



MALLU EDUARDA BATISTA

**MÓDULO WEB PARA O DIÁRIO DE CLASSE:
UMA IMPLEMENTAÇÃO VOLTADA ÀS NECESSIDADES DO
PROFESSOR**

LAVRAS – MG

MALLU EDUARDA BATISTA

MÓDULO WEB PARA O DIÁRIO DE CLASSE:
UMA IMPLEMENTAÇÃO VOLTADA ÀS NECESSIDADES DO PROFESSOR

Relatório Técnico apresentada à Universidade Federal de Lavras, como parte das exigências para a obtenção do título de Bacharel em Ciência da Computação.

Prof. Dra. Ana Paula Piovesan Melchiori
Orientadora

LAVRAS – MG

MALLU EDUARDA BATISTA

**MÓDULO WEB PARA O DIÁRIO DE CLASSE: UMA IMPLEMENTAÇÃO VOLTADA
ÀS NECESSIDADES DO PROFESSOR**

Relatório Técnico apresentada à Universidade
Federal de Lavras, como parte das exigências
para a obtenção do título de Bacharel em
Ciência da Computação.

APROVADA em 31 de janeiro de 2018.

Profa. Dra. Renata Teles Moreira	UFLA
Prof. Dr. José Monserrat Neto	UFLA
Karla Emanuella Veloso Pinto	Prefeitura Municipal de Lavras

Prof. Dra. Ana Paula Piovesan Melchiori
Orientadora

LAVRAS – MG

Para toda minha família e amigos. Em especial, para minha mãe Elizabete.

AGRADECIMENTOS

A Deus por me guiar e proteger por todos os caminhos.

Aos meus pais, pela educação, exemplo, apoio e força na realização desse sonho. Sem vocês nada disso seria possível ou teria sentido.

A toda minha família, pela de união e parceria. Meus irmãos Marcello e Marcos Paulo, por mais que me esgotem, transmitem muito amor. Minha tia Eliete, pelos anos de companheirismo, exemplo admirável e principal incentivadora de buscar e ir atrás dos meus objetivos. Minhas tias Elaine e Eliz que sempre me apoiaram em tudo. Minha vovó Maria Néria por todo carinho e amor.

Aos meus amigos e amigas: Ana Laura, Arthur, Danielle, Débora, Luana, Maria Fernanda, Nádia, Sanndy, Tatiane, pelo apoio nas mais diversas situações, presença nos melhores momentos, ombro nos piores e companheirismo sempre. Mas, principalmente por nunca desistirem de mim, apesar de tudo.

Aos meus queridos e incríveis amigos que ganhei durante faculdade; Elder, Gabriel e Matheus (White), pela atenção, longas horas de paciência e aprendizado, amizade e companheirismo durante essa jornada. Vocês foram essenciais na minha formação acadêmica e pessoal.

As minhas amigas Giovana, Carol e Laura, por me acolherem em casa nessa reta final, proporcionarem muitas alegrias, risadas, concelhos, broncas e por se tornarem indispensáveis no meu dia.

A professora Dra. Ana Paula pelos anos de acolhida, orientação, aprendizado, companheirismo e amizade.

A Universidade Federal de Lavras, ao departamento de Ciência da Computação e seu corpo docente, pelo suporte e aprendizado durante esses anos.

Se as nossas vontades não coincidirem, que a Sua prevaleça.

RESUMO

O diário de classe é um documento oficial que faz parte do processo de gestão escolar, e é de inteira responsabilidade do professor. O desenvolvimento de sistemas Web tem auxiliado na melhoria desse processo, contribuindo para a dinamização das tarefas, eficiência e modernização. Este trabalho teve como objetivo a criação e implementação de um módulo WEB para o diário de classe, visando ser um ambiente onde os professores cadastrem e administrem diários e suas respectivas informações, em um único lugar. O desenvolvimento foi feito usando a linguagem de programação PHP, padrão de arquitetura MVC, framework *Bootstrap* e as tecnologias HTML5, CSS3, JavaScript e JQuery. Como resultado, é disponibilizado a aplicação web para uso de professores, no âmbito de suas atribuições, agregando segurança, agilidade de armazenamento, e acesso á informação.

Palavras-chave: Diário de Classe. Professor. Módulo WEB. Gestão escolar.

ABSTRACT

The class diary is an official document that is part of the school management process, and is the teacher's entire responsibility. The development of Web systems has helped in the improvement of this process, contributing to the dynamization of tasks, efficiency and modernization. The development was done using a PHP programming language, MVC architecture standard, framework Bootstrap and as HTML5, CSS3, JavaScript and JQuery technologies. As a result, a web application is available for use by teachers, not its integrated assignments, storage agility and access to information.

Keywords: Class diary. Teacher. WEB module. School management.

LISTA DE FIGURAS

Figura 4.1 – Diagrama de caso de uso	21
Figura 4.2 – Modelo relacional do banco de dados	23
Figura 4.3 – Divisão dos diretórios - controller e model	24
Figura 4.4 – Divisão dos diretórios - view	24
Figura 4.5 – Arquivo "index.php"	25
Figura 4.6 – Arquivo "cadastroTela.php"	25
Figura 4.7 – Arquivo "assistenteInicialPT1.php"	25
Figura 4.8 – Arquivo "assistenteInicialPT2.php"	26
Figura 4.9 – Arquivo "assistenteInicialPT3.php"	26
Figura 4.10 – Arquivo "professorInterface.php"	26
Figura 4.11 – Arquivo "professorInterface.php- continuação da figura 4.10	27
Figura 4.12 – Classe "ProfessorBean.class.php"	27
Figura 4.13 – Classe "ProfessorDao.php"	28
Figura 5.1 – Tela inicial da aplicação	29
Figura 5.2 – Tela de cadastro de usuário	29
Figura 5.3 – Assistente inicial: parte 1	30
Figura 5.4 – Assistente inicial: parte 2	30
Figura 5.5 – Assistente inicial: parte 3	30
Figura 5.6 – Tela de login	31
Figura 5.7 – Pagina inicial do professor	31
Figura 5.8 – Controlador de ações	32
Figura 5.9 – Cadastrar Diário: parte 1	32
Figura 5.10 – Cadastrar Diário: parte 2	33
Figura 5.11 – Editar Diário: parte 1	33
Figura 5.12 – Editar Diário: parte 2	33
Figura 5.13 – Cadastro de Aluno	34
Figura 5.14 – Visualização dos Alunos	34
Figura 5.15 – Visualização do boletim do Aluno	35
Figura 5.16 – Cadastro de Aula	35
Figura 5.17 – Visualização das Aulas	36
Figura 5.18 – Listar presença dos Alunos na Aula	36

Figura 5.19 – Cadastro de Avaliacao	36
Figura 5.20 – Visualização das Avaliações	37
Figura 5.21 – Lançar nota de Avaliação	37

LISTA DE TABELAS

Tabela 4.1 – Requisitos funcionais	20
--	----

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.1.1	Objetivo geral	12
1.1.2	Objetivos Específicos	12
2	REFERENCIAL TEÓRICO	13
3	MATERIAL E MÉTODOS	17
3.1	Ferramentas utilizadas	17
3.1.1	Interface gráfica	17
3.1.2	Base de dados	17
3.1.3	Desenvolvimento das funcionalidades	18
3.1.4	Ferramentas auxiliares	18
4	DESENVOLVIMENTO DO SISTEMA	20
4.1	Especificação de requisitos	20
4.2	Projeto	20
4.2.1	Estruturação do banco de dados	21
4.3	Implementação	22
4.4	Testes	25
5	RESULTADOS E DISCUSSÃO	29
5.1	Módulo Professor	29
5.2	Componentes	30
5.2.1	Diário	31
5.2.2	Aluno	34
5.2.3	Aula	35
5.2.4	Avaliação	35
6	CONCLUSÕES	38
	REFERÊNCIAS	39

1 INTRODUÇÃO

A constante busca pela facilidade de acesso à informação, tem motivado cada vez mais o desenvolvimento de sistemas Web. Estes por sua vez, diferentemente do modelo tradicional, não precisam ser instalados e são acessíveis de qualquer lugar, por qualquer dispositivo conectado à internet. Atualmente, esse tipo de tecnologia é voltada para as mais diversas áreas e é empregada com várias finalidades, inclusive na educação.

No campo educacional, a gestão escolar retoma os conceitos de eficiência e melhoria no sistema de ensino, e é por meio dela que se busca organizar as áreas da escola. Nesse contexto, a adoção de sistemas para auxiliar a gestão escolar, automatizando o processo, traz benefícios como o aumento na produtividade dos funcionários, dados concretos e facilidade de acesso à informação.

Muitas escolas da atual rede de ensino brasileira, ainda adotam a utilização do Diário de papel pelos professores. Esse, é o documento oficial no qual são registradas todas as atividades pedagógicas abordadas, bem como o histórico de presença e notas de cada aluno, e é posteriormente repassado à secretaria ao final de cada período letivo, através da entrega da taleta. Tal processo é manual, lento e passível de erros, o que caracteriza a necessidade da adoção de um sistema mais eficiente e seguro a fim de melhorar o processo de gestão dessas informações.

Tomando como ponto de partida o estudo "Diários de Classe: traços históricos de um ensino de língua" de Menegolo e Cardoso, que analisou a aplicação real desse material, concluiu-se que as anotações realizadas pelos professores não condizem com as aulas lecionadas, mas sim com uma cultura escolar predeterminada que visa suprir as cobranças de um sistema de ensino frágil. Tal omissão corrobora para dificultar um diagnóstico preciso e correto sobre as falhas educacionais e as áreas que carecem de mudança imediata. Nesse sentido, a utilização de ferramentas tecnológicas, como sistemas Web, tem sido modelados para auxiliar na busca de uma solução prática e segura na tentativa de amenizar tais problemas.

A busca por ferramentas para auxiliar no problema abordado, levou a criação de projetos como o Educação Conectada¹, que desenvolveu um aplicativo móvel para dispositivos Android² que desempenha a mesma função do diário de papel supracitado. Entretanto foram identificados problemas na interface da aplicação, quanto as funcionalidades, e foram tratadas para melhor atender as necessidades do usuário final segundo padrões e técnicas de *design*

¹ <http://www.educacaoconectada.com.br/>

² <https://www.android.com/>

centrado no usuário, produzindo como resultado final, um novo protótipo de interface (LACERDINO; MELCHIORI, 2016).

1.1 Objetivos

1.1.1 Objetivo geral

Este trabalho tem como objetivo geral desenvolver um sistema Web para o perfil do professor, que funcione como um Diário de Classe virtual, substituindo o Diário de papel. A interface da aplicação será baseada nos resultados obtidos por Lacerdino e Melchiori (2016).

1.1.2 Objetivos Específicos

Os objetivos específicos norteiam:

- Estudo das ferramentas necessárias para o desenvolvimento da aplicação;
- Especificação de requisitos: levantamento de requisitos, diagrama de casos de uso e protótipo de interface;
- Projeto do software: definição dos módulos e componentes, criação do banco de dados e padrão de arquitetura MVC;
- Implementação: codificação;
- Teste: dinâmico, com validação das funcionalidades através da interface (teste de caixa branca) durante a implementação;

2 REFERENCIAL TEÓRICO

No desenvolvimento de *softwares* Web, a principal busca é a satisfação das necessidades dos usuários. A alta demanda por esse tipo de serviço é justificada pelo avanço das tecnologias de gestão da informação, e as facilidades que a utilização da internet trouxe em relação ao acesso e disponibilização de informação, praticidade e segurança (BARBOSA; SEPÚLVEDA; COSTA, 2009). Assim, o emprego dessa ferramenta está nas mais diversas áreas e com amplas propostas de utilização .

As facilidades alcançada com a utilização de *softwares* Web, vão além diversidade de ferramentas tecnológica que conseguem ter acesso a elas. Blatmann, Fachin e Varkavis (2000) ressalta que "O valor dos hiperdocumentos não está nos custos de manutenção mas na forma do usuário aprender mais rápido, localizar a informação mais rapidamente, ou acessar o conhecimento armazenado mais efetivamente.". Isso permite que qualquer tipo de usuário se beneficie de tal tecnologia.

Bax e Parreiras (2003) enfatizam os aspectos da descentralização da informação e os toma como justificativa para o desenvolvimento de sistemas de gestão de conteúdos na Web. Em um sistema onde o funcionamento gira em torno de vários módulos, melhorias são alcançada com a divisão das responsabilidades e a facilidade de acesso à elas, de acordo com sua função na hierarquia.

Atualmente, muito se discute acerca das atuais práticas de gestão da educação bem como metodologias que podem ser aplicadas e/ou inseridas no ambiente, para dinamizar os processos e promover a eficiência e democratização dessa prática. As principais problematizações norteiam a realidade educacional brasileira e a formação de profissionais (PARO, 2007).

Pensando em formas de intervenção no mundo da educação, o processo de gestão das organizações escolares requer etapas de profundas mudanças (BRITO, 2006). Essas, permeiam profissionais, alunos e toda prática pedagógica adotada dentro do processo de adaptação às mudanças, bem como a escolha das ferramentas e sua forma de utilização para obtenção de retornos positivos.

Devido à modernização tecnológica as organizações tem buscado alcançar processos de mudanças e inovações, visando qualidade e competitividade. Segundo Fonseca (2001), para alcançar esses objetivos as empresas devem se preocupar com o envolvimento de seus colaboradores no processo produtivo/serviço, e conseqüentemente a qualidade de vida dos mesmos, o que envolve condições de trabalho, novas formas de gestão, saúde física e mental, entre ou-

tras. No entanto, ao analisar o sistema educacional, percebemos que o método de trabalho do professor ainda é estagnado, dificultando a efetividade e organização das funções cabíveis a eles.

Associar o uso de sistemas Web à gestão das práticas pedagógicas é um desafio a ser enfrentado: de um lado estão os sistemas desenvolvidos para facilitar as tarefas diárias e moderniza-las ao mesmo tempo; do outro os professores, que alegam carecer de conhecimento frente a essas tecnologias (BRITO, 2013). Logo, um dos meios de solucionar o problema é aderir à substituição de ferramentas ultrapassadas utilizadas no processo educacional, a fim de facilitar o trabalho desses profissionais.

Essas ferramentas de auxílio, devem agir de forma a complementar o ensino tradicional, e jamais substituir a instituição principal; a escola. O professor por sua vez, deve incrementar em seu currículo e, simultaneamente em suas aulas, uma formação ampla para permear a difusão de conhecimento através da utilização dessas tecnologias (ALBINO; SOUZA, 2016). Sendo assim, tendem a complementar e modificar a prática pedagógica afim de surtirem facilidades e modernização de ferramentas e métodos. Entretanto, é importante ressaltar que o professor é ferramenta primordial nesse processo e carece de ferramentas de gestão que facilite tal metodologia.

Como exemplo de parte das atribuições de responsabilidade do professor, temos o preenchimento do Diário de Classe. Descrito segundo o Manual de Orientação de Zaponi, Monteiro e Torres (2009) é “um instrumento de registro do planejamento e do desenvolvimento das atividades pedagógicas do(a) professor(a), instrumento legal de registro das situações didáticas da vida escolar dos(as) estudantes, do acompanhamento das suas aprendizagens e do desempenho escolar. No diário de classe devem constar: a relação nominal dos(as) estudantes, em ordem alfabética, observações sobre o rendimento, frequência justificada e atitudes comportamentais; o planejamento das aulas, o registro dos conteúdos trabalhados em situação didática de cada bimestre e as atividades ou projetos especiais.”.

O diário é disponibilizado em forma de livro, onde o preenchimento deve ser feito manualmente e sem rasuras, sendo elas rubricadas e justificadas quando acontecido (TERESA, 1999). Como consequência, tal processo requer maior atenção, é passível de erros, repetitivo e cansativo. Nesse contexto, aplicações Web de boa qualidade podem ajudar e muito

Igor e Melchiori (2016) discorrem sobre o impacto da substituição de antigos métodos de gestão, enfatizando também acerca da problematização que tange sua utilização. Ao fazer a

análise de uma aplicação de gestão educacional desenvolvida no projeto Educação Conectada para substituição do Diário de Classe de papel nas escolas participantes, apontam falhas em relação a usabilidade da interface e conceitos técnicos que devem ser levados em consideração na criação de aplicações de qualidade.

A Engenharia de aplicações Web (*Web Engineering*), utiliza de princípios da engenharia de software para desenvolver aplicações Web de qualidade, que se baseiam na corretude e completude da aplicação, segundo as necessidades do usuário, para obtenção do produto final (PRESSMAN, 2011). Estas fases são: Especificação dos requisitos, Projeto, Implementação, Teste e Manutenção.

Dentro da engenharia de requisitos, o processo de especificação deve englobar a elicitação, modelo e análise dos requisitos levantados. Estes, tem por finalidade promover uma especificação clara, não ambígua e completa sobre as necessidades do cliente (KOSCIANSKI; SOARES, 2007).

A fase de projeto decide como a aplicação vai operar em termos de arquitetura, interface do usuário e base de dados. É através dessas definições que se estabelece e decide a viabilidade do projeto (PRESSMAN, 2011).

Modelagem e projeto do banco de dados, influenciam totalmente o comportamento e desenvolvimento da aplicação. O modelo relacional introduz as linguagens de consulta de alto nível como SQL (*Structured Query Language*), trazendo facilidades para programadores, é disponibilizado pela maior parte dos SGBDs (Sistemas Gerenciadores de banco de dados) e utiliza representação em tabelas (ELMASRI et al., 2005).

A maioria dos navegadores oferece suporte ao novo HTML5. A linguagem de marcação, combinada com os *styles* possibilitado pelo uso de CSS, agregam valor a aplicação desde o *design* à usabilidade, pelo usuário. *Framework*s como o Bootstrap deixam a aplicação responsiva e o *layout* comportado em qualquer tipo e tamanho de tela, revolucionando os antigos métodos (KIM, 2013).

Já na implementação, tem-se o código fonte da aplicação. Cabe ao programador, através do estudo de viabilização e disponibilidade das ferramentas, a escolha das tecnologias utilizadas, linguagem de programação e padrões de projeto. Nesse momento, deve ser levado em consideração questões como desempenho, usabilidade e tendencias de mercado (DANTAS et al., 2002).

Como linguagem de programação para Web, PHP se destaca pela vasta interação na dinamização de páginas estáticas, ser gratuito, orientado a objetos, embutido no html, executado no servidor, suportar vários bancos de dados e ser altamente portátil para execução em Windows, Linux ou Unix (NIEDERAUER, 2004).

Segundo Dantas (2002) "Podemos pensar em um padrão como a reutilização da essência de uma solução para determinados problemas similares.". Na construção de aplicações Web, a demanda pela utilização de arquiteturas de *softwares* que seguem um padrão crescem cada vez mais, pois facilita a legibilidade, manutenção e compreensão do código. O padrão MVC (*Model-view-controller*) propõe a separação da representação, da informação da sua apresentação (CAVALCANTI, 2006).

Contudo, além de disponibilizar uma vasta gama de áreas de utilização, softwares Web são desenvolvidos com base em fatores de desempenho, boas práticas de programação e padrões estruturais, para melhor alcançar os resultados propostos. Como ferramenta de auxílio à gerência escolar, tende a diminuir o acúmulo de responsabilidades com a separação em módulos, facilidade de acesso à informação, agilidade nos processos e maior segurança e integralidade dos dados (BARBOSA, 2004).

3 MATERIAL E MÉTODOS

Os materiais e métodos descritos, contextualizam parte do desenvolvimento do sistema Web proposto.

3.1 Ferramentas utilizadas

Aqui são descritas as ferramentas utilizadas na implementação do código-fonte, interface, base de dados e algumas auxiliares.

3.1.1 Interface gráfica

A linguagem de marcação criada pelo físico britânico Tim Berners-Lee, se tornou a mais utilizada para a construção de páginas da Web (HOY, 2011). Após diversas reestruturações, em sua grande maioria realizadas pela *World Wide Web Consortium* (W3C), o HTML - *HyperText Markup Language* (Linguagem de Marcação de Hipertexto) se encontra em sua quinta versão - HTML5. Segundo (FERREIRA; EIS, 2011) "Um dos principais objetivos do HTML5 é facilitar a manipulação do elemento possibilitando o desenvolvedor a modificar as características dos objetos de forma não intrusiva e de maneira que seja transparente para o usuário final."

Cascading Style Sheets - CSS¹ é utilizado para estilizar uma página, de forma a adicionar um *link* direto para um documento que contém toda a formatação de estilo utilizada. Possui sintaxe simples, com termos em inglês e possibilita declaração de seletores em blocos.

Bootstrap² possibilita a criação de projetos responsivos e móveis, utilizando ferramentas de código aberto e componentes de *front-end*. É voltado para o desenvolvimento com HTML, CSS, JS e utilização de *plugins* criados no jQuery³, além de disponibilizar uma extensa lista de componentes pré-construídos para uso (SPURLOCK, 2013).

3.1.2 Base de dados

MySQL WorkBench é um SGBD - Sistema Gerenciador de banco de dados, que fornece modelagem de dados e uso de SQL⁴ - *Structured Query Language*, ou Linguagem de Consulta Estruturada.

¹ www.css3.info/

² <https://getbootstrap.com/>

³ <https://jquery.com/>

⁴ Linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional)

3.1.3 Desenvolvimento das funcionalidades

Model-View-Controller - MVC é um padrão arquitetural formulado por Trygve Reenskaug em 1979, para definir quais são os elementos do software e como eles interagem entre si. Tal ferramenta busca proporcionar escalabilidade e eficiência da aplicação (KARAM; IBRAHIM, 2009). Seu principal benefício é isolar as regras de negócio da lógica de apresentação. Para reduzir o acoplamento e aumentar a coesão nas classes, três camadas propostas para serem independentes e podem ser descritas da seguinte maneira:

Model ou modelo funciona como regras de negocio. Gerencia elementos de dados, responde requisições do controlador e efetua alterações de estado.

View ou visão, é responsável por apresentar informações ao usuário, recebendo instruções do controlador e informações do modelo.

Controller ou controlador, é o intermediador entre as requisições do usuário, passadas da visão para as regras de negócio e/ou o contrário. É o único que possui conhecimento dessas duas camadas.

PHP⁵ - *Hipertext Preprocessor* é uma linguagem de *script open source*, amplamente utilizada, adequada para o desenvolvimento web e que pode ser embutida dentro do HTML. Delimitada por *tags* representando início e fim, permitindo alternar entre dentro e fora do "modo PHP"(TONU, 2012).

JavaScript⁶ é uma linguagem de programação interpretada. Criada em 1995, tida como a principal linguagem de programação para navegadores web, que utilizam o modelo cliente servidor, multiparadigma, com tipagem dinâmica, fraca e implícita (FLANAGAN, 2013).

3.1.4 Ferramentas auxiliares

Aqui abordadas, as ferramentas auxiliares no desenvolvimento do software, tem por objetivo manter o registro de alteração de dados e integrar outras ferramentas de desenvolvimento úteis em um único lugar.

Git⁷, pronuncia-se “git” ou “dit”, em inglês britânico é um sistema de controle de versão distribuído (livre e de código aberto), e um sistema de gerenciamento de código fonte

⁵ <http://www.php.net/>

⁶ <https://www.javascript.com>

⁷ <https://git-scm.com/>

com ênfase em velocidade e eficiência. Inicialmente foi projetado para desenvolvimento do Kernel Linux ⁸ mas teve sua adoção para diversos outros projetos (MOREIRA, 2016).

GitHub ⁹ é um serviço web que permite hospedar repositórios de projetos que utilizam Git . Assim ele facilita o trabalho em equipe e incentiva a colaboração com projetos *open source* (MOREIRA, 2016).

NetBeans IDE¹⁰ é um ambiente de desenvolvimento integrado, gratuito e de código aberto. Voltado para desenvolvedores de softwares. Aceita as principais linguagens de programação atuais como JAVA, PHP, JavaScript, HTML5, entre outras (TONU, 2012).

⁸ Forma a estrutura base do sistema operacional/sistema operativo GNU/Linux, que é um sistema operacional tipo unix.

⁹ <https://github.com/>

¹⁰ <https://netbeans.org/>

4 DESENVOLVIMENTO DO SISTEMA

Neste capítulo são abordadas as fases de desenvolvimento do software.

4.1 Especificação de requisitos

O módulo proposto visa satisfazer as necessidades do professor, quanto a utilização dos recursos do diário virtual. Isso impossibilita o compartilhamento de informações necessárias ao professor, como a lista de alunos por exemplo, que deveria ser fornecida por outro módulo (secretaria). Logo, a inserção desses dados fazem parte dos requisitos do sistema atual. Na Tabela 4.1 são descritos os requisitos funcionais levantados para o modo professor.

Tabela 4.1 – Requisitos funcionais

Identificador	Descrição
RF001	O software deve possibilitar adicionar um Diário
RF002	O software deve possibilitar a visualização do Diário
RF003	O software deve possibilitar a exclusão do Diário
RF004	O software deve possibilitar a visualização dos Alunos
RF005	O software deve possibilitar adicionar um novo Aluno
RF006	O software deve possibilitar excluir Aluno
RF007	O software deve possibilitar alterar Aluno
RF008	O software deve possibilitar a visualização das Aulas
RF009	O software deve possibilitar adicionar uma nova Aula
RF010	O software deve possibilitar excluir Aula
RF011	O software deve possibilitar alterar Aula
RF012	O software deve possibilitar a visualização das Avaliações
RF013	O software deve possibilitar adicionar uma nova Avaliação
RF014	O software deve possibilitar excluir Avaliação
RF015	O software deve possibilitar alterar Avaliação

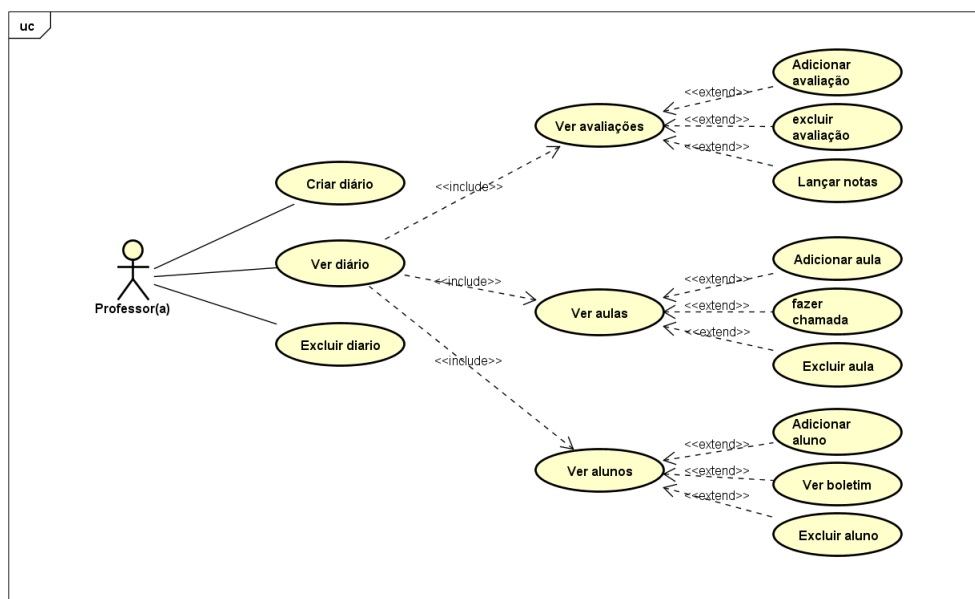
Estes, foram traduzidos para um diagrama de caso de uso, que por sua vez tem a função de documentar o que o sistema faz, no ponto de vista do usuário. Ele pode ser observado na Figura 4.1; o professor, que é o único usuário do módulo, consegue realizar tarefas (casos de uso) no sistema, para cada diário que ele possuir.

4.2 Projeto

O professor desempenha tarefas importantes com o uso do diário. São elas:

- Efetuar o registro da aula, informando a data e um resumo geral específico sobre o conteúdo lecionado nesse dia.

Figura 4.1 – Diagrama de caso de uso



- Registrar também qualquer ocorrência ou fato relevante que aconteça em sala de aula
- Realizar a chamada colocando "presença" ou "falta" para cada aluno matriculado na matéria referente a chamada.
- Transcrever as atividades avaliativas e informar o valor atribuído a cada uma delas, bem como a nota obtida por cada aluno nestas atividades.
- Fechar as notas de cada aluno ao final do período escolar, preenchendo a taleta (relatório).

Tais tarefas foram transcritas ao ambiente de desenvolvimento em formas de adaptações na produção da interface.

O módulo permite que cada professor acesse o conteúdo de seus diários. Para isso, é preciso efetuar *login* com *e-mail* e senha, possibilitando a criação de um novo cadastro, caso ainda não possua. Dentro de cada diário estão disponíveis cadastro e acesso à alunos, avaliações e aulas.

4.2.1 Estruturação do banco de dados

Após a definição dos principais requisitos do sistema e suas funcionalidades, foi estruturado uma base de dados para armazenar as informações necessárias para o funcionamento do módulo. Para isso, optou-se pelo modelo relacional, que possibilita criar entidades e o relacionamento entre elas.

As entidades modeladas (tabelas) representam as principais estruturas de dados necessárias para implementação das funcionalidades do sistema. São elas nomeadas de: Professor, Diário, Aluno, Avaliação, Nota, Chamada, Aula, PeríodoDeRegime e Turma. Tais nomes foram adotados, sugestivamente, como abstração dos objetos do mundo real para facilitar o entendimento. Assim, podemos dizer por exemplo que, nesse sistema, dadas as necessidades, o Professor (representado pela tabela professor) possui atributos (dados) "nome, e-mail, senha e uma imagem de perfil".

A estrutura final das entidades e do relacionamento entre elas, é apresentado na Figura 4.2. O script SQL (*Structured Query Language*, ou Linguagem de Consulta Estruturada) exportado, é utilizado pelo servidor para criação e disponibilização ao banco de dados para uso na aplicação.

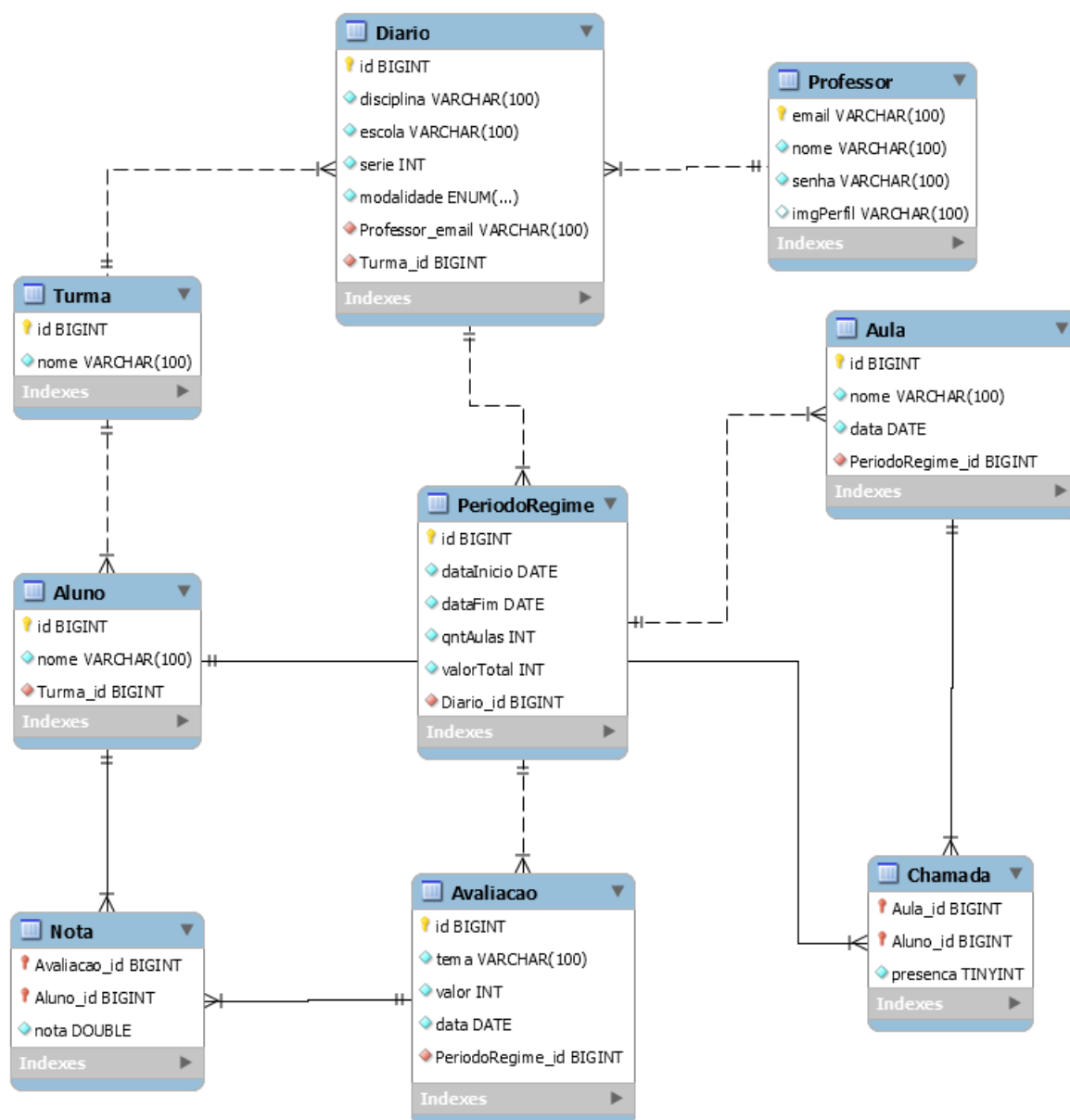
4.3 Implementação

Para desenvolvimento do software, adotou-se o uso da arquitetura *Model-view-controller* - MVC. Tal arquitetura propõe que seja feita a separação da representação da informação (dados), da interação do usuário (interface). A arquitetura da implementação do Diário de Classe encontra-se estruturada em três diretórios: *Model*, *View* e *Controller*. Tal divisão com seus respectivos arquivos são mostrados na Figura 4.3 e Figura 4.4.

No *Model*, estão os objetos e seus respectivos *gets* e *sets*, bem como as classes que estabelecem conexão ao banco de dados e as classes de acesso a dados (DAO - *Date Access Object*). Essa estruturação permite que quando ocorra mudança nos dados de qualquer objeto, seja fácil a notificação para classes que exercem controle sobre esses dados e o repasse para as classes responsáveis às "visões" dos usuários.

A interface se localiza no *View*. Ali estão todos os códigos em HTML, CSS e JS responsáveis pela renderização do conteúdo que interage com o usuário, imagens e fontes tipográficas. Por último, mas não menos importante, temos o diretório *Controller*, onde ficam as principais tarefas e requisições, controladas por comandos em PHP. Em exemplo: toda ação de cadastro de um "professor" é captada por um campo na interface *view* e transmitido para o *controller* "professorInterface.php" que é responsável por analisar os comandos e direcionar o caminho de cada tipo de ação para sua tarefa, esperar uma resposta e executar uma ação de retorno com essa resposta para a *view*.

Figura 4.2 – Modelo relacional do banco de dados



Ao entrar no sistema, o usuário se depara com duas opções: se cadastrar ou *logar* no sistema. Na primeira, o usuário insere dados de *e-mail*, senha, nome e imagem de perfil(opcional). Após prosseguir, na *view* os dados são encaminhados para *interfaceProfessor.php* que chama o *bean* responsável por registrar os dados. Esses dados são devolvidos a ele, que solicita a classe DAO (*Data Access Object*) que armazene os dados no banco. As Figuras 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, e 4.11 respectivamente, mostram os arquivos com o fluxo de dados descrito.

As estrutura das classes utilizadas no cadastro acima exemplificado, são mostradas nas figuras 4.12 e 4.13.

Figura 4.3 – Divisão dos diretórios - controller e model

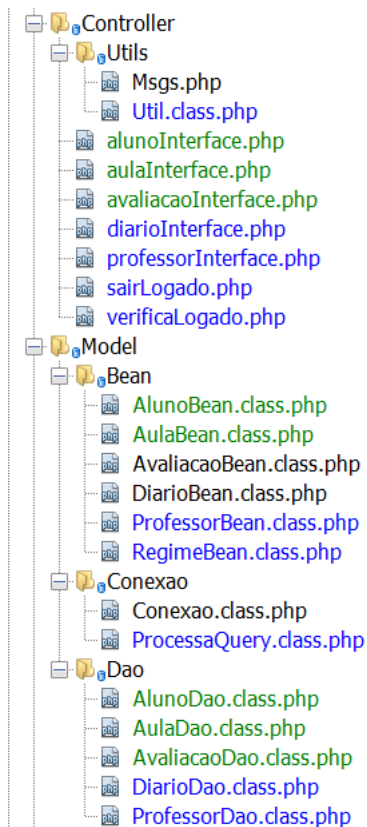


Figura 4.4 – Divisão dos diretórios - view

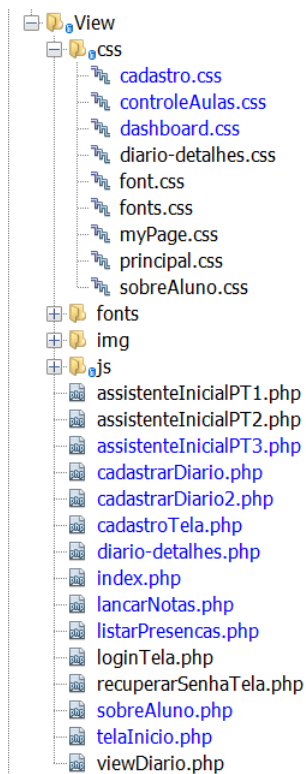


Figura 4.5 – Arquivo "index.php"

```

<!--Botoes-->
<div class="Container">
  <div class="row btn-initial">
    <div class="col-sm-4 col-sm-offset-2 col-md-4 col-md-offset-2 col-lg-4 col-lg-offset-2 col-xl-4 col-xl-offset-2">
      <a href="cadastroTela.php" class="btn btn-info" role="button">FAZER MEU CADASTRO</a>
    </div>
    <div class="col-sm-6 col-md-6 col-lg-6 col-xl-6">
      <a href="loginTela.php" class="btn btn-info btn-initial" role="button">ACESSAR MEUS DIÁRIOS</a>
    </div>
  </div>
</div>

```

Figura 4.6 – Arquivo "cadastroTela.php"

```

<div class="card card-container">
  <!-- 
  
  <p id="profile-name" class="profile-name-card">CRIAR NOVO USUÁRIO</p>
  <form class="form-signin" action="assistenteInicialPT1.php" method="POST">
    <span id="reauth-email" class="reauth-email"></span>
    <input type="email" id="inputEmail" class="form-control" placeholder="Email" name="email" required >
    <input type="password" id="inputPassword" class="form-control" placeholder="Senha" name="senha" required>

    <div id="remember" class="checkbox">
      <label>
        <input type="checkbox" name="lembrar" value="s"> Remember me
      </label>
    </div>

    <button class="btn btn-lg btn-primary btn-block btn-signin">ENTRAR</button>
  </form><!-- /form -->
  <a href="#" class="forgot-password">
    Esqueceu sua senha
  </a>
</div><!-- /card-container -->

<a href="loginTela.php">
  <button class="btn-cdst-rtrn btn btn-block">
    <span class="glyphicon glyphicon-chevron-left"> </span> JÁ POSSUO CADASTRO
  </button>
</a>

```

Figura 4.7 – Arquivo "assistenteInicialPT1.php"

```

<div class="col-sm-2 col-md-2 col-lg-2 col-xl-2">
  <div class="image"></div>
</div>
<div class="col-sm-10 col-md-10 col-lg-10 col-xl-10 padd top">
  <h2 class="cdst-right">
    <span id="title2"> SEJA MUITO BEM VINDO(A), </span><br/>
    <span id="subtitle">PROFESSOR(A) !</span>
  </h2>

  <p class="description text-justify">
    Esse assistente vai te ajudar a dar seus primeiros passos <br/>
    cadastrando-se e criando seus diários. Você pode clicar no <br/>
    botão fechar acima a qualquer momento para sair do assistente.<br/>
  </p>

  <form action="assistenteInicialPT2.php" method="POST" class="pull-right">
    <!-- hidden -->
    <input type="hidden" name="email" value="<?php echo $_POST['email']; ?>">
    <input type="hidden" name="senha" value="<?php echo sha1($_POST['senha']); ?>">

    <input type="submit" value="VAMOS LÁ!" class="btn btn-info" role="button" >
  </form>
</div>

```

4.4 Testes

Os testes feitos na aplicação, são baseados no conceito de "teste de caixa preta". Tal tipo de teste (também chamado de funcional ou de validação), tem por objetivo avaliar o comporta-

Figura 4.8 – Arquivo "assistenteInicialPT2.php"

```

<form class="form-signin" action="../../../Controller/professorInterface.php" method="POST" enctype="multipart/form-data">
  <!-- hidden -->
  <input type="hidden" name="acao" value="cadastrar">
  <input type="hidden" name="email" value="<?php echo $_POST['email']; ?>">
  <input type="hidden" name="senha" value="<?php echo $_POST['senha']; ?>">

  <div class="row ctn-dados">
    <div class="col-5 col-sm-5 col-md-5 col-lg-5 col-xl-5">
      <div id="ctn-foto" class="pull-center">
        <input type="file" name="arquivo" id="arquivo" >
        <!--<input type="submit" value="alterarImagem" -->
        <input type="hidden" value="<?php ?>">
      </div>
    </div>

    <div class="col-7 col-sm-7 col-md-7 col-lg-7 col-xl-7">
      <h3 class="cdst">Seu nome</h3>
      <input type="text" id="inputName" class="form-control-cdst btn-outline" placeholder="Ana Maria" name="nome" required>
    </div>
  </div>

  <div class="row">
    <div class="col-6 col-sm-6 col-md-6 col-lg-6 col-xl-6"> </div>
    <div class="col-6 col-sm-6 col-md-6 col-lg-6 col-xl-6">
      <button class="btn btn-info" role="button" type="submit">CONTINUAR
      <span class="glyphicon glyphicon-chevron-right pull-right"></span>
    </button>
  </div>
</form>

```

Figura 4.9 – Arquivo "assistenteInicialPT3.php"

```

require_once('verificaLogado.php');
require_once('Utils/Msgs.php');
require_once('Utils/Util.class.php');
require_once('.../Model/Bean/ProfessorBean.class.php');
require_once('.../Model/Dao/ProfessorDao.class.php');
$_DADOS = $_POST;

$acao = trim($_DADOS['acao']);
if(!isset($acao) || $acao == ''){//se houve algum problema na hora de receber a acao
  ?><script>alert('<?php echo $msgSemAcao; ?>');</script><?php
  header('Location: ../View/telaInicio.php');
}

```

Figura 4.10 – Arquivo "professorInterface.php"

```

switch($acao){
  case 'cadastrar':
    //ler dados
    $nome = Util::limpaString($_DADOS['nome']);
    $email = Util::limpaString($_DADOS['email']);
    $senha = Util::limpaString($_DADOS['senha']);

    //criar bean
    $professorBean = new ProfessorBean();
    $professorBean->setNome($nome);
    $professorBean->setEmail($email);
    $professorBean->setSenha($senha);

    //executa no banco
    $retorno = ProfessorDao::cadastrar($professorBean);
    if($retorno->status){//se tudo ocorreu bem
      //seta a session
      session_start();
      $_SESSION['email'] = $email;
      $_SESSION['nome'] = $nome;
      $_SESSION['imgPerfil'] = $semImagem;

      //seta os cookies se houver
      if(isset($_DADOS['lembrar'])){
        $_COOKIE['email'] = $email;
        $_COOKIE['nome'] = $nome;
        $_COOKIE['imgPerfil'] = $semImagem;
      }

      //redireciona
      ?>
      <script>
        window.location.replace("../View/assistenteInicialPT3.php");
      </script>
    <?php
  }else{

```

Figura 4.11 – Arquivo "professorInterface.php- continuação da figura 4.10

```

    }else{
        ?>
        <script>
            alert('<?php echo $retorno->resposta; ?>');
            window.location.replace("../View/cadastroTela.php");
        </script>
        <?php
    }
    break;

```

Figura 4.12 – Classe "ProfessorBean.class.php"

```

//Classe que representa a tabela Professor
class ProfessorBean{
    private $nome;
    private $email;
    private $senha;
    private $imgPerfil;
    //Retorna o nome do Professor
    public function getName(){
        return $this->nome;
    }
    //Altera o nome do Professor
    public function setName($valor){
        $this->nome = $valor;
    }
    //Retorna o email do Professor
    public function getEmail(){
        return $this->email;
    }
    //Altera o email do Professor
    public function setEmail($valor){
        $this->email = $valor;
    }
    //Retorna a senha do Professor
    public function getSenha(){
        return $this->senha;
    }
    //Altera a senha do Professor
    public function setSenha($valor){
        $this->senha = $valor;
    }
    //Retorna a imgPerfil do Professor
    public function getImgPerfil(){
        return $this->imgPerfil;
    }
    //Altera a imgPerfil do Professor
    public function setImgPerfil($valor){
        $this->imgPerfil = $valor;
    }
}

```

Figura 4.13 – Classe "ProfessorDao.php"

```

require_once('../Model/Conexao/ProcessaQuery.class.php');
//Classe que executa as acoes no banco
class ProfessorDao{
    //Conecta com o banco e cadastra o professor
    public static function cadastrar($bean){
        //cria a query
        $query = "INSERT INTO
                    Professor(nome,email,senha,imgPerfil)
                    VALUES
                    ('{$bean->getNome()}', '{$bean->getEmail()}', '{$bean->getSenha()}', NULL);";

        //executa
        return ProcessaQuery::executarQuery($query);
    }
    //Conecta com o banco e retorna o professor cujo dados batem
    public static function getProfessor($bean){
        //cria a query
        $query = "SELECT nome,imgPerfil
                    FROM Professor
                    WHERE email = '{$bean->getEmail()}' AND senha = '{$bean->getSenha()}';";

        //executa
        return ProcessaQuery::consultarQuery($query);
    }
    //Conecta com o banco e altera os dados do professor
    public static function alterar($bean){
        //cria a query
        $query = "UPDATE Professor
                    SET
                        nome = '{$bean->getNome()}'/*,
                        senha = '{$bean->getSenha()}',
                        imgPerfil = '{$bean->getImgPerfil()}'*/
                    WHERE email = '{$bean->getEmail()}';";

        //executa
        return ProcessaQuery::executarQuery($query);
    }
}

```

mento externo de cada componente, utilizando a saída obtida por uma entrada, com o resultado esperado.

Os testes foram feitos durante a fase de desenvolvimento para testar fluxo de dados, inspecionar condições, variáveis e outros, através da utilização da interface gráfica para validação.

5 RESULTADOS E DISCUSSÃO

5.1 Módulo Professor

O módulo implementado foi o do Professor. Para utilizar os componentes (Aluno, Aula, Avaliação e Diário), o sistema exige autenticação do usuário (Figura 5.6). Caso ainda não possua *login*, basta optar pela criação de um novo usuário como mostrado na Figura 5.1. O cadastro de professor é uma funcionalidade do módulo e é demonstrado nas Figuras 5.2, 5.3, 5.4 e 5.5. Após terminar seu cadastro pessoal, o professor cria seu primeiro diário, que será demonstrado posteriormente. Todas figuras aqui apresentadas, simulam o cadastro do professor fictício nomeado "Mallú Eduarda".

Finalizando esse processo, a página inicial é demonstrada na Figura 5.7.

Figura 5.1 – Tela inicial da aplicação



Figura 5.2 – Tela de cadastro de usuário

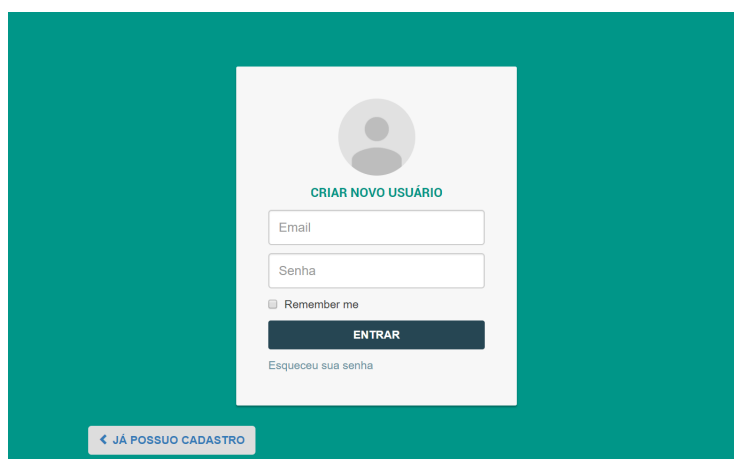


Figura 5.3 – Assistente inicial: parte 1



Figura 5.4 – Assistente inicial: parte 2

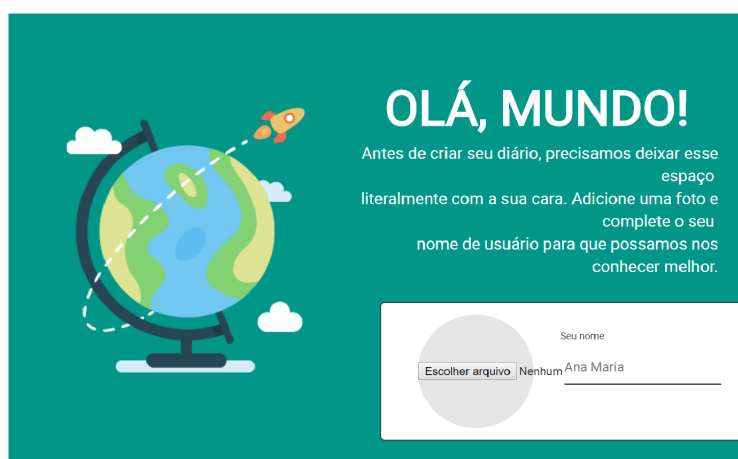


Figura 5.5 – Assistente inicial: parte 3



5.2 Componentes

Aqui são citados e explicados os componentes do sistema.

Figura 5.6 – Tela de login

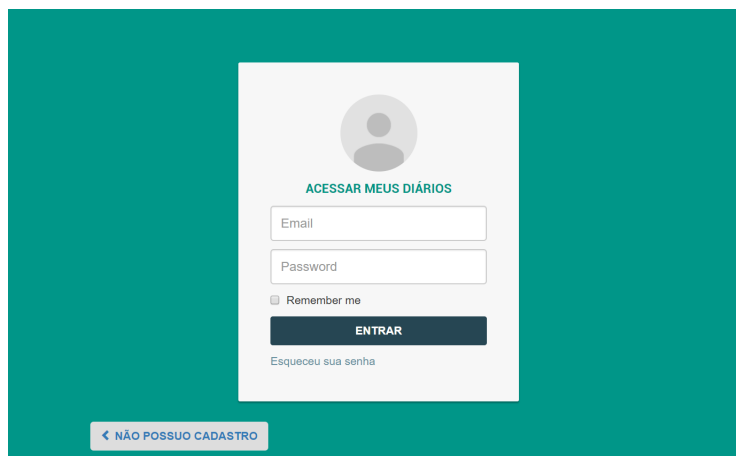
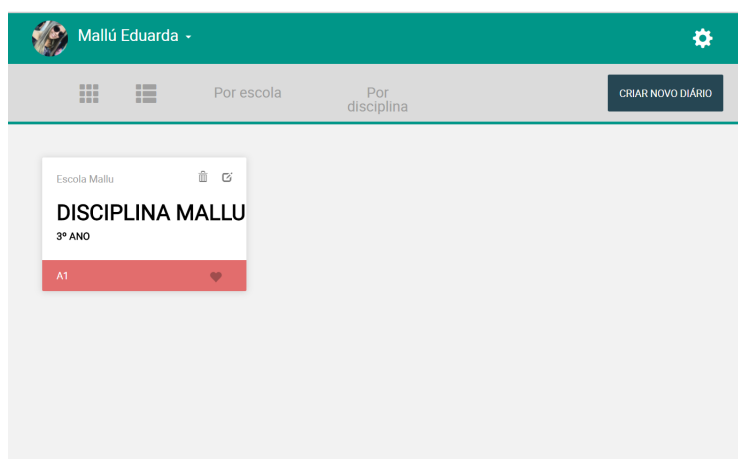


Figura 5.7 – Pagina inicial do professor



5.2.1 Diário

Como componente principal, portador de todas as informações relevantes ao registro da prática pedagógica do professor, o diário de classe desempenha papel fundamental e único no módulo. As funcionalidades são: cadastrar, editar, excluir e visualizar diário. As mesmas, serão numeradas e explicadas posteriormente.

A função que carrega as informações e controla as requisições do usuário dentro de cada diário, é mostrada na figura 5.8. Cada clique em uma função é redirecionado para uma ação.

1. Cadastrar:

Na tela principal do usuário, existe o botão para cadastro de um novo diário. As informações solicitadas condizem com as presente no diário de papel, que é utilizado pelos professores. São solicitados o nome da escola, nome da disciplina, modalidade, série, identificação da turma e regime de aulas, como mostrado na Figura 5.9. Após a

Figura 5.8 – Controlador de ações

```

4  $(document).ready(function () {
5      //carrega informacoes
6      getDiario();
7      getPeriodoRegimes();
8      getAulas();
9      getAvaliacao();
10     getAlunos();
11
12     //ouvintes dos botoes
13     $("#salvarAula").on("click", addAulaNoBanco);
14     $("#salvarAvaliacao").on("click", addAvaliacaoNoBanco);
15     $("#salvarAluno").on("click", addAlunoNoBanco);
16
17     $("#aulasBtn").on("click", aulasBtnFunc);
18     $("#avaliacaoBtn").on("click", avaliacaoBtnFunc);
19     $("#alunosBtn").on("click", alunosBtnFunc);
20
21
22
23     $("#alterarAula").on("click", alteraAulaNoBanco);
24     $("#alterarAvaliacao").on("click", alteraAvaliacaoNoBanco);
25     $("#alterarAluno").on("click", alteraAlunoNoBanco);
26 });

```

confirmação de preenchimento dos dados, eles são passados para próxima página, para completar o cadastro.

Nessa segunda parte, são solicitados dados referentes ao regime de aulas, definindo os períodos letivos, datas de início e fim para cada um, quantidade de aulas e o valor (em pontos) estipulado para o mesmo (Figura 5.10). Essas informações são passadas ao controlador, que capta a "ação" pelo método POST, trata os dados necessários, cria o bean, executa no banco através do dao e redireciona para tela correspondente. Em caso de erro nesse processo, o usuário é informado e redirecionado para recadastrar os dados. Do contrário, a tela inicial com os diários cadastrados é retornada.

Figura 5.9 – Cadastrar Diário: parte 1

Assistente de Criação de Diário

Cada diário pertence a uma escola e turma específica, por essa razão, precisamos que você informe o nome completo da Escola e os dados da turma que você gostaria de controlar nesse novo diário.

Ex: Escola Municipal Jardim Encantado
Série: 3
Modalidade de ensino: Ensino Fundamental
Identificação de turma: A

Nome completo da escola
Nome da escola

Nome da disciplina
Nome da disciplina

Modalidade
Ensino Primário

Série
1ª

Identificação da turma
Identificação Turma

Regime de aulas
Bimestral

CONTINUAR

2. Editar:

Ao optar pela edição do conteúdo, os dados são retornados para visão do usuário, advindos do banco de dados (Figuras 5.11 e 5.12).

Figura 5.10 – Cadastrar Diário: parte 2

Assistente de Criação de Diário

No regime bimestral, você pode adicionar até 4 períodos. Lembre-se que cada período tem uma data de início e fim. Caso queira fazer isso depois, clique no botão finalizar. Ao adicionar um novo período não se esqueça de clicar no botão confirmar (Você sempre poderá editar essas informações futuramente, caso cometa algum engano).

1?2?3? Bimestre	1?2?3? Bimestre	1?2?3? Bimestre	1?2?3? Bimestre
Início do período 22/01/2018	Início do período 22/01/2018	Início do período 22/01/2018	Início do período 22/01/2018
Fim do período 22/01/2018	Fim do período 22/01/2018	Fim do período 22/01/2018	Fim do período 22/01/2018
Quantidade de aulas	Quantidade de aulas	Quantidade de aulas	Quantidade de aulas
Valor do Bimestre	Valor do Bimestre	Valor do Bimestre	Valor do Bimestre

[Cancel e sair](#)
[Próximo >](#)

Figura 5.11 – Editar Diário: parte 1

Assistente de Edição de Diário

Cada diário pertence a uma escola e turma específica, por essa razão, precisamos que você informe o nome completo da Escola e os dados da turma que você gostaria de controlar nesse novo diário.

Ex: Escola Municipal Jardim Encantado
Série: 3
Modalidade de ensino: Ensino Fundamental
Identificação de

Nome completo da escola
Escola Mallu

Nome da disciplina
Disciplina Mallu

Modalidade
Ensino Médio

Série
3º

Identificação da turma
A1

Regime de aulas
Bimestral

CONTINUAR >

Figura 5.12 – Editar Diário: parte 2

Assistente de Edição de Diário

No regime bimestral, você pode adicionar até 4 períodos. Lembre-se que cada período tem uma data de início e fim. Caso queira fazer isso depois, clique no botão finalizar. Ao adicionar um novo período não se esqueça de clicar no botão confirmar (Você sempre poderá editar essas informações futuramente, caso cometa algum engano).

1 Bimestre	2 Bimestre	3 Bimestre	4 Bimestre
Início do período 22/01/2018	Início do período 20/03/2018	Início do período 16/05/2018	Início do período 23/08/2018
Fim do período 26/01/2018	Fim do período 26/04/2018	Fim do período 25/07/2018	Fim do período 25/10/2018
Quantidade de aulas 10	Quantidade de aulas 15	Quantidade de aulas 15	Quantidade de aulas 10
Valor do Bimestre 25	Valor do Bimestre 25	Valor do Bimestre 25	Valor do Bimestre 25

[Cancel e sair](#)
[Próximo >](#)

3. Excluir:

Ao excluir um diário, as informações são apagadas da base de dados.

4. Visualizar:

Os dados são dispostos para o usuário, em forma de acesso aos outros componentes. Para complementar as informações do diário, podem ser adicionados alunos, aulas e avaliações ao diário. A figura 4.14 mostra o carregamento da pagina inicial do diário.

5.2.2 Aluno

Os alunos de uma instituição, realizam seu cadastro por meio da secretaria. Esses dados são repassados diretamente para os professores, pela lista de chamada, através do diário de classe. O mesmo, possui acesso somente ao nome desses alunos.

Entretanto, nesse módulo, essas informações são inseridas manualmente como funcionalidade do sistema, pois não existe outro módulo senão o do professor. É possível cadastrar, editar, excluir, visualizar o boletim de cada aluno e visualizar os alunos.

Figura 5.13 – Cadastro de Aluno

Figura 5.14 – Visualização dos Alunos

Figura 5.15 – Visualização do boletim do Aluno

Boletim de
aluno teste
Média geral: 0%

FREQUÊNCIA

NOTAS

GERAR TABELA

5.2.3 Aula

Como parte das funções pedagógicas designadas ao professor, está o registro de aulas. Estas, devem possuir data e, uma descrição curta e objetiva sobre o conteúdo ministrado. Além de qualquer outra ocorrência de importância durante o período da aula.

Também é de responsabilidade do professor, efetuar a chamada, aluno por aluno, e registra-la com presença ou falta.

As funções de editar, cadastrar, excluir e visualizar são implementadas por meio de modais em java script que respondem a chamados da *view*.

Figura 5.16 – Cadastro de Aula

Diário - Aulas

Escola Mallu

DISCIPLINA MALLU
3º Ano | A1

Aulas

Avaliações

Alunos

Adicionar nova aula

Data da Aula
dd/mm/aaaa

Nome
Digite o nome da Aula

Período Regime
22/01/2018 -> 26/01/2018

Fechar Cadastrar

ADICIONAR AULA

5.2.4 Avaliação

Cada período letivo possui um valor. O somatório destes, corresponde ao total de cem pontos no ano letivo e é distribuído em avaliações no decorrer deste. Avaliações são métodos

Figura 5.17 – Visualização das Aulas

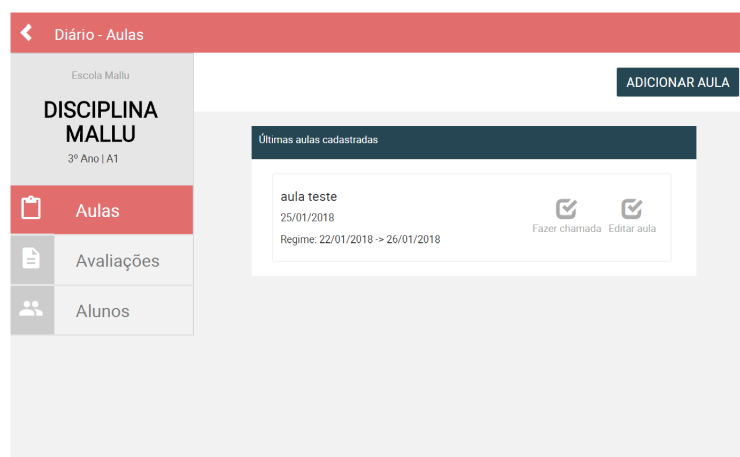
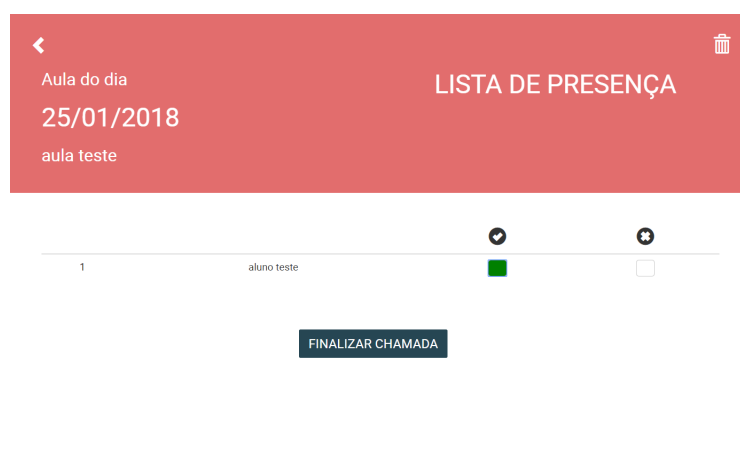


Figura 5.18 – Listar presença dos Alunos na Aula



que tem por objetivo dar nota as atividades propostas aos alunos e devem ser registradas no diário pelo professor.

Figura 5.19 – Cadastro de Avaliação

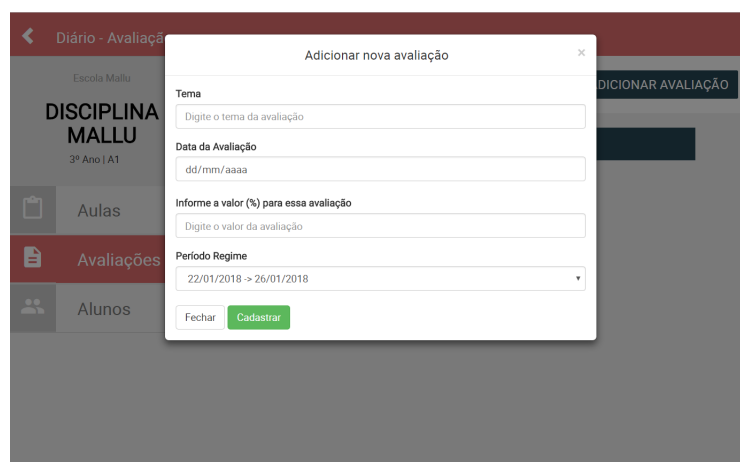


Figura 5.20 – Visualização das Avaliações

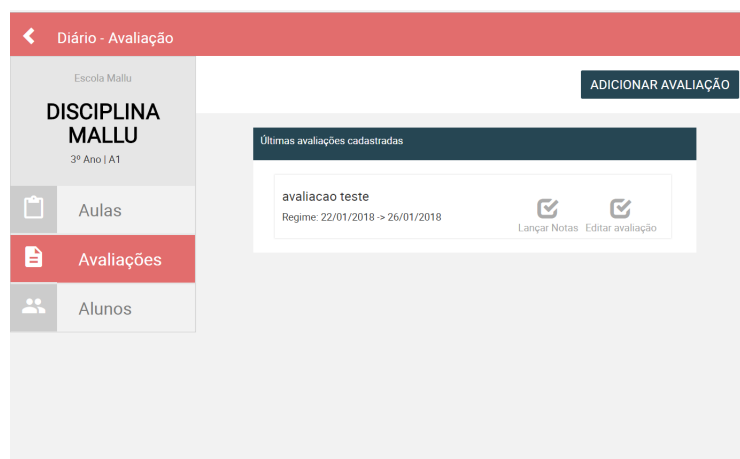


Figura 5.21 – Lançar nota de Avaliação

A interface possui um cabeçalho vermelho com o texto "Avaliação - Valor 30 pontos" e o ícone de uma lixeira. O nome da avaliação "avaliacao teste" e a data "25/01/2018" são exibidos. À direita, os detalhes da avaliação são listados: "Escola Mallu", "Disciplina: Disciplina Mallu", "Série: 3º Ano", "Turma: A1" e "Regime: 22/01/2018 -> 26/01/2018". Abaixo, uma tabela com o cabeçalho "aluno teste", "%", e "Nota" contém uma única linha com o nome do aluno e um campo de entrada para a nota. Um botão "CONCLUIR LANÇAMENTO" está posicionado na base da interface.

aluno teste	%	Nota
1		30

6 CONCLUSÕES

O módulo proposto para implementação foi desenvolvido e atua como proposta de substituição do Diário de Classe de papel, disponibilizando uma plataforma virtual com os requisitos levantados durante a especificação do projeto. É possível acessar a aplicação pelo endereço <http://mallusitediario.000webhostapp.com/View/index.php> . O código fonte está disponível na plataforma GitHub, pelo endereço: <https://github.com/mallueduardab/ModuloWebDiarioDeClasse>.

Como proposta de trabalhos futuros, é interessante a implementação de outros módulos viáveis a integralização da ferramenta com a escola, para auxílio na gestão escolar, como por exemplo, desenvolver os módulos para secretaria e direção.

Ressalta-se também a importância da realização de testes mais completos, com interação dos usuários finais.

REFERÊNCIAS

- ALBINO, R.; SOUZA, C. A. de. Avaliação do nível de uso das tics em escolas brasileiras: uma exploração dos dados da pesquisa “tic educação”. **Economia & Gestão**, v. 16, n. 43, p. 101–125, 2016.
- BARBOSA, E. F. **Uma contribuição ao processo de desenvolvimento e modelagem de módulos educacionais**. Tese (Doutorado) — PhD thesis, 2004.
- BARBOSA, R. R.; SEPÚLVEDA, M. I. M.; COSTA, M. U. P. da. Gestão da informação e do conhecimento na era do compartilhamento e da colaboração. **Informação & Sociedade**, Universidade Federal da Paraíba-Programa de Pós-Graduação em Ciência da Informação, v. 19, n. 2, 2009.
- BRITO, G. d. S. Tecnologias para transformar a educação. **Educar em Revista**, SciELO Brasil, n. 28, p. 279–282, 2006.
- BRITO, S. S. A. As mídias digitais e a prática pedagógica. **Eventos Pedagógicos**, v. 4, n. 1, 2013. ISSN 2236-3165. Disponível em: <<http://sinop.unemat.br/projetos/revista/index.php/eventos/article/view/1158>>.
- CAVALCANTI, M. **Arquitetura MVC**. 2006.
- DANTAS, A. et al. Suporte a padrões no projeto de software. **Caderno de Ferramentas do XVI Simpósio Brasileiro de Engenharia de Software, Gramado, Rio Grande do Sul, Brasil, Outubro**, 2002.
- ELMASRI, R. et al. **Sistemas de banco de dados**. Pearson Addison Wesley, 2005.
- FERREIRA, E.; EIS, D. **Html5: Curso w3c escritório brasil**. São Paulo, 2011.
- FLANAGAN, D. **JavaScript: O guia definitivo**. Tradução de João Eduardo Nóbrega Tortello. Revisão técnica de Luciana Nedel. 6. ed. Porto Alegre: O’Reilly, 2013. 1062 p.
- HOY, M. B. Html5: A new standard for the web. **Medical Reference Services Quarterly**, Routledge, v. 30, n. 1, p. 50–55, 2011. PMID: 21271452. Disponível em: <<https://doi.org/10.1080/02763869.2011.540212>>.
- KARAM, M.; IBRAHIM, M. A. Synchronous online help support with visual instruction aids for workflow-based mvc web applications: Proceedings of the 27th acm international conference on design of communication. **SIGDOC**, p. 105–114, 2009.
- KIM, B. Responsive web design, discoverability, and mobile challenge. **Library technology reports**, v. 49, n. 6, p. 29–39, 2013.
- KOSCIANSKI, A.; SOARES, M. dos S. **Qualidade de Software -2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. São Paulo: Novatec Editora, 2007.
- LACERDINO, I.; MELCHIORI, A. P. P. Implementação de métodos analíticos de usabilidade e prototipação para melhoria de aplicação móvel educacional: Estudo de caso do "diário de classe". **Monografia de Graduação, Universidade Federal de Lavras**, 2016.
- MOREIRA, J. P. Desbravando o git e o github. **Revista Educacional Interdisciplinar**, v. 5, n. 1, 2016.

NIEDERAUER, J. Desenvolvendo websites com php. **São Paulo: Novatec**, 2004.

PARO, V. H. **Gestão escolar, democracia e qualidade do ensino**. [S.l.]: Ática São Paulo, 2007.

PRESSMAN, R. S. Engenharia de software: uma abordagem profissional. 7ª edição. **Ed: McGraw Hill**, 2011.

SPURLOCK, J. **Bootstrap: Responsive Web development**. Printed in the United States os America: "O'Reilly Media, Inc.", 2013.

TERESA, M. Manual do professor. 1999.

TONU, M. H. **PHP Application Development with NetBeans**: Beginner's guide. Birmingham - Mumbai: Packt Publishing Ltd, 2012.