**Documentação de um**

**Web Sistema de Gestão Escolar**

**Versão 1.0**

Autores: Gutemberg Raposo da Silva, Wellysson Nascimento Rocha, Juan do Nascimento Gomes de Freitas

Revisor: Prof. Iury Oliveira Gomes

**2017**

**Documentação de um**

**Web Sistema de Gestão Escolar**

**Versão 1.0**

**ÍNDICE DETALHADO**

**Prefácio**

O objetivo deste documento é fornecer um roteiro para o desenvolvimento do sistema web de gestão escola utilizando os princípios do PHP orientada a objetos com notação

UML (Unified Modeling Language). É destinado a todos os alunos do Instituto Federal de Ciência e Tecnologia do Tocantins do curso de Análise e Desenvolvimento de Sistemas de Informação e Processamento de Dados, apoiando as disciplinas de Linguagem de Programação I e da disciplina de Banco de Dados I.

Esta é a versão 1.0 do documento é totalmente revisada para utilizar a notação UML. A escolha da orientação a objetos é devido à tendência de mercado, mas nada impede que o roteiro seja seguido no caso de opção pela Modelagem Estruturada (ver item Comentários sobre a Documentação).

No final deste documento há um glossário, os termos que constam no glossário são representados no documento pela letra G em azul.

Sugestões e Comentários podem ser enviados para [gtmbrgfox@gmail.com](mailto:gtmbrgfox@gmail.com)

**Modelo da Documentação**

Apresentamos um roteiro de documentação orientado ao desenvolvimento de um sistema web de gestão escolar utilizando notações de PHP, CSS, HTML e SQL, desde a fase inicial do projeto de gestão até a sua implantação. Para a criação dos diagramas aconselha-se a utilização de alguma ferramentas CASE (Astah Professional, BR Modelo, Workbench 6.3 CE). A documentação do código foi construída através do PHP Doc.

Roteiro:

1. Levantar dados das necessidades do usuário.
2. Construir documentação baseada em análise de dados
3. Apresentar proposta de construção do sistemas
4. Desenvolver o sistema
5. Testar o sistema
6. Corrigir o sistema
7. Lançar no mercado

**1. Introdução**

O mini sistema web de gestão escolar é um sistema multiplataforma que visa o gerenciamento dos processos escolares tais como: Alunos, Docentes, Salas de Aula, Disciplinas e Turmas. Este sistema está na versão 1.0 e não abrange todas as dimensões do sistema escolar vigente. Tem por objetivo atender as necessidade de gerenciamento escolar de uma escola regular.

**1.1. Tema**

Web Sistema de Gestão Escolar

Versão 1.0

**1.2. Objetivo do Projeto**

Objetivo geral: Atender a necessidade escolar no processamento e gerenciamento de dados

Objetivo específico: Gerenciar alunos, professores, salas de aula e turmas

**1.3. Delimitação do Problema**

O Web Sistema de Gestão Escolar v.1.0 ajuda no processamento e gerenciamento de dados dentro do ecossistema escolar. Neste sentido o Web Sistema de Gestão Escolar facilitará os processos educacionais dentro do ecossistema no que diz respeito ao tempo de planejamento, matrícula de discentes, organização do tempo dos docentes e da secretaria da escola. O Web Sistema de Gestão Escolar projeta índices e simplifica processos burocráticos, tornando o ecossistema mais dinâmico e interativo.

**1.4. Justificativa da Escolha do Tema**

Nossa escola para o desenvolvimento deste tema é fundamentada na necessidade de constante de implementação dos processos educacionais. Entendemos que que os estes necessitam de uma atenção contínua, portanto sistemas educacionais devem ser sempre implementados e reconfigurados para que possamos melhorar o ambiente escolar.

**1.5. Método de Trabalho**

O método a ser utilizado para realização do projeto foi o CRUD, o tipo de processo de desenvolvimento do web sistema segue padrões: **Finalidade** - reflete o que um padrão faz e **Escopo** - especifica se o padrão é aplicado à classe ou objeto. A modelagem a ser utilizada é totalmente Orientado a Objetos, seguindo os padrões de projeto e qualidade acima. Para acelerar a criação desse tipo de aplicação nós utilizamos o **Adianti framework**, ele traz uma série de componentes visuais prontos para a construção de formulários e datagrids. Dessa forma, nós podemos focar na lógica de negócio e não na construção da interface da aplicação.

**1.6. Organização do Trabalho**

Neste item deve-se descrever como o documento estará organizado.

**1.7. Glossário**

*Componente:* representa uma parte física da implementação de um sistema, que inclui código de software, com o objetivo de criar código de software coeso para sua reutilização e facilidade de manutenção.

*Ferramenta CASE (Computer Aided Software Engineering):* é uma ferramenta que auxilia no processo de desenvolvimento de software, ajudando a garantir a qualidade do projeto e facilitando a criação de modelos, documentos.

*Padrões de Projeto (design patterns):* são soluções simples para problemas específicos no projeto de software orientado a objetos. Padrões de projeto capturam soluções que foram desenvolvidas e aperfeiçoadas ao longo do tempo.

*Recursos alocados:* pessoas que irão trabalhar no projeto.

*Regras de negócio:* declarações e regras da política ou condição que deve ser satisfeita no âmbito do negócio.

*Requisito:* um requisito descreve uma condição ou capacidade à qual um sistema deve se adaptar, sejam necessidades dos usuários, um padrão ou uma especificação.

*Stored Procedures:* é uma rotina escrita através de comandos SQL, que tem como objetivo encapsular o processo de negócio e sua reutilização. As stored procedures ficam armazenadas no gerenciador de banco de dados.

*Usabilidade:* é a qualidade da interface homem-máquina, que permite que o usuário realize com eficiência e conforto as atividades a que o sistema se destina.

**2. Descrição Geral do Sistema**

Na tela de login o autor deverá logar com usuário e senha cadastrado no sistema, após o acesso autorizado o sistema direciona o ator para a tela inicial, se for servidor, a tela inicial deverá exibir dados como os números de alunos matriculados, docentes registrados, etc. O sistema usa um menu lateral com carregamento em jQuery para navegação entre as páginas das suas principais funções como cadastrar e listar salas e turmas, registrar e listar docentes, matricular e listar alunos.

**2.1. Descrição do Problema**

✓ Quem é afetado pelo sistema?  
 Todo o ecossistema escolar.

✓ Qual é o impacto do sistema?  
 Agilidade e rapidez nos processos.

✓ Qual seria uma boa solução para o problema?  
 Desenvolver e instalar um sistema multiplataforma orientado a objeto.

**2.2. Principais Envolvidos e suas Características**

**2.1.1. Usuários do Sistema**

Este sistema é destinado a instituições de ensino. Dentro dessas instituições temos tipos de usuários que utilizarão o sistema tais como: Alunos, Docentes e Corpo técnico.

**2.1.2. Desenvolvedores do Sistema**

Aqui temos 2 tipos de pessoas envolvidas em todo o desenvolvimento do sistema direta ou indiretamente.

1º temos a figura do analista - Ele se incubirá de fazer todo o levantamento de dados junto ao cliente e construirá o projeto de desenvolvimento.

2º temos a figura do desenvolvedor - Ele se incubirá de executar o projeto e desenvolverá o sistema baseado nele.

**2.3. Regras de Negócio**

As regras de negócio relevantes para o sistema web de gestão escolar possui algumas vantagens e desvantagens. As vantagens: o sistema web de gestão escolar pode ser facilmente vendido em qualquer lugar do Brasil e do mundo, ou seja, ele é extremamente escalável, de fácil manutenção e implementação, acessível e com interface amigável. As desvantagens: o sistema web de gestão escolar é um sistema multiplataforma online, ou seja, só funciona se tiver energia e internet e não possui versão off-line para desktop. Estima-se que o crescimento em adesão do sistema seja de 100% pois o seu custo-benefício é baixo, além de ser leve e não ocupa muito espaço na RAM.

**3. Requisitos do Sistema**

Logo abaixo descrevemos os requisitos do sistema web de gestão escolar. Neste caso o sistema web de gestão escola possui usuários / solicitantes reais para o levantamento de requisitos, utilizamos o modelo de documento de entrevista com usuários do RUP de Solicitações dos Principais Envolvidos.

**3.1. Requisitos Funcionais**

RF001 – Cadastrar e listar salas  
RF002 – Cadastrar e listar turmas  
RF003 – Registrar e listar docentes  
RF004 – Matricular e listar alunos  
RF005 – Cadastrar e listar disciplinas  
RF006 – Alterar e excluir salas  
RF007 – Alterar e excluir turmas  
RF008 – Alterar e excluir docentes

RF009 – Alterar e excluir alunos

RF010 – Alterar e excluir disciplinas

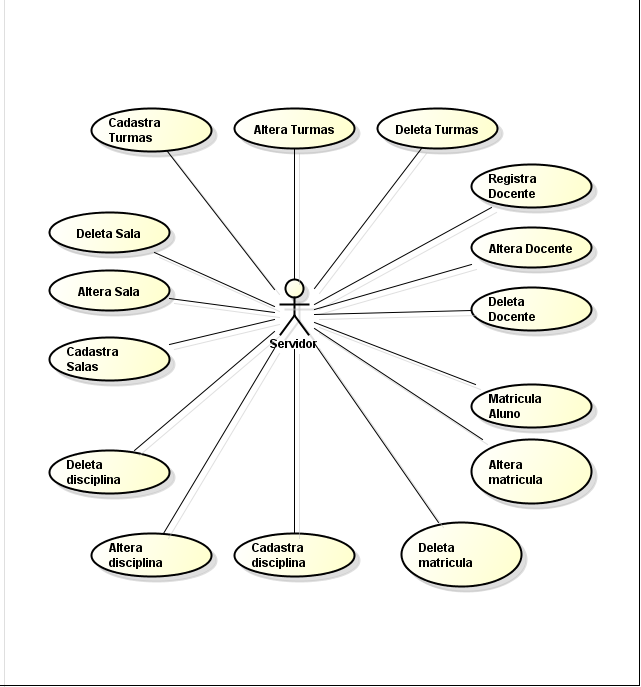


Diagrama de caso de uso do autor servidor

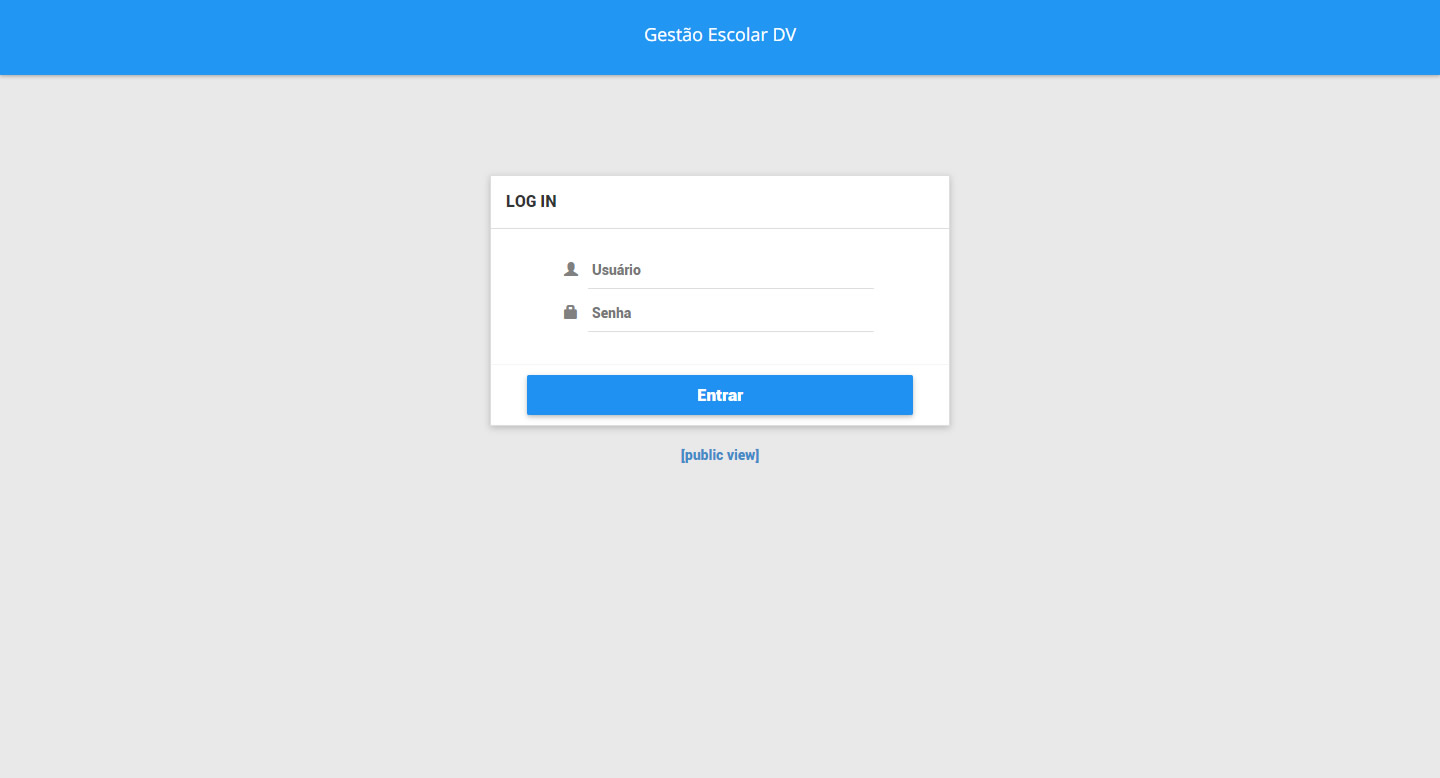
**3.2. Requisitos Não-Funcionais**

Desempenho ("Ao registrar um um docente ou matricular um aluno, os dados devem aparecer em, no máximo, 2 segundos").  
Disponibilidade ("O sistema estará disponível pelo menos 99,7% dos 7 dias das semanas durante 24 horas por dia").  
Flexibilidade ("O sistema deverá ser responsivo se adaptando a qualquer ecrã").  
Integridade/segurança ("Apenas usuários com privilégios de acesso de Servidor poderão fazer alterações no sistema").  
Confiabilidade ("Nenhuma informação do banco de dados pode ser perdido devido a falhas no sistema").  
Robustez ("Todas as variáveis de entrada terão valores null e tais valores serão usados sempre que dados de entrada estiverem faltando ou inválidos").  
Tolerância a falha ("O sistema deve fazer log dos dados cadastrados e/ou alterados via sistema").  
Usabilidade ("Um novo usuário com privilégios de administrador deverá ser capaz de cadastrar um novo usuário após não mais do que 30 minutos de orientação. Um novo usuário com privilégios de servidor terá a capacidade de cadastrar e fazer alterações nos dados de salas, turmas, docentes, alunos e disciplina.").  
Tipo de interface desejada ("O sistema deverá ser acessado completamente via browser HTTP/HTTS").  
Necessidades de internacionalização ("O produto será disponibilizado em inglês, mas de forma a permitir que versões em línguas latinas possam ser produzidas sem necessidade de ter acesso ao código fonte").  
Documentação necessária ("A documentação incluirá um Tutorial e um Manual de Referência").  
Preço da solução ("O produto deverá ser desenvolvido de forma a possibilitar um custo de produção de, no mínimo, R$0,00").  
Packaging ("O produto será hospedado exclusivamente pela pela empresa").

**3.3. Protótipo**

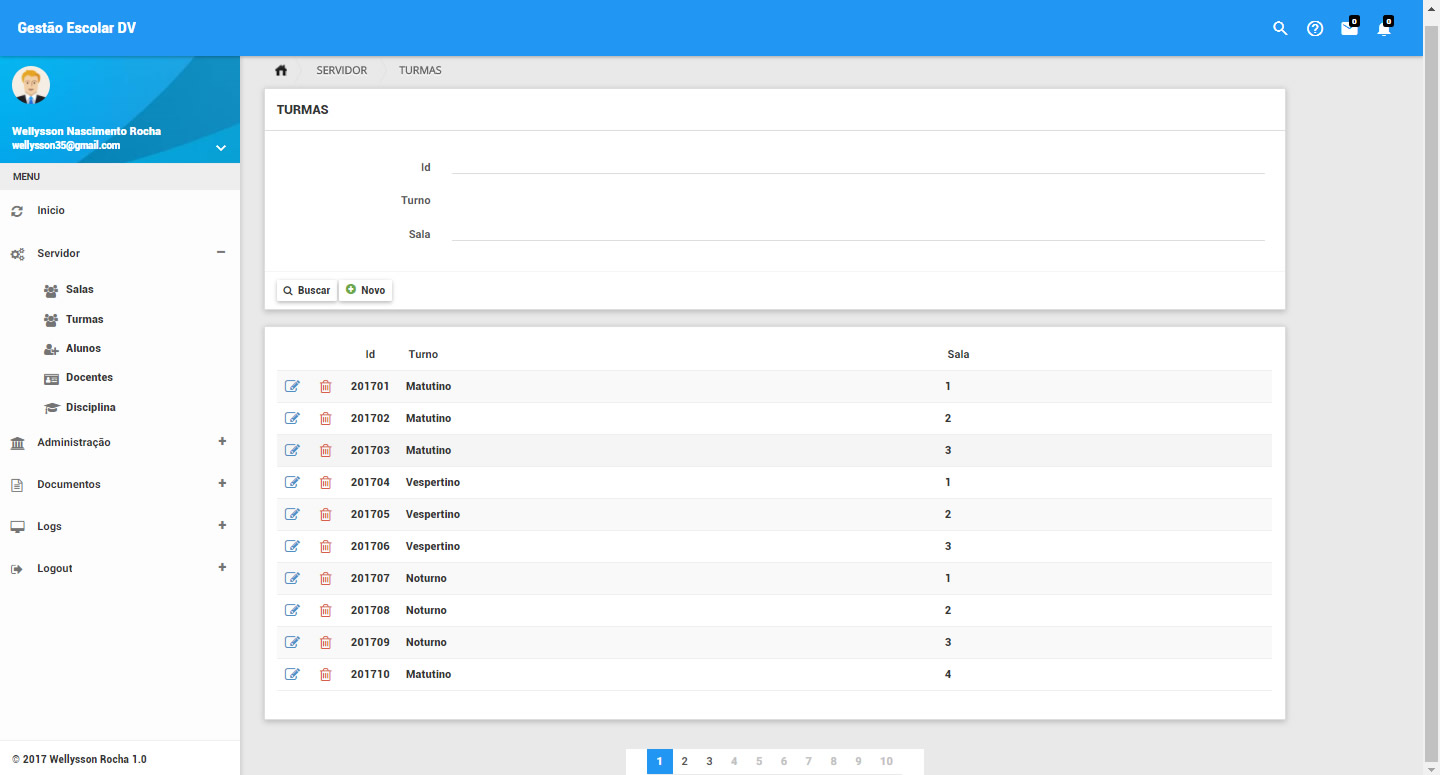
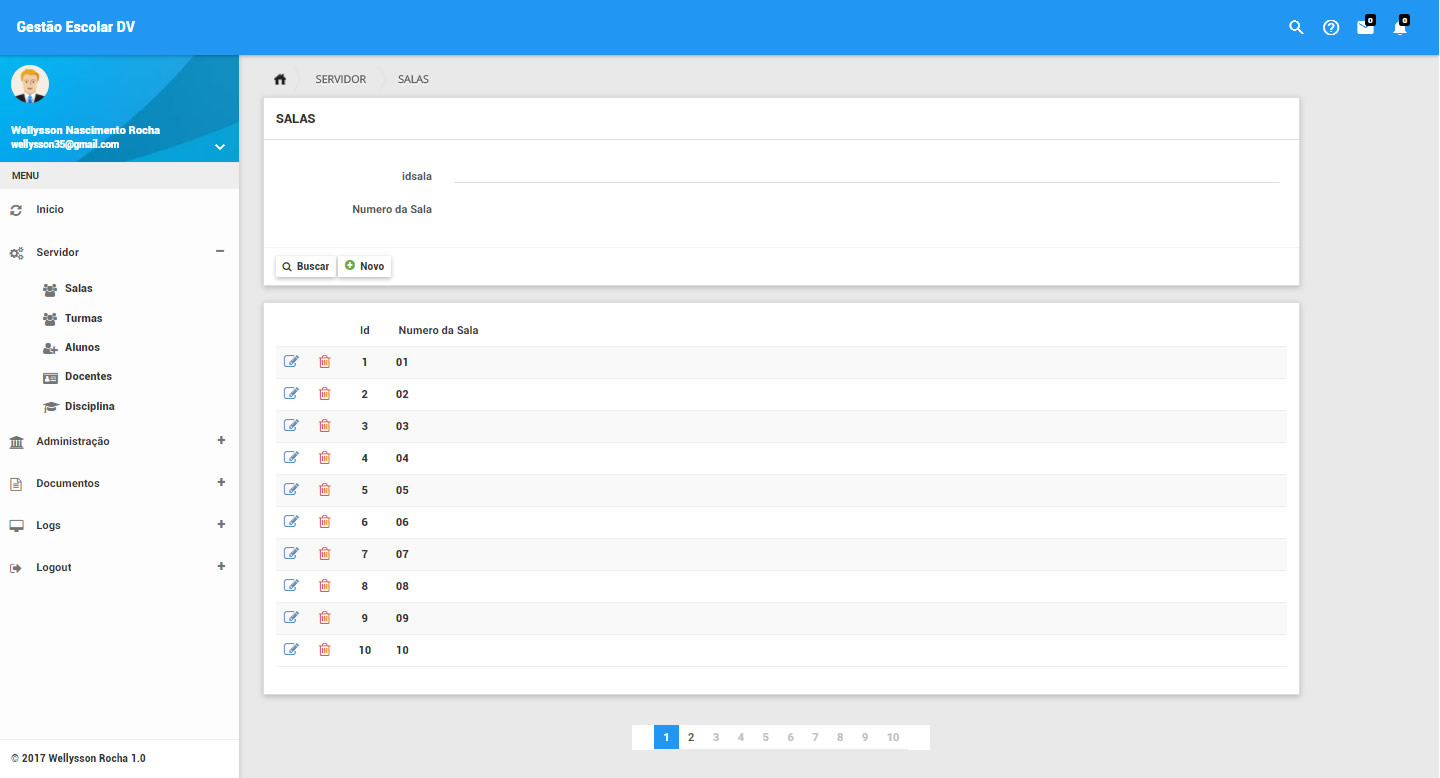
Os protótipos funcionais do sistema constituem a representação mais próxima do sistema a ser desenvolvido chamados de Protótipos de Alta Fidelidade. É possível simular o fluxo parcial das funcionalidades, permitindo a interação do usuário com a aparência visual, as formas de navegação e interatividade.

Seu desenvolvimento será realizado na fase final de desenvolvimento do sistema, pois utilizamos templates disponível da arquitetura open-source baseada em padrões de projeto, voltada para a criação ágil de aplicações de negócio em PHP, Adianti Framework.

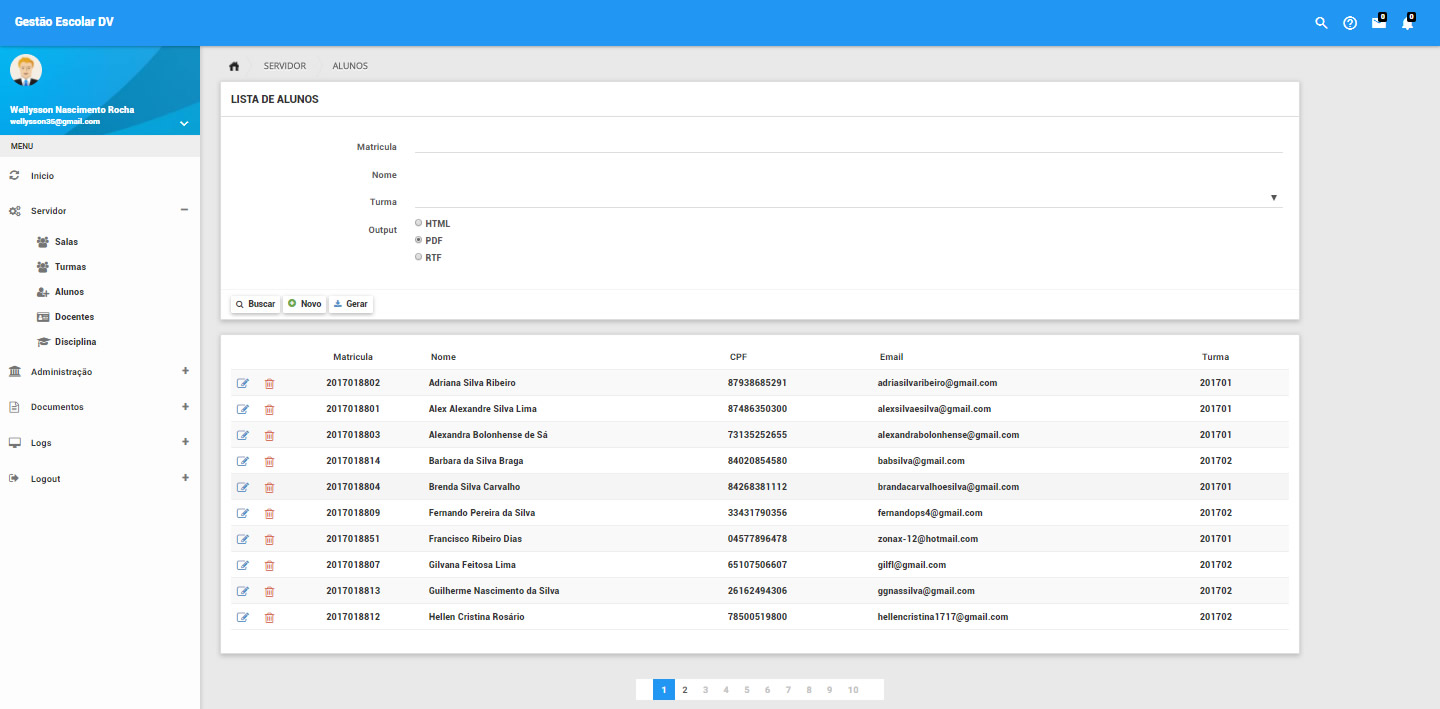
**3.3.1. Diagrama de Navegação  
Tela de Início - #01**

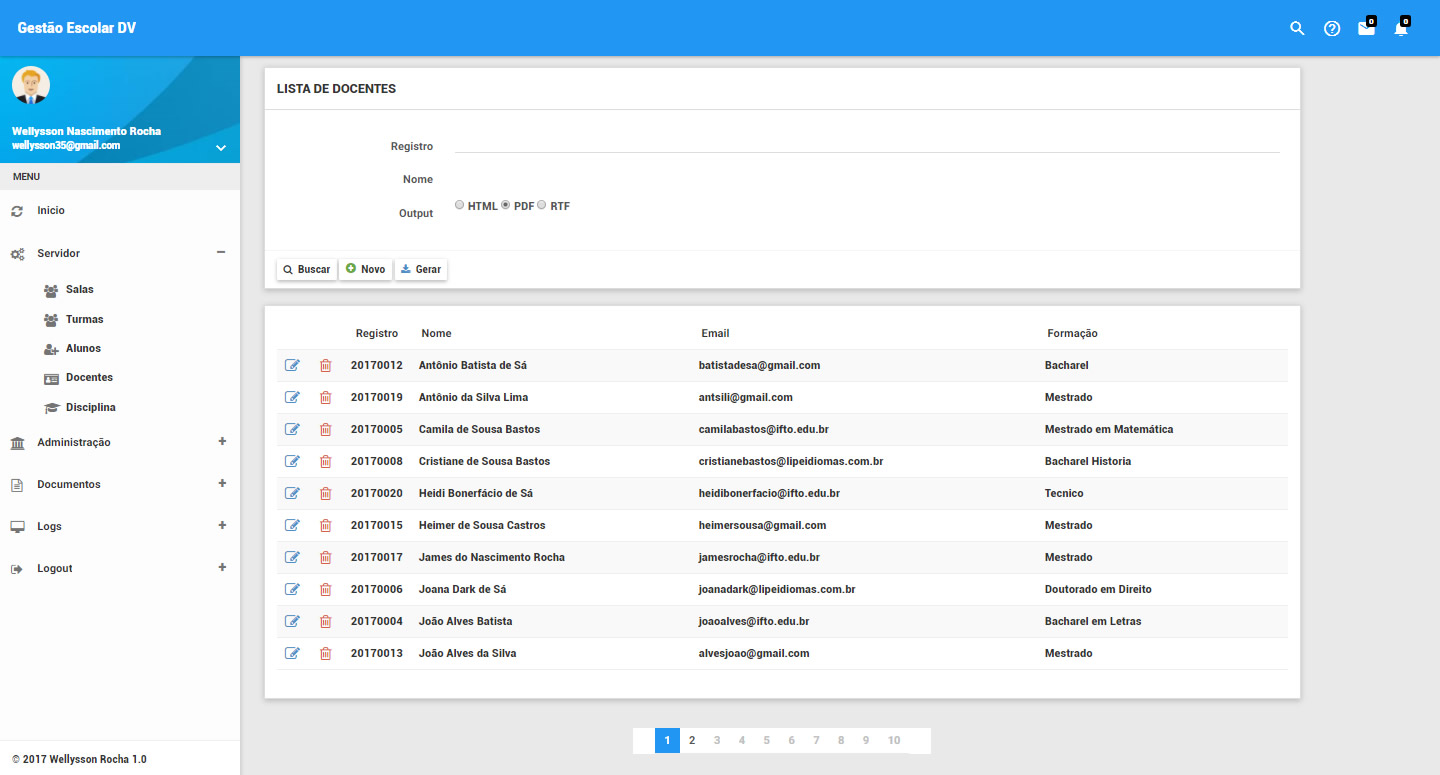
**Tela de Boas-vindas - #02**

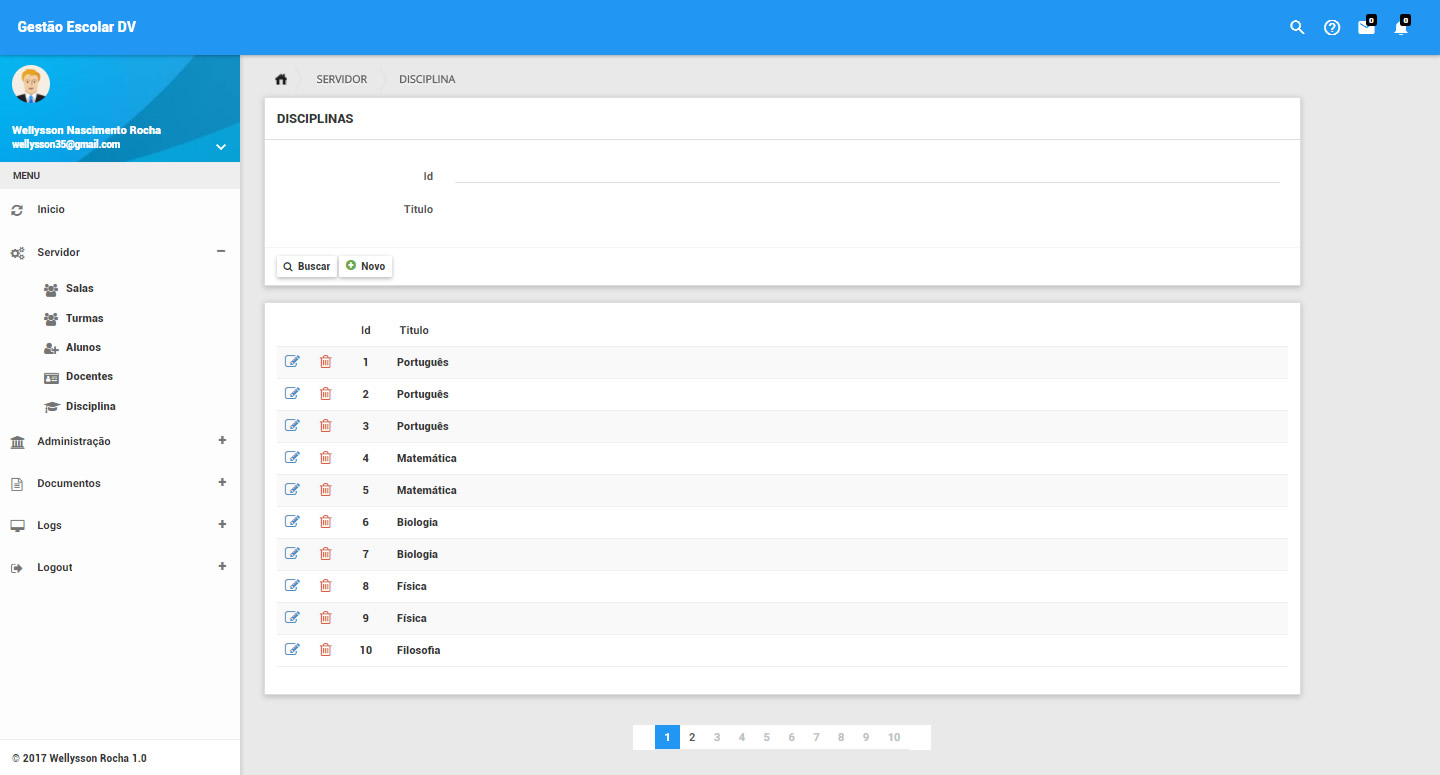


**Tela de Cadastro de Salas - #03  
  
Tela de Cadastro de Turmas - #04**

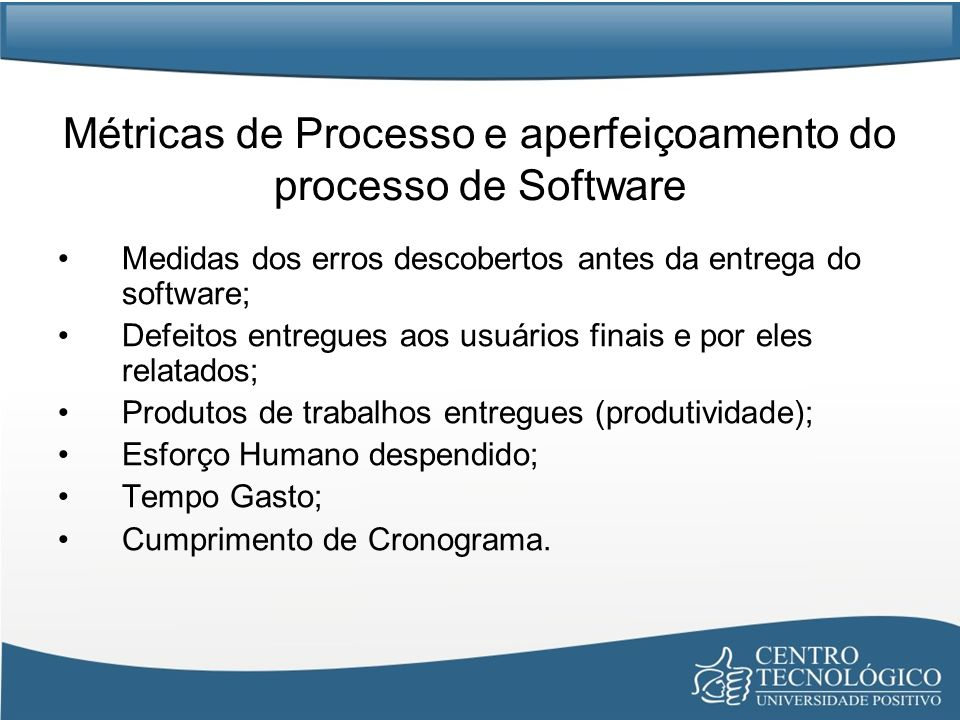
**Tela de Cadastro de Alunos - #05**

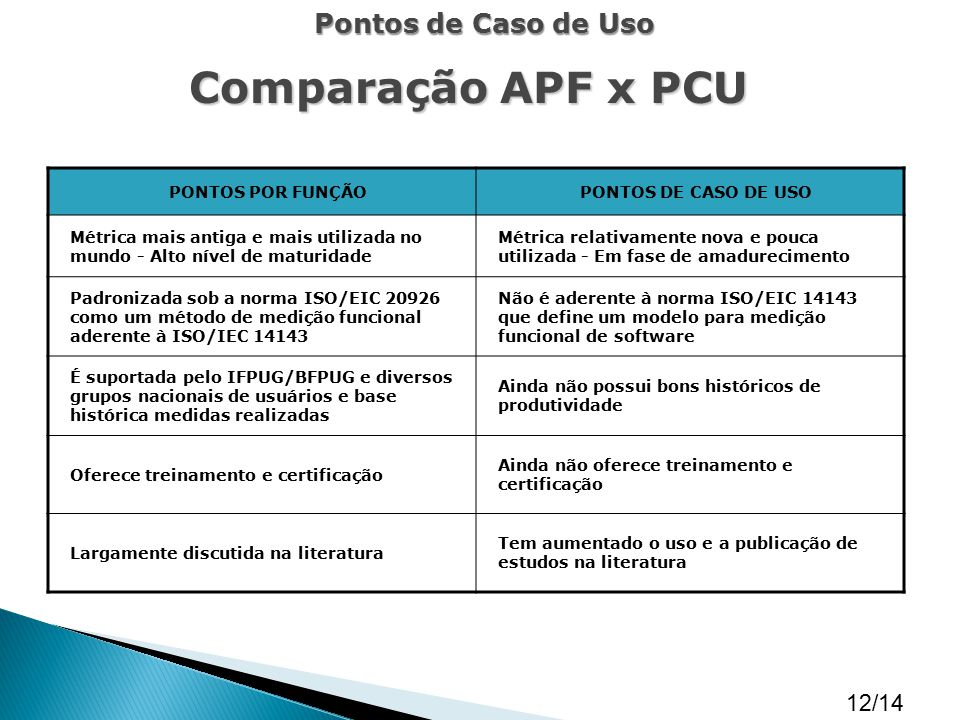
****

**Tela de Cadastro de Docentes - #06**

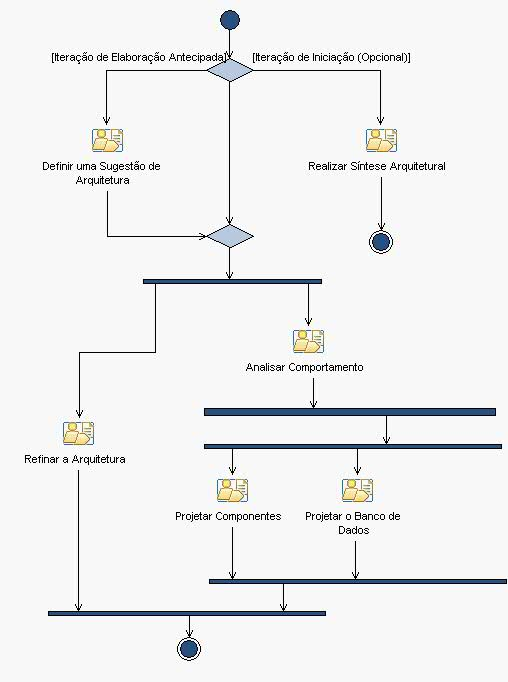
**Tela de Cadastro de disciplinas - #07**

**3.4. Métricas e Cronograma**



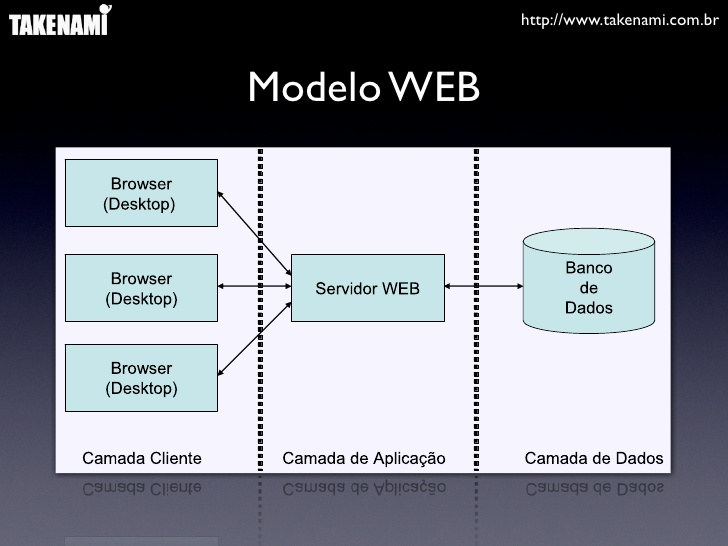


**4. Análise e Design**

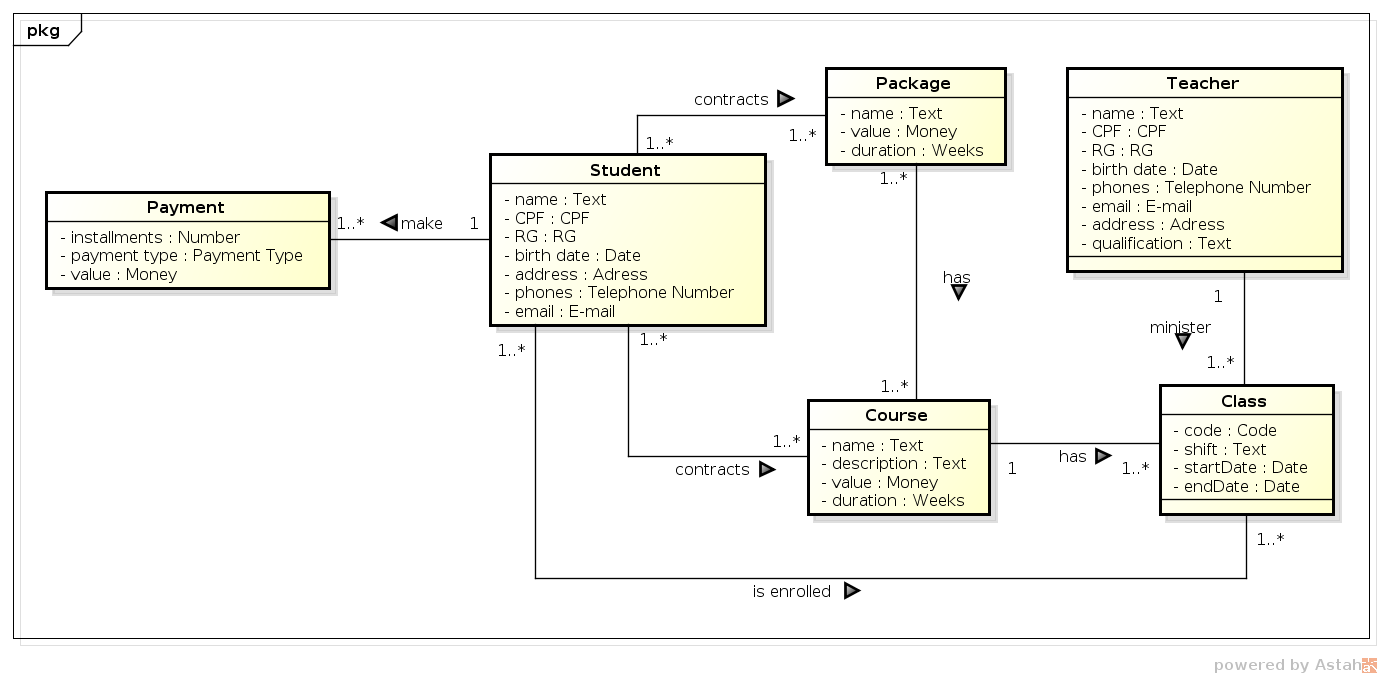


**4.1. Arquitetura do Sistema**

A análise arquitetural concentra-se em definir uma arquitetura sugerida e restringir as técnicas arquiteturais a serem utilizadas no sistema. Ela conta com a experiência obtida com sistemas ou domínios de problema semelhantes para restringir e enfocar a arquitetura, de modo que o esforço não seja desperdiçado na redescoberta arquitetural. A análise arquitetural é vantajosa principalmente ao desenvolver sistemas novos, nunca vistos antes; além disso, em sistemas nos quais já exista uma arquitetura bem-definida, ela poderá ser omitida.

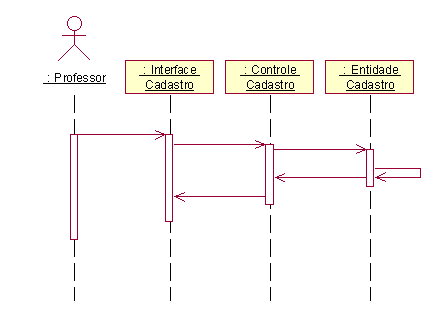


**4.2. Modelo do Domínio**



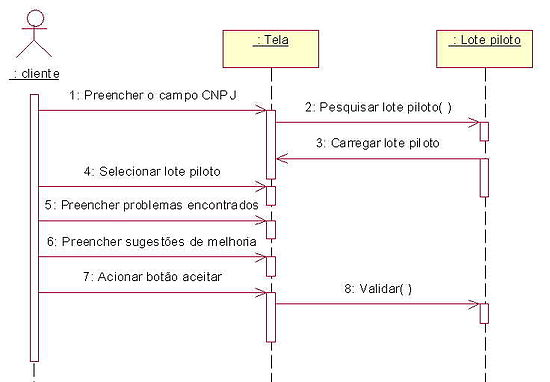
**4.3. Diagramas de Interação**

O diagrama de interação é composto pelos diagramas de seqüência e colaboração (comunicação, versão 2.0 UML) e modela os aspectos dinâmicos do sistema, mostrando a interação formada por um conjunto de objetos permitindo identificar mensagens que poderão ser enviadas entre eles.



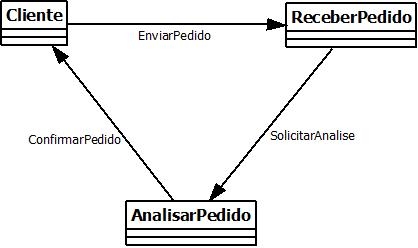
**4.3.1. Diagrama de Seqüência**

Um diagrama de seqüência representa interações de objetos organizadas em uma seqüência temporal, apresentando os objetos que participam da interação e a seqüência das mensagens trocadas. O diagrama de seqüência deve validar o diagrama de classes e vice-versa.

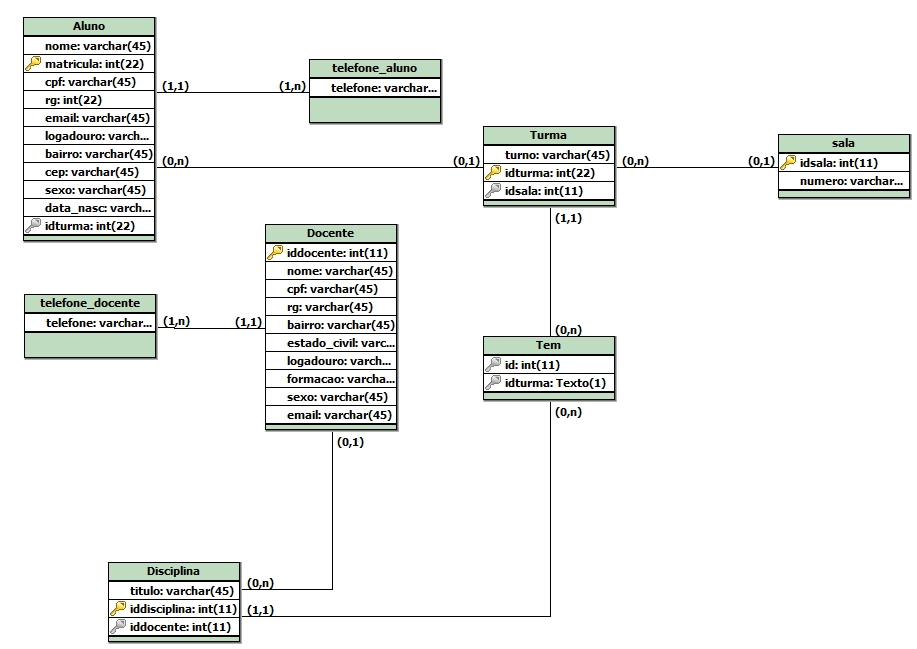
*.*

**4.3.2. Diagrama de Colaboração / Comunicação**

Um diagrama de colaboração descreve um padrão de interação entre objetos, apresentando os objetos que participam da interação bem como os seus links e mensagens trocadas.

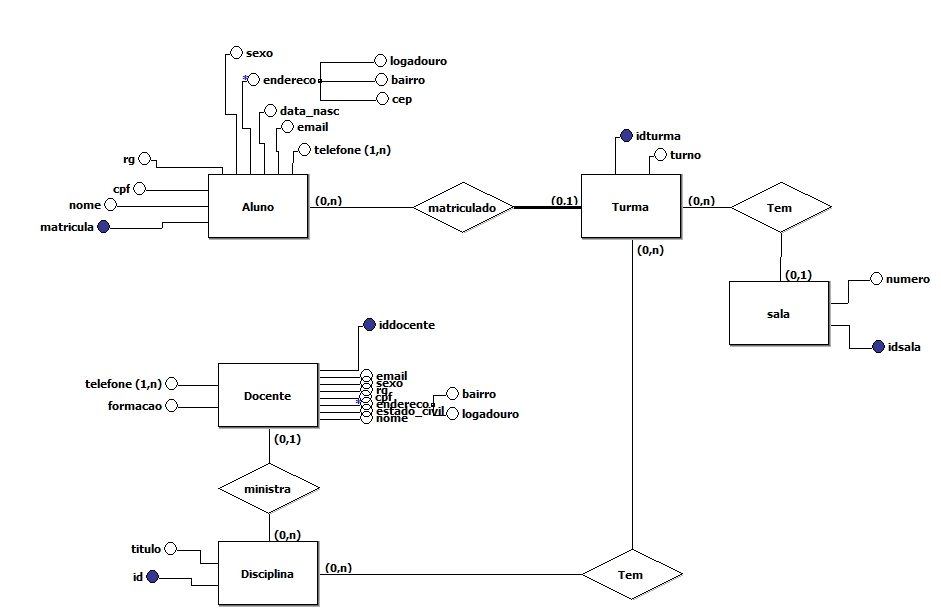


**4.4. Diagrama de Classes**



**4.5. Modelo de Dados**

**4.5.1. Modelo Lógico da Base de Dados**



**4.6.2. Criação Física do Modelo de Dados**

-- MySQL Workbench Synchronization  
-- Generated: 2017-12-16 09:25  
-- Model: Novo Modelo  
-- Version: 1.0  
-- Project: Projeto de mini-gestão Escolar  
-- Author: Wellysson Nascimento Rocha  
  
SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;  
SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;  
SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';  
  
ALTER TABLE `g\_e`.`Aluno`   
ADD INDEX `fk\_Aluno\_turma1\_idx` (`turma\_idturma` ASC),  
DROP INDEX `fk\_Aluno\_turma1\_idx` ;  
  
ALTER TABLE `g\_e`.`telefone\_aluno`   
ADD INDEX `fk\_telefone\_aluno\_Aluno\_idx` (`Aluno\_matricula` ASC),  
DROP INDEX `fk\_telefone\_aluno\_Aluno\_idx` ;  
  
ALTER TABLE `g\_e`.`turma`   
ADD INDEX `fk\_turma\_sala1\_idx` (`sala\_idsala` ASC),  
DROP INDEX `fk\_turma\_sala1\_idx` ;  
  
ALTER TABLE `g\_e`.`disciplina`   
ADD INDEX `fk\_disciplina\_docente1\_idx` (`docente\_iddocente` ASC),  
DROP INDEX `fk\_disciplina\_docente1\_idx` ;  
  
ALTER TABLE `g\_e`.`turma\_has\_disciplina`   
ADD INDEX `fk\_turma\_has\_disciplina\_disciplina1\_idx` (`disciplina\_iddisciplina` ASC),  
ADD INDEX `fk\_turma\_has\_disciplina\_turma1\_idx` (`turma\_idturma` ASC),  
DROP INDEX `fk\_turma\_has\_disciplina\_turma1\_idx` ,  
DROP INDEX `fk\_turma\_has\_disciplina\_disciplina1\_idx` ;  
  
ALTER TABLE `g\_e`.`telefone\_docente`   
ADD INDEX `fk\_telefone\_docente\_docente1\_idx` (`docente\_iddocente` ASC),  
DROP INDEX `fk\_telefone\_docente\_docente1\_idx` ;  
  
  
SET SQL\_MODE=@OLD\_SQL\_MODE;  
SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;  
SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**4.7. Ambiente de Desenvolvimento**

Apresentamos aqui os softwares de desenvolvimento:

* Netbeans
* Workbench
* Br modelo

Linguagem de programação:

* PHP
* HTML
* CSS
* UML
* SQL

Banco de dados:

* SQL server

Ferramentas:

* internet

Equipamentos de hardware:

* Computador
* tablet
* smartphone

Redes que sejam essenciais para o desenvolvimento do sistema:

* Wi-fi

**5. Implementação**

A etapa final na geração de código pelo compilador é a fase de otimização. Como o código gerado através da tradução orientada a sintaxe contempla expressões independentes, diversas situações contendo seqüências de código ineficiente podem ocorrer. O objeto da etapa de otimização de código é aplicar um conjunto de heurísticas para detectar tais seqüências e substituí-las por outras que removam as situações de ineficiência.

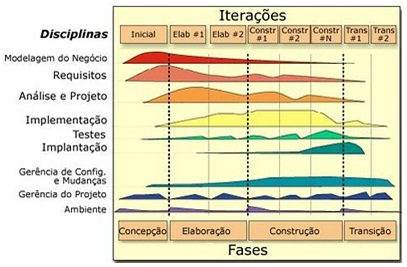
Algumas heurísticas de otimização são sempre aplicadas pelos compiladores. Por exemplo, se a concatenação de código gerado por duas expressões no programa original gerou uma situação de desvio incondicional para a linha seguinte, como em

<a>  
 goto \_L1  
\_L1: <b>

esse código pode ser seguramente reduzido com a aplicação da técnica de **eliminação de desvios desnecessários**, resultando em

<a>  
\_L1: <b>

Outra estratégia de otimização elimina os rótulos não referenciados por outras instruções do programa. Assim, se o rótulo \_L1 estivesse sendo referenciado exclusivamente por essa instrução de desvio, ele poderia ser eliminado em uma próxima aplicação das estratégias de otimização.

**6. Testes**

**6.1. Plano de Testes**

Fases:

* Concepção
* Elaboração
* Construção
* Transição

Disciplinas:

* Modelagem de Negócio
* Requisitos
* Análise e Projeto
* Implementação
* **Testes**
* Implantação
* Gerência de Configuração e Mudanças
* Gerência do Projeto
* Ambiente

**6.2. Execução do Plano de Testes**

O registro dos testes deve conter a identificação do sistema, o nome do realizador dos testes e a configuração do ambiente onde foi realizado o teste.

* Verificar a interação entre os objetos componentes do sistema;
* Verificar a integração correta de todos os componentes do sistema;
* Verificar se todos os requisitos foram implementados de maneira correta;
* Detectar o maior número possível de defeitos antes da fase de Implantação;
* Retestar todas as correções de defeitos e garantir que outras partes do sistema não foram afetadas por tais correções.

**7. Implantação**

Este capítulo tem como objetivo apresentar informações relevantes para a implantação e funcionamento do sistema.

**7.1. Diagrama de Implantação**

Neste item deve ser apresentado o diagrama de implantação que representa a parte física do sistema, exibindo os dispositivos, as máquinas de processamento em tempo de execução e os componentes que nelas serão instalados.

A seguir é apresentada a notação básica de um diagrama de implantação.

Processador

Processador

Processador

Dispositivo

*Notação básica do diagrama de implantação.*

**7.2. Manual de Implantação**

Este manual contém a descrição passo a passo de como deve ser realizada a instalação do sistema.

Para maiores detalhes pode-se consultar [aqui](http://www.uniondata.com.br/ajuda/manual_condominio.pdf).

**8. Manual do Usuário**

Este manual contém a descrição passo a passo de como utilizar o sistema.

Para maiores detalhes pode-se consultar [aqui](https://softwarepublico.gov.br/social/articles/0000/5463/manual_do_usuario_i_educar_Portabilis.pdf).

**9. Conclusões e Considerações Finais**

Neste projeto apresentamos o sistema web de gestão escolar versão 1.0. Este sistema é multiplataforma e portanto completamente online.

Usamos várias ferramentas para construí-lo, dentre elas o Netbeans, Workbench, DBDesigner 4, BR Modelo assim como frameworks como Adiante. Também usamos algumas das linguagens de programação mais usadas no mercado atualmente, como: PHP, HTML 5, SQL etc, tudo orientado a objeto.

Este projeto tem por objetivo maior a solução de problemas de gestão escolar e neste sentido aplica-se a toda e qualquer instituição de ensino que sofre com problemas de gestão escolar como, ineficácia, lentidão nos processos e organização de informação.

Não queremos ser pretensiosos ao ponto de dar como finalizado nosso projeto aqui, na verdade essa só e a versão 01 desse trabalho. Ainda há muito pela frente para chegarmos a um sistema completo. Desta forma pretendemos integrar vários outros módulos neste primeiro. A estrutura deste módulo é baseada nos sistemas de informação de informação de gestão - [SIG](https://pt.wikipedia.org/wiki/Sistema_de_informa%C3%A7%C3%A3o_de_gest%C3%A3o).

**Bibliografia**

FOWLER, M.; SCOTT.K. UML Essencial. Editora Bookman, 3a edição.

BOOCH, G.; RUMBAUGH, J.; JACOBSON. I. UML œ Guia do Usuário. Editora

Campus, 2000.

BOGGS, W.; BOGGS, M. Mastering UML com Rational Rose 2002. Editora Alta Books,

2002.

QUATRANI. T. Modelagem Visual com Rational Rose 2000 e UML. Editora Ciência

Moderna, 2001.

PRESSMAN, R. Software Engineering. A Practitioner ́s Approach. 5a edição, 2003.

McGrawHill.

SOMERVILLE, I. Engenharia de Software. Addison Wesley, 6a edição.

YOURDON, E Análise Estruturada Moderna. Editora Campus, 3a edição.

*Ferramenta Rational Unified Process.*

Site: http://prof.usjt.br/anapaula (disciplinas de Metodologia de Desenvolvimento de

Software e Engenharia de Software)