

# **Desenvolvimento de uma ferramenta de auxílio ao docente utilizando técnicas da engenharia de software**

**Maikel A. Vitancourt**

Curso de Ciência da Computação – Universidade Regional Integrada do Alto Uruguai e das Missões – Campus Santiago (URI – Santiago)  
Caixa Postal 97700-000 – Santiago – RS – Brasil

maikel-93@hotmail.com

**Abstract.** *Educational systems are increasingly present in educational institutions, studies show that if well used they provide good resources. The purpose of this work was to develop a software for the specific use of the teacher, which had its functionalities validated from a questionnaire to collect requirements, from which some techniques of software engineering were used to model, develop and test the system. The selected functionalities were the bank of files, bank of questions and control of class. With the tool developed, it is expected to meet the needs of the teacher, streamlining and automating pedagogical tasks and contributing to the digital inclusion of teachers.*

**Resumo.** *Sistemas educacionais estão cada vez mais presentes nas instituições de ensino, estudos demonstram que se bem utilizados fornecem bons recursos. A proposta deste trabalho foi desenvolver um software para uso específico do professor, o qual teve suas funcionalidades validadas a partir de um questionário para levantamento de requisitos, a partir disso foram utilizadas algumas técnicas da engenharia de software para modelar, desenvolver e testar o sistema. As funcionalidades selecionadas foram o banco de arquivos, banco de questões e controle de classe. Com a ferramenta desenvolvida espera-se que atenda às necessidades do docente, agilizando e automatizando tarefas pedagógicas e contribua para inclusão digital dos docentes.*

## **1. Introdução**

A utilização de Tecnologias da Informação e Comunicação (TICs) na educação brasileira foi um processo lento e conturbado. Valente (2004 apud BERGAMASCO, 2013) afirma que no Brasil o uso da informática na área educacional teve uma defasagem no tempo e a sua disseminação levantou a questão sobre a validade de utilização de tais recursos como necessidade na educação.

Segundo Bridge & Appleyard (2008 apud DALMON et al., 2012) softwares educacionais são desenvolvidos para alcançar vantagens pedagógicas do ponto de vista da docência. Exemplos eficazes de aplicação de informatização na área educacional são os que proporcionam agilidade ou automatização em procedimentos repetitivos, porém para alcançar tais vantagens, é necessário que seja desenvolvido e implantado softwares de auxílio no âmbito acadêmico.

De acordo com Valente (2004 apud BERGAMASCO, 2013) são necessários apenas quatro ingredientes básicos: o computador, o software educativo, o aluno e o professor capacitado para usar o computador como meio educacional. Por mais que a implantação de tecnologias nesse meio seja simples, é necessária ação de alguma parte

interessada em contribuir para a solução do problema.

A UNESCO (2013 apud BERGAMASCO, 2013) alerta sobre as consequências dessa exclusão digital, pontuando que o Brasil precisa investir na formação dos professores para utilizar as TICs na educação. Desta forma este trabalho teve como objetivo o desenvolvimento de software para educação e introduzir os docentes na utilização de TICs, de forma a facilitar as tarefas desenvolvidas diariamente no meio acadêmico.

A metodologia utilizou uma técnica da engenharia de software e abordou uma técnica de elicitação de requisitos que possibilitou identificar os requisitos do software, especificar os requisitos, assim provendo uma base para projetar e modelar o software, buscando utilizar boas práticas de programação.

Para levantamento dos requisitos o trabalho utilizou a aplicação de um questionário que foi escolhido devido à facilidade de aplicação. O questionário levou ao usuário a possibilidade de responder se uma funcionalidade citada é útil para seu dia a dia ou ainda se tinha ideia de outra funcionalidade que não estava no questionário.

Com o resultado do questionário foi possível definir o escopo que se adequou ao cronograma de tempo disponível, seguido do projeto e modelagem do sistema, utilizou-se linguagem UML (*Unified Modeling Language*) e o modelo relacional, que foi usado para representar o projeto de banco de dados.

A partir do planejamento do software foi desenvolvido uma ferramenta para auxiliar o docente. Durante o desenvolvimento buscou-se utilizar boas práticas de programação, como conceitos de orientação a objetos e recomendações de padrões do PHP (*Hypertext Preprocessor*), como a PSR-1 (*PHP Standard Recommendation*), PSR-2, PSR-4 e para validação dos requisitos foram realizados testes funcionais.

O objetivo do presente trabalho se dispôs da seguinte forma, o objetivo geral se baseou em desenvolver uma ferramenta de software para auxiliar os docentes, utilizando técnicas da engenharia de software para definir as funcionalidades, modelar diagramas e testar o sistema finalizado.

A estrutura do trabalho se dispõe da seguinte forma, a seção 2 inclui uma breve descrição de técnicas e ferramentas para desenvolvimentos de softwares que foram utilizadas buscando desenvolver um software de qualidade. A seção 3 apresenta os trabalhos relacionados ao assunto educação e tecnologia, dos quais são citados sistemas como o Moodle, Tidia-AE e TelEduc, que proporcionaram ideias de como a tecnologia pode auxiliar tanto professor como aluno. A seção 4 aborda o desenvolvimento do trabalho, levantamento de requisitos, definição do escopo, projeto e modelagem, implementação do software e testes.

## **2. Técnicas e ferramentas para desenvolvimento de softwares**

Softwares são programas de computador que propõem funcionalidades a seus usuários, segundo Sommerville (2011) o mundo moderno necessita do mesmo, que contribui em sua parte fornecendo serviço de controle e gerenciamento de informações, incluindo também sistemas e infraestruturas a níveis nacionais.

Segundo Pressman (2016) o desenvolvimento de software necessita de conceitos que se atentem com todos os aspectos de produção de software, propondo conceitos e técnicas de desenvolvimento que visam não só um sistema final de qualidade como também um sistema de fácil manutenção.

Segundo Sommerville (2011) o objetivo principal da engenharia de software é focar em atividades da produção de software, desde o seu planejamento, até posteriormente com o mesmo em uso e na sua manutenção. Os processos de sua produção não são as únicas preocupações durante o desenvolvimento do software, pois softwares são abstratos e intangíveis, não se regendo pelas leis da física ou processos de manufatura, também são identificados: a necessidade de incluir conceitos de gerenciamento de projeto para apoiar o desenvolvimento do software e os programas que podem programas podem se tornar extremamente complexos e sem restrição em seu tamanho, daí surge a engenharia de software.

Pressman (2016) ressalta a importância de examinar o software e suas características em geral, pois este se difere da construção de objetos físicos, sendo mais um elemento lógico. O autor descreve as operações da engenharia de software, abordando, de forma sistemática e disciplinada, o processo de desenvolvimento e de manutenção de software, aprofundando-se e aperfeiçoando essas práticas.



**Figura 1. Camadas da engenharia de software. PRESSMAN (2016, p. 16)**

As camadas da engenharia de software vistas na Figura 1, demonstram o foco na qualidade, sendo a base da engenharia de software os processos, definindo tecnologias para o desenvolvimento de software de forma racional, utilizando metodologias efetivas e também constituindo-se de todo gerenciamento do projeto. Os métodos são informações técnicas e princípios que governam as diversas áreas da tecnologia, incluindo suas atividades e restrições.

## **2.1. Processo de software**

Processos de software são atividades utilizadas independentemente do tamanho ou complexidade do projeto. Segundo Sommerville (2011) esses processos são um conjunto de atividades que produzem um software, no entanto, também se aplica à extensões e modificações em sistemas em produção. Os processos de software são versáteis, mas todos devem incluir quatro atividades essenciais, dentre elas: Especificação de software: Elicitação de requisitos, definição das funcionalidades e restrições de funcionamento; Projeto e implementação de software: Modelagem, definição de arquitetura de sistema e implementação do software; Validação de software: Validação do software através de testes e pelo uso cliente; Evolução de software: Melhoria contínua no software a fim de atender necessidades de mudança do cliente.

Pressman (2016), por outro lado, descreve um conjunto de atividades que estabelecem o alicerce para um processo de software, sendo aplicáveis cinco atividades genéricas: Comunicação: Previamente, antes que qualquer tipo de tarefa, deve haver uma comunicação com o cliente, com a intenção de entender o objetivo do software;

Planejamento: O planejamento é uma espécie de mapa que serve para guiar a equipe, este mapa é denominado plano de projeto de software, este mapa descreve tarefas técnicas, riscos, recursos e produtos que devem ser resultados de um cronograma de trabalho; Modelagem: A modelagem é um esboço, o qual serve para dar uma ideia geral do projeto, geralmente é explícita em forma de modelos utilizados na engenharia de software; Construção: Atividade que se baseia em construir o software através dos modelos criados no processo anterior; Entrega: A entrega pode ser parcial ou completa e é avaliada pelo retorno do cliente.

Essas atividades citadas, de acordo com Sommerville (2011) podem ser utilizadas tanto para programas pequenos e simples, como para programas complexos. Essas atividades servem como base para a criação dos modelos de processo de software que serão abordados na próxima seção.

## 2.2. Modelos de processo de software

Segundo Sommerville (2011), modelos de processo de software, também chamados de paradigmas de processo ou *frameworks* de processo, são representações simplificadas do funcionamento dos processos de software. Cada modelo contém sua particularidade e sequência de atividades.

Um exemplo de modelo utilizado é o ciclo de desenvolvimento incremental, que segundo Sommerville (2011) atua como parte fundamental das abordagens ágeis, sendo também mais performático que o modelo cascata para a maioria dos tipos de sistema. Esse modelo se utiliza das atividades de especificação, desenvolvimento e validação, gerando a cada iteração das três atividades um incremento ao software, esse incremento devendo ser uma funcionalidade.

## 2.3. Engenharia de requisitos

A atividade de identificar, compreender e restringir os requisitos do sistema se encaixa na atividade de especificação do software, também chamada de engenharia de requisitos, essa atividade tem como objetivo principal obter um documento referente aos requisitos acordados com os clientes, a Figura 2 demonstra o fluxograma de atividades do processo.

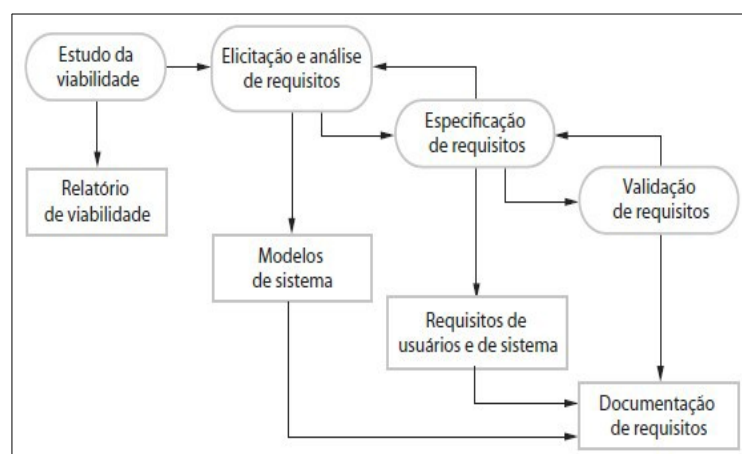


Figura 2. Processo de engenharia de requisitos.  
SOMMERVILLE (2011, p. 24)

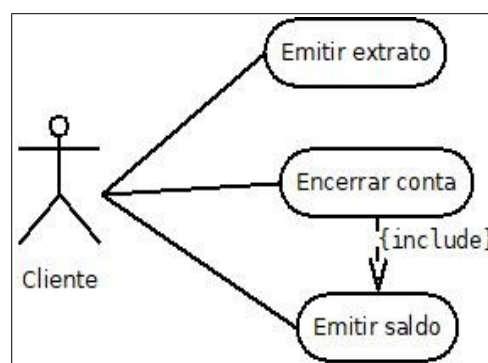
Segundo Sommerville (2011) a engenharia de requisitos emprega quatro atividades preponderantes, dentre elas: Estudo de viabilidade: Fase em que se identifica a possibilidade de cumprir as expectativas do usuário utilizando recursos computacionais, devendo ser um estudo de viabilidade rápido e de baixo custo; Elicitação e análise de requisitos: Fase na qual se utiliza diversas técnicas para elicitação dos requisitos como, análise de cenário, aplicação de questionários, *brainstorming*, entrevista, prototipagem, etc; Especificação de requisitos: Fase em que se refina os dados obtidos na fase anterior, gerando dois tipos de requisitos, funcionais e não funcionais, geralmente com informações abstratas, nesta fase deve haver uma maior especificação dos mesmos.; Validação de requisitos: Fase em que se analisa os requisitos especificados, validando quanto a possibilidade de sua aplicação, além de que nesta fase deve identificar os possíveis erros que existam no documento de requisitos.

Sommerville (2011) ainda argumenta que essa atividade não é feita apenas uma vez, podendo surgir a necessidade de novos requisitos durante o projeto, portanto esse processo pode se repetir de forma intercalada aplicando a ideia de melhoria contínua nos processos e no software.

Os resultados gerados pelo processo de engenharia de requisitos são documentos que irão ser utilizados pela fase de modelagem e projeto que será citado na próxima seção.

## 2.4. Modelagem de software

A modelagem de software contribui para obter uma visão do sistema de uma forma abstrata, descrevendo aspectos estruturais e comportamentais, segundo Guedes (2011) os modelos descrevem aspectos físicos do sistema, com um nível apropriado de detalhe. O autor ainda argumenta que por mais simples que o projeto seja, deve-se modelar antes de iniciar sua implementação, pois qualquer projeto tem perspectivas de crescimento e é necessário que exista uma documentação de sua estrutura.

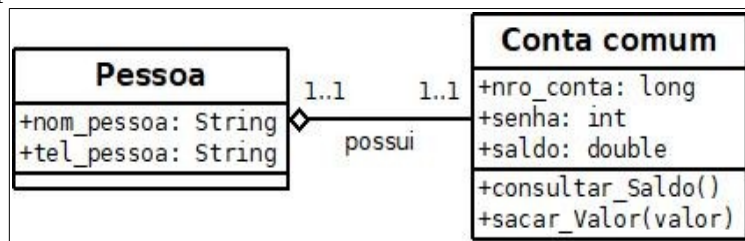


**Figura 3. Exemplo de diagrama de casos de uso de um sistema bancário. Adaptado de GUEDES (2011, p. 31)**

A UML ou Linguagem de Modelagem Unificada é uma linguagem visual utilizada na modelagem de software, segundo Guedes (2011) a UML se tornou a linguagem padrão para modelagem de software. Essa notação tem como objetivo auxiliar engenheiros de software a representarem graficamente a estrutura de sistemas. A linguagem contém um grupo de diagramas com objetivos de demonstrar múltiplas visões do sistema, segundo o autor cada diagrama analisa o sistema, ou uma parcela dele. Os diagramas de caso de uso é utilizado durante a fase de levantamento e análise

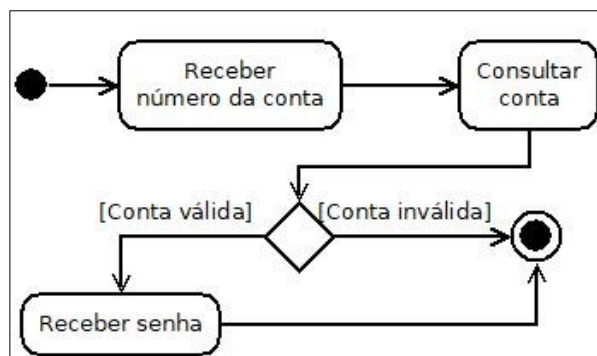
de requisitos, o autor também afirma que este diagrama serve para representar atores (usuários do sistema ou outros sistemas), serviços e funcionalidades.

Como exemplo, a Figura 3 demonstra um diagrama de casos de uso de um banco, sendo seus atores, clientes e funcionários representados pela figura simples e rudimentar da anatomia de um ser humano, já suas funcionalidades são representadas através de elipses.



**Figura 4. Exemplo de diagrama de classes de um sistema bancário. Adaptado de GUEDES (2011, p. 32)**

O diagrama de classes segundo Guedes (2011) é o mais utilizado e um dos mais importantes, servindo de apoio à orientação a objetos, este diagrama tem a função de representar a estrutura das classes, seus atributos, métodos e suas respectivas relações e trocas de informações. A Figura 4 representa um modelo de classes, construído para um sistema bancário, como pode ser observado serve para complementar o diagrama de casos de uso exibido na Figura 3.

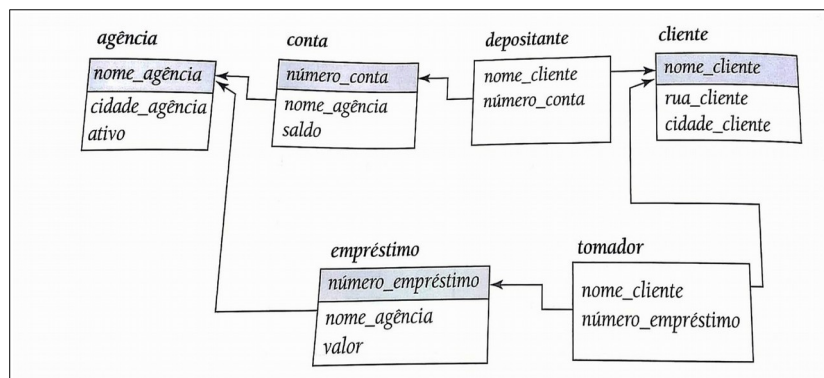


**Figura 5. Exemplo de diagrama de atividades de um sistema bancário. Adaptado de GUEDES (2011, p. 37)**

O diagrama de atividades, apresentado na Figura 5 representa os passos percorridos até a conclusão de uma atividade. A partir destes quatro tipos de modelagem já é possível ter uma visão abstrata do funcionamento de um software, sendo assim possível implementá-lo.

## 2.5. Modelagem de banco de dados

A modelagem de banco de dados, segundo Date (2003) tem como motivação o problema de compreensão limitada do que significam os dados. A abordagem é feita através da identificação de conceitos que sejam úteis quando se trata do mundo real. O autor relaciona com o mundo real, que é composto de entidades e que as mesmas podem ser classificadas com seus respectivos tipos. Estas entidades ainda podem ser distintas entre si a partir de propriedades atributivas e se relacionarem entre outras entidades através de relacionamentos.



**Figura 6. Exemplo de representação de um modelo relacional.**  
**SILBERSCHATZ (2006, p. 30)**

Silberschatz (2006) define os modelos relacionais como os principais modelos utilizados para facilitar o trabalho do programador, comparando-o com outros modelos. O autor também afirma que os modelos relacionais consistem em uma coleção de tabelas nomeadas individualmente, sendo suas colunas as propriedades e cada linha da tabela um dado inserido. Um exemplo de modelo relacional criado para uma agência bancária pode ser visto na Figura 6.

## 2.6. HTML

O HTML (*HyperText Markup Language* – Linguagem de Marcação de Hipertexto) é a linguagem mais básica da web. Segundo a Documentação do HTML (2018), ele serve para definir conteúdos de páginas web e tem suas funcionalidades complementadas pelo CSS (*Cascading Style Sheets* – Folhas de estilos em cascata) e pelo JavaScript. A Documentação também cita que o HTML utiliza marcações para envolver textos e imagens, facilitando o entendimento e a renderização da página no navegador.

Para aprofundar os conhecimentos em HTML é recomendado ler a documentação em <https://developer.mozilla.org/pt-BR/docs/Web/HTML>.

## 2.7. CSS

O CSS é uma linguagem de estilo utilizada para complementar apresentação de documentos HTML, XML, SVG ou XHTML. Segundo a documentação do CSS, ele é uma das principais linguagem da web, sendo especificado pela W3C.

Para aprofundar conhecimento em CSS é recomendado ler a documentação no site <https://developer.mozilla.org/pt-BR/docs/Web/CSS>.

## 2.8. Javascript

JavaScript ou de forma abreviada JS, é uma linguagem leve e interpretada, segundo a Documentação do JavaScript (2018) ela é utilizada em navegadores web e outros ambientes fora deles, como node.js, Apache CouchDB e Adobe Acrobat, sendo uma linguagem script de multiparadigma que suporta estilos de programação orientados a objetos, imperativos e funcionais.

O JavaScript é comumente utilizado no lado do cliente da web, ou seja, os navegadores. Porém, atualmente, sua aplicação se voltou para o lado servidor com node.js. A Documentação ainda afirma que é uma linguagem fácil de se aprender, porém poderosa, sendo muito utilizada para adaptar comportamentos de páginas web

dinâmicas.

Para se aprofundar nos recursos disponíveis na linguagem JavaScript acesse o site da documentação no site <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>.

## **2.9. PHP**

O PHP é uma linguagem muito utilizada no desenvolvimento web, segundo a Documentação do PHP (2018) é um código utilizado no servidor, gerando HTML para enviar como resposta para o navegador. A documentação também afirma que ele é extremamente simples para níveis iniciantes, mas oferece muitos recursos a níveis avançados. Apesar do foco em *scripts* do lado servidor o PHP oferece mais funcionalidades.

Além de atuar como interpretador de código em servidores web, pode ser utilizado em scripts de linha de comando e até para desenvolvimento de aplicações desktop com o PHP-GTK (PHP-GIMP *Toolkit*), que disponibiliza ferramentas para criação de ferramentas multiplataformas. A vantagem de utilizar a linguagem é a gama de sistemas operacionais compatíveis, segundo a Documentação do PHP, os sistemas Linux, sistemas baseados em Unix, sistemas da Microsoft, Mac OS, RISC OS e provavelmente outros.

Para se aprofundar mais nas funcionalidades acesse a documentação completa da linguagem PHP disponível no site <https://secure.php.net/docs.php>.

## **2.10. MySQL**

Segundo a Documentação do MySQL (2018), o MySQL é a plataforma de banco de dados de código aberto mais confiável em uso atualmente. Isso é alcançado através da rápida entrega, das multithreads, acessos multiusuários e sua robusta linguagem *Structured Query Language* (SQL).

Para aprofundar os conhecimentos nas funcionalidades e serviços do MySQL acesse o site <https://dev.mysql.com/doc/refman/8.0/en/introduction.html>.

## **2.11. Bootstrap**

O Bootstrap é utilizado para criar projetos web responsivos através de componentes *front-end*. É uma ferramenta gratuita de desenvolvimento HTML, CSS e Javascript. Em sua documentação, que é encontrada no site <http://getbootstrap.com.br/docs/4.1/getting-started/introduction/> demonstra suas funcionalidades e componentes que podem ser inclusos em seus projetos, tornando o desenvolvimento mais ágil, tendo que os elementos gerador a partir do padrão são adaptativos a diferentes tamanhos de tela.

# **3. Educação e tecnologia**

O uso da tecnologia busca o aumento da eficiência em atividades humanas, principalmente nas que necessitam produtividade. As TICs criaram uma atração no meio educacional, criando novas possibilidades que diminuíram aspectos de espaço e distância. Porém, Bergamasco (2013) afirma que o processo de inclusão de tecnologias no âmbito educacional foi muito conturbado, levantando questionamentos sobre a eficiência e a prioridade de sua inclusão no ensino. Por esse motivo a UNESCO (2013 apud BERGAMASCO, 2013) declara que o Brasil precisa investir na utilização de tecnologias por professores.



Segundo Costa (2004) o uso da informática na educação vem sendo objeto de estudo e pesquisa gerando várias discussões. Devido a isso as primeiras iniciativas no Brasil foram adotadas na década de setenta, quando se começou a discutir a importância do computador, tendo como base os modelos americanos de informática na educação. As justificativas para a introdução da informática na educação são diversas e dependem da forma de utilização, tanto na interação aluno-professor como na utilização para apoio a tarefas pedagógicas.

Silva (2014) argumenta que antes do surgimento tecnológico havia uma busca pelo aumento dos conhecimentos produzidos em sala de aula, a tecnologia por si só teve uma contribuição no desenvolvimento do país e na capacidade humana de expandir o conhecimento. O autor também afirma que com a facilidade obtida através das tecnologias, se pode aprimorar métodos de educação arcaicos, visto que devemos acompanhar a evolução e adaptarmos os métodos de ensino a realidade atual.

A revolução tecnológica surte efeito dentro das salas de aula, Moreira (2007) assegura que os gestores e professores devem encarar os desafios trazidos pelo uso das TICs nas escolas, pois há nelas uma capacidade de agregar mudanças significativas nas práticas escolares, tanto dos docentes como dos discentes. Há um desejo de que o professor invista em atualização nessas novas tecnologias para aplicar esses princípios que visam reinventar a escola, principalmente com foco em aumentar a qualidade do ensino.

Segundo Oliveira (2013) as TICs tem grande influência em organizações como a escola, a adaptação do indivíduo na sociedade, requer de forma fundamental o conhecimento de mecanismos que a tecnologia impõem. Os professores, sendo principais mediadores da aprendizagem do aluno, devem se inteirar da evolução tecnológica a fins de repassar este conhecimento. Além do professor, é indispensável que as instituições invistam em equipamentos necessários, propiciando acesso a alunos e professores.

Silva (2014) reconhece que o educador e a escola ao compreenderem os aspectos atuais da sociedade podem facilitar no crescimento educacional, modernizando os ambientes escolares e utilizando tecnologias, criando uma esfera que facilite a busca de informações. Do ponto de vista pedagógico, a formação do docente é essencial, pois ele é quem guiará o aluno na utilização das TICs no aprendizado. Essa formação deve ser focada em instruir o professor, para que se capacite a ensinar o uso da tecnologia e ensine usar a tecnologia para aperfeiçoar os conhecimentos.

Como visto durante esta seção, há diversos problemas a serem resolvidos para que haja uma integração de tecnologias na educação, dentre eles, a resolução de problemas de infraestrutura, capacitação de funcionários para que se tenha uma boa monitoria e manutenção dos equipamentos, capacitação dos docentes para uso da tecnologia e estudos sobre o uso da tecnologia em ambientes de aprendizagem.

### **3.1. Trabalhos relacionados**

Os Ambientes Virtuais de Aprendizagens (AVAs) estão sendo utilizados pelo sistema de educação como forma de apoio e suporte às atividades docentes. Ao serem utilizados, dão suporte à aprendizagem presencial, aquela que ocorre em sala de aula, por apresentarem inúmeras possibilidades de concretizar o processo de ensino e aprendizagem do educando. Dentre eles encontram-se disponíveis algumas ferramentas como: Moodle, TelEduc e Tidia-AE, que serão descritas nessa seção.

O Moodle, dessa forma, oferece ao professor, a possibilidade do gerenciamento de um curso à distância, provendo o planejamento, implementação e gestão do aprendizado à distância, permitindo inclusive o uso em cursos semipresenciais ou para a publicação de materiais que complementem os cursos presenciais. As vantagens da utilização desse tipo de ferramenta são inúmeras, se comparadas aos recursos que a maioria dos professores utiliza em sala de aula.

O Moodle, dentre muitos aspectos positivos, proporciona ao educador, por exemplo: maior interação entre os alunos, por ocasião da criação de fóruns para discussão; o professor pode utilizar de vários recursos de aprendizagem, para o mesmo conteúdo, que ficará disponível para o aluno acessá-lo quando assim quiser; facilita o lançamento das notas, além de oferecer melhor controle sobre entrega das atividades exigidas; aulas complementares podem ser disponibilizadas, reforçando o conteúdo; os trabalhos, individuais ou em grupos podem ser gravados e enviados para avaliação; o chat do moodle proporciona ao aluno a tranquilidade de sanar dúvidas em tempo real.

Outra ferramenta para ensino a distância é o TelEduc, pelo qual é possível realizar cursos com a utilização da internet. O TelEduc, segundo Ribeiro e Mendonça (2007 apud FRANCISCATO et al, 2008, p. 3), é uma plataforma *Open Source*, um ambiente de suporte EaD. O seu desenvolvimento é realizado de acordo com as necessidades, tanto tecnológico como metodológicas, por desenvolvedores do Núcleo de Informática Aplicada à Educação (NIED) da Universidade Estadual de Campinas (UNICAMP). Em termos de apresentação, o ambiente TelEduc é flexível e está dividido em duas partes: as ferramentas e o conteúdo correspondente à ferramenta selecionada.

Amorim (2005) destaca alguns dos recursos disponíveis para alunos e educadores, destacam-se os seguintes: Disponibilização das atividades a serem realizadas durante o curso; Disponibilização de material de apoio; Perguntas frequentes relacionadas ao curso; Correio eletrônico interno; Criação de grupos de discussão. Para os formadores, essa ferramenta apresenta: Intermap: permitindo a visualização e a interação dos participantes do curso; Administração: permite aos formadores disponibilizar materiais nas diversas ferramentas do ambiente, tendo a possibilidade, ainda de configurar opções em algumas delas.

O Tidia-AE (Tecnologia da Informação no Desenvolvimento da Internet Avançada – Aprendizado Eletrônico) vem em auxílio ao ensino presencial e eletrônico, apresentando um ambiente colaborativo que gerencia cursos e atividades de aprendizagem, ele reúne ferramentas de software desenvolvidas especialmente para auxiliar os usuários (alunos, professores, pesquisadores...) que usando um navegador da web criam um portal que reúne as necessidades de aprendizado através das ferramentas disponíveis.

Segundo Gomes et al (2018) o Tidia-AE surgiu com o objetivo auxiliar na pesquisa e no desenvolvimento de ferramentas de apoio ao ensino e aprendizagem, tendo como ferramentas disponíveis tanto para administração, coordenação e comunicação, o ambiente também permite que o educador crie cursos e o gerencie de forma colaborativa. Além do perfil pessoal, apresenta a possibilidade da visualização de uma agenda compartilhada; interação entre os usuários; realização de testes; disponibilizar e compartilhar conteúdos didáticos e atividades, entre outros.

Segundo Santos Junior (2011) a utilização de ferramentas da web fornece alguns benefícios para o educador, na visão de um pesquisador, são: Suporte a diferentes tipos de aprendizagem: cooperativa, orientada por discussão, centrada no sujeito, por projetos,

por desafios/problemas/casos; Suporte para desenvolvimento de práticas pedagógicas interdisciplinares e transdisciplinares; Possibilita disseminar informação para um grupo de pessoas.

Dentre os três trabalhos citados, se destacam as mediações pedagógicas, ou seja, foco no relacionamento professor-aluno, criando funcionalidades que oferecem ambientes de interação, compartilhamento de materiais e controle de entrega de atividades. Estas ferramentas se adaptam muito bem em modalidades de ensino EaD e semipresenciais.

A relação destes trabalhos com o aqui proposto é o auxílio a educação, porém a abordagem é referente as atividades do docente, automatizando e dando agilidade em tarefas repetitivas, para que preste um trabalho de qualidade aos alunos, assim disponibilizando mais tempo de atuação direta professor-aluno, criando benefícios ao docente, mas também aos alunos. Partindo dessa ideia a aplicação de um questionário para identificar as atividades que possam ser automatizadas, se torna favorável, dando valia da ideia aqui apresentada.

#### **4. Desenvolvimento do projeto**

O desenvolvimento do projeto se baseou em cinco etapas, são elas: levantamento de requisitos, definição do escopo, modelagem, implementação e testes. Essas etapas serão descritas nas seções seguintes.

##### **4.1. Levantamento de requisitos**

A primeira etapa aplicada no desenvolvimento do projeto foi o levantamento de requisitos, nessa fase foi criado um questionário contendo alguns requisitos base para o respondente indicar se a funcionalidade seria útil ou não no seu dia a dia.

Também foi disponibilizado campos de texto aberto para que o respondente escrevesse sua ideia de funcionalidade caso houvesse e também um campo de questionamento sobre o conhecimento de tais funcionalidades em algum software já existente.

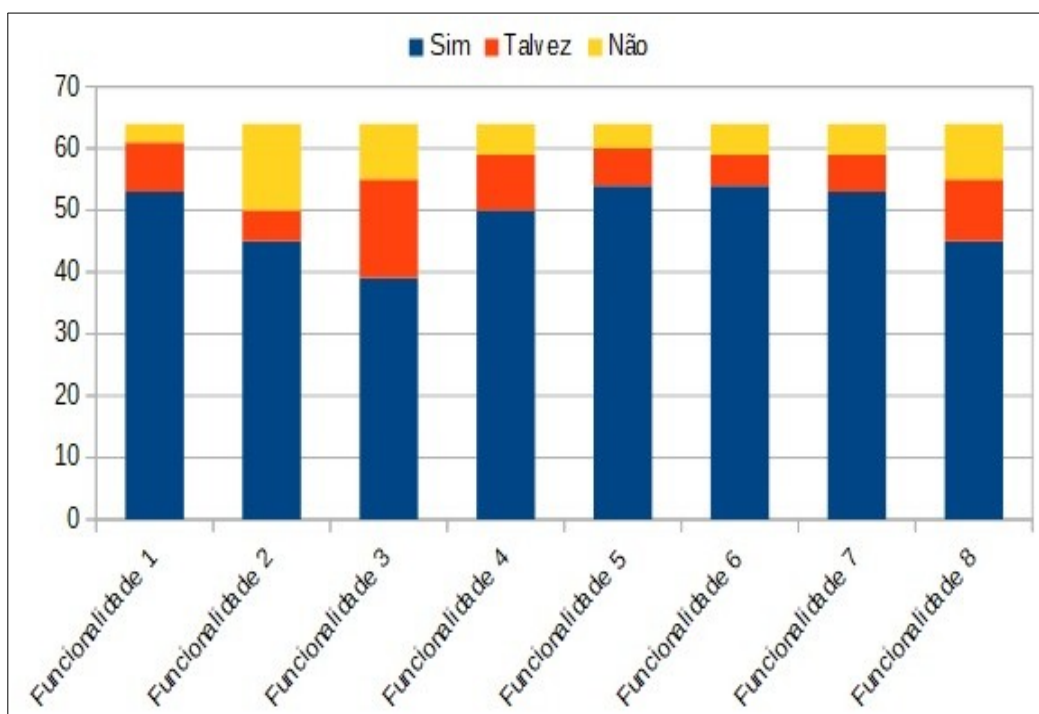
Este questionário foi aplicado em dois meios, eletrônico e físico. O meio eletrônico foi aplicado utilizando uma ferramenta do Google chamada *Google Forms*, que possibilita a criação de questionários e facilita a distribuição e apuração das respostas na internet. Já o meio físico foi aplicado em uma escola localizada em São Borja, chamada Centro Escolar de Turno Integral Municipal – CETIM.

O questionário foi aplicado com o seguinte título: Questionário de funcionalidades para um software de auxílio ao docente. Este questionário tem como finalidade identificar funcionalidades que facilitariam as atividades diárias de um docente.

As questões aplicadas e as respostas obtidas serão dispostas sequencialmente:

- Funcionalidade 1 – Banco de questões, utilizado para armazenar questões de provas e revisões;
- Funcionalidade 2 – Geração de provas de forma automatizada a partir das questões armazenadas no banco de questões;
- Funcionalidade 3 – Compartilhamento de questões e provas com outros

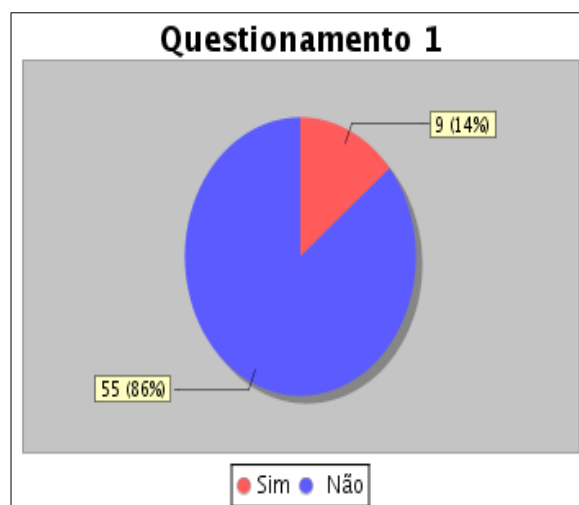
professores que utilizem a ferramenta;



**Figura 7. Respostas de respostas coletadas divididas por funcionalidades. Criado pelo Autor.**

- Funcionalidade 4 – Controle de classe (Gerenciamento de alunos, notas, presença);
- Funcionalidade 5 – Aplicativo para dispositivos móveis, que a partir dos dados cadastrados no controle de classe, disponibilize uma ferramenta offline (Sem necessidade de internet) para controle de presença em sala de aula;
- Funcionalidade 6 – Banco de arquivos, no qual o docente armazenaria o que lhe fosse conveniente;
- Funcionalidade 7 – Registro de tarefas (Provas, reuniões e compromissos em geral) com lembrete via e-mail;
- Funcionalidade 8 – Registro de plano de aula com geração de relatórios.
- Questionamento 1 – Você conhece uma ferramenta que disponibilize alguma dessas funcionalidades?

Os resultados dos questionários são apresentados em forma de gráfico na Figura 7 e Figura 8, sendo elas respectivamente, funcionalidade 1 a 8 e questionamento 1. A aplicação do questionário foi efetuada entre o período de abril e maio de 2018, o total contabilizado foram 64 respostas, sendo 10 coletadas por meio físico e 54 pelo eletrônico. As respostas coletadas serviram para dar embasamento nas decisões da segunda etapa, que será descrita na próxima seção.



**Figura 8. Gráfico de respostas coletadas do Questionamento 1. Criado pelo autor.**

#### 4.2. Definição do escopo

A segunda fase do desenvolvimento teve como objetivo definir o escopo, ou seja, quais funcionalidades seriam implementadas, essa etapa foi aplicada entre junho e julho de 2018. Com as respostas do questionário foi possível verificar a necessidade dos professores, buscando identificar quais se adaptariam melhor ao objetivo de auxiliar o docente.

Outro ponto levado em questão foi a pesquisa realizada pelo Núcleo de Informação e Coordenação do Ponto BR (NIC.br) onde foram levantados dados sobre os recursos providos pelas TICs que são utilizados pelos professores. Os dados coletados nessa pesquisa podem ser vistos na Figura 9.



**Figura 9. Pesquisa de recursos usados por professores. Adaptado de NIC.br (2015, pg. 155)**

Após as observações foram definidos que as seguintes funcionalidades seriam desenvolvidas:

- O acesso ao sistema deve ser restrito a usuários cadastrados;
- Banco de arquivos, aonde o usuário possa armazenar seus arquivos;
- Controle de classe, aonde o usuário possa cadastrar seus alunos, turmas e possibilite registrar presença e notas;
- Banco de questões colaborativo, com compartilhamento de questões entre os usuários.

Os requisitos funcionais e não funcionais serão descritos na Tabela 1 e na Tabela 2 que seguem no texto:

**Tabela 1. Requisitos não funcionais**

<b>Requisitos não funcionais</b>
O sistema deve funcionar na plataforma web.
O sistema deve ser responsivo.
O sistema deve restringir o acesso somente para usuários que tenham criado conta.

**Tabela 2. Requisitos funcionais**

<b>Requisitos funcionais</b>
O sistema deve disponibilizar uma forma para que o usuário se cadastre utilizando e-mail e senha.
O sistema deve disponibilizar uma forma que caso o usuário esqueça a senha, possa recuperá-la via e-mail.
O sistema deve permitir que o usuário gerencie as suas categorias para os arquivos (Cadastro, edição e exclusão)
O sistema deve permitir que o usuário gerencie os seus arquivos (Cadastro, edição, exclusão e <i>uploads</i> ).
O sistema deve permitir que o usuário gerencie seus alunos. (Criação, edição, exclusão)
O sistema deve permitir que o usuário gerencie suas turmas (Cadastro, edição, exclusão).
O sistema deve permitir que o usuário vincule alunos às suas turmas, também deve conter uma forma de desvincular o aluno.
O sistema deve permitir que o usuário defina a quantidade de avaliações que sua turma terá, mínimo uma, máximo três na criação da turma, sendo posteriormente bloqueado a alteração dessa informação.
O sistema deve permitir que o usuário defina presença/falta dos alunos vinculados a uma turma em determinada data.
O sistema deve ter pré-cadastrado ciências, disciplinas e conteúdos até o terceiro nível hierárquico da classificação do CNPQ, para uso na classificação das questões do banco de questões.
O sistema deve permitir que o usuário cadastre questões objetivas selecionando um conteúdo na qual a questão é relacionada e podendo inserir no máximo cinco alternativas, assinalando qual é a correta.
O sistema deve permitir que o usuário cadastre questões dissertativas selecionando um conteúdo na qual a questão é relacionada.

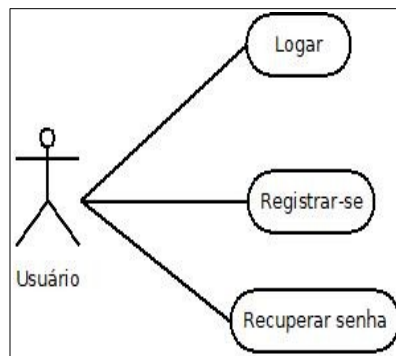
O sistema também deve permitir que o usuário altere e exclua questões criadas com o seu usuário.

O sistema deve exibir um *Feed* de questões que possibilite o usuário visualizar todas as questões cadastradas no sistema e disponibilize filtros de busca dentre os dados itens pré-cadastrados.

A definição do escopo gerou uma ideia básica sobre as funcionalidades que aqui foram especificadas, serão modeladas na etapa que será descrita na seção seguinte.

#### 4.3. Modelagem

Neste título serão descritas as funcionalidades, utilizando representações gráficas, como o Modelo Entidade Relacionamento e Diagramas UML, dentre os variados tipos de diagramas, foram escolhidos um do tipo estrutural, o de classe e dois comportamentais, caso de uso e de atividades.



**Figura 10. Caso de uso: Acesso ao sistema. Criado pelo Autor.**

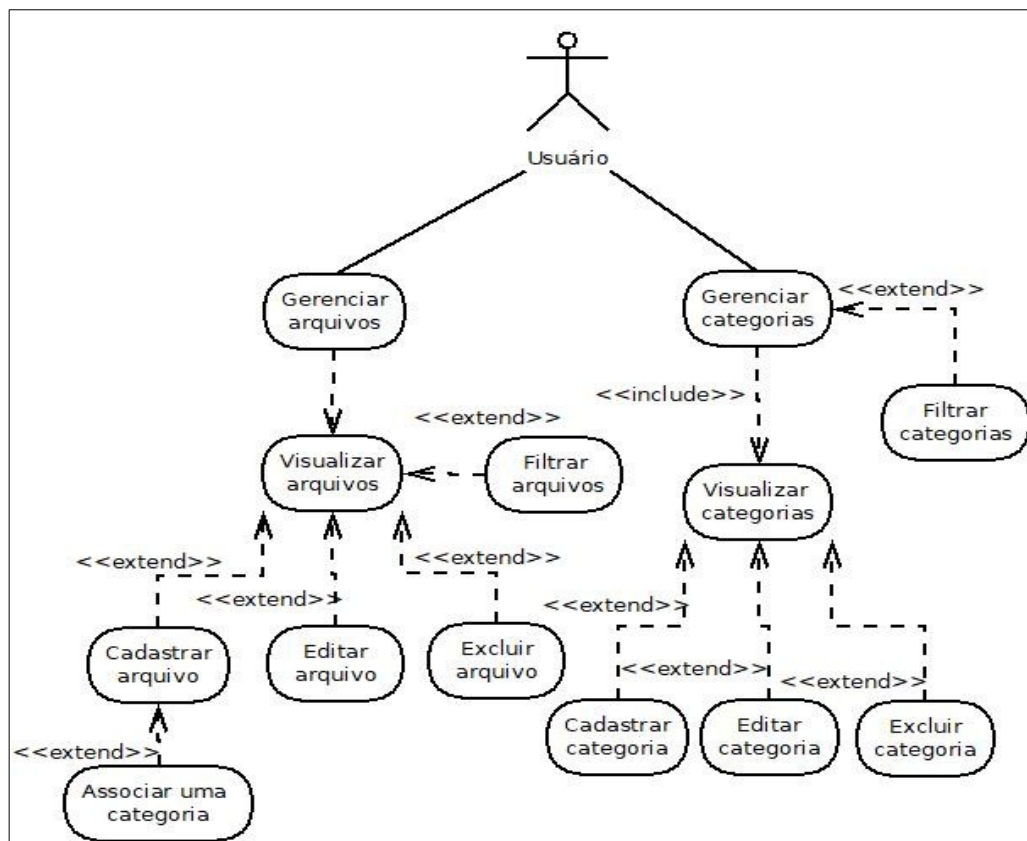
Para melhor disposição das imagens, foi definido que as imagens serão dispostas separadas por funcionalidades, visto que as ferramentas são modulares e há pouca ou quase nenhuma interferência entre uma e outra.

Na criação do modelo entidade relacionamento das funcionalidades foi utilizada a ferramenta *MySQL Workbench*. A notação das cardinalidades são representadas por losangos, sendo que os totalmente brancos representam relacionamentos um para um e os bicolores (preto e branco) um para n. Para a criação dos Diagramas UML foi utilizado um software chamado Dia.

Para restrição de acesso ao sistema será usado uma única tabela chamada usuário. Ela contém os dados que serão usados para validar o acesso no sistema e todas as funcionalidades do sistema são dependentes dessa tabela, partindo da ideia que serão criados dados derivando sua chave estrangeira.

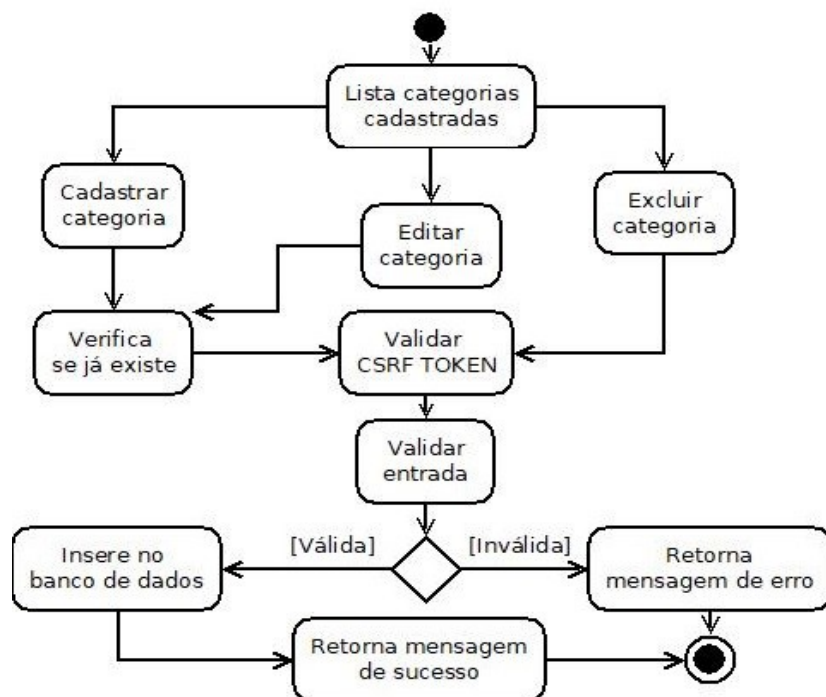
O caso de uso apresentado na Figura 10 representa as três etapas disponíveis, como logar, que é a inserção de dados válidos para acesso ao sistema; registrar-se, que disponibiliza a criação de uma conta para acesso ao sistema e recuperar senha que envia um link para alteração de senha para o e-mail do usuário em caso de esquecimento da senha.

As funcionalidades providas pelo módulo Banco de arquivos estão representadas no diagrama de caso de uso da Figura 11. Este caso de uso tem como pré-condição o login no sistema e possibilita o gerenciamento das categorias e a gerência de arquivos, das quais as funcionalidades são: Cadastro de categoria; Visualizar categorias; Filtrar



**Figura 11. Caso de uso: Banco de arquivos. Criado pelo Autor.**

pela descrição da categoria Edição de categoria; Exclusão de categoria; Cadastrar arquivo com a possibilidade de vinculá-lo com uma categoria ou não; Editar arquivo com a possibilidade de vinculá-lo com uma categoria ou não; Excluir arquivo.



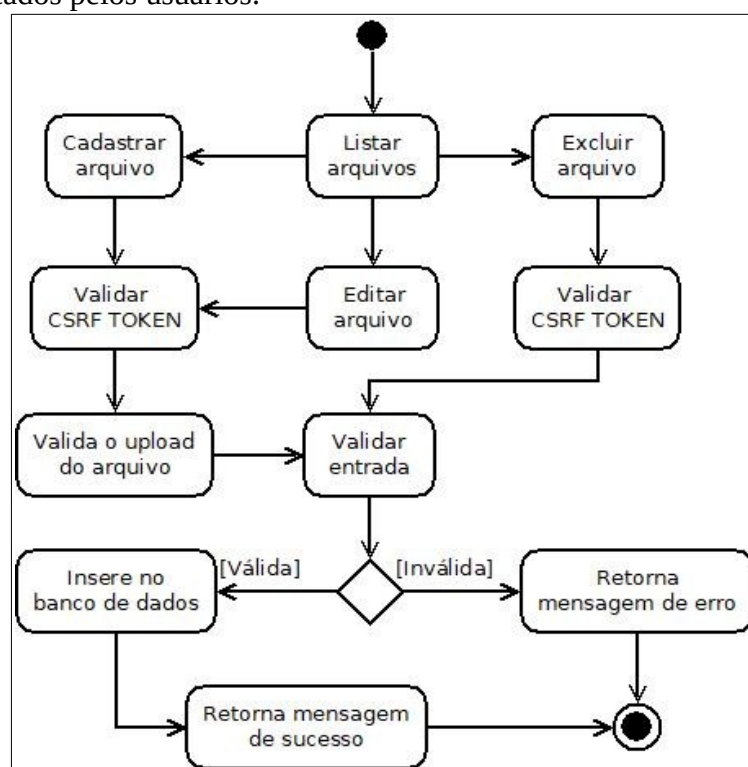
**Figura 12. Diagrama de atividades do gerenciamento de categorias do banco de arquivos. Criado pelo Autor.**



Essas categorias serão posteriormente usadas para cadastrar um arquivo no sistema, no qual deve obrigatoriamente vincular uma categoria ao arquivo, para fins de organização e busca textual.

A Figura 12 representa os fluxos de atividades disponíveis para gerenciar as categorias referentes ao módulo banco de arquivos, listar, filtrar, cadastrar, editar e excluir categorias.

Todos os fluxos de alteração de dados devem contar com a implementação de um *CSRF Token* para segurança dos dados, além de evitar dupla requisição em formulários aplicados pelos usuários.

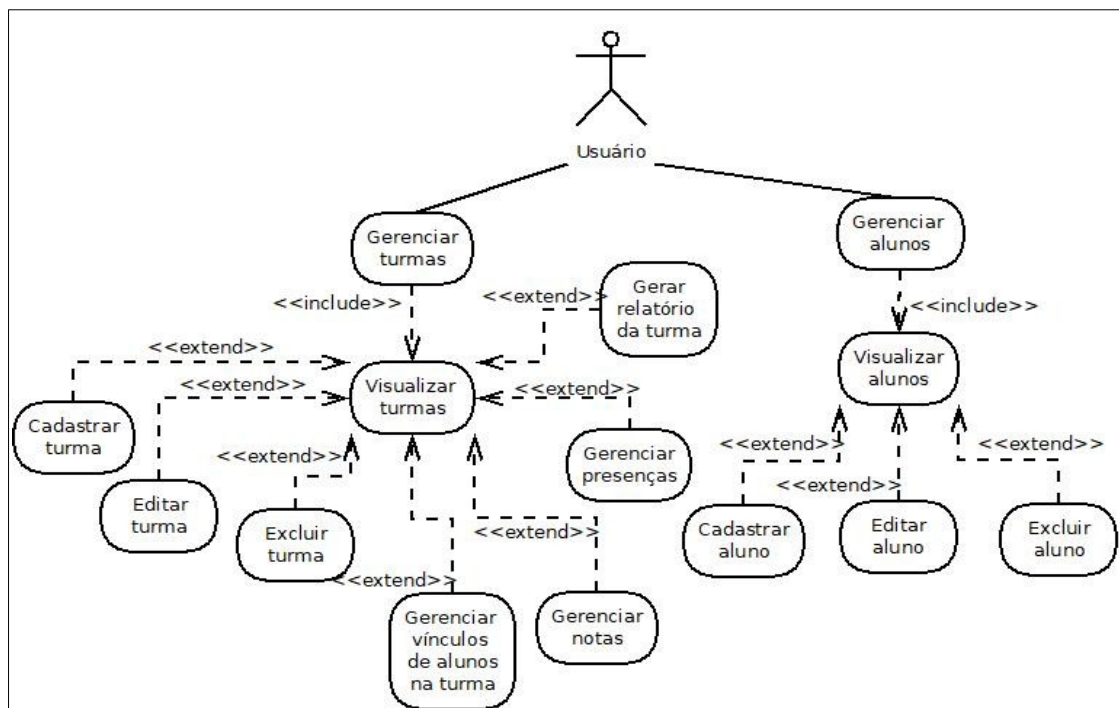


**Figura 13. Diagrama de atividades do gerenciamento de arquivos do banco de arquivos. Criado pelo Autor.**

A Figura 13 representa os possíveis fluxos de atividades disponíveis para gerenciar os arquivos referentes ao módulo banco de arquivos, listar, filtrar, cadastrar, editar e excluir arquivos.

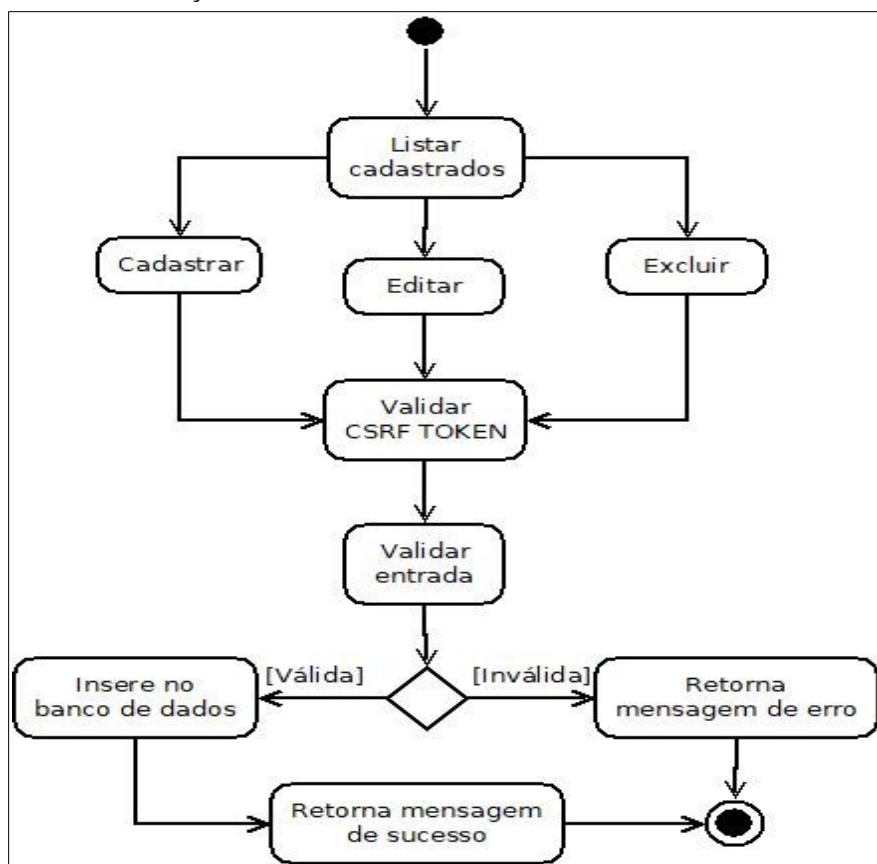
O módulo banco de arquivos conta com o uso de três tabelas contando a tabela relacionada com o acesso ao sistema, seguindo a seguinte lógica de que um usuário pode ter vários arquivos e várias categorias e um arquivo pode ter ou não uma categoria, ou seja, uma tabela categoria e uma tabela arquivo.

As funcionalidades providas pelo módulo Banco de questões estão representadas no caso de uso da Figura 14. Este caso de uso tem como pré-condição estar logado no sistema e possibilita o gerenciamento das categorias e a gerência de arquivos, das quais as funcionalidades são: Cadastro de aluno; Visualizar alunos; Filtrar aluno pelo nome do aluno; Edição de aluno; Exclusão de aluno; Cadastro de turma; Visualizar turma; Filtrar turma pela descrição da turma; Edição de turma; Exclusão de turma; Associar aluno a uma turma; Desvincular aluno de uma turma; Realizar diário de presença referente a um dia; Editar diário de presença referente a um dia; Excluir diário de presença referente a



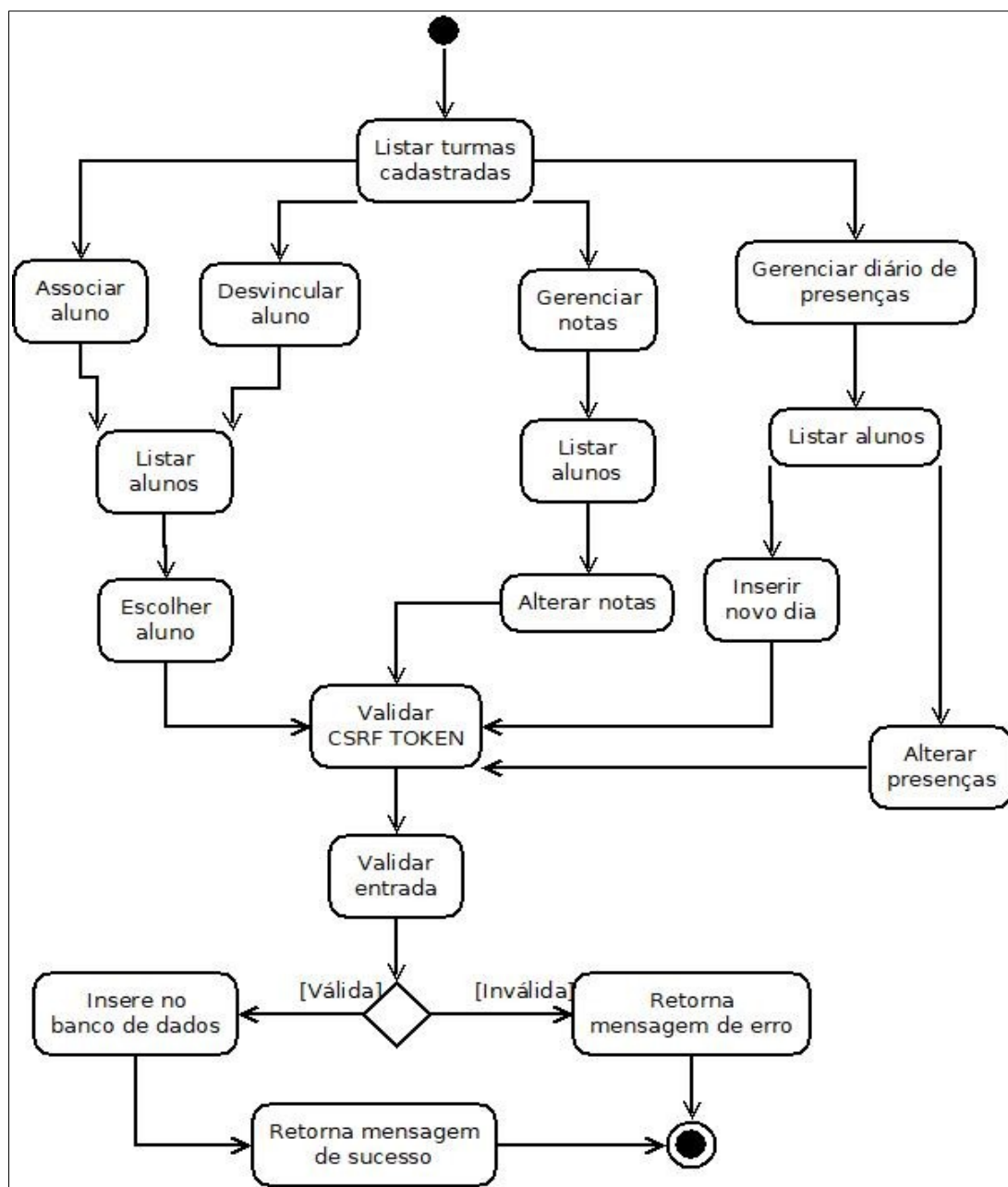
**Figura 14. Caso de uso: Controle de classe. Criado pelo Autor.**

um dia; Registrar notas de avaliações; Alterar notas de avaliações; Relatório da turma. Editar notas de avaliações; Gerar relatório da turma.



**Figura 15. Diagrama de atividades do gerenciamento de turmas e alunos do controle de classe. Criado pelo Autor.**

A Figura 15 demonstra o fluxo de atividades tanto para o gerenciamento de alunos do módulo controle de classe, como para o gerenciamento de turma, podendo ser dispostos no mesmo diagrama pela semelhança das operações básicas o cadastro, edição e exclusão, tendo como diferencial somente os dados inseridos.

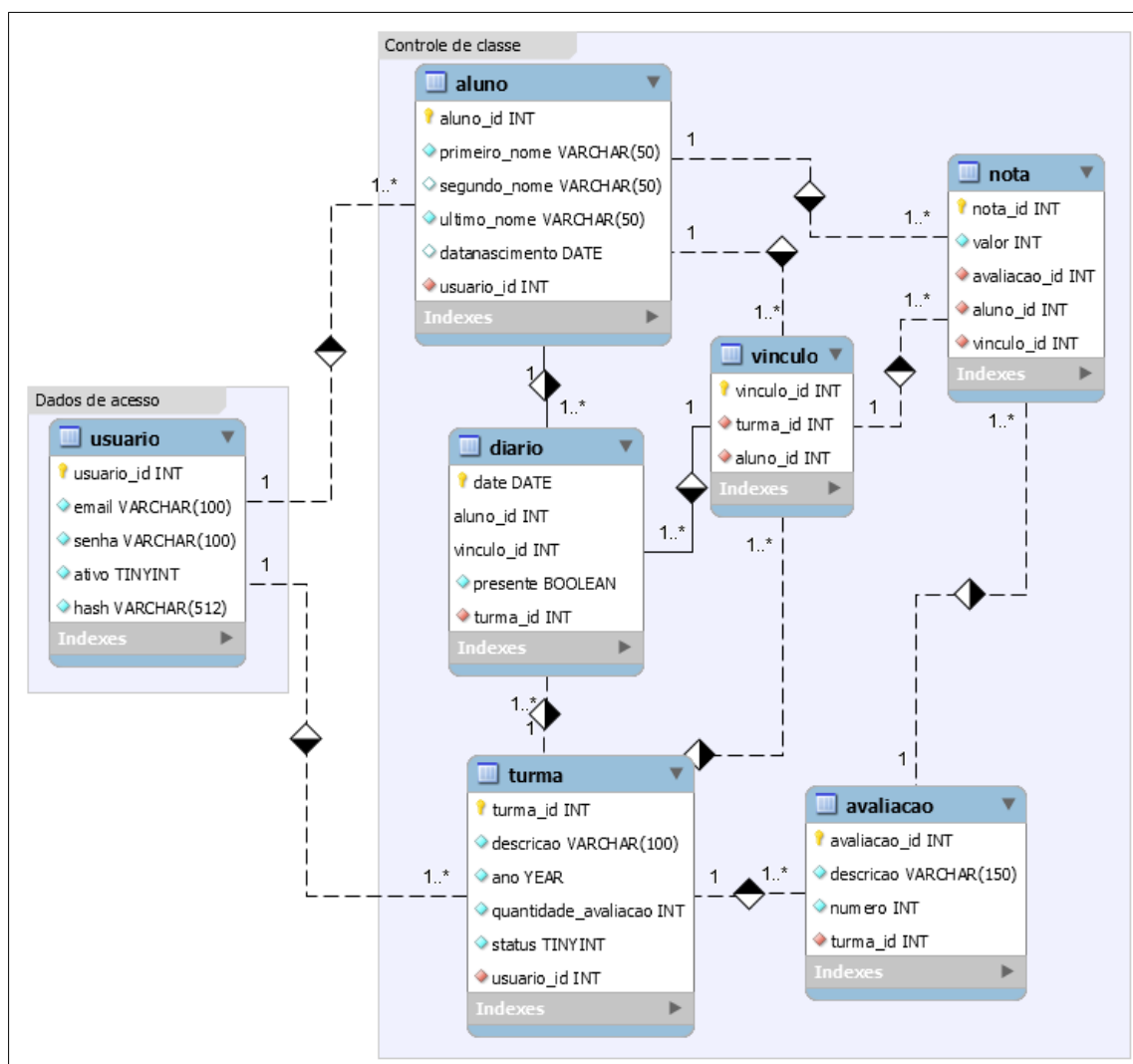


**Figura 16. Diagrama de atividades do gerenciamento do controle de classe. Criado pelo Autor.**

A Figura 16 demonstra o fluxo de atividades para efetuar o controle de classe dentro do módulo, sendo os processos básicos para o gerenciamento, associar um aluno, desvincular um aluno, alterar notas referentes a uma avaliação, inserir um novo diário de presença referente a um dia, alterar presenças de um dia já cadastrado.

O módulo Controle de Classe conta com o uso de 7 tabelas contando a tabela referente ao usuário do sistema, o modelo relacional deste módulo está disposto na

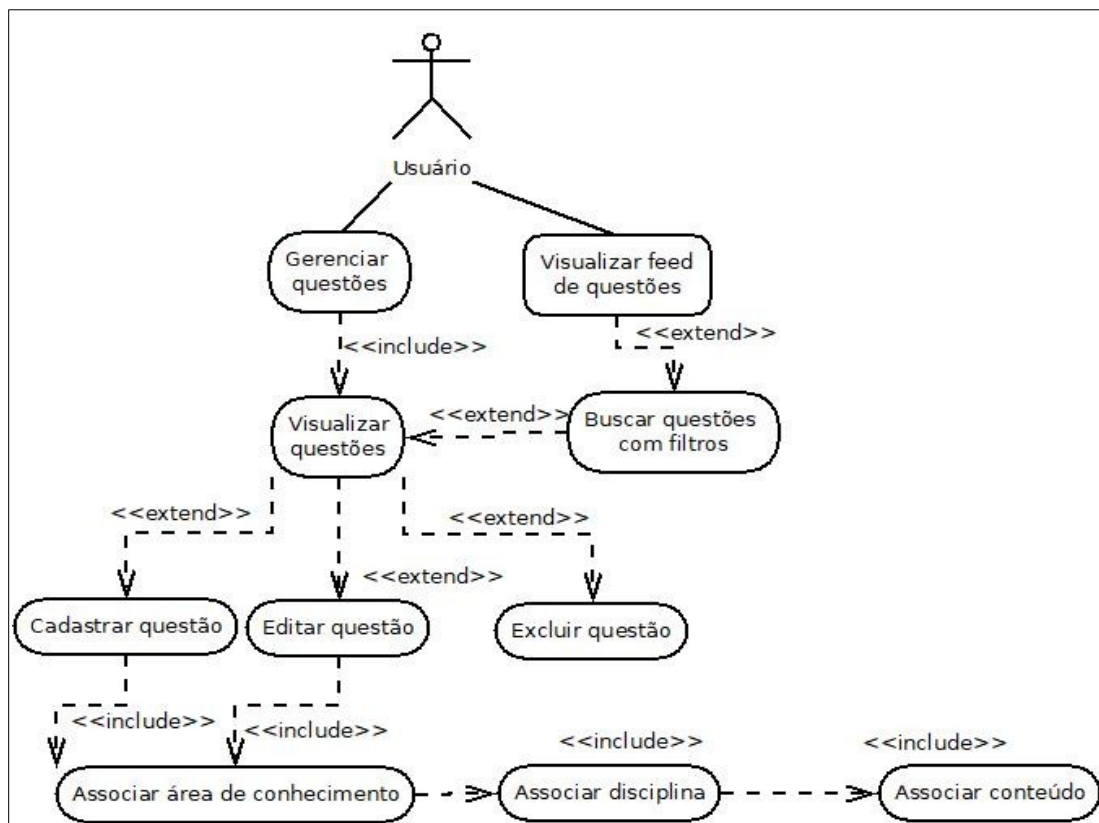
Figura 17 e representa a lógica entre as tabelas deste módulo.



**Figura 17. Modelo relacional do módulo controle de classe. Criado pelo Autor.**

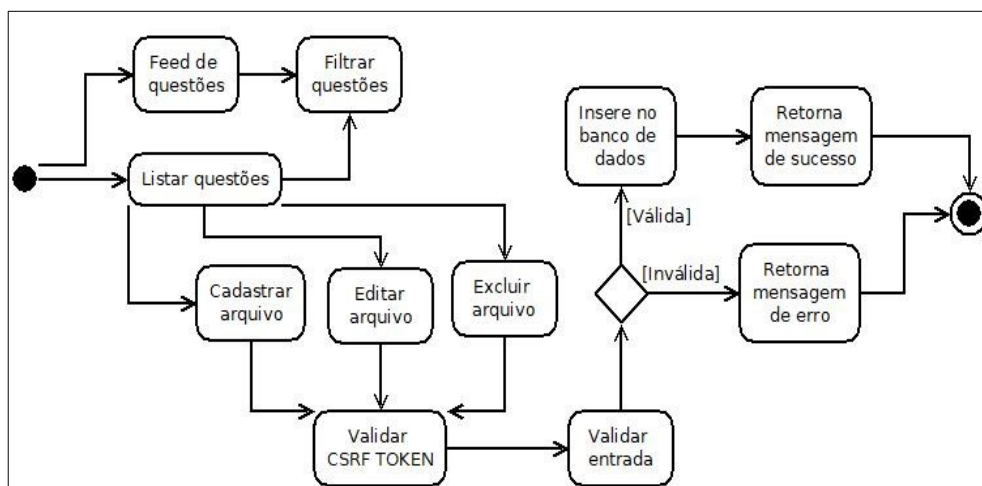
Neste esquema, um usuário do sistema pode ter vários alunos e turmas, sendo que os alunos podem ser vinculados a várias turmas. Uma turma pode ter de uma a três avaliações, seguindo a regra de negócio e cada avaliação terá várias notas referentes aos alunos da turma. Cada turma terá seus diários de presença referentes a cada dia de aula, sendo que cada aluno vinculado a turma deve ter presença ou falta no dia.

As funcionalidades providas pelo módulo Banco de questões estão representadas no diagrama de caso de uso da Figura 18. Este caso de uso tem como pré-condição o login no sistema e tem como funcionalidades: Cadastro de questão vinculando área de conhecimento e selecionando o tipo da questão, objetiva ou descritiva; Edição de questão, podendo alterar opções que foram cadastradas de forma incorreta, exceto o tipo da questão; Excluir questão; Visualização de todas as questões do sistema em um feed de questões; Filtro de busca para as questões, por área de conhecimento, disciplina e conteúdo. Esse módulo tem como objetivo facilitar a tarefa de criação de provas e possibilitar o compartilhamento de questões entre os usuários do sistema, visando facilitar a criação de provas e simulados.



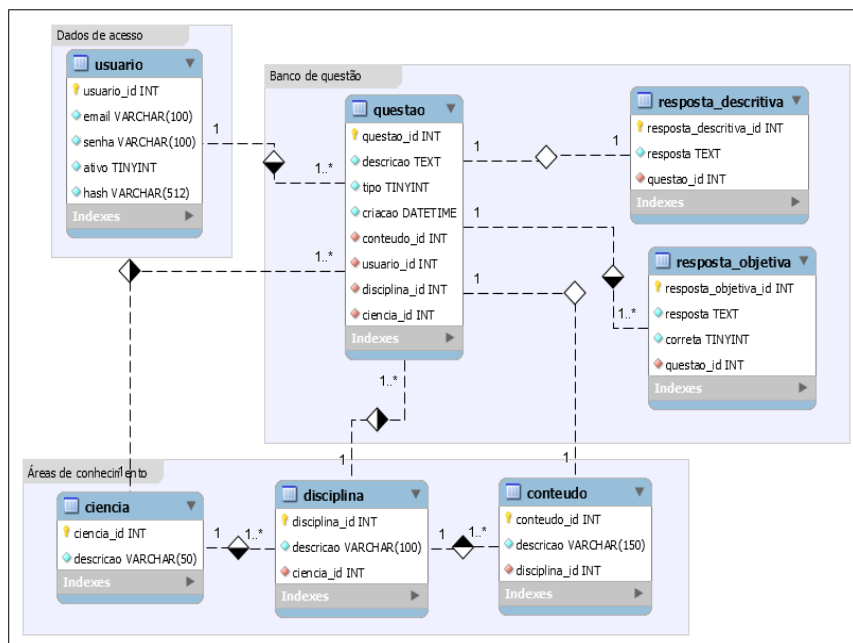
**Figura 18. Caso de uso: Banco de questões. Criado pelo Autor.**

A Figura 19 demonstra o fluxo de atividades para efetuar o gerenciamento de questões dentro do módulo, sendo os processos básicos para o gerenciamento, cadastrar questão e vincular com uma área de conhecimento, uma disciplina e um conteúdo, editar questão, excluir questão, visualizar *feed* de questões e a possibilidade de aplicar filtros de busca.



**Figura 19. Diagrama de atividades do gerenciamento do banco de questões. Criado pelo Autor.**

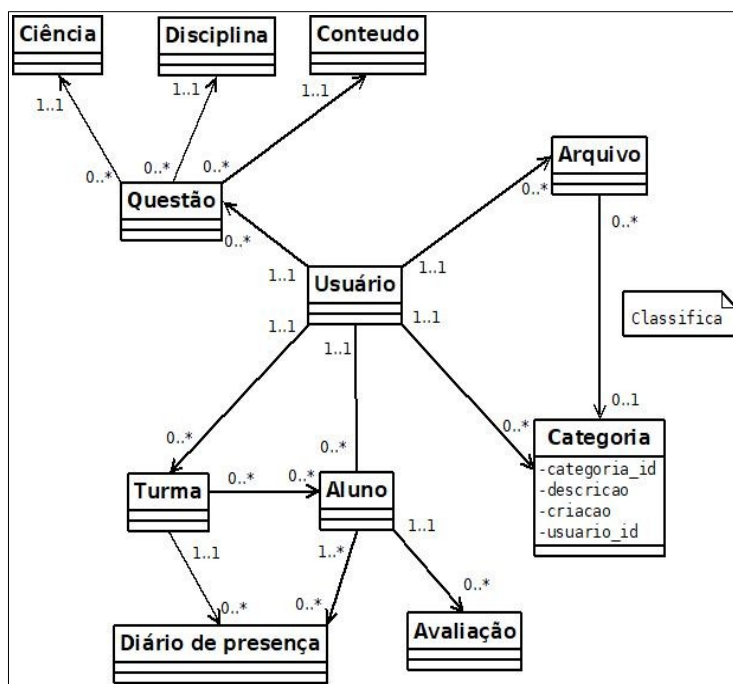
O módulo Banco de Questões conta com o uso de 7 tabelas contando a de usuário para acesso ao sistema, sendo que um usuário pode cadastrar questões, tanto descritivas como objetivas, devendo obrigatoriamente vinculá-la a um conteúdo, por questões de organização do sistema e para possibilitar a aplicação de filtros de busca.



**Figura 20. Modelo relacional do módulo banco de questões. Criado pelo Autor.**

O modelo relacional desse módulo está disposto na Figura 20, sendo que pela regra de negócio os dados das tabelas ciência, disciplina e conteúdo devem ser pré cadastrados no sistema via banco de dados seguindo os requisitos funcionais. Por fim, o diagrama de classes da Figura 21 propõe uma visão estrutural das classes do sistema.

Após a criação da documentação de cada módulo, foi possível iniciar a implementação do software, auxiliando o cumprimento dos requisitos e fluxos necessários.



**Figura 21. Diagrama de classes do sistema. Criado pelo Autor.**

#### 4.4. Implementação do software

A implementação do software foi feita utilizando linguagem PHP, Javascript, linguagem de marcação HTML, CSS, *framework* Bootstrap, biblioteca PHPMailer, biblioteca PHP *Data Object*, biblioteca CKEditor, biblioteca *DataTables* e ícones disponibilizados pelo *FontAwesome* e banco de dados MySQL. Também foi utilizado o paradigma de desenvolvimento orientado a objetos e arquitetura se baseou no padrão MVC (*Model-View-Controller*).

O desenvolvimento utilizou boas práticas de programação seguindo as Recomendações de Padrões PHP utilizando a PSR-1, PSR-2 e o *autoload* da PSR-4, o que trouxe uma facilidade na codificação. Esses padrões auxiliaram na estética do código e na facilidade de vinculação de arquivos através do *autoloader*.

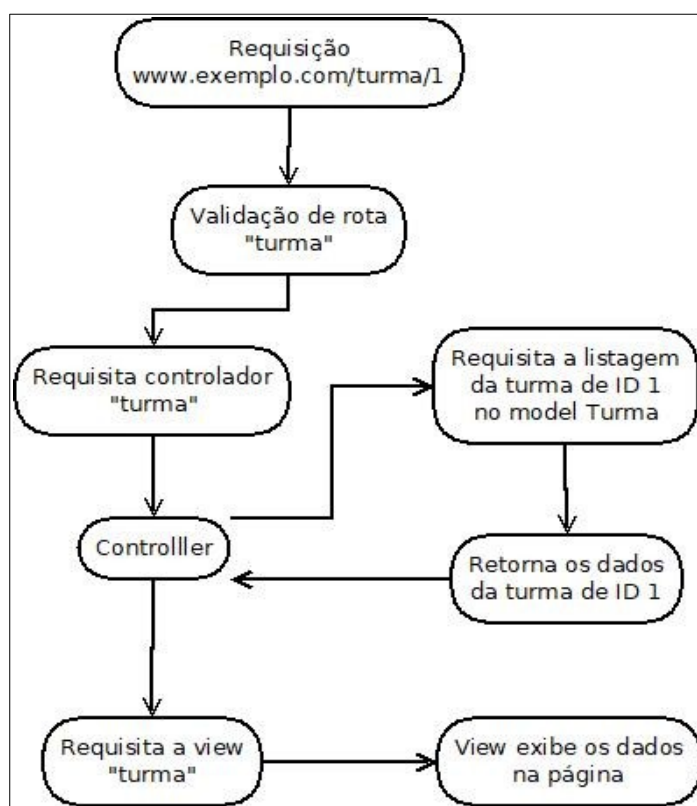


Figura 22. Exemplo de funcionamento do MVC implementado no desenvolvimento do sistema.

A arquitetura utilizada é conhecida como *Model-View-Controller* devido a sua forma de uso, aonde o *controller* recebe uma rota, por exemplo, `http://localhost/turma` onde a turma representa a rota requisitada. A partir dessa rota o *controller* deve ter uma função, na qual trata o pedido, que pode utilizar uma *Model*, que nesta arquitetura representa as classes utilizadas pelo sistema e retorna informações para o usuário, podendo ser através de texto formatado como JSON ou pela requisição de uma *view* que nesse caso são *templates* criados com HTML, CSS, Bootstrap e Javascript.

A Figura 22 auxilia na exemplificação do que foi implementado e representa um funcionamento lógico real. Através da requisição da rota é realizada uma validação, a fim de evitar que uma rota não existente seja chamada. Com a rota validada o controlador procura por uma função que responde a rota, nesse caso como é passado



uma variável de controle, logicamente é feito a requisição de uma turma de identificador 1.

Através disso o controlador cria a *model* chamada Turma e requisita os dados da turma com o identificador indicado, após obter a resposta o controlador organiza os dados recebidos e o envia para uma *view* que é exibida na página. A regra implementada no núcleo do MVC é de que a rota, a função correspondente no controlador e a *view* utilizada devem conter a mesma representação textual.

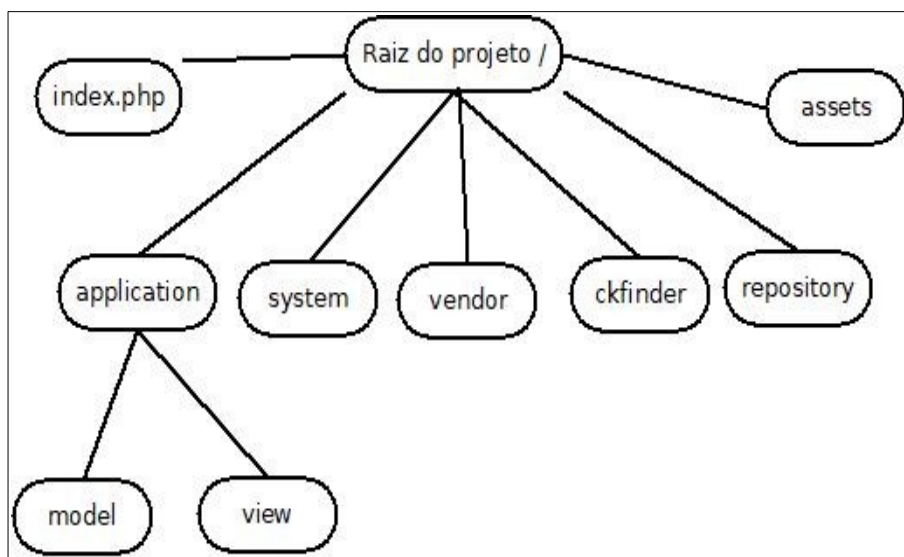


Figura 23. Estrutura de pastas do projeto

Com a utilização do MVC é possível criar uma organização de pastas que referenciam seus respectivos dados. A Figura 23 representa a estrutura, onde o `index.php` é o responsável por manipular todas as outras requisições. A pasta *application* contém dentro dela outras duas pastas, *model* que contém as classes e a *view* que contém os *templates*, já a pasta *system* engloba a lista de rotas dentro do arquivo `Router.php`, o controlador no `Controller.php` e o arquivo de configurações nomeado `Systeacher.php` que é responsável por configurar o caminho dos diretórios, o nome do sistema, versão, a codificação dos caracteres e o padrão de horário do PHP.

Seu software de auxílio

Acesse o sistema

Email:

Senha:

Entrar    Registre-se    Esqueci minha senha

A imagem mostra uma interface de login com um fundo escuro. No topo, o título 'Seu software de auxílio' é exibido em branco. Abaixo dele, o texto 'Acesse o sistema' aparece em uma fonte menor. Seguem dois campos de entrada: 'Email:' e 'Senha:', ambos com labels em branco e campos de texto brancos. Na base, há três botões arredondados em azul: 'Entrar', 'Registre-se' e 'Esqueci minha senha'.

Figura 24. Tela de login. Criado pelo autor.



A pasta *vendor* compreende todos os arquivos relacionados ao *autoloader*, a pasta *ckfinder* é responsável por armazenar os arquivos enviados por *upload* pela biblioteca CKEditor que terá seu uso explicado mais a frente. A pasta *repository* armazena os *uploads* feitos pelo módulo banco de arquivos e por fim a pasta *assets* que possui arquivos CSS e Javascript utilizados pelo sistema.

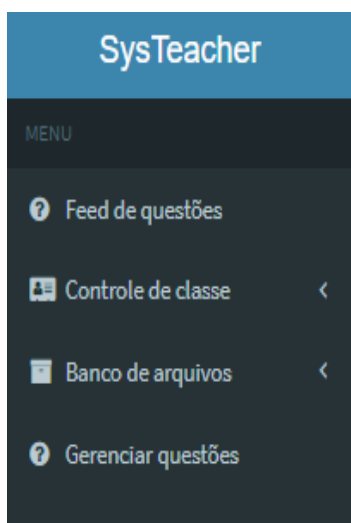
Inicialmente foram desenvolvidas as funcionalidades de acesso ao sistema, tais como: Acessar o sistema, Registre-se e Esqueci minha senha que são funcionalidades básicas em sistemas web. Tanto a função Registre-se que possibilita que o usuário se cadastre no sistema como a função Esqueci minha senha, possibilita que o usuário altere sua senha utilizando a API PHPMailer que segundo PHPMailer (2018) é uma API que facilita o envio de e-mails através do PHP e um servidor de envio de e-mails, utilizando orientação a objeto e uma documentação enxuta.

Na Figura 24 é possível visualizar a simplicidade do *template* utilizado para o acesso do usuário no sistema. A lógica codificada para o sistema foi que, inicialmente o usuário precisa se registrar, inserindo seu e-mail, no qual se os dados inseridos no cadastro forem válidos receberá em sua caixa de entrada um link que ativará sua conta. Em caso de esquecimento, o e-mail da conta deve ser inserido na funcionalidade Esqueci minha senha e também será enviado para caixa de entrada um link para troca de senha.

```
public function recuperarSenha($email, $hash)
{
    $this->mailer->addAddress($email);
    $this->mailer->addReplyTo('maikel-93@hotmail.com', 'SYSTEACHER');
    $assunto = utf8_decode(SYSTEM_NAME);
    $this->mailer->Subject = $assunto;
    $texto = "
    <p>Acesse o link abaixo para alterar sua senha.</p><br>
    {$this->link}recuperarsenha/{$hash}
    ";
    $mensagem = utf8_decode($this->pattern($texto));
    $this->mailer->Body = $mensagem;
    $mensagemSemHTML = utf8_decode($texto);
    $this->mailer->AltBody = $mensagemSemHTML;
    if(!$this->mailer->send()) {
        if (ENVIRONMENT == "development") {
            var_dump($this->mailer->ErrorInfo);
            exit();
        }
        return false;
    } else {
        return true;
    }
}
```

Figura 25. Função para envio de e-mail de recuperação de senha utilizando PHPMailer.

Essas funcionalidades foram implementadas utilizando uma *hash* única por requisição, que são apagadas após concluir a operação, ou no caso de recuperação de senha, apagado após um login válido. A facilidade do uso do PHPMailer pode ser visto na Figura 25, essa figura demonstra que fora as validações, com apenas 13 linhas é possível definir o destinatário, o remetente, o assunto e enviar um e-mail, até menos dependendo o tipo de indentação utilizado.



**Figura 26. Menu lateral do sistema.**

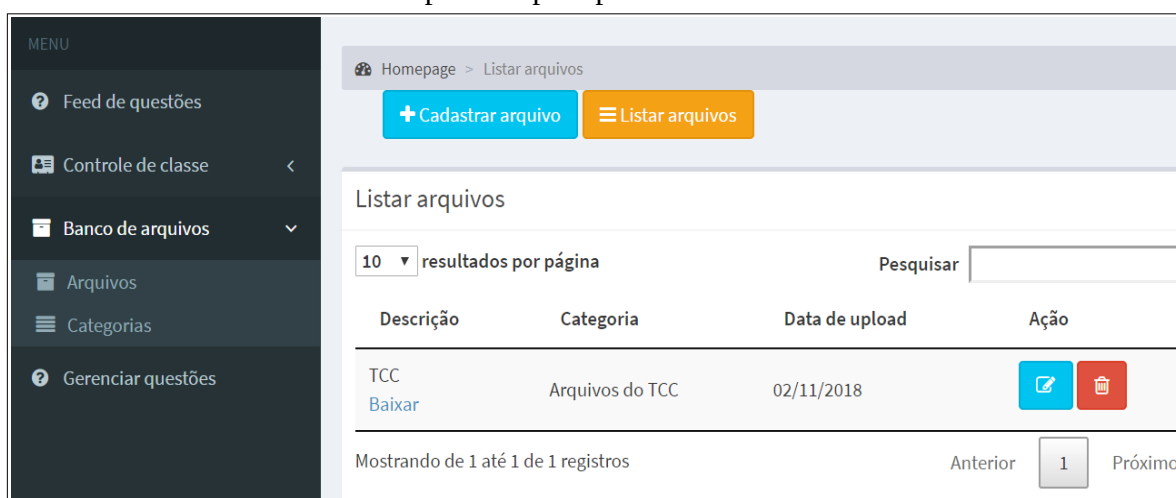
Outro padrão adotado pelo sistema são as mensagens de retorno aos usuários conhecidas como *Flash Messages* que utilizam um retorno através de variáveis de sessão, elas são definidas após o processamento e exibidas na tela, como demonstra a Figura 27. Todas as mensagens definidas como um erro são exibidas em um fundo vermelho, já as mensagens de sucesso nas operações em um fundo verde.

A navegação entre as funcionalidades do sistema são feitas através de dois menus. A Figura 26 exhibe o menu responsável pela navegação entre os três módulos e o *feed* de questões. Este menu fica disposto na lateral esquerda do sistema durante todas as operações realizadas dentro do sistema, facilitando a interação entre os módulos. O segundo menu é exibido dentro dos módulos, localizado na parte superior como também pode ser visto na Figura 27. Esse menu realiza interações entre as funcionalidades de cada módulo e fica localizado na parte superior da interface.

O sistema também utiliza a biblioteca conhecida como *DataTables* que é responsável por estilizar as tabelas de exibição de dados e utiliza JavaScript para criar interações como, limitar resultados por página, paginação, busca textual de informações e renderizar a tabela de forma responsiva para plataformas *mobile*.



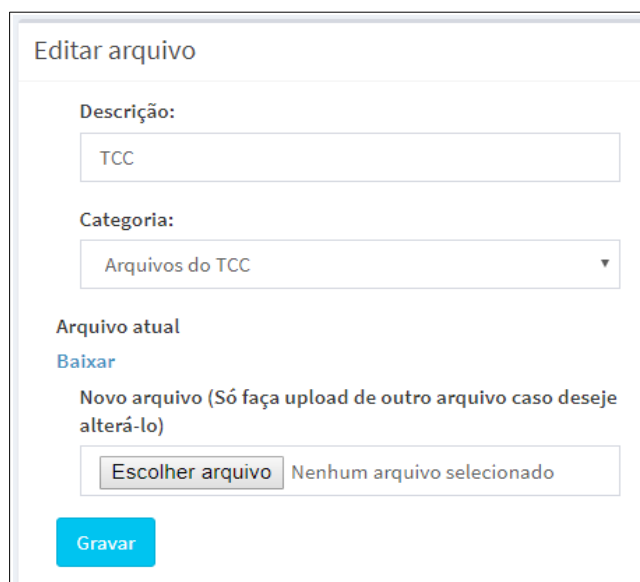
**Figura 27. Exemplo de Flash Messages de erro.**



**Figura 28. Exemplo de interface dos módulos.**

A Figura 28 demonstra a disposição dos elementos durante a listagem de dados

de um módulo, neste exemplo, o banco de arquivos. Nesta figura é possível visualizar como são arranjados os botões que são responsáveis por ações individuais de cada linha de dados, no exemplo, impressão, edição e exclusão. As cores e os ícones adotados pelos módulos do sistema seguem o mesmo padrão, buscando tornar prático a rotina do usuário.

A imagem mostra um formulário web intitulado "Editar arquivo". O formulário contém os seguintes elementos: um campo de texto rotulado "Descrição:" com o valor "TCC"; um menu suspenso rotulado "Categoria:" com a opção selecionada "Arquivos do TCC"; uma seção rotulada "Arquivo atual" que contém um link azul "Baixar"; um texto "Novo arquivo (Só faça upload de outro arquivo caso deseje alterá-lo)"; um botão "Escolher arquivo" e o texto "Nenhum arquivo selecionado"; e um botão azul "Gravar" na base do formulário.

**Figura 29. Exemplo de formulário de alteração de dados.**

A Figura 28 também demonstra parcialmente o uso do módulo banco de arquivos. Nesta figura é possível ver um arquivo nomeado TCC inserido junto a um hiperlink Baixar que é responsável pelo download do arquivo, a categoria no qual o arquivo foi vinculado, a data na qual foi efetuado o *upload* e os botões de ação. Já na Figura 29 é possível visualizar o padrão dos formulários de alteração dos dados, no exemplo a edição de um arquivo. Os outros formulários de edição seguem o mesmo padrão, replicando os dados que foram cadastrados e oferecendo a possibilidade de alteração.

A lógica funcional do módulo banco de arquivos se dá da seguinte forma, o usuário pode cadastrar categorias para vincular aos seus arquivos, assim facilitando a busca. A listagem exibe os arquivos mais recentes no topo da lista, também exibindo sua categoria, se existente, e o link para baixar. Já a edição viabiliza a alteração da descrição do arquivo, da categoria e também do arquivo.

O sistema utiliza funções responsáveis por manipular o banco de dados em todas os módulos implementados e essas utilizam a biblioteca PHP Data Object (PDO), que é responsável pela interação da aplicação com um banco de dados, neste caso MySQL e por criar uma abstração na questão de validação de dados.

A Figura 30 representa uma função de atualização de dados, aonde é feita a criação do script MySQL (linha 280 até 298). Como podemos observar, dentro do *script* temos atribuições precedidas de dois pontos (:), exemplo na linha 282 da imagem, que são posteriormente validadas e alteradas por um valor enviado por parâmetro. Essa função de validação limita as chances de uma injeção de SQL, que poderia comprometer a segurança do sistema, na linha 300 temos um exemplo, onde a referência:descricao (dois pontos acompanhada da palavra descricao, sem acento e ç) é alterada por uma *String* que está subscrita na variável \$post['descricao'].

```

280 $sql =
281 "update arquivo set
282     descricao = :descricao,";
283 if ($arr_upload) {
284     $sql .= "
285         caminho = :caminho,
286         tipo = :tipo,
287         tamanho = :tamanho,
288         nome_arquivo = :nome_arquivo,
289     ";
290 }
291 $sql .= "
292     categoria_id = :categoria_id
293 where
294     usuario_id = :usuario_id
295 and
296     arquivo_id = :arquivo_id
297 limit 1
298 ";
299 $consulta = $this->database->prepare($sql);
300 $consulta->bindParam(':descricao', $post["descricao"], PDO::PARAM_STR);
301 if ($arr_upload) {
302     $consulta->bindParam(':caminho', $arr_upload["filemoved"], PDO::PARAM_STR);
303     $consulta->bindParam(':tipo', $arr_upload["filetype"], PDO::PARAM_STR);
304     $consulta->bindParam(':tamanho', $arr_upload["filesize"], PDO::PARAM_STR);
305     $consulta->bindParam(':nome_arquivo', $arr_upload["filename"], PDO::PARAM_STR);
306 }
307 if (
308     $post["categoria_id"] == null ||
309     $post["categoria_id"] == ""
310 ) {
311     $consulta->bindParam(':categoria_id', $post["categoria_id"], PDO::PARAM_NULL);
312 } else {
313     $consulta->bindParam(':categoria_id', $post["categoria_id"], PDO::PARAM_INT);
314 }

```

Figura 30: Função de atualização de arquivo no banco de dados.

No módulo controle de classe tem-se duas partes distintas, a primeira e fundamental é responsável pelo gerenciamento dos alunos ligados ao usuário, nesse gerenciamento é possível cadastrar alunos, distinguindo-os pelos seus nomes e pelas

Cadastrar aluno

Primeiro nome:

Segundo nome:

Último nome:

Data de nascimento:

Figura 31. Cadastro de alunos.

datas de nascimento, como podemos observar na Figura 31, ainda, a listagem e a edição seguem o mesmo padrão anteriormente descrito no módulo banco de arquivos.

O gerenciamento de turmas, é a segunda parte do módulo, o cadastro de turma se baseia em uma descrição textual e a referência ao ano. A partir disso, deve-se gerenciar os alunos que participarão da turma, sendo essa parcela, dependente do cadastro de no mínimo um aluno.



**Figura 32. Gerenciamento de turmas.**

Como podemos visualizar na Figura 32, a listagem disponibiliza botões em azul, que descrevem textualmente as ações pelas quais são responsáveis. Nesse ponto a lógica requer que no gerenciamento de alunos seja realizado o vínculo destes à sua turma.

Nome	Data de nascimento	Estado
José Juca	30/05/1980	Vincular na turma
Maikel Vitancourt	15/04/1993	Desvincular da turma

Mostrando de 1 até 2 de 2 registros

Anterior 1 Próximo

**Figura 33: Vínculo de alunos**

Conforme podemos analisar na Figura 33 o vínculo de alunos na turma é feito através de um em um botão referente ao aluno desejado, da mesma forma também pode ser feito remoção do mesmo.

Com o número desejado de alunos vinculados a turma, o gerenciamento de provas e diários de presença é viável. As provas podem ser gerenciadas clicando no botão referente ao número da prova, conforme exposto na Figura 32. Isso disponibiliza ao usuário a lista dos usuários com vínculo na turma e um campo referente a cada usuário para inserção da nota, de acordo com a Figura 34. A mesma interface é usada para alteração dos registros, visto que se houver notas já cadastradas ela oferece a

alteração.

### Gerenciar Notas da prova - Turma - Turma 1 - 2018

Descrição:

Pesquisar

Nome	Nota
Maikel Vitancourt	<input type="text" value="80"/>
José Juca	<input type="text" value="100"/>

Mostrando de 1 até 2 de 2 registros

Gravar

**Figura 34. Gerenciamento de notas.**

Os diários de presença estão disponíveis no botão nomeado gerenciar diários, como pode ser observado na Figura 32. Ao acessar este link, é disponibilizado ao usuário a listagem dos diários já cadastrados, se houver, ordenados pela data mais recente no topo. O registro das presenças é feito através de um cadastro, aonde o usuário deve inserir uma data ainda não cadastrada na turma e assinalar se o aluno estava presente ou ausente. Nesta interface também é gerado uma simples contagem de presenças e faltas do dia, conforme a Figura 35.

+ Cadastrar diário

≡ Listar diários

≡ Voltar

### Listar diários - Turma 1 - 2018

10 resultados por página

Pesquisar

Data	Faltas	Presenças	Ação
09/11/2018	0	1	<div><div></div><div></div></div>
02/11/2018	0	1	<div><div></div><div></div></div>

Mostrando de 1 até 2 de 2 registros

Anterior

1

Próximo

**Figura 35. Gerenciamento de diários de presença.**

O controle de classe também disponibiliza relatórios, gerais e individuais, que podem ser visualizados através de um clique no botão amarelo com um ícone de impressão, conforme pode ser visto na Figura 32. O relatório geral é o primeiro a ser exibido, dentro dele é possível gerar uma impressão ou acessar o relatório individual de

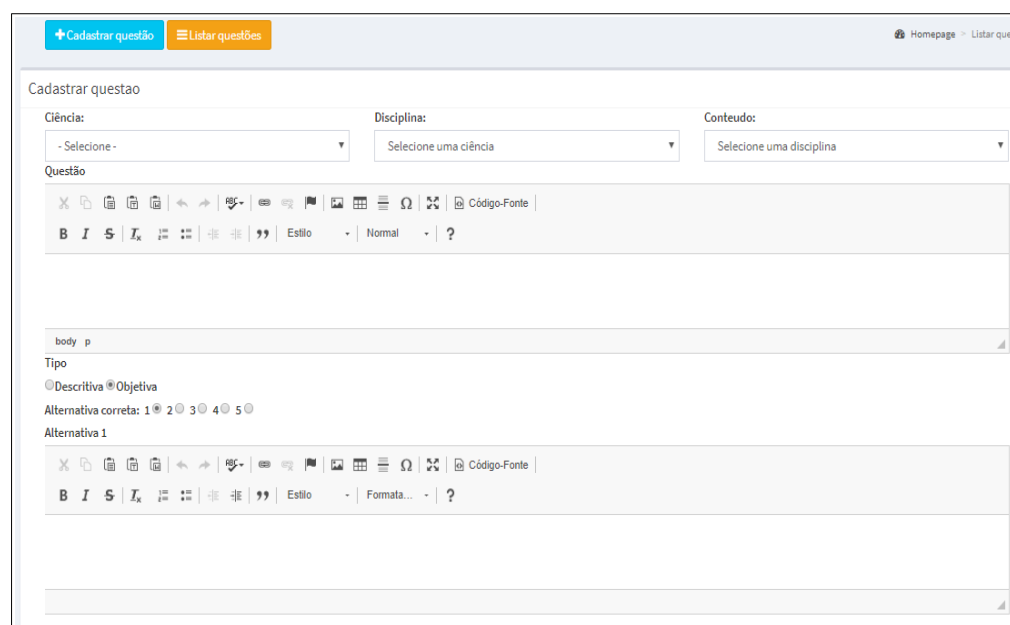
um aluno. A Figura 36 demonstra um relatório geral.



Nome	Faltas	Presenças	Média
José Juca <a href="#">Relatório individual</a>	0	0	0.00
Maikel Vitancourt <a href="#">Relatório individual</a>	0	2	90.00

**Figura 36. Relatório geral da turma.**

O último módulo implementado é o banco de questões, que conta com a possibilidade de cadastrar questões descritivas e objetivas e tem como objetivo principal o armazenamento e o compartilhamento de questões entre professores, dado esse motivo, além do gerenciamento, o módulo conta com um *feed* de compartilhamento de questões.



**Figura 37. Formulário de cadastro de questões.**

Para incrementar os dados relacionados as questões foi utilizado uma tabela de áreas de conhecimento do Conselho Nacional de Pesquisa (CNPQ) aonde foram pré cadastradas no sistema as ciências, disciplinas e o primeiro nível de classificação do conteúdo. Isso servirá para organização, classificação e busca das questões, visto ser obrigatória a associação das questões nos itens citados.

Para o cadastro das questões, é necessário informar a ciência, disciplina e conteúdos. Há dois tipos de questões disponíveis, descritivas e objetivas, sendo que

respectivamente, uma habilita um campo para a resposta correta e a segunda proporciona cinco campos para alternativas, sendo obrigatório selecionar qual alternativa está correta, conforme a Figura 37.

```
106 <label>Alternativa 2</label>
107 <textarea class="form-control"
108   name="objetiva[2]"
109   id="alternativa_2"></textarea>
110 <script>
111   var editor = CKEDITOR.replace( 'alternativa_2' );
112   CKFinder.setupCKEditor( editor );
113 </script>
114 <br>
```

Figura 38. Exemplo de chamada da biblioteca CKEditor.

A alteração nas informações de questões cadastradas também é possível através de um formulário semelhante ao disposto na Figura 37, onde não é possível alterar o tipo da questão e a classificação, devido à estrutura criada no banco de dados.

```
158 <script>
159 $('#ciencia').change(function(e){
160   var ciencia = $("#ciencia").val();
161   var select = document.getElementById("select-disciplina");
162   $('#loading-disciplina').html('Aguarde, carregando...');
163   $.getJSON('/getdisciplina/'+ciencia, function (dados){
164     if (dados.length > 0){
165       $('#select-disciplina').html("<option value=''>- Selecione -</option>");
166       dados.forEach(function(valor, chave){
167         var option = document.createElement("option");
168         option.setAttribute("value", valor.disciplina_id);
169         var txtnode = document.createTextNode(valor.descricao);
170         option.appendChild(txtnode);
171         select.appendChild(option);
172         $('#loading-disciplina').html("");
173       });
174     }else{
175       $('#loading-disciplina').html("Erro ao carregar as disciplinas, selecione novamente a ciência!");
176     }
177   })
178 })
```

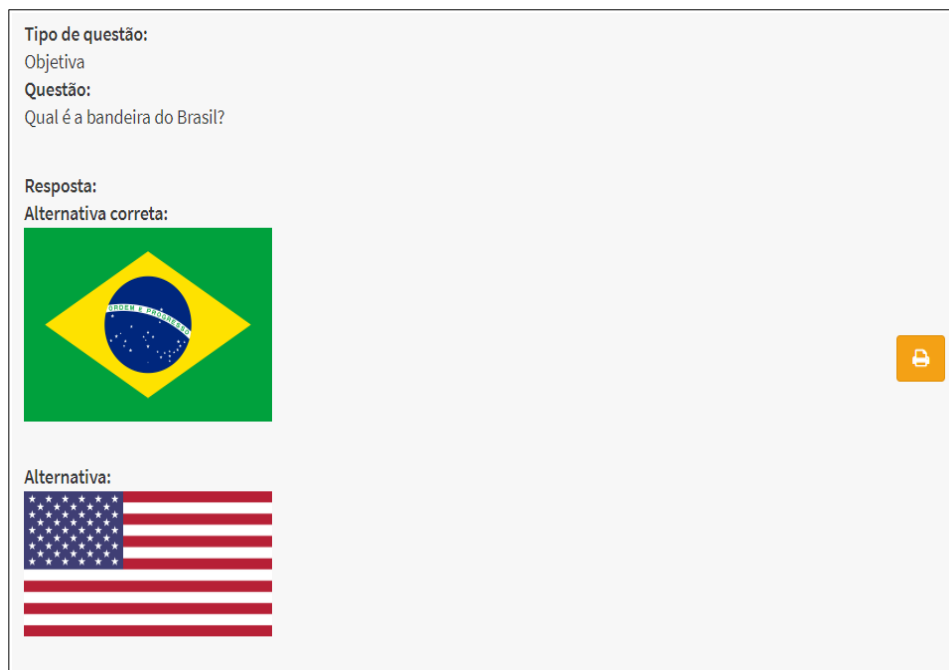
Figura 39. Requisição Ajax.

Também é possível inserir imagens nas questões e respostas, isso foi implementado com a utilização da biblioteca CKEditor, que oferece recursos de manipulação de texto para um *textarea*. Como pode ser visto nas linhas 111 e 112 da Figura 38, a chamada da biblioteca é feita através de duas linhas, na qual a função deve receber por parâmetro um campo do tipo *textarea*.

Para as hierarquias de classificação foram desenvolvidas requisições Ajax, o exemplo da Figura 39, demonstra uma requisição, que através de uma ciência passada por parâmetro e aguarda um retorno em formato JSON com as disciplinas, como podemos ver na linha 163 e renderiza o campo de seleção, linha 166 até 173.

As requisições em Ajax também são utilizadas no *Feed* de Questões para filtrar a busca. O diferencial entre os links no menu Banco de Questões e *Feed* de questões é que respectivamente, o primeiro é utilizado para gerenciar as questões cadastradas pelo usuário, já, o segundo disponibiliza uma busca entre todas as perguntas contidas no banco de dados.





**Figura 40. Exemplo de questão no Feed de Questões.**

A Figura 40 serve como demonstração do *Feed* de questões, com utilização de imagens e duas alternativas com resposta. Nessa interface também é possível gerar uma tela para impressão.

Após o desenvolvimento dos módulos foram realizados testes funcionais, que serão descritos na próxima seção.

#### 4.5. Testes

Para assegurar o funcionamento do sistema, foram feitos testes funcionais, a fim de validar se estão conforme os requisitos funcionais. O roteiro utilizado nos testes foi o seguinte: Inserção passando dados válidos; Inserção passando dados inválidos; Alteração passando dados válidos; Alteração passando dados inválidos; *Upload* de arquivo; *Upload* de arquivo para substituição do antigo; Exclusão de dados; Passagem de parâmetro incorreto; Reutilização de CSRF Token; Checagem do redirecionamento de links.

Os testes foram aplicados em um ciclo de repetição até que as funcionalidades se adequassem aos requisitos funcionais e que os dados fossem cadastrados no banco de dados de forma correta. A Figura 41 demonstra como foi identificado o funcionamento correto da inserção de dados cumprimento o que era requisitado.

questao_id	descricao	tipo 1 Descritiva 2 Objetiva	criacao	conteudo_id	usuario_id	disciplina_id	cie
3	<p>Quem descobriu o Brasil?</p>	1	2018-11-11 20:41:46	289	1	68	
4	<p>Qual é a bandeira do Brasil?</p>	2	2018-11-11 21:21:02	361	1	96	

**Figura 41: Verificação de inserções no banco de dados. Criado pelo Autor.**

## 5. Considerações finais

Este trabalho teve como propósito auxiliar o docente em suas tarefas do dia a dia, aumentando sua produtividade por facilitar suas atividades através do desenvolvimento de um software. O sistema também visa proporcionar o acesso às TICs pelos professores, demonstrando as possibilidades criadas através da inclusão de tecnologia nas áreas educacionais.

A solução proposta pelo trabalho, foi realizar um levantamento de requisitos para desenvolver um sistema para auxiliar docentes. As funcionalidades definidas foram o banco de arquivos, banco de questões e controle de classe, que tiveram relação com as respostas do questionário aplicado e da pesquisa realizada pelo NIC.br, que tinha como objetivo descobrir os recursos mais utilizados pelos professores na internet. A partir do questionário aplicado, verificou-se que há uma pequena quantidade de softwares que focam o docente como usuário final, ou se existem são pouco divulgados no meio educacional, pois de acordo com as respostas, uma baixa quantidade de educadores tem conhecimento sobre sistemas que contemplem essas funcionalidades.

Durante o desenvolvimento do sistema foram utilizados conhecimentos obtidos durante a graduação em Ciência da Computação, que auxiliaram na adoção padrões que permitiram desenvolver o software de uma forma eficiente, dentre estes conhecimentos estão: Técnicas de engenharia de software; Gerenciamento de banco de dados; Programação orientação a objetos; Padrão MVC que facilitou o desenvolvimento devido ao esquema de sua arquitetura; dentre outros.

O trabalho também constata oportunidades para desenvolvedores de software criarem sistemas focados no auxílio ao docente, sendo que boa parte dos professores não tem conhecimento sobre sistemas que podem auxiliar no seu dia a dia. A pesquisa feita pelo NIC.br dá embasamento as funcionalidades aqui desenvolvidas e também cria um leque de novas oportunidades e ideias para trabalhos futuros, como implementação de novas funcionalidades listadas na pesquisa e também no desenvolvimento de ferramentas mobile que contemplem as funcionalidades aqui desenvolvidas.

## Referências

- Alencar, S. Andréia et al. “O Moodle como ferramenta didática” Disponível em: <<http://www.periodicos.letras.ufmg.br/index.php/ueadsl/article/viewFile/2919/2878>>. Acesso em: 20 jun. 2018.
- Amorim, A. Joni et al. “Uso do Teleduc como um recurso complementar no ensino presencial” Associação Brasileira de Educação a Distância. Disponível em: <<http://seer.abed.net.br/index.php/RBAAD/article/view/171/51>>. Acesso em: 20 jun. 2018.
- Bastos, André, Berardi, G. C. Rita e Silveira, A. Ricardo. (2005) “Webduc: Uma proposta de ferramenta de avaliação formativa no ambiente TelEduc. v. 2 Nº 1” Disponível em: <<http://seer.ufrgs.br/renote/article/viewFile/13690/8034>>. Acesso em: 20 jun. 2018.
- Bergamasco, C. Elizabeth, Bergamasco, C. C. Leila e Brandão, O. Leônidas (2013) “A utilização das Tecnologias de Informação e Comunicação na Educação Infantil: avanços e desafios” In: Anais do SBIE.
- Bootstrap. Disponível em: <<http://getbootstrap.com.br/>>. Acesso em: 04 mai. 2018.

- Costa, José W. e Oliveira, Maria A. M. (2004) “Novas Linguagens e novas tecnologias: educação e sociabilidade” Petrópolis: Vozes.
- Dalmon, L. Danilo, Brandão, A. F. Anarosa e Brandão, O. Leônidas (2012) “Uso de Métodos e Técnicas para Desenvolvimento de Software Educacional em Universidades Brasileiras” In: Anais do SBIE.
- Date, C. J. (2003) “Introdução a sistemas de banco de dados” Rio de Janeiro: Elsevier.
- Documentação do CSS, Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 04 mai. 2018.
- Documentação do HTML, Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 04 mai. 2018.
- Documentação do JavaScript, Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 04 mai. 2018.
- Documentação do MySQL, Disponível em: <<https://dev.mysql.com/doc/>>. Acesso em: 04 mai. 2018.
- Documentação do PHP, Disponível em: <<https://secure.php.net/docs.php>>. Acesso em: 04 mai. 2018.
- Franciscato, Fábio Teixeira., Ribeiro, Patric da S., Mozzaquatro, Patricia M., Medina, Roseclea D. “Avaliação dos Ambientes Virtuais de Aprendizagem Moodle, TelEduc e Tidia-AE: um estudo comparativo” CINTED-UFRGS Novas Tecnologias na Educação. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/14509>>. Acesso em: 21 jun. 2018.
- FTEaD. “Formação de Tutores em Educação a Distância. Tidia-AE” Disponível em: <<http://proec.ufabc.edu.br/uab/index.php/roteiros/roteiro5/19-fteadinicio/fteadaulas/141-aula54>>. Acesso em junho de 2018.
- Gomes, Eduardo H., Pimentel, Edson P., Kleinschmidt, João H. “Segurança em Sistemas de E-learning: uma Análise do Ambiente Tidia-AE/Sakai. Universidade Federal do ABC” Instituto Federal de Educação Ciência e Tecnologia de São Paulo. Disponível em: <<http://ehgomes.com.br/images/pdf/2011-comtel-seguranca-gomeseh.pdf>>. Acesso em: 21 jun. 2018.
- Guedes Gilleanes T. A. (2011) “UML 2: uma abordagem prática” São Paulo: Novatec Editora, 2ª edição.
- Mendonça, Gilda Aquino de Araújo. “As tecnologias na Educação à Distância” Instituto Federal de Educação, Ciência e Tecnologia de Goiás. Goiânia-GO, 04/2014. Disponível em: <<http://www.abed.org.br/hotsite/20-ciaed/pt/anais/pdf/115.pdf>>. Acesso em: 22 jun. 2018.
- Moreira, Antonio F. B., Sonia K. (2007) “Contemporaneidade, educação e tecnologia” Campinas: Educação & Sociedade.
- Nascimento. Michelle R. L. do, (2012) “A inserção das tecnologias nas escolas e a cultura escolar” Rio de Janeiro: UERJ.
- NIC.br (2015) “TIC Educação 2015: Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras” Disponível em

<[https://www.nic.br/media/docs/publicacoes/2/TIC\\_Edu\\_2015\\_LIVRO\\_ELETRONICO.pdf](https://www.nic.br/media/docs/publicacoes/2/TIC_Edu_2015_LIVRO_ELETRONICO.pdf)> Acesso em: 22 Ago. 2018.

Oliveira, Elsa D. (2013) “Tecnologia e educação” São Paulo: PUC – SP.

Pais, Luiz C. (2005) “Educação escolar e as tecnologias da informática” Belo Horizonte: Autêntica.

Paula Filho, Wilson de Pádua (2009) “Engenharia de software: fundamentos, métodos e padrões” Rio de Janeiro, LTC, 3ª edição.

PHP-FIG “Recomendações de Padrões PHP” Disponível em <<https://www.php-fig.org/psr/>> Acesso em: 28 Ago. 2018.

PHPMailer “API de envio de e-mails PHP” Disponível em <<https://github.com/PHPMailer/PHPMailer>> Acesso em: 28 Ago. 2018.

Pinto, Aparecida. M. (2012) “As novas tecnologias e a educação” Revista Portal Anped sul.

Pressman, Roger S. (2016) “Engenharia de software: Uma abordagem profissional” Porto Alegre: AMGH, 8ª edição.

Silberschatz, Abraham. (2006) “Sistema de banco de dados” Rio de Janeiro: Elsevier.

Silva, Renildo F. Da, Correa, Emilce S. (2014) “Novas tecnologias e educação: A evolução do processo de ensino e aprendizagem na sociedade contemporânea” Educação e Linguagem, 2014.

Sommerville, Ian. (2011) “Engenharia de software” São Paulo: Pearson Prentice Hall, 9ª edição.