Salvar Notas
	10. O sistema verifica os dados
	11. O sistema cadastra ou altera o resultado do exame para cada aluno com sucesso

UC10 – Controlar a Frequência dos Alunos

Tabela 10: Caso de uso - Controlar a Frequência dos Alunos

Caso de Uso 10 - Controlar a Frequência dos Alunos	
Descrição Geral:	Esse caso permite adicionar, alterar e visualizar a frequência dos alunos
Atores:	Administrador
Pré-condições:	O ator deve ter completado o UC1 com sucesso
	O sistema deve ter pelo menos um aluno cadastrado
	O sistema deve ter pelo menos uma turma cadastrada
Pós-Condições:	
Fluxo Principal:	
	1. O Ator clica no menu Gerenciar Frequência Escolar
	2. O sistema carrega um submenu contendo os itens: Ver lista de Presença e Adicionar Presença
	3. O ator seleciona a opção Ver Lista de Presença
	4. O sistema carrega uma tabela contendo todos os alunos cadastrados. Sendo que em cada linha a um botão "Ver" para visualizar a frequência do aluno da respectiva linha.

	5. O ator clica no botão "Ver"
	6. O sistema carrega uma página contendo a frequência do aluno marcado em um calendário.
Fluxo Alternativo A:	Adicionar Frequência
	1a. No passo 3 o ator seleciona a opção Adicionar Frequência
	2a. O sistema mostra um formulário de pesquisa com dois campos: Turma e Data
	3a. O ator preenche a turma e a data em que deseja realizar a chamada e clica em pesquisar
	4a. O sistema carrega uma tabela com os alunos matriculados nesta turma podendo o ator marcar a frequência individualmente de cada um deles
	5a. O ator marca a frequência dos alunos e clica em salvar
	6a. O sistema salva a frequência dos alunos no banco de dados e volta para o passo 2a

4 PROJETO E IMPLEMENTAÇÃO DO SISTEMA

Neste capítulo abordaremos as fases do projeto e implementação dos módulos Administrador e do Aluno. O módulo do Professor será deixado para trabalhos futuros. A seguir apresentaremos as ferramentas utilizadas, os diagramas conceituais e de projeto, a modelagem banco de dados e a interface do sistema.

4.1 FERRAMENTAS UTILIZADAS

Para que possamos desempenhar bem nossas atividades e atingir nossos objetivos neste trabalho, além de conhecermos bem a Engenharia de Software e várias linguagens e técnicas de programação, também é importante utilizarmos boas ferramentas de desenvolvimento web.

À vista disso, as ferramentas escolhidas para desenvolver o sistema foram:

- Framework CodeIgniter;
- MySQL e MySQL Workbench;
- XAAMP (PHP, MySQL e Apache);
- Bootstrap e AdminLTE;
- JetBrains PhpStorm.

4.1.1 Framework CodeIgniter

O CodeIgniter é um poderoso e popular, de acordo com Gráfico 1 é o segundo mais utilizado, framework de desenvolvimento de aplicações PHP com um Footprint (tamanho ocupado na memória RAM por uma aplicação) muito pequena, que pode ser usado para desenvolver websites. É uma estrutura de código aberto. Tem um conjunto muito rico de funcionalidades, o que aumentará a velocidade do

trabalho de desenvolvimento de sites. O CodeIgniter foi criado pelo EllisLab e agora é um projeto do British Columbia Institute of Technology [12].

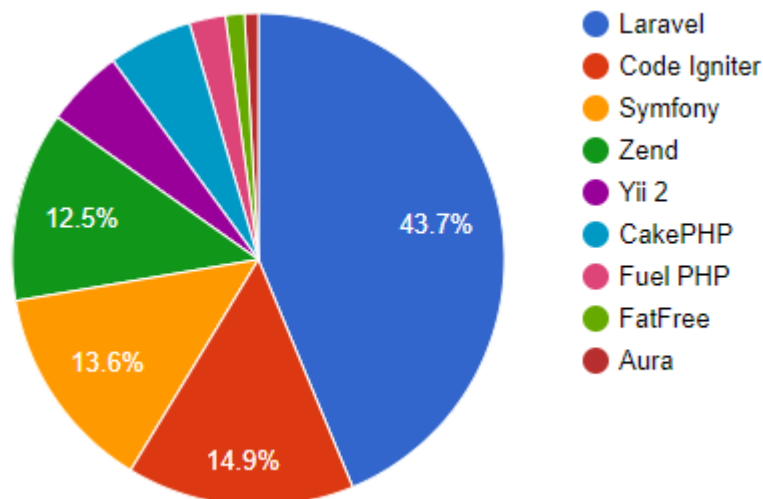


Gráfico 1: Estatísticas de frameworks PHP utilizados em projetos

O CodeIgniter fornece um conjunto de bibliotecas e funções para conexão com o banco de dados e execução de várias operações. Como enviar e-mails, fazer upload de arquivos, gerenciar sessões etc.

Ele usa o padrão desenvolvimento de software MVC (Modelo – Visão – Controlador) que permite uma grande separação entre a lógica e a apresentação, conforme explicado no Capítulo 2. No CodeIgniter eles têm as seguintes funções:

- O **Modelo** representa suas estruturas de dados. Geralmente, suas classes de modelo contêm funções que ajudam a recuperar, inserir e atualizar informações no banco de dados;
- A **Visualização** é a informação que está sendo apresentada a um usuário. Uma Visualização é uma página da web, mas no CodeIgniter, uma Visualização também pode ser um fragmento de página como um cabeçalho ou rodapé;
- O **Controlador** serve como intermediário entre o Modelo, a Visualização e quaisquer outros recursos necessários para processar a solicitação HTTP e gerar uma página da web.

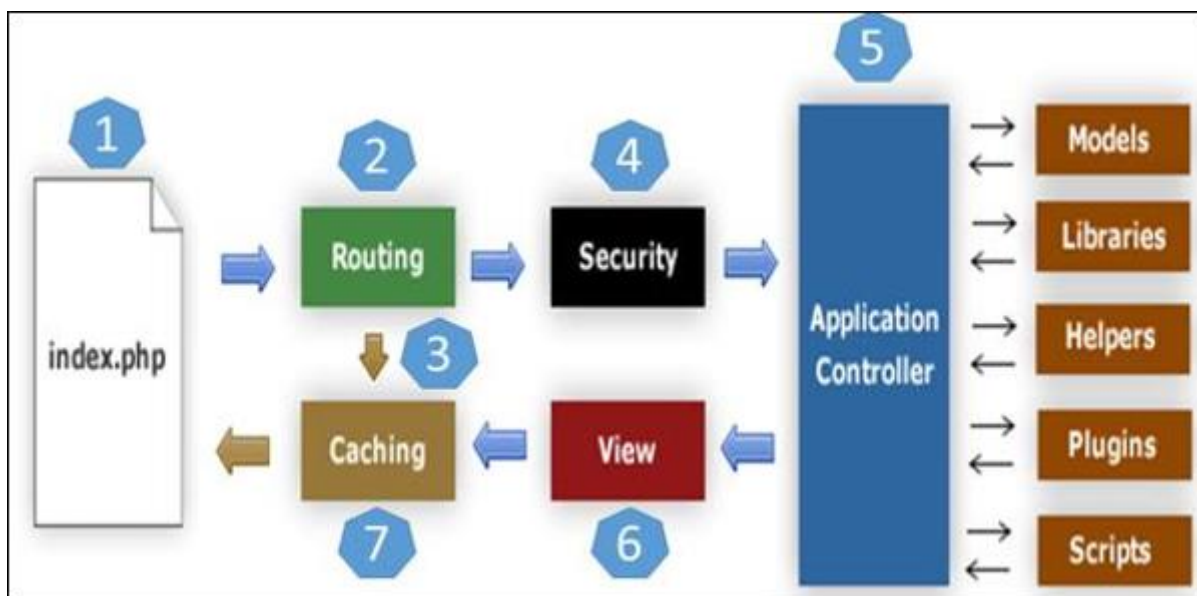


Figura 7: Arquitetura e fluxo de dados do CodeIgniter

Como mostrado na Figura 7 no passo 1, sempre que um pedido chega ao CodeIgniter, ele irá primeiro para página “**index.php**”.

Na segunda etapa o Roteamento (Routing) irá decidir se passa a solicitação para etapa 3 para Armazenar no Cache (Caching) ou para a etapa 4 para realizar a verificação de segurança. Se a página já existir no cache, o Roteamento passará a solicitação para a etapa 3 e a resposta retornará ao usuário. Mas se ela não existir, o Roteamento passará a página para etapa 4 para realizar verificações de segurança e depois para a etapa 5.

Na etapa 5 o Controlador (*Application Controller*) carrega os Modelos (*Models*), Bibliotecas (*Libraries*), Assistentes (*Helpers*), Plugins e Scripts necessários e os passa para a Visualização (View).

Na etapa 6 a Visualização renderizará a página solicitada com os dados disponíveis e a transmitirá para a etapa 7, o Armazenamento no Cache. Na etapa 7, como a página solicitada não foi armazenada em cache antes, agora ela será armazenada para essa página ser processada rapidamente para solicitações futuras.

4.1.2 MySQL e MySQL Workbench

O MySQL é um sistema de gerenciamento de banco de dados relacionais (SGBDR) de código aberto mais popular do mundo, que possibilita a entrega econômica de aplicativos de banco de dados confiáveis, de alto desempenho e escaláveis, com base na Web e incorporados [13].

Um banco de dados relacional é uma coleção estruturada de dados com relacionamentos predefinidos. Pode ser desde uma simples lista de compras até uma galeria de imagens ou uma vasta quantidade de informações em uma rede corporativa. Esses itens são organizados como um conjunto de tabelas com colunas e linhas. Para adicionar, acessar e processar dados armazenados em um banco de dados de computador, você precisa de um sistema de gerenciamento de banco de dados como o MySQL e uma linguagem específica de domínio, neste caso o SQL.

Já o MySQL Workbench é uma ferramenta visual de design e consultas de banco de dados MySQL. O MySQL Workbench fornece modelagem de dados, desenvolvimento de SQL e ferramentas de administração abrangentes para configuração de servidor, administração de usuários, backup e muito mais. No sistema proposto nós utilizamos principalmente sua função de modelagem ER.

4.1.3 XAMPP (PHP, MySQL e Apache)

XAMPP é um software multiplataforma de código aberto desenvolvido por *ApacheFriends*. O pacote de software do XAMPP contém distribuições do Apache para o servidor Apache, MySQL, PHP e Perl.

O XAMPP consiste basicamente um servidor local que pode ser utilizado no seu próprio desktop ou laptop e como a maioria das implantações reais de servidores da Web usa os mesmos componentes que o XAMPP, a transição de um servidor de teste local para um servidor ativo também é extremamente fácil. [14]

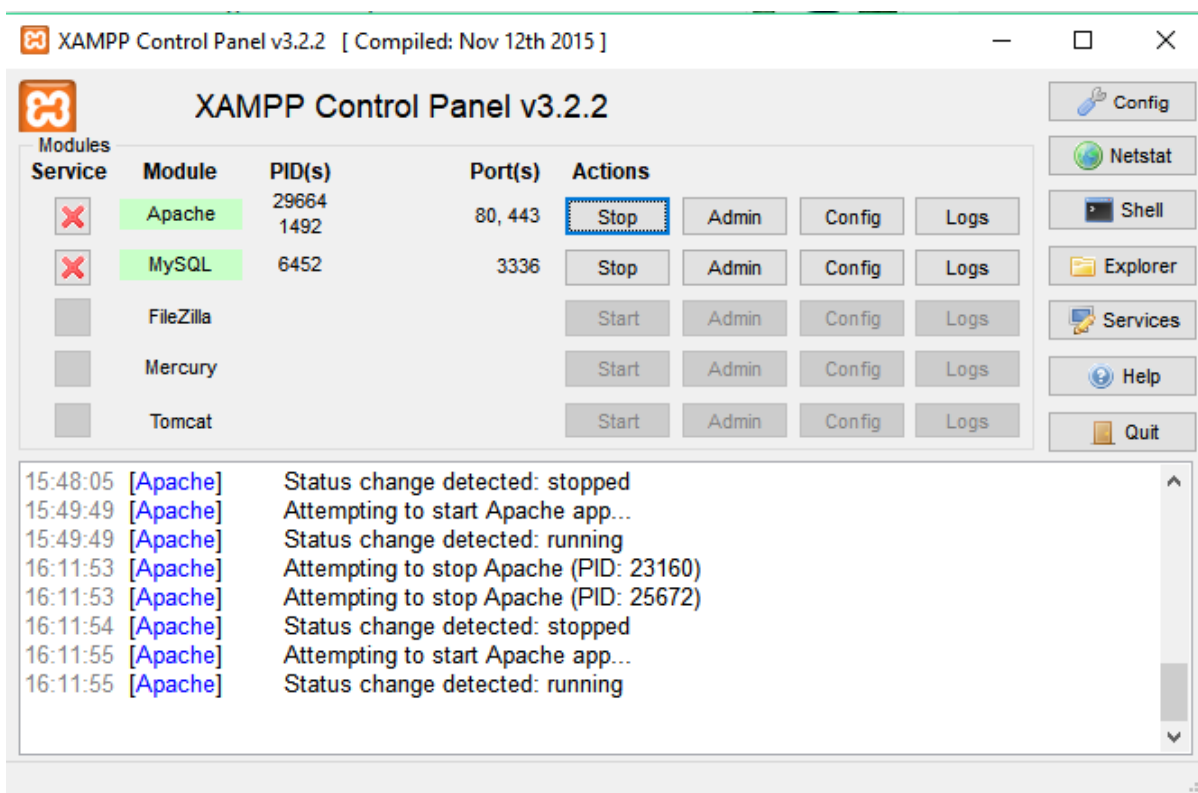


Figura 8: XAMPP

4.1.4 Bootstrap

O Bootstrap é um framework Javascript de código aberto desenvolvido pela equipe do Twitter. É uma combinação de código HTML, CSS e Javascript projetado para ajudar a criar componentes da interface do usuário e frontend para sites e aplicações web baseando-se em modelos de design para a tipografia, formulários, botões, navegação e outros componentes e extensões. O Bootstrap também foi programado para suportar HTML5 e CSS3 [15].

E AdminLTE é um painel (Template) de administração, de código aberto, construído sobre Bootstrap 3. Neste projeto ele foi utilizado para construir as interfaces gráficas de modo a agilizar o processo de implementação do sistema.

Na Figura 9 temos uma ilustração do AdminLTE.

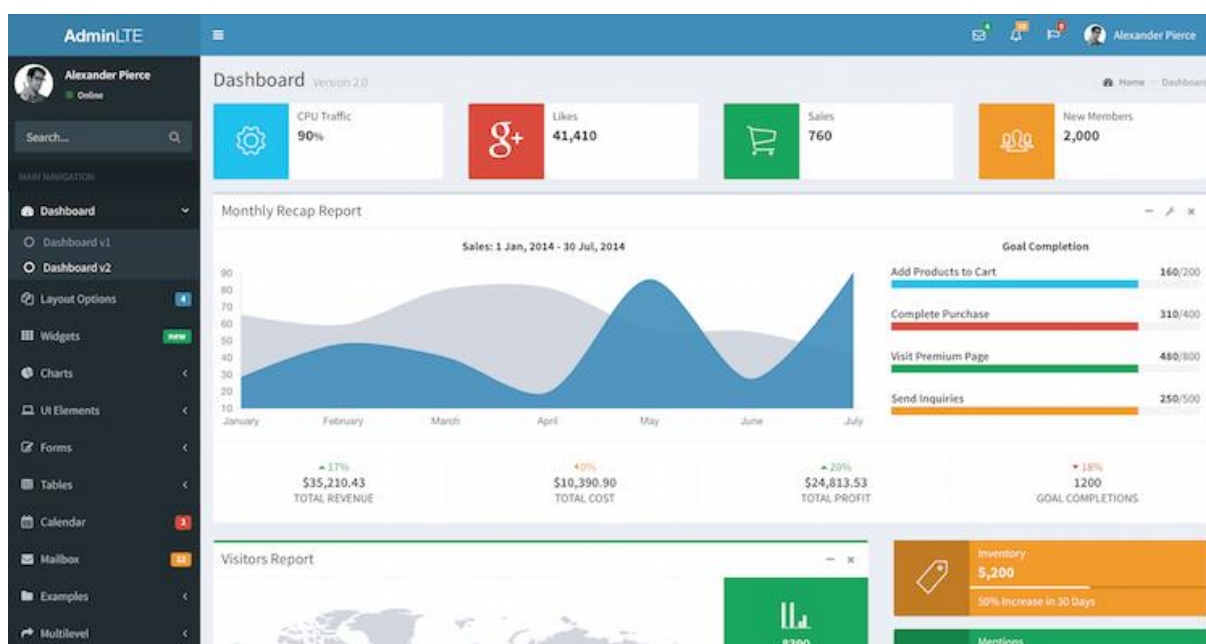


Figura 9: Template AdminLTE

4.1.5 JetBrains PhpStorm

JetBrains PhpStorm é um IDE de plataforma cruzada comercial (ambiente de desenvolvimento integrado) para PHP criado na plataforma IntelliJ IDEA da JetBrains.

PhpStorm fornece um editor para PHP, HTML e Javascript com análise de código em tempo real, prevenção de erros e refatoração automatizada para código PHP e Javascript. A complementação de código do PhpStorm suporta PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1 e 7.2 (projetos modernos e legados), incluindo geradores, rotinas, a palavra-chave finally, lista em foreach, namespaces, closures, traits e short array syntax. Inclui um editor de SQL completo com resultados de consulta editáveis. Além disso permite a integração com Xdebug e Zend Debugger, com sistemas de controle de versão (Git por exemplo), com banco de dados e com o Framework CodeIgniter [16].

Na Figura 10 temos uma ilustração de sua interface gráfica.

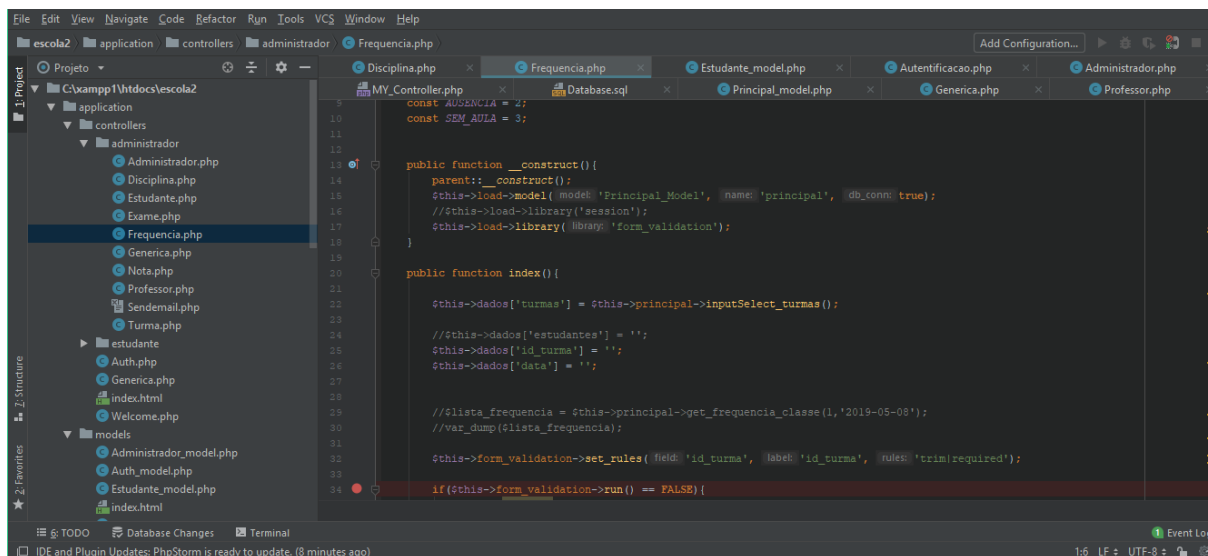


Figura 10: Interface PhpStorm

4.2 DIAGRAMA DE CLASSES CONCEITUAL

O diagrama de classes conceitual é usado para entender e analisar um domínio do problema além de permitir a captura do conhecimento sobre a funcionalidade desejada do sistema. Ao analisar o diagrama de casos de uso do sistema, muitos conceitos importantes podem ser descobertos. Esses conceitos são representados como classes no modelo conceitual. Eles representam a estrutura das informações que serão gerenciadas pelo sistema de acordo com as regras de negócios levantadas no Capítulo 3.

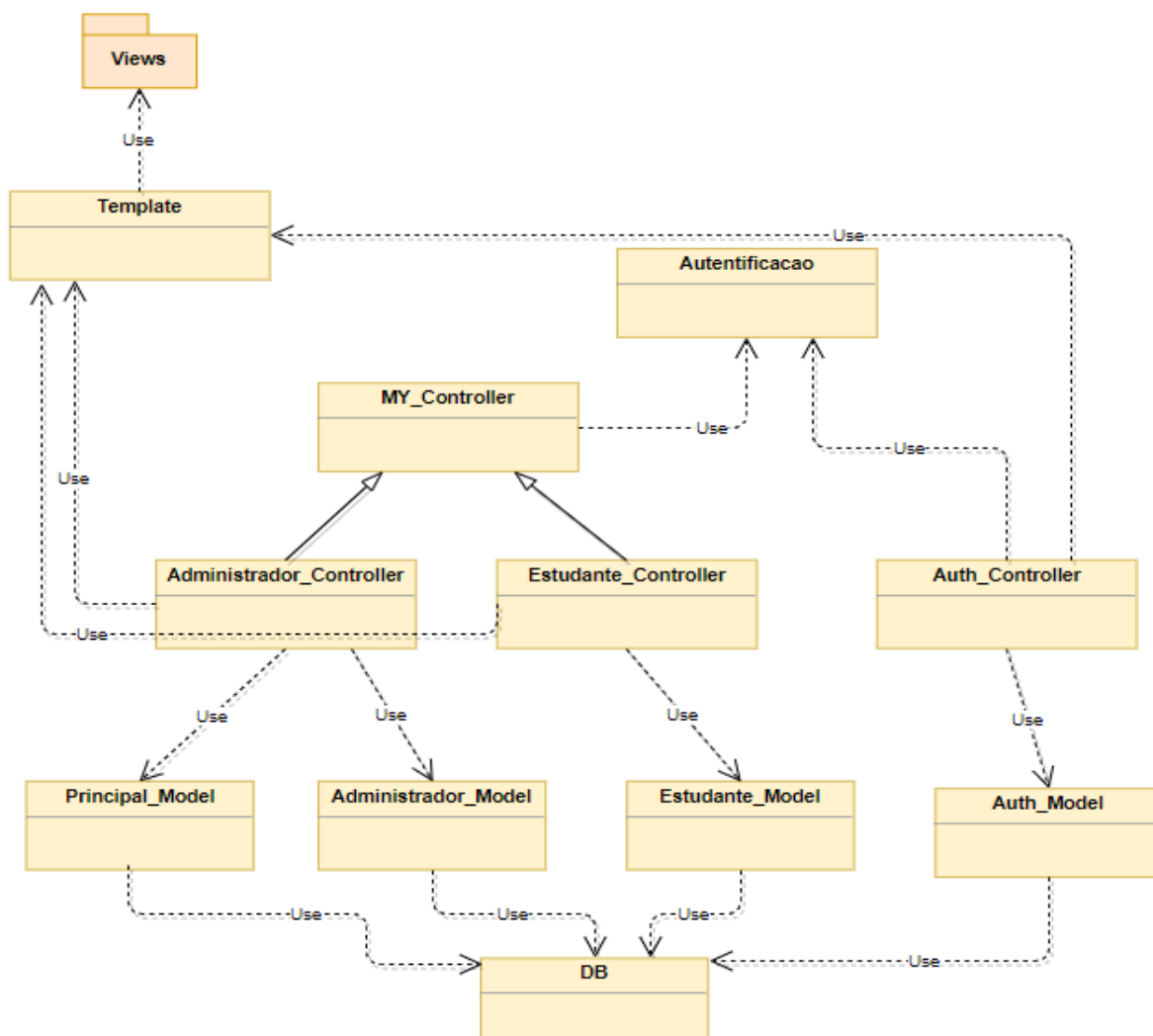


Figura 11: Diagrama conceitual de classes

Na Figura 11 mostramos um diagrama de classes conceitual de baixa nível de abstração, pois nós já levamos em consideração que a arquitetura que será utilizada (MVC). Nele nós mostramos as principais classes dos sistemas, suas interações (herança, agregação, associação) e as dependências mais relevantes. Porém foram omitidos as bibliotecas auxiliares, as operações e os atributos.

As classes “_Controller” são classes do tipo controle e neste projeto serão responsáveis por fazer o intermédio entre a camada de visualização e de modelo respondendo ao que deve ser feito a cada requisição específica.

As classes “_Model” serão as classes responsáveis pela persistência e operações com o banco de dados, por exemplo, de consulta, atualizar e adição. Para isso esta classe utiliza a classe auxiliar **DB**.

A classe Template é responsável pela renderização das Views do sistema.

E o pacote View é responsável por especificar as páginas ou fragmentos de páginas carregadas pelas classes de controle, via a classe Template, e apresentar para o usuário.

A classe Autenticação contém os métodos responsáveis pela autenticação (e-mail e senha) e pelas informações do tipo autorização dos mesmos.

A classe MY_Controller é o coração do sistema sendo responsável pelo direcionamento e carregamento do módulo que o sistema deve habilitar. Essa escolha é baseada no tipo de usuário que entra no sistema de acordo com a regra do usuário habilitando somente funcionalidades do sistema adequadas ao perfil. Sendo que as classes Administrador_Controller e Aluno_Controller são especializações desta mesma classe.

4.3 DIAGRAMA DE CLASSES DE PROJETO

O diagrama de classes de projeto é um refinamento do modelo conceitual sendo uma versão mais detalhada em que cada classe representa um objeto real do sistema. Nele podemos mostrar a estrutura e o comportamento de uma ou mais classes, além de mostrar informações como associações, métodos e atributos. Na prática esse diagrama pode ser construído conforme a fase do projeto avança.

Os diagramas de classes deste sistema foram desenhados com a ferramenta StarUML.

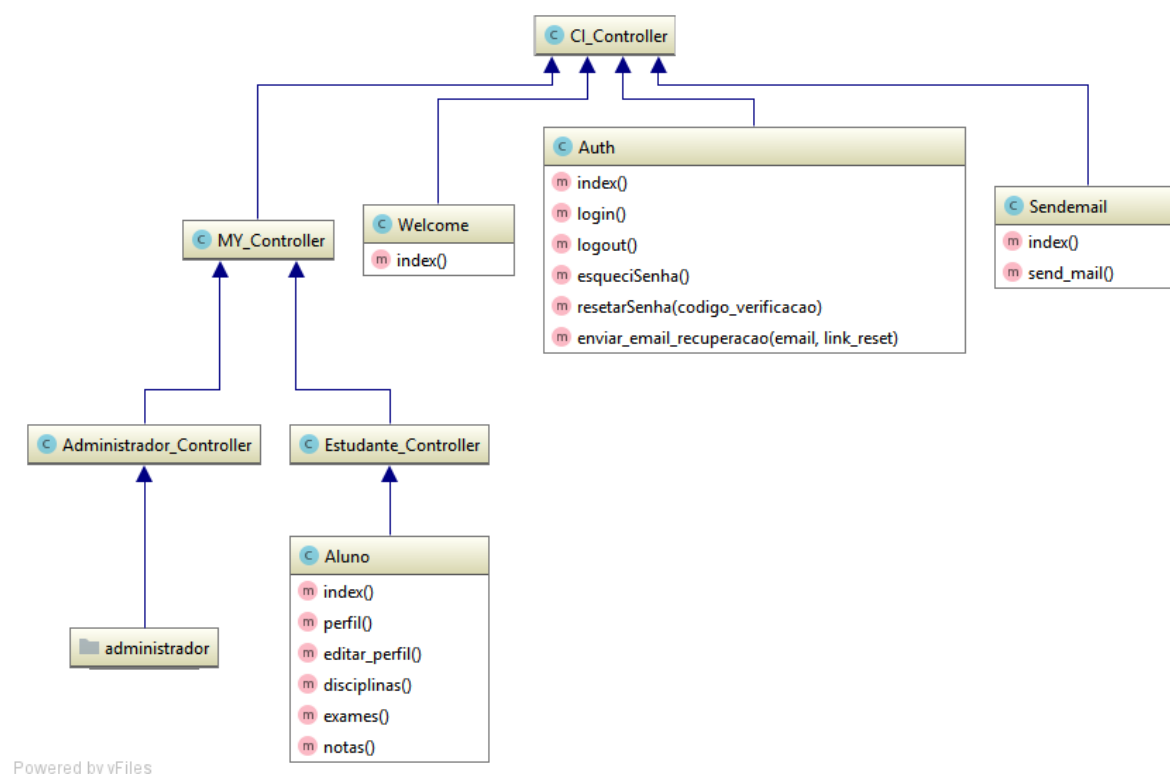
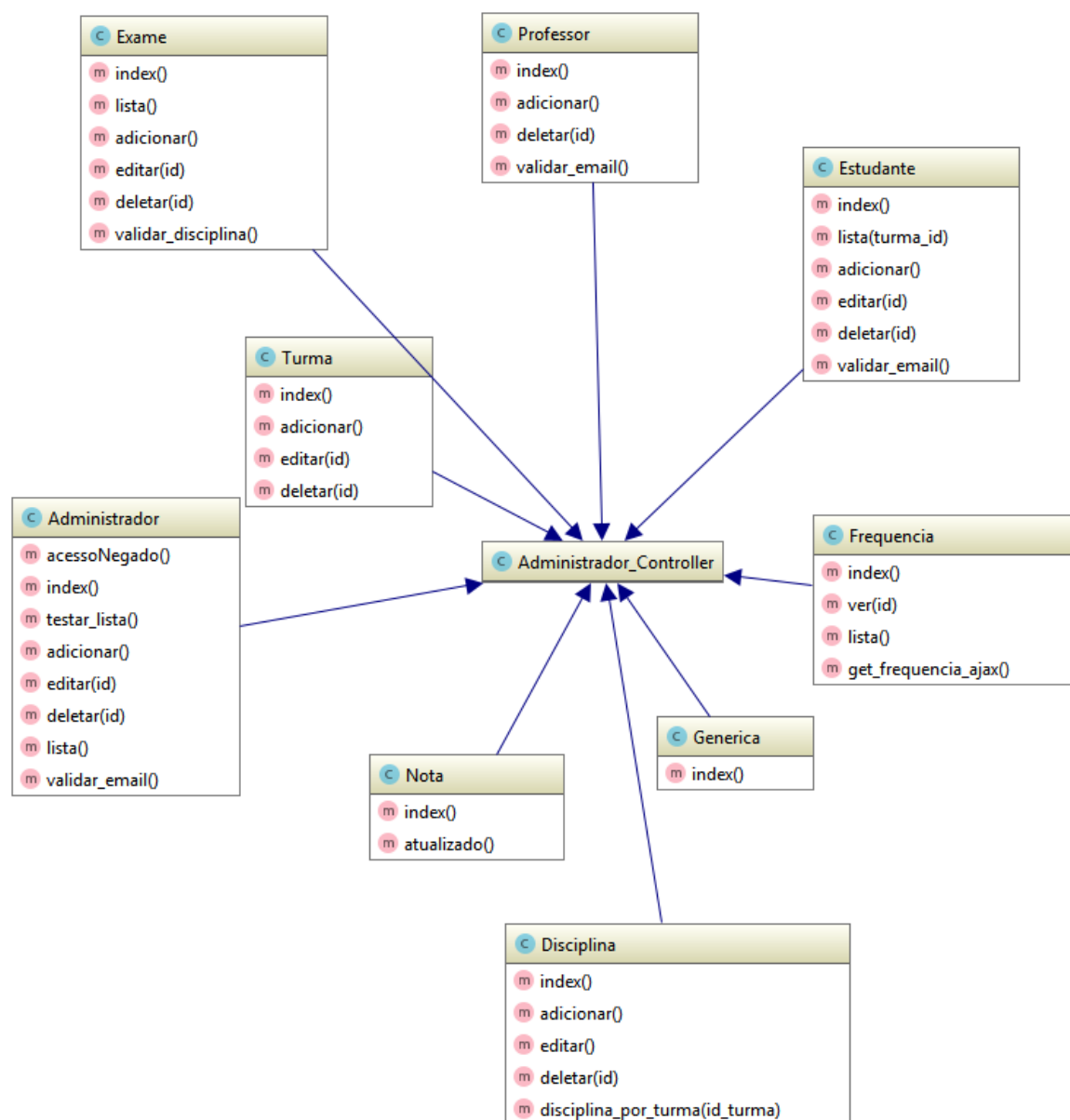


Figura 12: Diagrama de classes de controle

Conforme ilustrado na Figura 12, classe `CI_Controller` é uma superclasse pertencente ao core do CodeIgniter, em que cada biblioteca é designada. Temos a classe `My_Controller`, descrita anteriormente, `Welcome` responsável pela página inicial e a classe `Sendemail` responsável pelo envio e conexão com servidores de e-mail.

A classe `Auth` controla a autenticação inicial do usuário, além de conter os métodos para a recuperação e reset da senha dos usuários.

A classe `Aluno` é uma espécie componente do módulo estudante (`Estudante_Controller`), sendo ela uma especialização da mesma. Futuramente serão adicionados outros componentes.



Powered by yFiles

Figura 13: Diagrama de Classes do Administrador_Controller

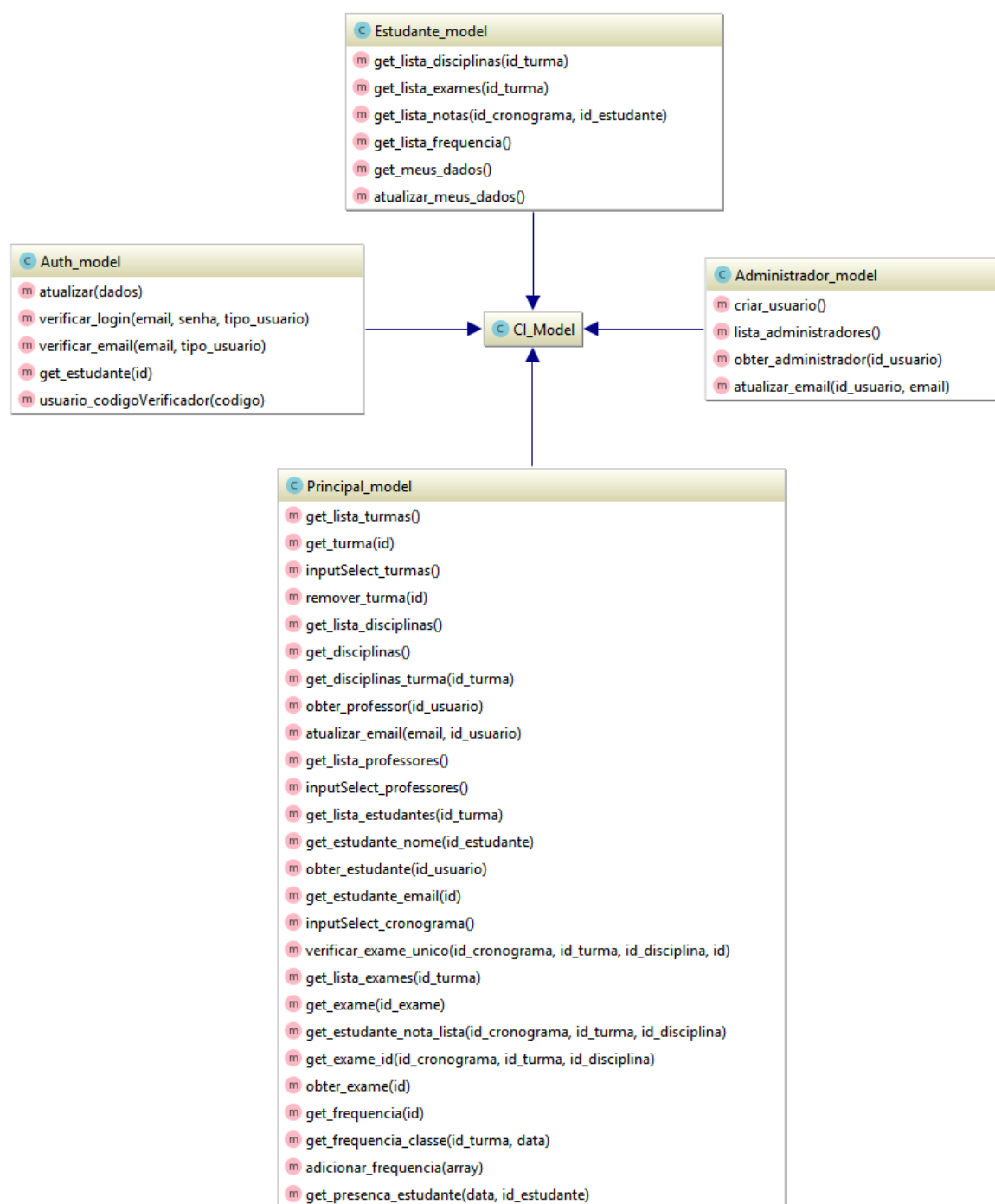
Conforme ilustrado na Figura 13, a classe **Administrador_Controller** é a classe genérica dos componentes (classes) utilizados pelo módulo de administração sendo ela a responsável por controlar quando cada componente deve ser solicitado.

Entre as responsabilidades de cada componente temos as seguintes:

- Classe **Disciplina**: Responsável por gerenciar as disciplinas;
- Classe **Estudante**: Responsável por gerenciar os estudantes;
- Classe **Professor**: Responsável por gerenciar os professores;

- Classe Administrador: Responsável por gerenciar os administradores.
- Classe Turma: Responsável por gerenciar as turmas;
- Classe Exame: Responsável por gerenciar os exames;
- Classe Frequência: Responsável por gerenciar a frequência dos estudantes;
- Classe Nota: Responsável por gerenciar as notas dos exames para cada estudante;
- Classe Genérica: Classe Default utilizada quando nenhuma outra classe é carregada sendo ela um requisito do framework.

Na Figura 14 ilustramos o diagrama de classes da camada Modelo. Conforme podemos visualizar, as classes `Administrador_Model`, `Principal_Model`, `Auth_Model` e `Estudante_Model` são especializações da superclasse `CI_Model`.



Powered by yFiles

Figura 14: Diagrama de classes da camada Modelo

4.4 MODELO LÓGICO DE DADOS

Um modelo de lógico de dados representa a arquitetura de dados e a organização de forma gráfica, sem levar em conta a implementação física ou a tecnologia do sistema de gerenciamento de banco de dados envolvida no armazenamento dos dados. Um modelo de dados lógico fornece todas as informações sobre as várias entidades e os relacionamentos entre as entidades presentes em um banco de dados.

Normalmente um modelo lógico de dados contém representações de entidades e atributos, relacionamentos, identificadores exclusivos e domínios.

A ilustração da Figura 15 representa graficamente as entidades e seus relacionamentos utilizados neste projeto. Para sua elaboração foi utilizada a ferramenta MySQL WorkBench.

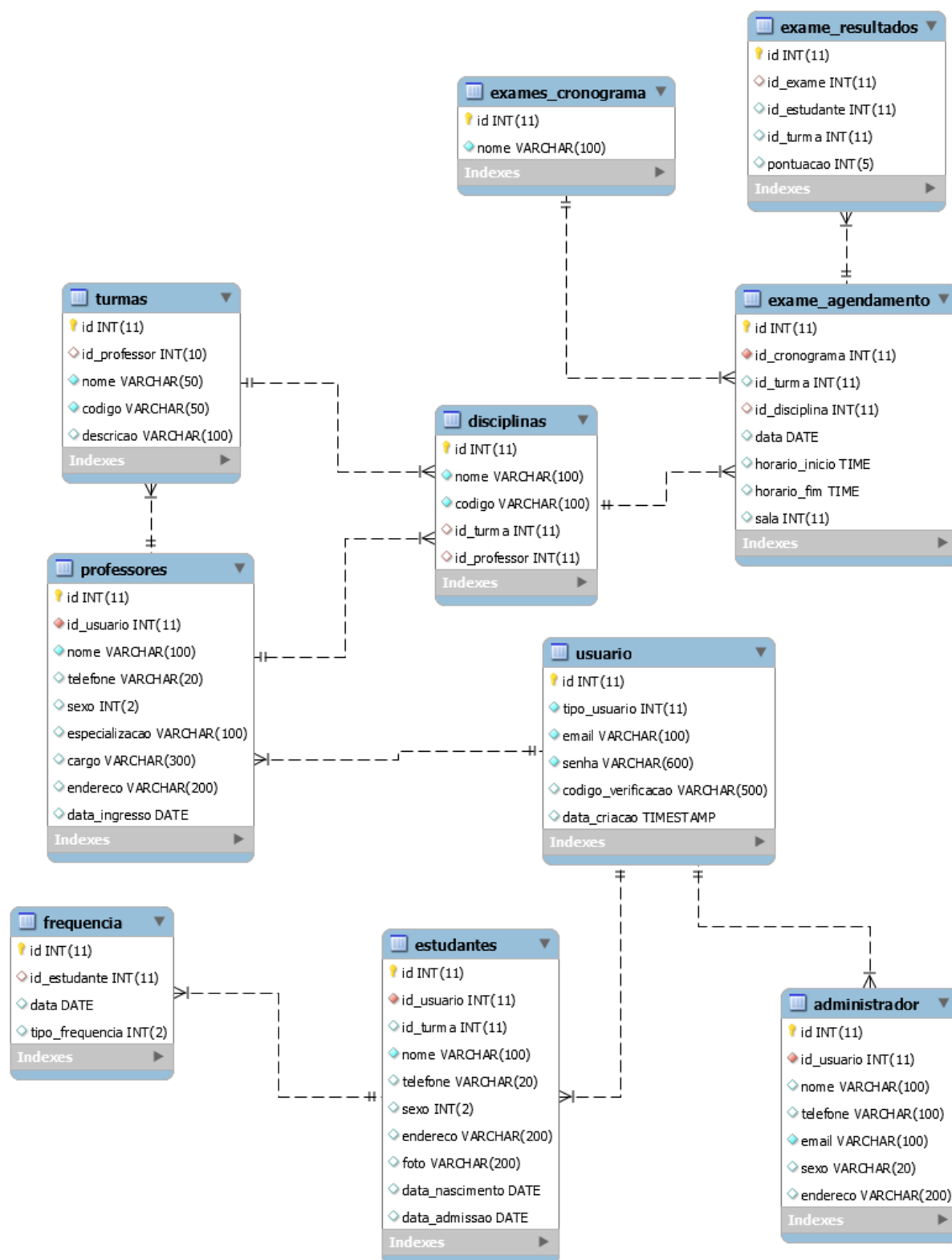


Figura 15: Modelo do Banco de Dados

O código SQL para criação das tabelas do BD se encontra no Apêndice A.

4.5 INTERFACE DO SISTEMA

Neste Capítulo será apresentando um protótipo da interface do sistema, com seus respectivos layouts, mostrando através das principais telas, suas funcionalidades, validações e interações do usuário com o sistema.

4.5.1 Tela de Login e Recuperação de Senha

O protótipo da interface de login do Sistema de Gerenciamento Escolar apresenta o seguinte layout:

- Título:** Sistema de Gerenciamento Escolar
- Instrução:** Insira os detalhes de Login
- Campos de Entrada:**
 - Email:** Campo de texto com a label "Email" e uma mensagem de validação "Este campo é requerido." com ícone de envelope.
 - Senha:** Campo de texto com a label "Senha" e uma mensagem de validação "Este campo é requerido." com ícone de cadeado.
- Seleção de Perfil:** Menu suspenso com a label "Selecione uma opção" e uma lista de opções: "Administrador", "Estudante" e "Professor".
- Botão de Login:** Botão azul com o texto "Entrar".
- Link de Recuperação:** Link azul com o texto "Eu esqueci minha senha".

Figura 16: Interface de Autenticação

Conforme a Figura 16, a primeira tela do aplicativo é uma interface de autenticação, nela nós contamos com duas opções.

A primeira opção é fazer o login de modo a possibilitar o acesso ao sistema. Para isso devemos informar os valores de E-mail e Senha, além de selecionar o nível de privilégio do usuário. O sistema conta com 3 níveis: Administrador, Estudante e Professor. Com os campos preenchidos e pressionando-se o botão Acessar, o usuário obtém acesso ao sistema carregando a página inicial correspondente ao seu nível de privilégio.

Na segunda, temos um link identificado como “Esqueci minha senha”, nele podemos fazer a solicitação de recuperação de senha, conforme a Figura 18. Para usa essa funcionalidade, primeiro o usuário o formulário com E-mail e Tipo de Usuário, se a informação a informação for válida é enviado um e-mail de recuperação de senha para o usuário com um link e as instruções. Este link redireciona para uma página do sistema responsável pela mudança de senha. Para segurança desse processo é utilizado código único criptografo que está atrelado ao link de recuperação e relacionado ao usuário no sistema, para trabalhos futuros será colocado um limite de tempo nesse código único.

Sistema de Gerenciamento Escolar

JANELA 1
Recuperar senha

Insira o email cadastrado

Email

Selecione uma opção

- Selecione uma opção
- Administrador
- Estudante
- Professor

Enviar

[VOLTAR](#)

Sistema de Gerenciamento Escolar

JANELA 2

Insira a nova Senha

Senha

Senha

Mudar Senha

Figura 17: Telas de Recuperação de Senha

4.5.2 Tela Principal da Administração

Essa página é exibida depois da autenticação (login) bem-sucedida do Administrador. Além disso, somente Administradores podem acessar essa página. O layout dessa página pode ser observado na Figura 18.

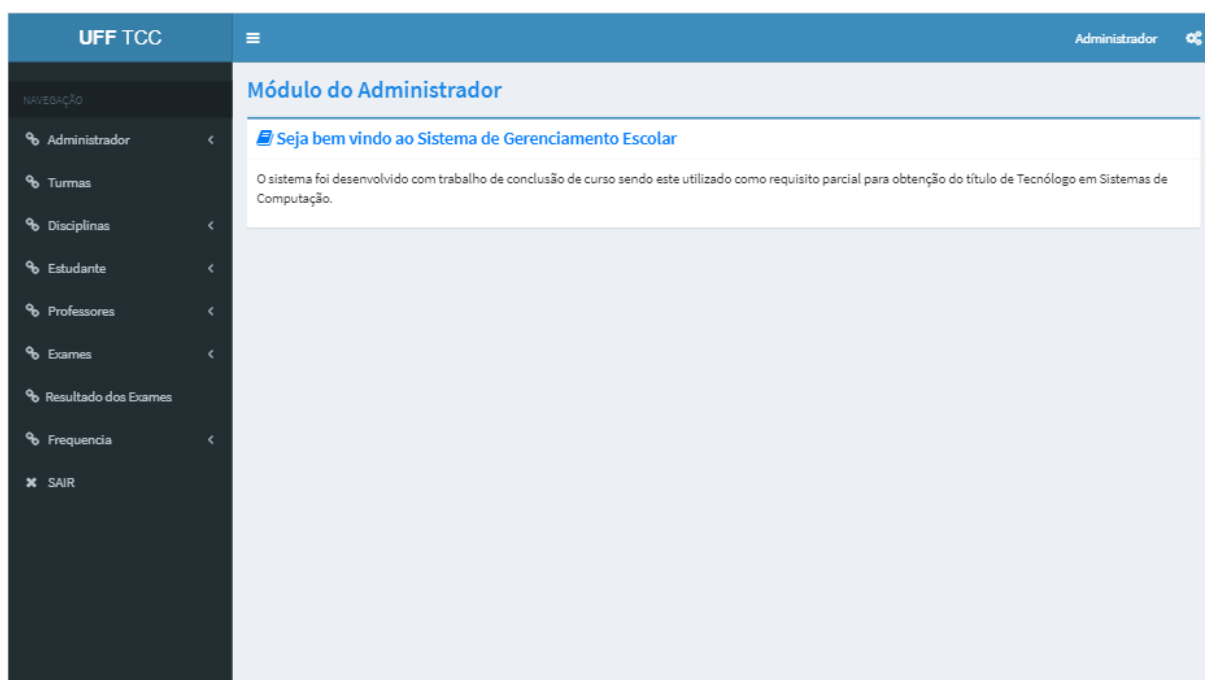


Figura 18: Tela Inicial da Administração

4.5.3 Interface de Listagem Genérica

A interface de Listagem Genérica é um modelo de layout utilizado para exibição, em formato de tabela, dos dados de todos administradores, turmas, disciplinas, estudantes, exames e professores cadastrados no banco de dados do sistema.

Cada uma dessas tabelas tem uma coluna de Ação, em que dependendo da entidade, ela poderá ter os seguintes botões com as respectivas funções:

- Ver: Serve para visualizar os dados do objeto em questão;
- Editar: Serve para editar os dados do objeto em questão;
- Deletar: Serve para deletar o objeto do banco de dados.

E ainda, no cabeçalho de cada tabela, temos os campos Paginação, em que limitamos quantas linhas queremos mostrar por página e o campo pesquisar, que ao ser preenchido pesquisará em todas as colunas o dado desejado e depois mostrará o resultado na Tabela.

Na tela de listagem de administradores temos as colunas de Posição, Nome, E-mail, Telefone e Ação, conforme ilustrado na Figura 19

Posição	Nome	Email	Telefone	Ação
1	ADMINISTRADOR1	heturi.pereira@gmail.com	(55) 4455-55555	Ver Editar Deletar
2	ADMINISTRADOR2	superadmin2856@gmail.com	(55) 4455-55555	Ver Editar Deletar
3	ADMINISTRADOR3	daniel1308.pereira@gmail.com	(55) 4455-55555	Ver Editar Deletar
4	ADMINISTRADOR4	daniel8568.pereira@gmail.com	(55) 4455-55555	Ver Editar Deletar
5	ADMINISTRADOR5	superadmin25864@gmail.com	(55) 4455-55555	Ver Editar Deletar
6	ADMINISTRADOR6	heturi25.pereira@gmail.com	(24) 2999-65566	Ver Editar Deletar
7	ADMINISTRADOR7	heturi29.pereira@gmail.com	(55) 4455-55555	Ver Editar Deletar

Figura 19: Tela de listagem de Administradores

Na tela de listagem de turmas temos as colunas de Código, Turma, Professor Responsável, Descrição e Ação, conforme ilustrado na Figura 20.

UFF TCC Administrador

NAVEGAÇÃO

- Administrador
- Turmas
- Disciplinas
- Estudante
- Professores
- Exames
- Resultado dos Exames
- Frequencia
- SAIR

Turmas

+ Adicionar Turma

10 resultados por página Pesquisar

Código	Turma	Professor Responsável	Descrição	Ação
2	1º Série	Professor2	teste	Editar Deletar
408	3º Série	Professor1		Editar Deletar
vgr	2º Série	Professor1		Editar Deletar

Mostrando de 1 até 3 de 3 registros

Anterior 1 Próximo

Figura 20: Tela de listagem de Turmas

Na tela de listagem de disciplinas temos as colunas de Nome da Disciplina, Código, Turma, Professor e Ação, conforme ilustrado na Figura 21.

UFF TCC Administrador

NAVEGAÇÃO

- Administrador
- Turmas
- Disciplinas
- Estudante
- Professores
- Exames
- Resultado dos Exames
- Frequencia
- SAIR

Disciplinas

+ Adicionar Disciplina

10 resultados por página Pesquisar

Nome da Disciplina	Código	Turma	Professor	Ação
Geografia	geo		Professor3	Editar Deletar
Historia	HIS101		Professor1	Editar Deletar
Matematica	Mat	1º Série	Professor2	Editar Deletar

Mostrando de 1 até 3 de 3 registros

Anterior 1 Próximo

Figura 21: Tela de listagem de Disciplinas

Na tela de listagem de estudantes temos as colunas de Número, Nome, E-mail, Telefone, Turma e Ação. Além disso, há um campo onde podemos filtrar os estudantes de acordo com a turma em que se encontra cadastrado, conforme ilustrado na Figura 22.

Estudantes

Selecione a Turma

Adicionar Estudante

Por página

Pesquisar

		Nome	Email	Telefone	Turma	Ação
		Estudante3	estudante3@gmail.com	5555	1º Série	Ver Editar Deletar
10008		DANIEL BARBOSA PEREIRA	estudante4@gmail.com		1º Série	Ver Editar Deletar
10009		DANIEL B PEREIRA	estudante5@gmail.com		1º Série	Ver Editar Deletar
10010		Estudante_login	estudante_login@gmail.com	(24) 8565-65658	1º Série	Ver Editar Deletar

Mostrando de 1 até 4 de 4 registros

Anterior 1 Próximo

Figura 22: Tela de listagem de Estudantes

E na tela de listagem de exames temos as colunas de Avaliação, Turma, Disciplinas, Data, Horário de Início, Horário de Fim, Sala e Ação, conforme ilustrado na Figura 23.

Avaliação	Turma	Disciplina	Data	H. Início	H. Fim	Sala	Ação
1º Bimestre	1ª Série	Matematica	2019-04-09	00:30:00	02:00:00	508	Editar Deletar
3º Bimestre	1ª Série	Matematica	2019-04-25	01:30:00	04:00:00	508	Editar Deletar

Mostrando de 1 até 2 de 2 registros

Anterior 1 Próximo

Figura 23: Tela de listagem de Exames

4.5.4 Tela de cadastro de Administradores

A tela de cadastro de administradores exibe um formulário com o objetivo de realizar o cadastro de novos administradores. Neste formulário os campos Nome, E-mail, Telefone, Sexo e Senha são obrigatórios e o Endereço é opcional, conforme ilustrado na Figura 24.

Caso o e-mail já esteja cadastrado no banco de dados, o sistema retorna uma mensagem de erro solicitando que o usuário escolha outro e-mail.

The screenshot shows a web application interface for 'UFF TCC'. On the left is a dark sidebar with a 'NAVEGAÇÃO' menu containing items like 'Administrador', 'Turmas', 'Disciplinas', 'Estudante', 'Professores', 'Exames', 'Resultado dos Exames', 'Frequencia', and 'SAIR'. The main content area has a blue header with 'UFF TCC' and 'Administrador'. Below the header, the title 'Adicionar Administrador' is displayed. The form contains several input fields: 'Nome*' (required), 'Email*' (required, highlighted with a red border and the message 'Este campo é requerido.'), 'Telefone*', 'Endereço', 'Sexo*' (a dropdown menu with '--Selecionar--'), and 'Senha' (password field). A blue 'Adicionar' button is at the bottom of the form.

Figura 24:Tela de cadastro de Administradores

4.5.5 Tela de cadastro de Turmas

O objetivo dessa tela é realizar o cadastro de novas turmas vinculando-as ao professor responsável atendendo a RN7. É necessário cadastrar o Nome, o Número, o Professor Responsável e uma Descrição sendo este último campo opcional, vide Figura 25.

UFF TCC

NAVEGAÇÃO

- Administrador
- Turmas
- Disciplinas
- Estudante
- Professores
- Exames
- Resultado dos Exames
- Frequencia
- SAIR

Adicionar Turma

Nome da Turma*:

Número*:

Professor Responsável*:

Descrição:

Adicionar

localhost/escola2/administrador/turma/index2.html

Figura 25: Tela de cadastro de Turmas

4.5.6 Tela de cadastro de Disciplinas

O objetivo dessa tela é realizar o cadastro de novas disciplinas vinculando-as tanto ao professor responsável por lecionar a disciplina tanto como a turma que ela pertence. Em outras palavras, essa tela vai cadastrar as disciplinas lecionadas em cada turma.

No formulário apresentado os campos Nome, o Código da Disciplina, Professor Responsável e Turma são obrigatórios conforme ilustrado na Figura 26.

NAVEGAÇÃO

- Administrador
- Turmas
- Disciplinas
- Estudante
- Professores
- Exames
- Resultado dos Exames
- Frequencia
- SAIR

Adicionar Disciplina

Nome*:

Nome

Código da Disciplina*:

Professor Responsável*:

Professor2

Turma*:

2º Série

Adicionar

Figura 26: Tela de cadastro de Disciplinas

4.5.7 Tela de cadastro de Estudantes

O objetivo dessa tela é realizar o cadastro de novos alunos vinculando-os a turma em que será inscrito. Em trabalhos futuros, será permitido que usuários não autenticados realizem a pré-matrícula do estudante, informando os dados dele e enviando seus respectivos documentos.

No formulário apresentado os campos Nome, E-mail, Turma, Sexo, Data de Nascimento, Data de Admissão e são obrigatórios e os campos Telefone, Endereço são opcionais. Além disso, no campo Foto o usuário pode fazer o upload de uma foto do estudante, desde que, seja respeitado as limitações em relação ao tipo e tamanho da imagem estabelecida pelo sistema.

Na Figura 27, podemos visualizar o leiaute desta tela.

Adicionar Estudante

FORMULÁRIO

Nome:*

Email:*

Turma:*

Telefone:

Endereço:

Sexo:*

Data de nascimento:*

Data de Admissão:*

Senha:*


Foto 

Foto Somente .jpg, .png ou .gif (Max 300kB);

Figura 27: Tela de cadastro de Estudantes

4.5.8 Tela de cadastro de Professores

Essa tela exibe um formulário com o objetivo de realizar o cadastro de novos professores. Neste formulário os campos Nome, E-mail, Telefone, Sexo, Cargo, Data de Ingresso e Senha são obrigatórios e os campos Endereço e especialização são opcionais, conforme ilustrado na Figura 28.

Figura 28: Tela de cadastro de Professores

4.5.9 Tela de cadastro de Estudantes

O objetivo dessa tela é realizar o cadastro dos Exames que serão realizadas pelos estudantes. Esses exames são divididos pelo período de aplicação (bimestre), pela turma em que será aplicada e pela disciplina correspondente conforme estabelecido nos requisitos do sistema (vide Capítulo 3). Ou seja, os mesmos são vinculados a Avaliação (Bimestre de aplicação), a Turma e a Disciplina.

No formulário apresentado na tela, os campos Avaliação, Turma, Disciplina, Data, Horário de Início, Horário de Fim e N° da Sala são obrigatórios. Em relação ao

campo Disciplina do tipo *Select*, ele é populado dinamicamente via Ajax de acordo com a Turma escolhida no campo anterior.

Na Figura 29, podemos visualizar o leiaute desta tela.

A interface 'Adicionar Exame' apresenta um menu lateral de navegação com as seguintes opções: Administrador, Turmas, Disciplinas, Estudante, Professores, Exames, Resultado dos Exames, Frequencia e SAIR. O formulário principal contém os seguintes campos obrigatórios:

- Avaliação*:** Dropdown menu com a opção selecionada '1º Bimestre'.
- Turma*:** Dropdown menu com a opção selecionada '1º Série'.
- Disciplinas*:** Dropdown menu com a opção selecionada 'Matematica'.
- Data*:** Campo de texto para a data da avaliação.
- Horário de Início*:** Campo de texto para o horário de início.
- Horário de Término*:** Campo de texto para o horário de término.
- Nº da Sala*:** Campo de texto para o número da sala.

Um botão azul 'Adicionar' está localizado na base do formulário.

Figura 29: Tela de cadastro de Exames

4.5.10 Interface de Resultado Acadêmico

O objetivo desta interface é permitir o lançamento e a alteração das notas recebidas pelos alunos nas disciplinas que estejam matriculados.

Na Figura 30 temos o leiaute da primeira tela desta interface que consiste de um formulário de pesquisa cujo objetivo é procurar no banco de dados o exame em que deseja atribuir a pontuação dos alunos. Neste formulário temos os campos Avaliação, Turma e disciplina, sendo este último populado via Ajax baseado na turma

escolhida no campo anterior. Caso o exame não seja encontrado, o sistema apresentar uma mensagem de erro com os seguintes dizeres: “A avaliação pesquisada não foi encontrada no sistema. Por favor crie o exame primeiro”.

The screenshot shows the 'Resultado Acadêmico' (Academic Result) page in the UFF TCC system. The interface includes a dark sidebar on the left with a 'NAVEGAÇÃO' (Navigation) menu containing links for 'Administrador', 'Turmas', 'Disciplinas', 'Estudante', 'Professores', 'Exames', 'Resultado dos Exames', 'Frequencia', and 'SAIR'. The main content area has a header with 'UFF TCC' and 'Administrador'. Below the header, the title 'Resultado Acadêmico' is displayed. A search section titled 'Selecione os Parâmetros' contains three dropdown menus: 'Avaliação*' (set to '1º Bimestre'), 'Turma*' (set to '1º Série'), and 'Disciplina*' (set to 'Matematica'). A 'Pesquisar' (Search) button is located to the right of these filters.

Figura 30: Tela resultado Acadêmico

Entretanto, em caso positivo, o sistema abre uma tela para incluirmos ou alteramos a pontuação dos alunos, conforme ilustrado na Figura 31.

Gerenciar Notas

Matematica

Identificação	Foto	Nome	Pontuação
10002		Estudante3	<input type="text" value="50"/>
10008		DANIEL BARBOSA PEREIRA	<input type="text" value="75"/>
10009		DANIEL B PEREIRA	<input type="text" value="55"/>
10010		Estudante_login	<input type="text" value="80"/>

Mostrando de 1 até 4 de 4 registros

[Salvar Notas](#)

Figura 31: Tela de gerenciamento de Notas

4.5.11 Interface de Frequência

A interface de Frequência é responsável pelo gerenciamento da frequência escolar dos estudantes. A tela principal desta interface, conforme ilustrado na Figura 32, é composta de:

- Uma tabela com a listagem de todos os alunos matriculados
- Um campo do tipo select populado com todas as turmas cadastradas. O objetivo deste campo é filtrar os dados da tabela de alunos por turma;
- Do botão “Adicionar Frequência”.

Frequencia dos Alunos

Selecione a Turma Adicionar Frequencia

10 resultados por página Pesquisar

Número	Foto	Nome	Email	Telefone	Turma	Frequencia
10002		Estudante3	estudante3@gmail.com	5555	1ª Série	
10008		DANIEL BARBOSA PEREIRA	estudante4@gmail.com		1ª Série	
10009		DANIEL B PEREIRA	estudante5@gmail.com		1ª Série	
10010		Estudante_login	estudante_login@gmail.com	(24) 8565-65658	1ª Série	

Mostrando de 1 até 4 de 4 registros Anterior 1 Próximo

Figura 32: Tela principal de Frequência

Na tabela com a lista de alunos, além dos campos de Número, Foto, E-mail, Telefone e Turma, temos também o campo Frequência que contém o botão Ver. Ao clicar neste botão o usuário é redirecionado para a página de visualização da frequência escolar do estudante.

Na Figura 33 temos o layout desta tela, onde a cor vermelha no calendário representa “ausência”, a cor verde “presença” e a cor cinza que neste dia não teve aula.

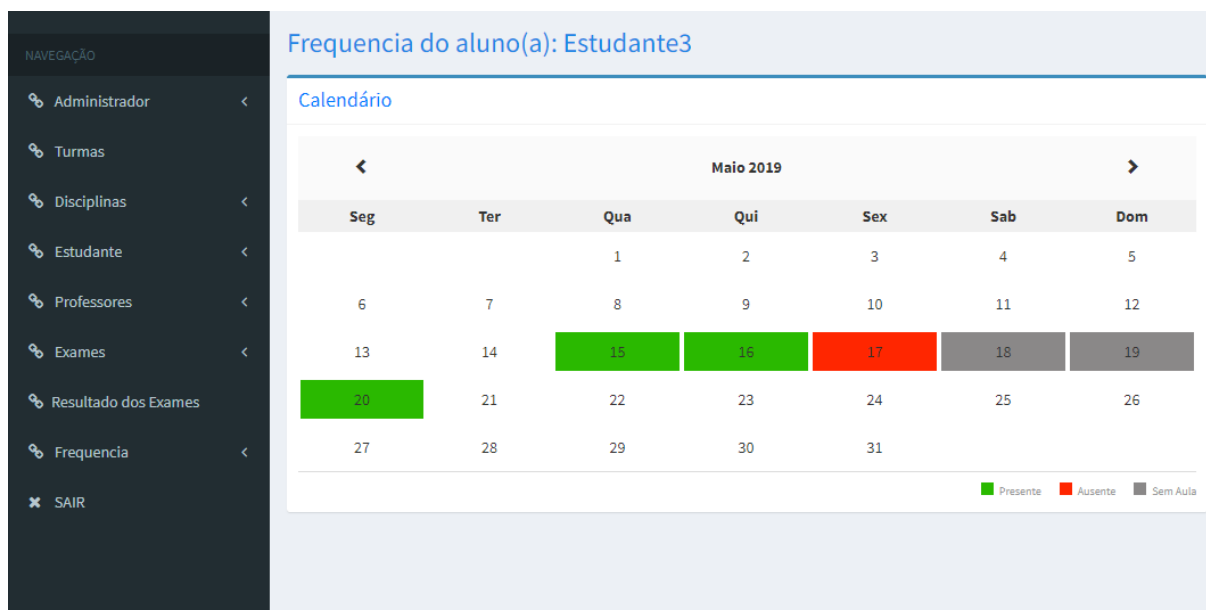


Figura 33: Tela de visualização Frequência escolar

Para realizar a chamada, o usuário deve clicar no botão “Adicionar Frequência”. Dessa forma, o sistema abre uma tela de pesquisa, onde o usuário escolhe a turma e a data em que realizará a chamada e clicar em Pesquisar. Logo em seguida, o sistema carregará uma lista de estudantes onde o usuário poderá cadastrar a frequência individualmente para cada indivíduo, conforme ilustrado na Figura 34.



Figura 34: Tela de Cadastro da Frequência Escolar

4.5.12 Interfaces do módulo Estudante

A página inicial do módulo Estudante é apresentada aos alunos cadastrados no sistema após estes realizarem o login através da interface de autenticação. Sendo semelhante a página inicial de Administração (vide Figura 35), esta possui a mesma regra de restrição quanto ao tipo de usuário, ou seja, somente usuários com status de Aluno tem autorização para visualizar esta página.

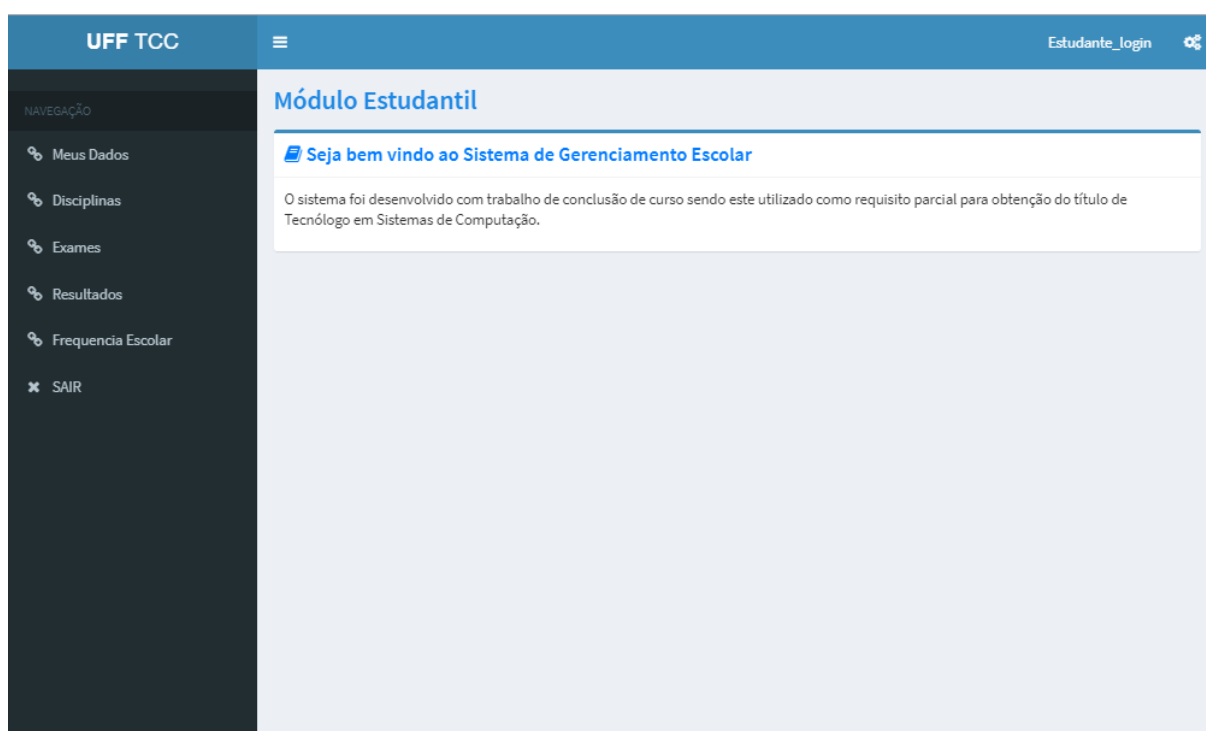
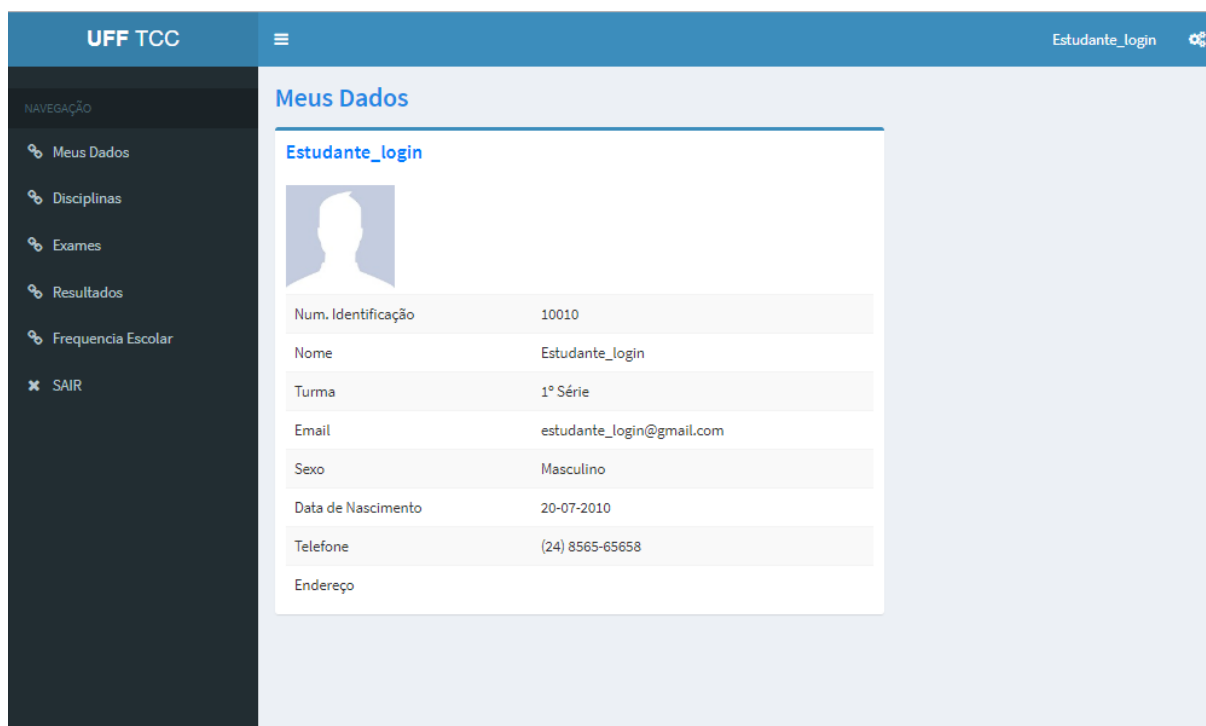


Figura 35: Página inicial do Estudante

Conforme podemos visualizar na Figura 35 no lado esquerdo da tela temos o menu de navegação. Este menu apresenta, além da opção de Sair do sistema, mais cinco opções: Meus Dados, Disciplinas, Exames, Resultados e Frequência Escolar. Cada uma dessas opções implementa uma interface específica, cujas funcionalidades descreveremos brevemente nos parágrafos abaixo.

O primeiro item do menu, Meus Dados, é responsável por mostrar, ao aluno, os seus dados cadastrados no sistema, conforme mostrado na Figura 36.



The screenshot shows the 'Meus Dados' (My Data) page in the UFF TCC system. The page has a blue header with 'UFF TCC' on the left and 'Estudante_login' on the right. A dark sidebar on the left contains a 'NAVEGAÇÃO' (Navigation) menu with options: 'Meus Dados', 'Disciplinas', 'Exames', 'Resultados', 'Frequencia Escolar', and 'SAIR'. The main content area is titled 'Meus Dados' and features a profile card for 'Estudante_login'. The card includes a placeholder for a profile picture and a table of personal information.

Num. Identificação	10010
Nome	Estudante_login
Turma	1º Série
Email	estudante_login@gmail.com
Sexo	Masculino
Data de Nascimento	20-07-2010
Telefone	(24) 8565-65658
Endereço	

Figura 36: Tela Meus Dados

A tela de Disciplina é responsável por mostrar ao aluno, as disciplinas em que ele se encontra inscrito, ou seja, as disciplinas da turma que ele está cadastrado. Na Figura 37 podemos visualizar o leiaute desta tela.

UFF TCC	Estudante_login
NAVEGAÇÃO	Minhas Disciplinas
Meus Dados	
Disciplinas	
Exames	
Resultados	
Frequencia Escolar	
SAIR	

Nome da Disciplina	Código	Professor
Matematica	Mat	Professor2

Mostrando de 1 até 1 de 1 registros

Figura 37: Tela Minhas Disciplinas

A tela de Exames mostra os exames agendados para aquele período letivo ao aluno. Na Figura 38 podemos visualizar o leiaute desta tela.

UFF TCC	Estudante_login
NAVEGAÇÃO	Exames Agendados
Meus Dados	
Disciplinas	
Exames	
Resultados	
Frequencia Escolar	
SAIR	

Avaliação	Disciplina	Data	H. Inicio	H. Fim	Sala
1º Bimestre	Matematica	09-04-2019	00:30:00	02:00:00	508
3º Bimestre	Matematica	25-04-2019	01:30:00	04:00:00	508

Mostrando de 1 até 2 de 2 registros

Figura 38: Tela Exames Agendados

A tela de Resultados mostra a pontuação obtida nas avaliações realizadas pelo aluno. Na Figura 39 podemos visualizar o leiaute desta tela.



Avaliação Acadêmica		
Resultados do Primeiro Bimestre		
Nome da Disciplina	Código	Pontuação
Matematica	Mat	80
Resultados do Segundo Bimestre		
Nome da Disciplina	Código	Pontuação
Nenhum registro encontrado		
Resultados do Terceiro Bimestre		
Nome da Disciplina	Código	Pontuação
Nenhum registro encontrado		
Resultados do Quarto Bimestre		
Nome da Disciplina	Código	Pontuação
Nenhum registro encontrado		

Figura 39: Tela Avaliação Acadêmica

A tela de Frequência mostra a frequência escolar do aluno através de um calendário mensal. Na Figura 40 podemos visualizar o leiaute desta tela.

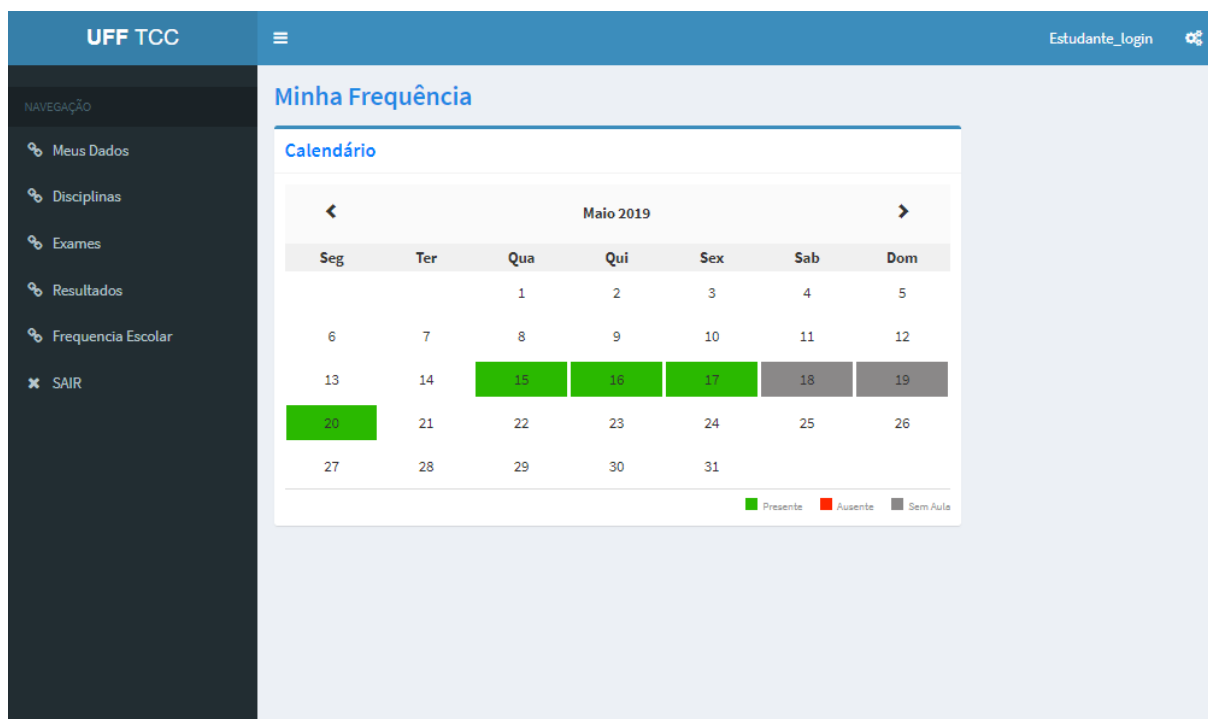


Figura 40: Tela Minha Frequência

5 CONCLUSÕES E TRABALHOS FUTUROS

Os recursos trazidos pela Tecnologia da Informação são evidentes. A automatização de processos burocráticos da instituição de ensino, além de reduzir custos, permitiu uma melhor utilização do tempo dos funcionários, permitindo-os se concentrar no objetivo final da escola: a aprendizagem efetiva por parte dos alunos. Assim, um software de gestão escolar faz com que a instituição de ensino funcione de forma integrada e independente.

Dessa forma, o intuito maior deste trabalho foi construir o núcleo do sistema, onde os principais requisitos foram identificados, analisados, projetados e, de modo que, novas funcionalidades fossem adicionadas no futuro com uma maior facilidade.

Utilizando-se do modelo incremental, foi possível a separação dos requisitos em módulos, o que permitiu otimizar o tempo além de possibilitar que os novos requisitos ou melhorias, que apareciam durante a implementação, fossem adicionados ao sistema sem gerar um retrabalho dos módulos anteriores.

Além disso, a utilização de técnicas padronizadas como a modelagem em MVC e a utilização de ferramentas conceituadas e testadas, como o CodeIgniter e Bootstrap, permitiram que o sistema fosse implementado de forma mais eficiente, integrada e segura.

O sistema, apesar de não estar acabado, se encontra plenamente funcional, visto a metodologia incremental utilizada para o processo de desenvolvimento do software proposto neste trabalho. Sendo assim, será apresentado abaixo, algumas sugestões de trabalhos futuros:

- Codificar e testar o módulo **Professor** conforme modelado neste trabalho;
- Permitir a pré-matrícula de novos Alunos por usuários não autenticados, permitindo o envio de documentos;
- Permitir que professores adicionem deveres escolares no sistema;

- Implementar uma função para gerar relatórios de frequência, contabilizando presenças, e faltas;
- Implementar uma função para gerar o histórico escolar autenticado com validação;
- Implementar um sistema mensagens geradas pelo sistema para informar o estudante sobre ações realizadas ou lembretes. Exemplo: “Prova de Matemática foi marcada para...”, “Novo dever de casa adicionado”, “Reunião no próximo dia” e etc.;
- Refinar o sistema de controle de usuário;
- Refinar o leiaute do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

1. IEEE. **IEEE Transactions on Software Engineering**. [S.I.]: IEEE Computer Society, 1992.
2. CARVALHO, A. M. **Introdução à Engenharia de Software**. São Paulo: Editora da Unicamp, 2001.
3. PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. 6ª. ed. [S.I.]: Mc Graw Hill, 2011.
4. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 12207**. Rio de Janeiro. 1998.
5. FALBO, R. D. A. **Engenharia de Software**, 2005. Disponível em: <https://www.inf.ufes.br/~falbo/files/ES/Notas_Aula_Engenharia_Software.pdf> . Acesso em: 18 fev. 2019.
6. SOMMERVILLE, I. **Engenharia de software**. 8º. ed. São Paulo: Pearson, 2007.
7. BOOCH, G.; RUMBAUGH, J. **UML Guia do Usuário**. 1º. ed. São Paulo: Elsevier, 2006.
8. MELO, A. C. **Desenvolvendo aplicações com UML 2.2 do conceito à implementação**. 3ª. ed. Rio de Janeiro: Brasport, 2010.
9. SANTOS,. **Possibilidades e limitações da arquitetura mvc (model –view – controller) com ferramenta ide (integrated development environment)**. Universidade José do Rosário Vellano. Alfenas, p. 56. 2010.
10. DEXTRA. Requisito ou Regra de Negócio? **Dextra**, 2019. Disponível em: <<https://dextra.com.br/pt/requisito-ou-regra-de-negocio/>>. Acesso em: 21 abr. 2019.
11. ALENQUER, P. L. Regras de negócio para análise em ambientes OLAP, 2013. Disponível em: <<http://www.cin.ufpe.br/~sfa/Regras%20de%20Neg%3cio%20para%20An%e1lise%20em%20Ambientes%20OLAP.pdf>>. Acesso em: 20 abr. 2019.

12. CODEIGNITER. CodeIgniter Documentation. **CodeIgniter**, 2019. Disponível em: <<https://codeigniter.com/docs>>. Acesso em: 02 maio 2019.
13. ORACLE. MySQL 8.0 Reference Manual Including MySQL NDB Cluster 8.0. **Oracle MySQL**, 2019. Disponível em: <https://docs.oracle.com/cd/E17952_01/mysql-8.0-en/introduction.html>. Acesso em: 02 maio 2019.
14. APACHEFRIENDS. What is XAMPP? **XAMPP**, 2019. Disponível em: <<https://www.apachefriends.org/index.html>>. Acesso em: 20 abr. 2019.
15. BOOTSTRAP. The Bootstrap Blog. **Bootstrap**, 2019. Disponível em: <<https://blog.getbootstrap.com/>>. Acesso em: 02 maio 2019.
16. INTELLIJ IDEA. The Lightning-Smart PHP IDE. **Jetbrains**, 2019. Disponível em: <<https://www.jetbrains.com/phpstorm/>>. Acesso em: 20 abr. 2019.
17. MACORATTI, J. Padrões de Projeto : O modelo MVC - Model View Controller, 2012. Disponível em: <http://www.macoratti.net/vbn_mvc.htm>. Acesso em: 18 abr. 2019.
18. ARAUJO, E. S. **Desenvolvimento de Software**, 2015. Disponível em: <http://everson.com.br/files/Desenvolvimento_de_Software_-_impressao.pdf>. Acesso em: 15 abr. 2019.

APÊNDICE

APÊNDICE A – CÓDIGO DE CRIAÇÃO DAS TABELAS DO BD

```
/*
*MySQL Script generated by MySQL Workbench
*MySQL Workbench Forward Engineering
*****/

/*!40101 SET NAMES utf8 */;

/*!40101 SET SQL_MODE=' ' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
/*Table structure for table `administrador` */

DROP TABLE IF EXISTS `administrador`;

CREATE TABLE `administrador` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_usuario` int(11) NOT NULL,
  `nome` varchar(100) DEFAULT NULL,
  `telefone` varchar(100) DEFAULT NULL,
  `email` varchar(100) NOT NULL,
  `sexo` varchar(20) DEFAULT NULL,
  `endereco` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_usuario` (`id_usuario`),
  CONSTRAINT `administrador_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `usuario` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=latin1;

/*Table structure for table `disciplinas` */

DROP TABLE IF EXISTS `disciplinas`;

CREATE TABLE `disciplinas` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(100) NOT NULL,
  `codigo` varchar(100) NOT NULL,
  `id_turma` int(11) DEFAULT NULL,
  `id_professor` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_turma` (`id_turma`),
  KEY `id_professor` (`id_professor`),
  CONSTRAINT `disciplinas_ibfk_1` FOREIGN KEY (`id_turma`) REFERENCES `turmas` (`id`),
  CONSTRAINT `disciplinas_ibfk_2` FOREIGN KEY (`id_professor`) REFERENCES `professores` (`id`)
```

```

) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

/*Table structure for table `estudantes` */

DROP TABLE IF EXISTS `estudantes`;

CREATE TABLE `estudantes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_usuario` int(11) NOT NULL,
  `id_turma` int(11) DEFAULT NULL,
  `nome` varchar(100) NOT NULL,
  `telefone` varchar(20) DEFAULT NULL,
  `sexo` int(2) DEFAULT NULL,
  `endereco` varchar(200) DEFAULT NULL,
  `foto` varchar(200) DEFAULT NULL,
  `data_nascimento` date DEFAULT NULL,
  `data_admissao` date DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_usuario` (`id_usuario`),
  CONSTRAINT `estudantes_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES
`usuario` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=10010 DEFAULT CHARSET=latin1;

/*Table structure for table `exame_agendamento` */

DROP TABLE IF EXISTS `exame_agendamento`;

CREATE TABLE `exame_agendamento` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_cronograma` int(11) NOT NULL,
  `id_turma` int(11) DEFAULT NULL,
  `id_disciplina` int(11) DEFAULT NULL,
  `data` date DEFAULT NULL,
  `horario_inicio` time DEFAULT NULL,
  `horario_fim` time DEFAULT NULL,
  `sala` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_disciplina` (`id_disciplina`),
  CONSTRAINT `exame_agendamento_ibfk_1` FOREIGN KEY (`id_disciplina`) REFE-
RENCES `disciplinas` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;

/*Table structure for table `exame_resultados` */

DROP TABLE IF EXISTS `exame_resultados`;

CREATE TABLE `exame_resultados` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_exame` int(11) DEFAULT NULL,
  `id_estudante` int(11) DEFAULT NULL,
  `id_turma` int(11) DEFAULT NULL,
  `pontuacao` int(5) DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `Unico` (`id_exame`,`id_estudante`),
  CONSTRAINT `exame_resultados_ibfk_1` FOREIGN KEY (`id_exame`) REFERENCES
`exame_agendamento` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=48 DEFAULT CHARSET=latin1;

/*Table structure for table `exames_cronograma` */

```



```

DROP TABLE IF EXISTS `exames_cronograma`;

CREATE TABLE `exames_cronograma` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

/*Table structure for table `frequencia` */

DROP TABLE IF EXISTS `frequencia`;

CREATE TABLE `frequencia` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_estudante` int(11) DEFAULT NULL,
  `data` date DEFAULT NULL,
  `tipo_frequencia` int(2) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_estudante` (`id_estudante`),
  CONSTRAINT `frequencia_ibfk_1` FOREIGN KEY (`id_estudante`) REFERENCES
`estudantes` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=34 DEFAULT CHARSET=latin1;

/*Table structure for table `professores` */

DROP TABLE IF EXISTS `professores`;

CREATE TABLE `professores` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_usuario` int(11) NOT NULL,
  `nome` varchar(100) NOT NULL,
  `telefone` varchar(20) DEFAULT NULL,
  `sexo` int(2) DEFAULT NULL,
  `especializacao` varchar(100) DEFAULT NULL,
  `cargo` varchar(300) DEFAULT NULL,
  `endereco` varchar(200) DEFAULT NULL,
  `data_ingresso` date DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_usuario` (`id_usuario`),
  CONSTRAINT `professores_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES
`usuario` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

/*Table structure for table `turmas` */

DROP TABLE IF EXISTS `turmas`;

CREATE TABLE `turmas` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_professor` int(10) DEFAULT NULL,
  `nome` varchar(50) NOT NULL,
  `codigo` varchar(50) NOT NULL,
  `descricao` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_professor` (`id_professor`),
  CONSTRAINT `turmas_ibfk_1` FOREIGN KEY (`id_professor`) REFERENCES `pro-
fessores` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

/*Table structure for table `usuario` */

```

```

DROP TABLE IF EXISTS `usuario`;

CREATE TABLE `usuario` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `tipo_usuario` int(11) NOT NULL,
  `email` varchar(100) NOT NULL,
  `senha` varchar(600) NOT NULL,
  `codigo_verificacao` varchar(500) DEFAULT NULL,
  `data_criacao` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=36 DEFAULT CHARSET=utf8;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```