



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

Avalog 2.0 - Sistema *Web* para Geração de Avaliações

Bruno Araujo Camarda

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Bruno Rabello Monteiro

**Julho, 2019
João Monlevade–MG**

Bruno Araujo Camarda

Avalog 2.0 - Sistema *Web* para Geração de Avaliações

Orientador: Bruno Rabello Monteiro

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Julho de 2019

C172a Camarda, Bruno Araujo.
Avalog 2.0 [manuscrito]: Sistema web para geração de avaliações / Bruno Araujo Camarda. - 2019.

51f.:

Orientador: Prof. MSc. Bruno Rabello Monteiro.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Avaliação. 2. Sistemas de recuperação da informação. 3. Sistemas de coleta automática de dados. 4. Aplicações web. I. Monteiro, Bruno Rabello. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 004.775

Catálogo: ficha.sisbin@ufop.edu.br

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

Avalog 2.0 - Sistema *Web* para Geração de Avaliações

Bruno Araujo Camarda

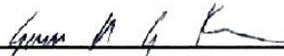
Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:



Bruno Rabello Monteiro
Mestre
DECSI - UFOP



Fernando Bernardes de Oliveira
Doutor
Examinador
DECSI - UFOP



George Henrique Godim da Fonseca
Doutor
Examinador
DECSI - UFOP

João Monlevade, 10 de julho de 2019

Agradecimentos

Agradeço primeiramente aos meus pais que sempre fizeram o possível para que eu tivesse a oportunidade de chegar até aqui.

Aos meus amigos, por me proporcionarem momentos felizes, engraçados e boas memórias.

Ao Marcos, Diego, Bia, Guilherme e Júlia, por serem aqueles aos quais eu corro para pedir socorro nos momentos ruins e por sempre estarem ali por mim.

Ao meu orientador Bruno, pela paciência e compreensão, mesmo quando eu sumia sem dar notícias.

“If one could run without getting tired, I don’t think one would often want to do anything else.”

— C.S.Lewis (1898 – 1963),
in: The Chronicles of Narnia: The Last Battle

Resumo

Neste trabalho é apresentado o Avalog 2.0, um sistema *web* que gera avaliações de forma automatizada, e tem por intuito reduzir o tempo e o trabalho de professores no processo de elaboração de provas e testes. Ele armazena questões e respostas em um banco de dados, e permite que, informados os critérios como conteúdo a ser abordado, quantidade de questões e tempo, avaliações sejam geradas de modo automatizado. O sistema foi desenvolvido em linguagem *PHP* com o *framework* Laravel e banco de dados relacional *MySQL*. Os resultados obtidos foram satisfatórios, atingindo os objetivos propostos, embora com algumas limitações.

Palavras-chaves: Avaliação, questão, assunto, geração automática, sistema *web*.

Abstract

This work presents the Avalog 2.0, a web system that generates exams in an automated way, and aims to reduce the time and the work of teachers in the process of elaboration of evaluations and tests. It stores questions and answers in a database, and allows that, informed the criteria such as content to be addressed, number of questions and time, assessments are generated in an automated way. The system was developed in PHP language with the Laravel framework and MySQL relational database. The results were satisfactory, reaching the proposed objectives, although with some limitations.

Key-words: Evaluations, question, subjects, automatic generation, web system.

Lista de ilustrações

Figura 1 – As funções básicas de um sistema de informação	16
Figura 2 – Funcionamento básico de um sistema web	17
Figura 3 – Exemplo de Diagrama de Sequência de Sistema	18
Figura 4 – Representação de um SGBD	19
Figura 5 – Exemplo de tabela do Modelo Relacional	20
Figura 6 – Tela de Visão Geral de Avaliação	23
Figura 7 – Tela de Início SM Provas	23
Figura 8 – Tela de inserção de questões Easy Test Maker	24
Figura 9 – Tela de inserção de questões Avalog	25
Figura 10 – Caso de Uso - Cadastrar Questão	28
Figura 11 – Diagrama de Caso de Uso	29
Figura 12 – Diagrama de Sequência - Gerar Avaliação	30
Figura 13 – Diagrama Entidade-Relacionamento	31
Figura 14 – Arquitetura utilizada pelo Laravel	32
Figura 15 – Tela de Login Avalog 2.0	33
Figura 16 – Tela de Cadastro de Assunto - Avalog 2	33
Figura 17 – Tela de Cadastro de Questão - Avalog 2	34
Figura 18 – Indicativo de Botão de Impressão - Avalog 2	35
Figura 19 – Tela de Geração Avaliação - Avalog 2	36
Figura 20 – Caso de Uso - Cadastrar Assunto	46
Figura 21 – Caso de Uso - Gerar Gabarito	47
Figura 22 – Caso de Uso - Gerar Avaliação	48
Figura 23 – Diagrama de Sequência - Cadastrar Questão	49
Figura 24 – Diagrama de Sequência - Cadastrar Assunto	49
Figura 25 – Avaliação Gerada - Avalog 2	50
Figura 26 – Avaliação Gerada - Avalog 2	51

Lista de tabelas

Tabela 1 – Comparativo entre trabalhos relacionados	25
Tabela 2 – Requisitos Funcionais do Sistema	27
Tabela 3 – Requisitos Não-funcionais do Sistema	27

Lista de abreviaturas e siglas

BD Banco de Dados

MVC *Model View Controller*

SGBD Sistema de Gerência de Bancos de Dados

SI Sistema de Informação

SIW Sistema de Informação baseado em Web

SQL Linguagem de Consulta Estruturada

TI Tecnologia da Informação

UML Linguagem de Modelagem Unificada

Sumário

1	INTRODUÇÃO	13
1.1	Problema	13
1.2	Objetivos	14
1.3	Organização do trabalho	14
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Sistemas de Informação	16
2.2	Linguagem de Modelagem Unificada	18
2.3	Banco de Dados	19
2.4	Linguagem de Programação PHP	21
2.4.1	Laravel	21
2.5	Avaliações de aprendizagem	21
2.6	Trabalhos relacionados	22
2.6.1	Easy LMS	22
2.6.2	SM Provas	23
2.6.3	Easy Test Maker	24
2.6.4	Avalog	24
3	ESPECIFICAÇÕES DO SISTEMA	26
3.1	Análise de Requisitos	26
3.2	Casos de Uso	27
3.3	Diagrama de Sequência	29
3.4	Diagrama Entidade-Relacionamento	30
4	AVALOG 2.0	32
4.1	Cadastro de Questões e Assuntos	33
4.1.1	Editor de Texto	34
4.2	Geração de Avaliações	35
4.2.1	Algoritmo Gerador de Avaliações	36
5	CONCLUSÃO	39
5.1	Trabalhos futuros	40
	REFERÊNCIAS	42

APÊNDICES	45
APÊNDICE A – CASOS DE USO	46
APÊNDICE B – DIAGRAMAS DE SEQUÊNCIA	49
APÊNDICE C – AVALIAÇÕES GERADAS	50

1 Introdução

A avaliação de aprendizagem é uma tarefa didática amplamente utilizada e que tem grande importância no processo de ensino. De acordo com [Zanon e Freitas \(2007\)](#) avaliar é um método para adquirir e processar evidências necessárias para melhorar a aprendizagem do aluno. [Gasparin \(2011\)](#) acrescenta que a avaliação deve ser a manifestação efetiva de que o aluno adquiriu um conhecimento e que este se tornou um instrumento de discernimento da realidade e de transformação social.

Como a avaliação é uma das etapas da aprendizagem, é necessário que ela esteja de acordo com as intenções do processo de ensino, coerente com o que foi trabalhado e passado no contexto educacional. Através de sua aplicação, é possível relacionar e comparar os resultados obtidos pelos alunos com os objetivos propostos pelo professor, de modo a compreender as dificuldades e progressos na educação e garantir uma forma de alavancar o desenvolvimento social e humano. ([GASPAR, 2009](#))

Para um aluno a avaliação é como um instrumento que o leva a pensar, buscar novas maneiras e recursos para resolver um problema, e assim, atingir conhecimento. Esse conhecimento não é apenas uma transmissão de informação, mas a síntese dela, ou seja, sua percepção do que ele aprendeu e como ele irá aplicá-lo no cotidiano ([ARAN et al., 2000](#)). [Oliveira e Chadwick \(2001\)](#) complementam que alunos que são avaliados recorrentemente e de forma correta tendem a obter melhores resultados.

Cabe aos professores a elaboração da avaliação, seguindo seus próprios critérios e métodos avaliativos. Ela é composta por questões escolhidas em livros e outras fontes, ou criadas por eles, conforme conteúdos tratados durante as aulas. O professor assume, então, o papel de avaliador ao mesmo tempo em que é o responsável pelo o que está sendo avaliado. ([GATTI, 2013](#)).

Este trabalho visa desenvolver um sistema *web* que facilite esse processo de elaboração, dando ao professor autonomia e condições de criar avaliações de forma rápida e prática.

1.1 Problema

Elaborar uma avaliação é comum à rotina de um professor e pode ser uma tarefa cansativa e demorada ([PERRENOUD, 1999](#)). É preciso levar em consideração os critérios que serão usados, o tipo de avaliação (objetivas e/ou subjetivas), qual conteúdo será abordado, o número de questões e o tempo de resolução de cada uma, entre outros critérios.

Essa tarefa tem evoluído, passando de um modo totalmente manual para o uso de computadores conectados à rede, equipados com *softwares* editores de texto e impressoras. Porém, o processo em si pode, por muitas vezes, ainda ser cansativo e repetitivo. O reuso de questões já utilizadas e a formatação de toda a avaliação em si é um trabalho que pode ter um custo de tempo considerável.

Nesse cenário, um sistema que consiga gerar avaliações automaticamente, recebendo como parâmetros algumas configurações como a dificuldade de cada questão e o tempo necessário para realizar a prova, pode ser uma solução para essa tarefa. Tal sistema pode requerer um pouco mais de trabalho inicialmente, mas pode trazer benefícios para o professor nas próximas elaborações, como otimização do seu tempo, a praticidade e a confiabilidade da não repetição de questões.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver um sistema *web* para geração automática de avaliações. Cada avaliação será definida e configurada conforme a solicitação do usuário, com base em questões previamente cadastradas em um banco de dados.

Especificamente, pretende-se:

- Desenvolver uma interface de usuário intuitiva, ou seja, que permita que o usuário interaja com facilidade com o sistema, e que seja no idioma Português;
- Garantir a customização e formatação das questões e das avaliações visando uma estrutura mais organizada;
- Garantir que cada usuário tenha acesso apenas aos seus dados, assuntos e questões cadastrados em sua conta;
- Permitir uma interface responsiva, que entregue ao usuário visões adaptadas para os mais diversos sistemas e tamanhos de tela.

1.3 Organização do trabalho

Esse trabalho é organizado e distribuído em 5 capítulos, conforme descrito a seguir.

Neste capítulo, foi apresentada uma breve introdução, a problemática, o contexto e os objetivos definidos durante o desenvolvimento deste trabalho.

No Capítulo 2 é realizada as descrições e definições sobre as tecnologias e conceitos utilizadas no desenvolvimento, assim como a apresentação de trabalhos correlatos.

O Capítulo 3 apresenta as especificações do sistema, o levantamento de requisitos, os diagramas de caso de uso e o projeto do banco de dados.

Já o Capítulo 4 mostra o sistema em si, seus recursos e funcionalidades.

Por fim, as considerações finais e as sugestões de trabalhos futuros podem ser vistas no Capítulo 5.

2 Revisão Bibliográfica

Este capítulo apresenta os conceitos e definições acerca dos assuntos envolvidos durante o desenvolvimento do trabalho e uma análise comparativa entre trabalhos relacionados.

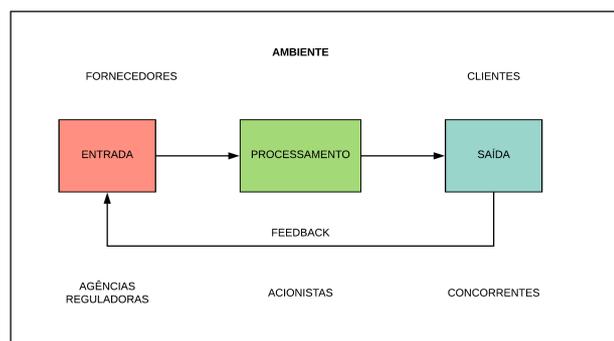
2.1 Sistemas de Informação

Diante de um mercado muito competitivo, as organizações têm investido em tecnologias para atingir seus objetivos, tendo, assim, que se modernizarem para acompanharem a evolução do mercado. Nesse sentido, mesmo ambientes educacionais e de ensino podem fazer uso de um Sistema de Informação (SI) para melhorar a eficiência dos seus processos.

De acordo com [Laudon e Laudon \(2011\)](#),

Um SI pode ser definido tecnicamente como um conjunto de componentes interrelacionados que coleta (ou recupera), processa, armazena e distribui informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização. Além de dar suporte à tomada de decisão, à coordenação e ao controle, esses sistemas também auxiliam os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e criar novos produtos. (p. 12)

Figura 1 – As funções básicas de um sistema de informação

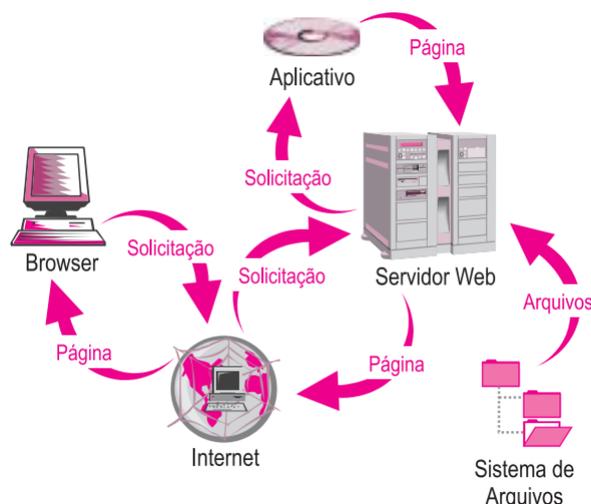


Fonte: Adaptada de [Laudon e Laudon \(2011, p. 13\)](#)

Esses sistemas contêm três atividades ou funções básicas, responsáveis por produzir as informações que as organizações utilizam em seu processo de tomada de decisões. São elas: a entrada, o processamento e a saída, representadas na Figura 1. A entrada é atividade de juntar e coletar dados. O processamento agrupa os dados coletados, analisa-os e transforma-os em algo com mais significado, ou seja, informação. Por sua vez, a saída retorna os resultados aos seus usuários para que sejam utilizadas conforme o seu propósito.

Um Sistema de Informação baseado em Web (SIW) é um sistema de informação que pode ser acessado através da *World Wide Web* (THALHEIM, 2013), para oferecer informações e serviços para usuários ou outros sistemas, independente de plataforma e localização. De acordo com Junior e Vidal (2006, p. 236), um SIW busca atender usuários identificados e que possuem visões específicas do sistema, e são geralmente relacionados a outros sistemas e bancos de dados, como mostra a Figura 2.

Figura 2 – Funcionamento básico de um sistema web



Fonte: Junior e Vidal (2006, p. 235)

O crescimento no número de aplicações voltadas para a *web* aumenta, principalmente devido à facilidade e menores custos de usabilidade, uma vez que as organizações não precisam investir em *hardware*, *software* ou atualizações, como mostra Alcântara (1998):

A grande conectividade da rede, aliada à facilidade de uso e acesso, à independência de plataforma obtida com os navegadores e a um modelo de disponibilização de aplicações que diminui os custos relacionados a implantação e distribuição de novas versões, transformaram a *web* na plataforma ideal para o desenvolvimento de aplicações com alto grau de escalabilidade. (p. 5)

De acordo com Junior e Vidal (2006, p. 431) um SIW é composto por muitas tecnologias diferentes, desde linguagens de programação a ferramentas de banco de dados e regras de negócios, o que exige um alto nível de especialização dos profissionais de Tecnologia da Informação (TI), pois seria difícil que um mesmo desenvolvedor tivesse conhecimento sobre todas as tecnologias utilizadas. Uma desvantagem disso é que, algumas vezes, é necessária a contratação de mais de uma empresa de TI para que um projeto seja concluído.

Estes conceitos estão presentes no desenvolvimento deste trabalho, que é um SIW. Pode-se perceber as três atividades, como já vistas na Figura 1. A entrada é composta

pelas questões, assuntos e configurações definidas pelo usuário no uso do sistema, o processamento é o tratamento e lógica que irá gerar a saída, ou seja gerar a avaliação ao final no *browser* do usuário.

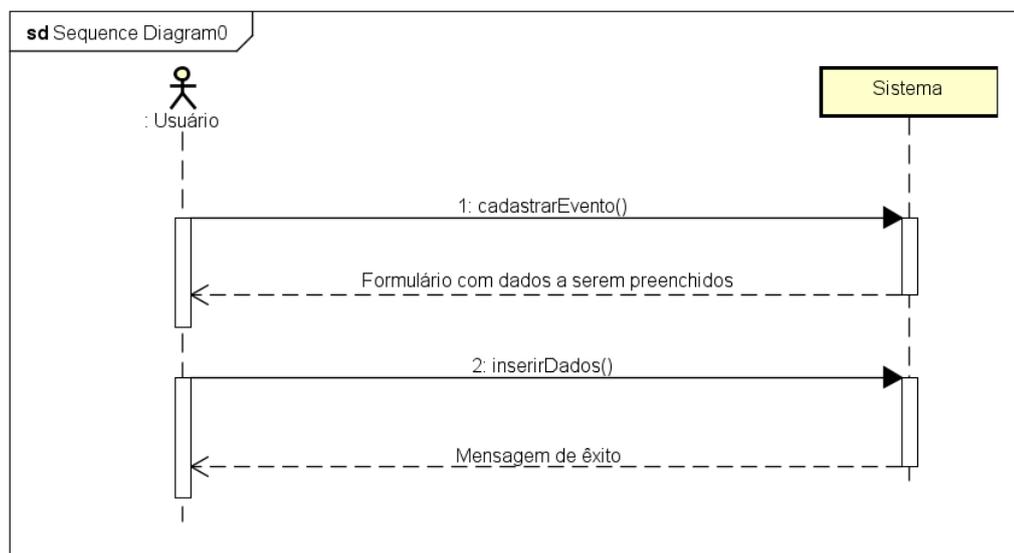
2.2 Linguagem de Modelagem Unificada

É característico do desenvolvimento dos sistemas de *software* terem a sua complexidade proporcional ao tamanho do sistema, fazendo-se, assim, necessário um planejamento inicial (BEZERRA, 2015). Essa necessidade faz com que haja a definição de um modelo, uma perspectiva ampla de como o sistema será construído. Para isso, o uso de uma notação padrão, textual e visual, é importante.

A Linguagem de Modelagem Unificada (UML)¹ é uma linguagem visual, independente de linguagens de programação, usada na elaboração da estrutura de um projeto de *software*, aplicada na sua visualização, construção e documentação, permitindo traduzir todos os processos da criação de um sistema de forma padronizada. Segundo Booch, Rumbaugh e Jacobson (2006), a UML possibilita uma forma-padrão para a definição de um projeto de arquitetura para sistemas, onde informa funções, processos, esquemas de bancos de dados, entre outros aspectos.

De acordo com Bezerra (2015), a UML é composta por elementos com sintaxe e semântica. A sintaxe é dada pela forma visual que o elemento possui, seja ela um círculo, um retângulo ou outra forma qualquer. A semântica, por sua vez, representa o propósito de cada elemento e quando ele deve ser utilizado.

Figura 3 – Exemplo de Diagrama de Sequência de Sistema



Fonte: Elaborado pelo próprio autor

¹ <https://www.uml.org/>

Na terminologia **UML**, os documentos criados durante a modelagem dos sistemas são chamados artefatos, que podem ser definidos pela utilização de diagramas, conforme mostra o exemplo na Figura 3, que representa um diagrama de sequência. Esse artefato demonstra as interações do usuário com o sistema (mensagens trocadas, processos etc) ao longo do tempo, mostrando como o sistema e suas classes se comportam e se complementam.

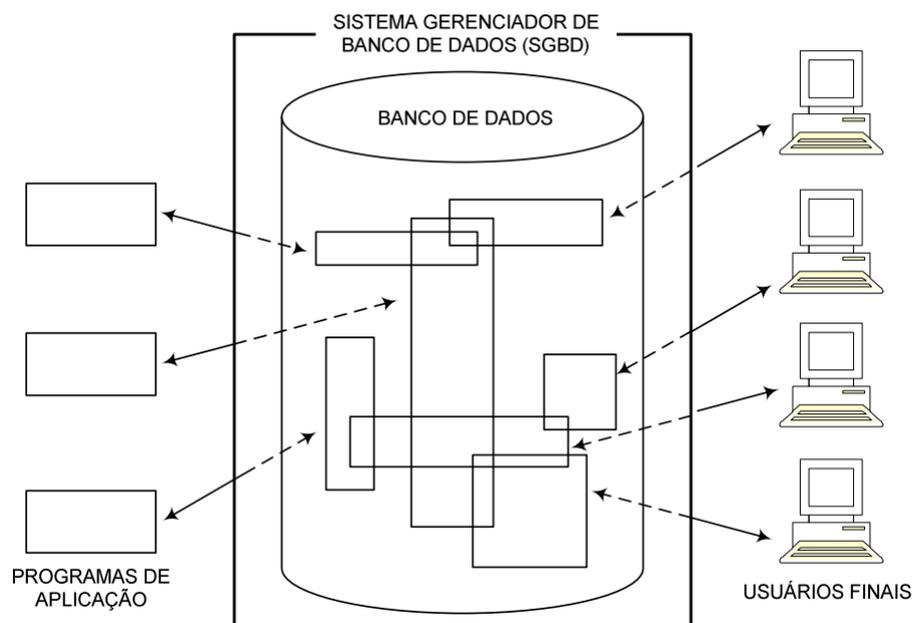
Para o desenvolvimento deste trabalho foram utilizados alguns artefatos UML, que serão apresentados mais detalhadamente no Capítulo 3: Análise de requisitos, Casos de Uso e Diagrama de Sequência.

2.3 Banco de Dados

Um Banco de Dados (**BD**) é um conjunto de elementos relacionados que contêm registros sobre algum objeto, mantendo-os organizados e consistentes, de forma a construir informação. De acordo Heuser (2009, p. 22), um **BD** é um agrupamento de arquivos que tem relações entre eles e que auxiliam na obtenção dos objetivos dos usuários, sendo uma peça importante de um **SI**.

Os grandes problemas da utilização de um **BD** são, de acordo com Silberschatz, Sundarshan e Korth (2016, p. 3): redundância e inconsistência de dados, dificuldade no acesso, isolamento, problemas de integridade e segurança. Esses problemas podem ser eliminados (ou reduzidos) através da utilização de um Sistema de Gerência de Bancos de Dados (**SGBD**).

Figura 4 – Representação de um SGBD



Fonte: Adaptado de Date (2004, p. 6)

Para [Elmasri e Navathe \(2011, p. 3\)](#), um **SGBD** pode ser dito como a união de vários programas e sistemas que permitem a criação e manutenção de um **BD**. "O SGBD é, portanto, um sistema de software de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre vários usuários e aplicações". A Figura 4 ilustra, o funcionamento de um **SGBD**, através de quatro componentes: dados integrados e compartilhados, hardware, software e usuários finais.

O modelo de um **BD** representa a sua estrutura lógica, determinando relações e restrições entre os dados e informando como serão acessados e armazenados. Pela definição de [Date \(2004, p. 14\)](#), "um modelo de dados é uma definição abstrata, autônoma e lógica dos objetos, operadores e outros elementos que, juntos, constituem a *máquina abstrata* com a qual os usuários interagem". Existem vários modelos de dados, porém, para este trabalho, será tratado apenas o relacional.

Para [Silberschatz, Sundarshan e Korth \(2016, p. 39\)](#), um **BD** relacional é uma coleção de tabelas unicamente nomeadas, onde cada linha (tupla, na terminologia matemática) representa uma relação entre um conjunto de valores, como representada pela Figura 5, que representa os dados de vários funcionários em uma tabela.

Figura 5 – Exemplo de tabela do Modelo Relacional

id	nome	dept_nome	salário
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Fonte: [Silberschatz, Sundarshan e Korth \(2016, p. 40\)](#)

Para a manipulação e consulta, os bancos de dados relacionais utilizam a Linguagem de Consulta Estruturada (**SQL**). Embora seja chamada como uma linguagem de consulta, a **SQL** pode fazer muito mais do que apenas consultar. ([SILBERSCHATZ; SUNDARSHAN; KORTH, 2016, p. 57](#)). Ela possui uma lista abrangente de comandos para definição de dados, consultas e atualizações e é responsável pelo sucesso do uso de bancos de dados, sendo utilizada por uma grande parte das organizações ([ELMASRI; NAVATHE, 2011, p. 149](#)).

2.4 Linguagem de Programação PHP

PHP² é uma linguagem de programação gratuita e de código aberto utilizada no desenvolvimento de aplicações para a *web*. Ela difere das outras linguagens, como a HTML, pois tem uma capacidade de interagir e transformar os *websites* em páginas mais dinâmicas. (NIEDERAUER, 2016, p. 21). Ela permite a incorporação de códigos HTML e é um facilitador para conexão com banco de dados do lado do servidor, sendo a maior parte do que ela realiza, invisível aos usuários.

Para Converse e Park (2003, p.10 e seg), dentre as razões para se utilizar PHP estão: o fato de ser gratuita e de código-aberto, não utilizar de *tags*, de ser estável e rápida. Além disso, a PHP trabalha bem com outras linguagens, adiciona novos recursos regularmente e é popular.

Lerdorf et al. (2002, p.10) mostra que PHP é uma linguagem muito flexível, simples e poderosa, capaz de ser interpretada em todos os sistemas operacionais. Por isso ela é tão utilizada e defendida na comunidade.

2.4.1 Laravel

Laravel³ é um *framework* para desenvolvimento em ambiente *web*, baseado na arquitetura *Model View Controller* (MVC), escrito em PHP. (MCCOOL, 2012, p. 3). Criado como uma forma rápida, livre e de código aberto, cujo objetivo inicial é permitir um desenvolvimento mais organizado e facilitado (HE, 2015/01, p. 301).

Laravel fornece todas as ferramentas necessárias para o início de uma programação em PHP, além de prover convenções que reduzem a quantidade de código que um desenvolvedor precisa escrever e, caso seja novo, entender (STAUFFER, 2019, p. 2).

2.5 Avaliações de aprendizagem

Além dos contextos técnicos envolvendo linguagens de programação e bancos de dados, também é importante considerar os conceitos que envolvem o processo de avaliação da aprendizagem, uma vez que fazem parte da criação e implementação deste trabalho.

Em sua obra, Darsie (1996, p. 48) diz que a avaliação é uma atividade integrada e indivisível a todo tipo de ação que procure atribuir melhorias e mudanças. Assim, ela faz parte da ação educativa, surgindo e atuando como um instrumento de impulso da educação e não apenas como uma simples forma de mensurá-la.

² <https://php.net/>

³ <https://laravel.com/>

Luckesi (2000, p. 3) mostra ainda que não é possível tomar uma decisão sem uma análise e um diagnóstico e não existe diagnóstico sem um processo para validá-lo. Ou seja, antes de mais nada é preciso quantificar e qualificar o estado de algo, baseado em suas especificações, para que seja possível compreender e agir da melhor maneira possível.

O processo de avaliação é algo que depende e deriva de regras formais e informais e deve levar em consideração atributos inter e intrapessoais. De acordo com Pacheco (1998),

As tarefas escolares que dizem respeito à avaliação do aluno são explícita e implicitamente contextualizadas pelas competências curriculares da administração central, da escola e do professor e ainda pelas relações deste com os alunos, os encarregados de educação e os colegas. Não é sem razão que os processos avaliativos do professor são explicados por uma prática que é comum a todos os professores ou a professores de áreas curriculares específicas. (p. 117).

2.6 Trabalhos relacionados

A seguir, são apresentadas informações e descrições de trabalhos anteriormente desenvolvidos e que possuem, de algum modo, finalidade parecida aos objetivos proposto para este trabalho.

2.6.1 Easy LMS

O Easy LMS⁴ (QUIZWORKS, 2019) é um sistema para gestão de aprendizagem que conta com versões gratuitas e pagas.

Para sua utilização é preciso criar uma conta, definir o plano e seguir por alguns tutoriais. O sistema é completamente em Português e tem uma interface intuitiva para o usuário, com diversas imagens, seções de ajuda e até mesmo um *chat online*.

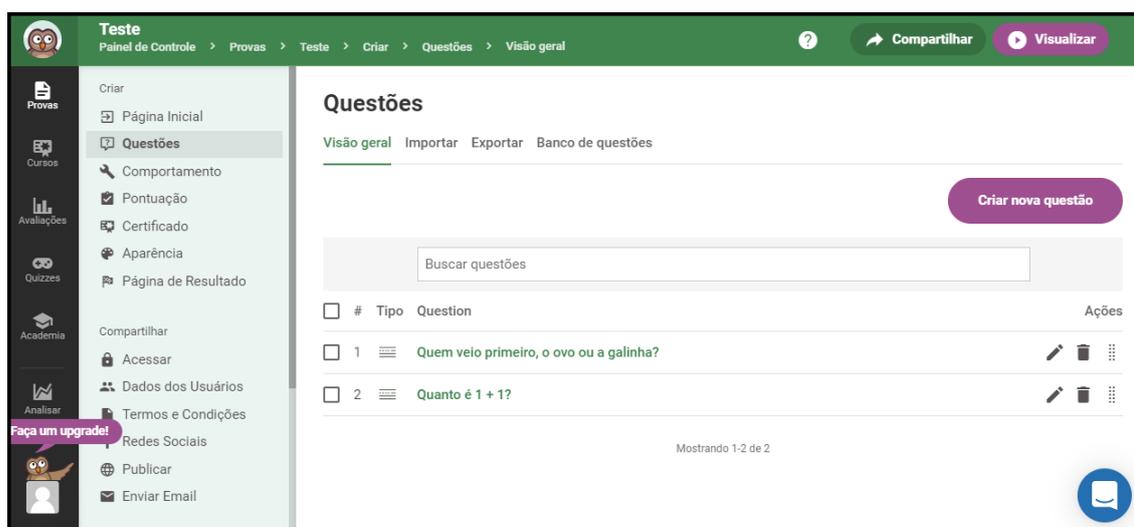
O sistema permite a criação e armazenamento *online* de avaliações e questões (múltipla escolha, espaços em branco, questão aberta e questão com imagens). No entanto, é permitido apenas a criação da avaliação, com a inserção manual de questões. A repetição de questões em mais de uma avaliação não é realizada.

Para o usuário que gostaria de ter acesso a um banco de questões pré-definidas pela comunidade, há a possibilidade de contratar o serviço a parte.

A Figura 6 apresenta a tela de visão geral do Easy LMS, mostrando duas questões cadastradas para uma avaliação.

⁴ <https://www.easy-lms.com>

Figura 6 – Tela de Visão Geral de Avaliação



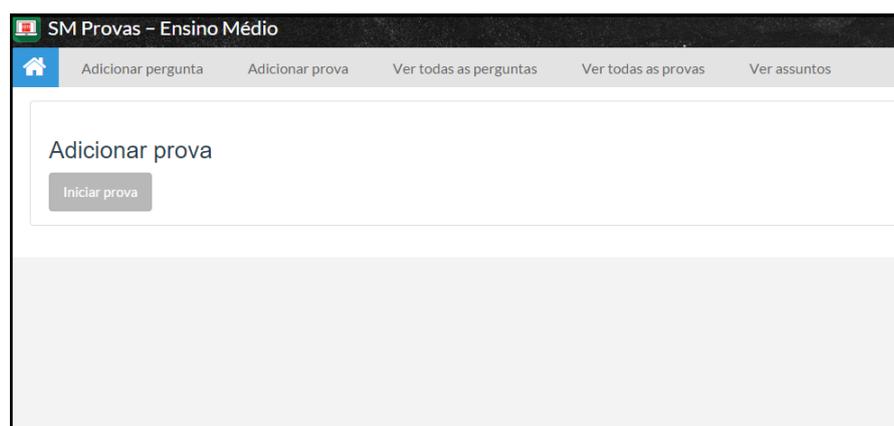
Fonte: [Quizworks](#) (2019)

2.6.2 SM Provas

O SM Provas⁵ ([SMBRASIL, 2019](#)) é um *software desktop*, gratuito e em português, usado para a criação de avaliações de forma simplificada.

Com o sistema instalado é possível criar uma nova prova, como mostra a Figura 7. Também é possível selecionar uma disciplina, um assunto e selecionar questões pré-cadastradas ou inserir novas questões manualmente.

Figura 7 – Tela de Início SM Provas



Fonte: [SMBrasil](#) (2019)

As provas são geradas com as questões selecionadas e há a opção de inserir cabeçalhos e de gerar versões diferentes para a prova, onde as questões têm a ordem alterada.

Entretanto, por se tratar de um sistema *desktop*, há a limitação de se precisar de um computador com o sistema instalado. Além disso, há a separação dos sistemas em

⁵ <http://simplificabrasil.smeducacao.com.br/>

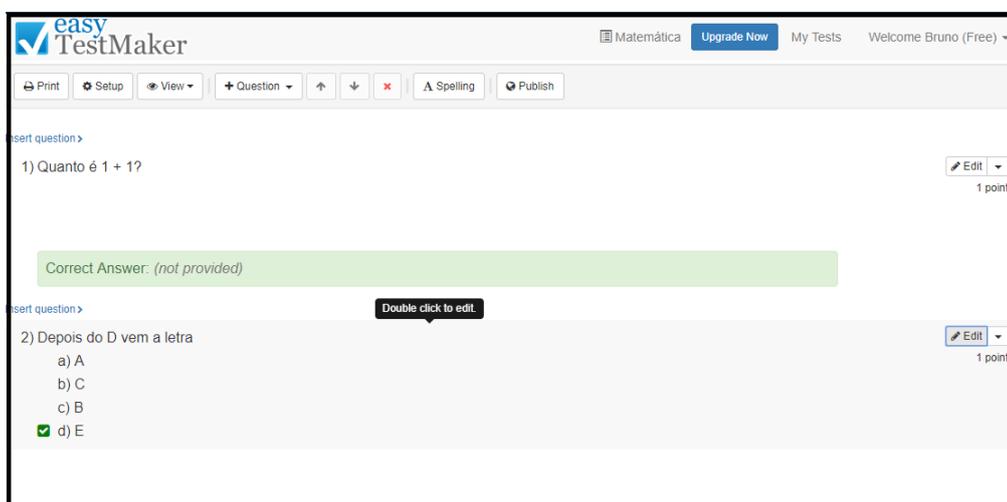
níveis de ensino (Médio e Fundamental), restringindo o uso de questões à aquele tipo.

2.6.3 Easy Test Maker

O Easy Test Maker⁶ (EASYTESTMAKER, 2019) é um sistema gerador de avaliações gratuito e que pode ser acessado através da *web*.

O sistema exige uma conta para ser acessado e não tem tradução para o português, ficando limitado ao inglês. Nele é possível criar e gerenciar avaliações de forma manual, como mostra a Figura 8, mas as questões ficam limitadas a cada prova, não podendo ser aproveitadas posteriormente. As questões podem ser de alguns tipos, como múltipla escolha ou abertas, mas não há a funcionalidade de inserir imagens, gráficos e tabelas.

Figura 8 – Tela de inserção de questões Easy Test Maker



Fonte: EasyTestMaker (2019)

Há ainda a opção de adquirir planos de versões comerciais, acrescentando algumas funcionalidades como a permissão para inserir imagens nas questões, alterar a ordem e também aumentar o número máximo de avaliações que podem ser armazenadas.

2.6.4 Avalog

Avalog (DELFINO, 2010) é um *software* gratuito, produto de um trabalho de conclusão de curso, e que foi tomado como base para o desenvolvimento deste trabalho. Ele é totalmente em português e gera avaliações de forma automática, bastando apenas que o usuário cadastre as questões em um banco de dados.

As questões podem ser abertas ou fechadas, além de possuírem algumas configurações como dificuldade e tempo médio de resposta, conforme mostra a Figura 9. Com as questões cadastradas, é possível solicitar ao *software* que gere automaticamente, ou seja,

⁶ <https://www.easytestmaker.com/>

que selecione as melhores questões para preencher a avaliação. Há ainda a possibilidade de editar e conferir as questões já cadastradas, bem como assuntos e respostas.

Figura 9 – Tela de inserção de questões Avalog



Fonte: Delfino (2010, p. 56)

Por ter sido usado como base para o trabalho juntamente com os outros trabalhos relacionados, a Tabela 1 exibe um comparativo entre eles, de acordo com alguns requisitos considerados importantes.

Tabela 1 – Comparativo entre trabalhos relacionados

Características	Easy LMS	SM Provas	Easy Test Maker	Avalog
Gratuito	Versão de Testes	Sim	Versão limitada	Sim
Cadastro de questões	Sim	Sim	Sim	Sim
Reuso de questões	Não	Sim	Não	Sim
Versão para web	Sim	Não	Sim	Não
Geração automática	Não	Não	Não	Sim
Configurações de questões	Sim	Não	Não	Sim
Formatação/Imagem	Não	Formatação	Versão Paga	Imagem

Fonte: Elaborado pelo próprio autor

3 Especificações do Sistema

O principal objetivo deste trabalho é a construção de um sistema para geração automática de avaliações, com base em questões previamente cadastradas e parâmetros informados pelos usuários. Dado o uso do Avalog (DELFINO, 2010), o sistema foi nomeado *Avalog 2.0*. O sistema foi projetado para permitir o cadastro de assuntos; o cadastro de questões objetivas e discursivas; opções de personalização e formatação, incluindo o uso de imagens e tabelas; além da definição de parâmetros como o tempo estimado para resolução da avaliação e o grau de dificuldade de cada questão, por exemplo.

Um dos primeiros requisitos levantados foi a necessidade de ser um sistema *web*, que fosse totalmente em português e que tivesse opções para formatar os textos das questões. Esses requisitos foram levantados e analisados baseados nos trabalhos relacionados descritos na Seção 2.6 e nos objetivos (Seção 1.2). Para tal fim foram gerados alguns artefatos da Linguagem de Modelagem Unificada (UML) e de Engenharia de Software, tais como: Análise de Requisitos, Diagrama de Casos de Uso, Diagrama de Sequência, Diagrama de Classe e Diagrama Entidade-Relacionamento. Os artefatos foram gerados com o Astah UML Students¹.

3.1 Análise de Requisitos

A etapa de levantamento de requisitos é importante, pois é nela onde será compreendido o problema e definidas as funcionalidades que irão suprir as necessidades do usuário. De acordo com Bezerra (2015), os requisitos podem ser considerados funcionais, que são as funcionalidades do sistema, e não-funcionais, dadas como as características de qualidade do sistema.

A Tabela 2 apresenta os requisitos funcionais do Avalog 2.0. Cada requisito teve sua prioridade classificada em: Essencial, Importante ou Desejável. Mais detalhes dessa classificação pode ser vista em Bezerra (2015).

¹ <http://astah.net/>

Tabela 2 – Requisitos Funcionais do Sistema

Nome	Descrição	Prioridade
Geração Automática	Capacidade de entregar ao usuário avaliações geradas de forma automática	Essencial
Cadastro de Questões	Capacidade de armazenar questões em um banco de dados	Essencial
Cadastro de Assuntos	Capacidade de armazenar assuntos em um banco de dados	Essencial
Geração de Gabaritos	Sistema deve entregar ao usuário um gabarito para cada avaliação gerada	Importante
Avaliações Editáveis	Sistema deve permitir que usuário edite a avaliação antes de imprimir	Desejável
Avaliações diferentes	O sistema deverá ser capaz de gerar avaliações diferentes a cada vez	Importante

Fonte: Elaborado pelo próprio autor

Os requisitos não-funcionais do sistema, por sua vez, estão mostrados na Tabela 3.

Tabela 3 – Requisitos Não-funcionais do Sistema

Nome	Descrição	Prioridade
Língua: Português	O Sistema deve ser todo em Português	Importante
Autenticação	Sistema só poderá ser acessado por usuários logados	Essencial
Editor de Texto	Capacidade de inserir questões estilizadas e formatadas através de um editor de texto	Desejável
Acesso Multiplataforma	Sistema deve ser compatível com os diversos sistemas e softwares disponíveis	Desejável
Integridade do Dados	Dados devem ser armazenados com precisão e consistência	Essencial
Controle de Acesso	Cada usuário só deverá ter acesso a dados pertinentes a ele	Essencial

Fonte: Elaborado pelo próprio autor

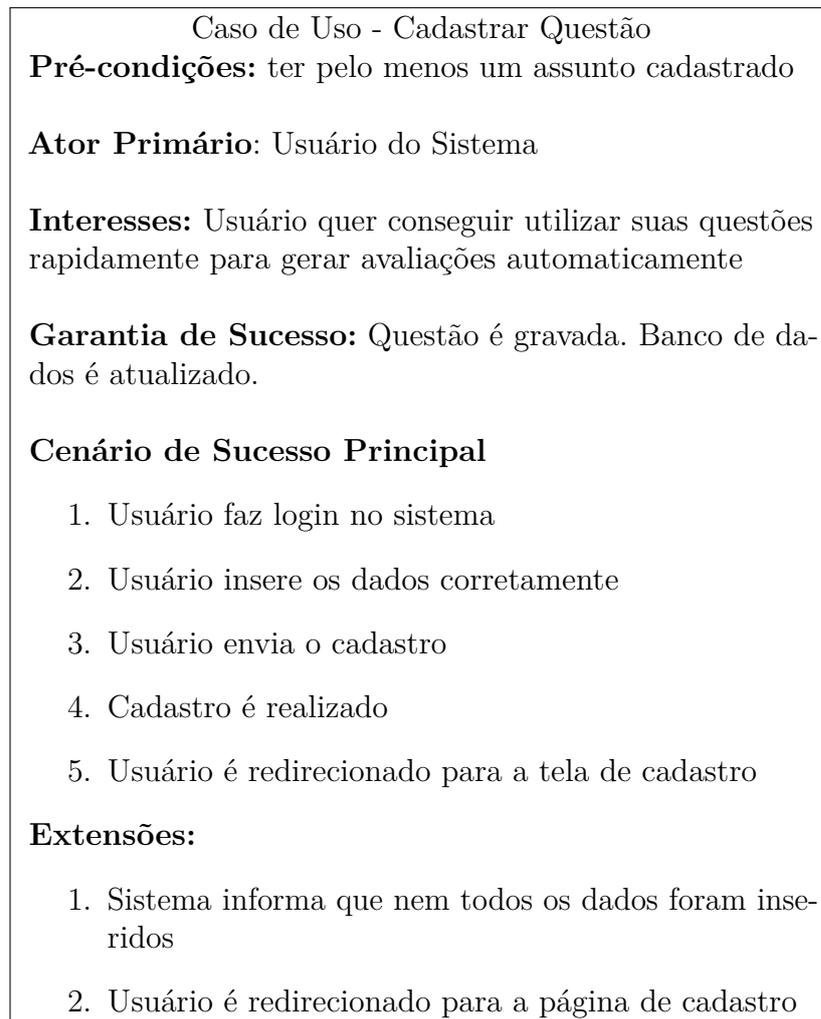
3.2 Casos de Uso

De acordo com Fowler (2014, p. 104), um caso de uso é uma técnica para demonstrar os requisitos funcionais de um sistema, descrevendo suas interações com os usuários e com o próprio sistema, de forma narrativa. Booch, Rumbaugh e Jacobson (2006) afirmam

ainda que os casos de uso são uma forma de ajudar a validar a arquitetura e verificar se o sistema está de acordo com seus requisitos ainda em seu desenvolvimento.

Não existe um padrão para os casos de uso e diferentes formatos funcionam para diferentes problemas. O importante é que eles sejam capazes de expressar aquilo que se propõem (FOWLER, 2014, p. 105). A Figura 10 apresenta o caso de uso Cadastrar Questão, onde o ator é o professor. Outros casos de uso do sistema podem ser encontrados no Apêndice A

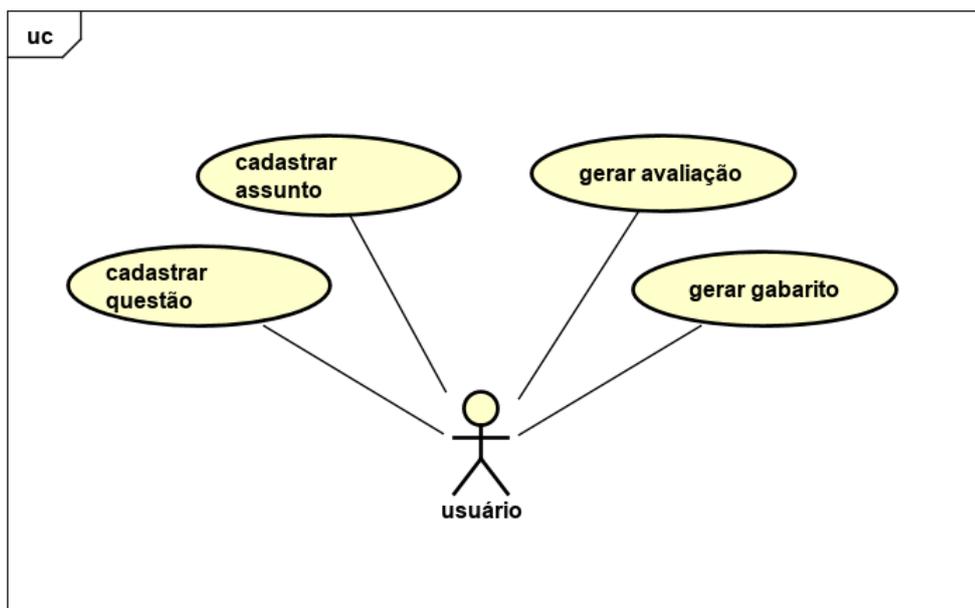
Figura 10 – Caso de Uso - Cadastrar Questão



Fonte: Elaborado pelo próprio autor

A Figura 11 apresenta um diagrama de caso de uso, que demonstra as funcionalidades e as interações do sistema do ponto de vista do usuário. Nele não são apresentados detalhes técnicos que apontam como o sistema deve fazer para funcionar.

Figura 11 – Diagrama de Caso de Uso



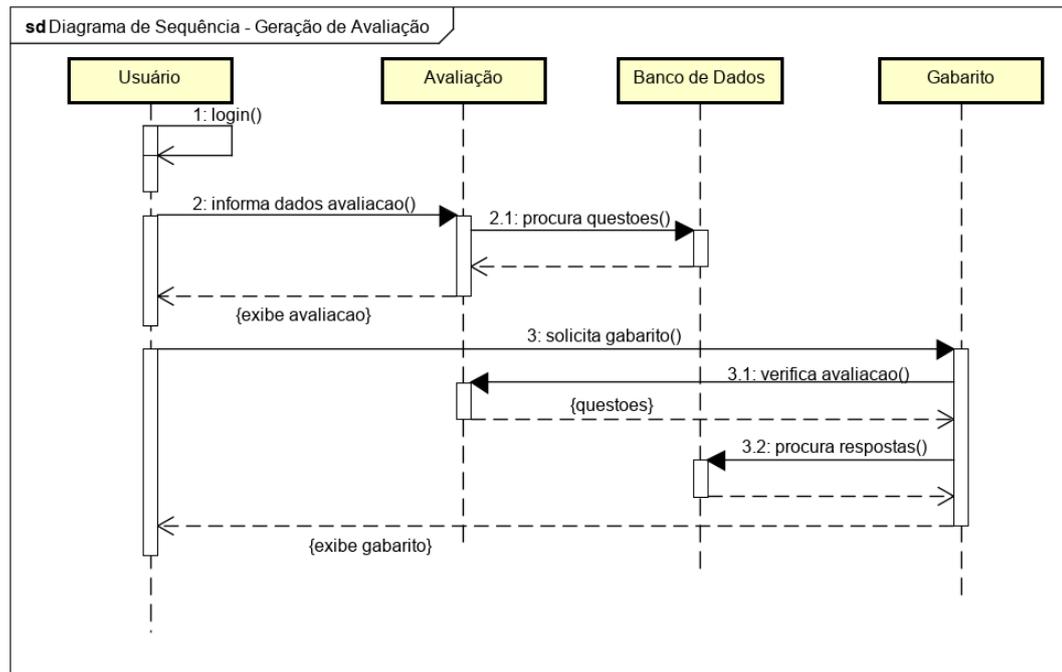
Fonte: Elaborado pelo próprio autor

3.3 Diagrama de Sequência

Um diagrama de sequência é focado na ordem temporal dos eventos. Ele é utilizado para demonstrar a visão dinâmica de um sistema e deve ser criado baseado nos casos de uso. Assim, pode-se dizer que esse diagrama desenha os casos de uso de um sistema, mostrando a comunicação entre os objetos que o compõem (BOOCH; RUMBAUGH; JACOBSON, 2006).

A Figura 12 apresenta um diagrama de sequência para o caso de uso *Gerar Avaliação*, onde é possível ver cronologicamente como o sistema se comporta e se comunica para gerar uma avaliação. Os demais diagramas utilizados neste trabalho estão apresentados no Apêndice B.

Figura 12 – Diagrama de Sequência - Gerar Avaliação



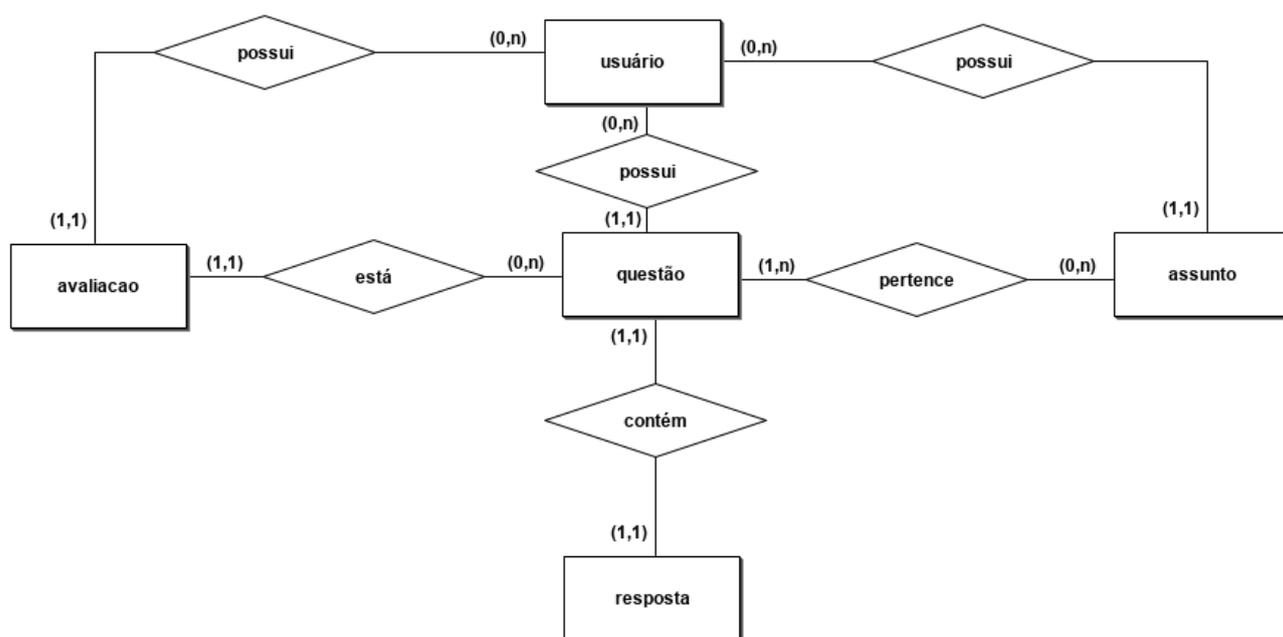
Fonte: Elaborado pelo próprio autor

3.4 Diagrama Entidade-Relacionamento

Um diagrama Entidade-Relacionamento permite representar a estrutura lógica de um banco de dados relacional, mostrando como os componentes (entidades), se relacionam entre si dentro do sistema, demonstrando suas exigências e dependências (SILBERSCHATZ; SUNDARSHAN; KORTH, 2016).

Esses diagramas são normalmente gerados para uso em bancos de dados relacionais. A Figura 13 ilustra o diagrama entidade-relacionamento elaborado para este trabalho. Vale ressaltar que ele não se encontra nos padrões propostos pelo *Laravel*, mas sua utilização foi permitida através de modificações e ajustes no código fonte do sistema.

Figura 13 – Diagrama Entidade-Relacionamento



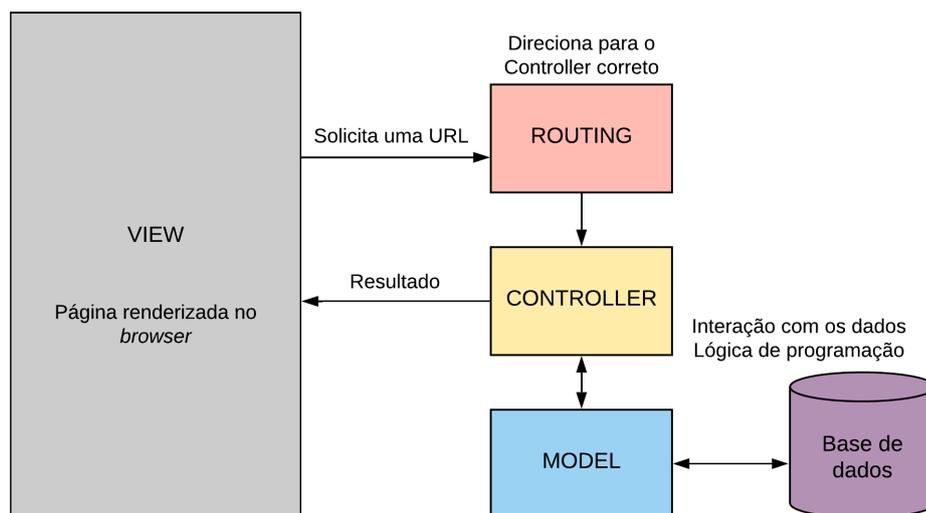
Fonte: Elaborado pelo próprio autor

4 Avalog 2.0

Após as análises de requisitos, definições de objetivos e estudos dos conceitos e tecnologias que seriam aplicadas, foi implementado o sistema web Avalog 2.0, baseado no Avalog (DELFINO, 2010).

O sistema permite o cadastro de questões discursivas e objetivas (múltipla escolha), cadastro de assuntos, a geração de avaliações automaticamente, utilizando um algoritmo desenvolvido como uma melhoria do exposto em Delfino (2010), gabaritos para elas, além de uma maneira de visualizar as avaliações criadas. Ele está na linguagem Português e em ambiente *web*, inclusive com suporte a dispositivos móveis.

Figura 14 – Arquitetura utilizada pelo Laravel



Fonte: Adaptado de Blumenstein et al. (2017, p. 2)

Para o seu desenvolvimento foi utilizado o *framework* PHP Laravel, que utiliza a arquitetura MVC. Essa arquitetura separa a lógica de programação da interface do usuário. Isso é feito separando a aplicação em três partes: *Models* que gerenciam comportamentos e dados fundamentais da aplicação, *Views* que exibem a interface com os dados renderizados e *Controllers* que recebem entradas de usuários e processam essas entradas chamando interações com os *models* e *views*. Essa arquitetura está descrita de forma simplificada na Figura 14.

Como apoio para o desenvolvimento foi utilizado o *software* editor de código-fonte, livre e de código aberto, Visual Studio Code versão 1.35.1, juntamente com o SGBD relacional, MySQL.

Caso o usuário ainda não esteja logado, a tela inicial será a tela de *login*, conforme

mostra a Figura 15. Em caso de perda de senha, será possível solicitar a recuperação por *email*. O *login* será realizado através de *email* e senha e, caso o usuário não esteja cadastrado, poderá se registrar, inserindo seu nome completo, *email* e senha.

Figura 15 – Tela de Login Avalog 2.0

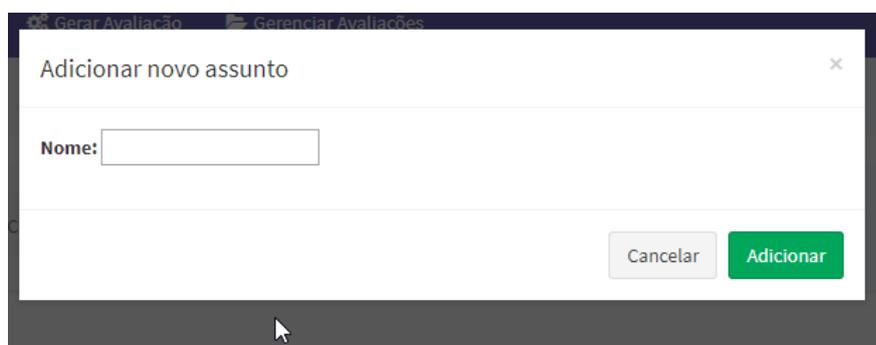
A screenshot of the login page for Avalog 2.0. The page has a light gray background. At the top center, the text "Avalog 2.0" is displayed in a bold, dark font. Below this, a white rectangular box contains the login form. The form has the heading "Entre para iniciar uma nova sessão". It features two input fields: the first is for an email address, with "bruno@gmail.com" entered; the second is for a password, with "....." entered. To the right of each field is a small icon (an envelope for email and a padlock for password). Below the password field is a checkbox labeled "Me manter conectado" which is currently unchecked. To the right of the checkbox is a blue button labeled "Entrar". At the bottom of the form, there are two links: "Esqueci minha senha" and "Registrar um novo membro".

Fonte: Elaborado pelo próprio autor

4.1 Cadastro de Questões e Assuntos

Para o caso de o usuário já estar logado, ou após efetuar *login*, a tela inicial será a tela de cadastro de questão. Além do menu na parte superior, comum à todas as telas, será apresentado os passos para o cadastro. Inicialmente é preciso inserir um ou mais assuntos ou então cadastrar um novo para poder usá-lo, conforme a Figura 16.

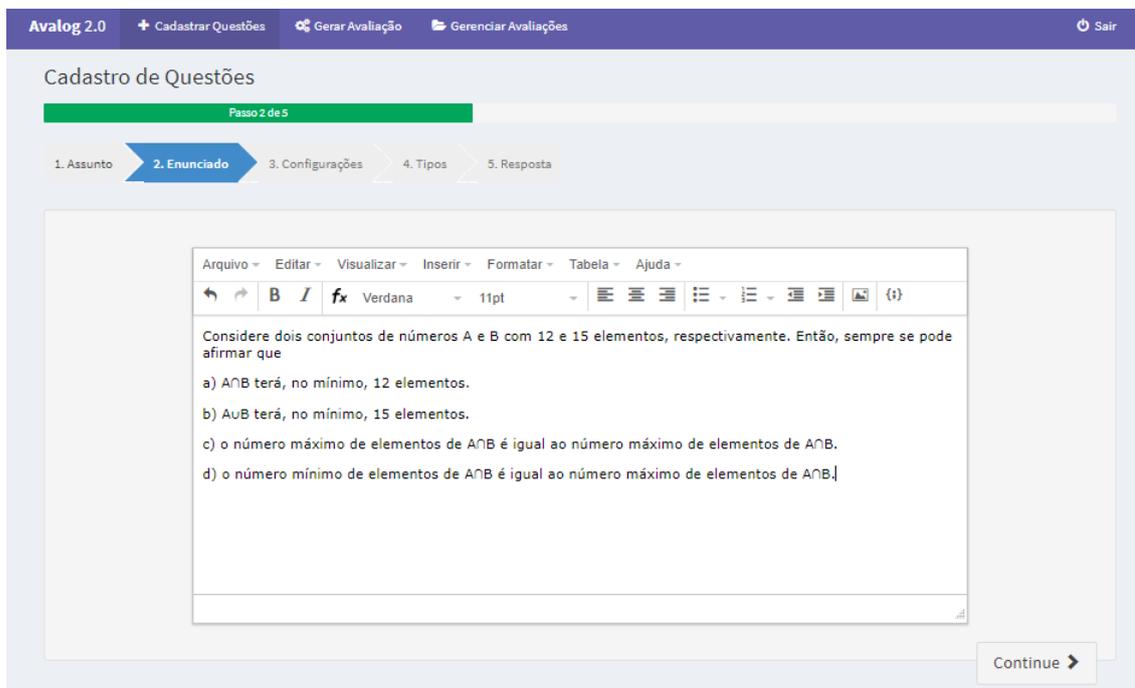
Figura 16 – Tela de Cadastro de Assunto - Avalog 2

A screenshot of a web browser window showing a modal dialog box titled "Adicionar novo assunto". The dialog has a close button (an 'x' icon) in the top right corner. Inside the dialog, there is a label "Nome:" followed by an empty text input field. At the bottom right of the dialog, there are two buttons: a gray "Cancelar" button and a green "Adicionar" button. The browser's address bar and tabs are visible at the top of the window.

Fonte: Elaborado pelo próprio autor

Em seguida, o usuário precisa inserir o enunciado através do editor de texto presente no próprio sistema, algumas configurações como tempo médio de resolução, dificuldade e, com base na escolha do tipo de questão (discursiva ou objetiva), a resposta ou uma alternativa correta, conforme mostra a Figura 17.

Figura 17 – Tela de Cadastro de Questão - Avalog 2



Fonte: Elaborado pelo próprio autor

4.1.1 Editor de Texto

Uma das características do Avalog 2.0 é o seu editor de texto. Ele foi implementado utilizando a ferramenta *TinyMCE*, um editor de texto *online* e de código aberto, escrito em *Javascript*. Ele permite a formatação completa de textos, inserção de imagens, tabelas, caracteres especiais, estilos, equações e códigos. Através dele é possível inserir um texto formato e previamente copiado de outro local (Internet, Arquivos de Texto etc) e ainda assim manter suas configurações.

Ele gera um código HTML que é salvo no banco de dados como sendo o enunciado da questão. Esse mesmo código pode ser reinterpretado pelo editor futuramente ou diretamente pelo *browser*, exibindo o texto no estilo em que foi concebido.

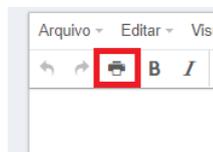
Infelizmente ele não dá suporte ao *upload* de imagens diretamente do computador, sendo necessário inserir imagens que já estejam armazenadas em algum servidor pela *Internet*. Há a possibilidade de comprar um *plugin* para adicionar essa funcionalidade, mas para os objetivos desse trabalho, não se viu necessário.

Caso acessado por um dispositivo móvel, configurado para exibir o *site* de acordo com este ambiente, o editor de texto se tornará uma versão mais compacta, com menos funcionalidades adicionais, mas que atende a maioria das necessidades do usuário.

4.2 Geração de Avaliações

Uma vez que assuntos e questões estejam cadastradas é possível gerar uma nova avaliação. Para isso, basta acessar a tela representada pela Figura 19, inserir um nome, selecionar um ou mais assuntos e definir algumas configurações, como tempo, dificuldade, tipo (Lista de Exercícios, Avaliação e Exame Especial), e o número de questões de cada nível (Fácil, Médio e Difícil). Em caso de sucesso, a tela será redirecionada para um editor de texto, desta vez já contendo as questões selecionadas automaticamente. Esse passo é importante, pois permite ao usuário um nível maior de personalização, dando a ele a oportunidade de fazer alguns ajustes nas questões antes de imprimir. Para imprimir, basta selecionar o botão referente à impressora, como na Figura 18. Em alguns sistemas, pode ser preciso configurar a página para remover os metadados que aparecerão no cabeçalho e rodapé.

Figura 18 – Indicativo de Botão de Impressão - Avalog 2



Fonte: Elaborado pelo próprio autor

Logo abaixo da avaliação há o botão *Gerar Gabarito*, que retorna um arquivo no formato *PDF* contendo as respostas para as questões que constam na avaliação gerada, na ordem em que foram apresentadas inicialmente. Essas respostas podem ser textos ou alternativas, de acordo com o tipo de questão selecionada.

Vale lembrar que em alguns casos há a possibilidade de a avaliação conter menos questões que a quantidade informada pelo usuário. Isso se dará devido tanto a um número pequeno de questões cadastradas no sistema quanto às definições de tipo de avaliação. Esse processo será descrito melhor na Seção 4.2.1.

É importante entender que a qualidade das avaliações depende muito mais da diversidade do banco de dados do que do sistema Avalog 2.0 em si. Um usuário que possuir um número pequeno de questões cadastradas dificilmente terá avaliações diferentes geradas uma após a outra utilizando os mesmos critérios.

Figura 19 – Tela de Geração Avaliação - Avalog 2

Avalog 2.0 + Cadastrar Questões Gerar Avaliação Gerenciar Avaliações Sair

Gerar Avaliação

Gere sua avaliação conforme suas configurações.
Selecione um ou mais assuntos, defina o tempo e o número de questões

Nota: Leve em consideração o tipo de avaliação que deseja gerar e suas limitações. Em alguns casos, o número de questões pode ser menor do que o definido

Nome
Lista de Matemática 2019/02

Assuntos
x Álgebra x Conjuntos x Matemática

Tipo
Lista de Exercícios

Tempo
60

Quantidade de Questões Fáceis
2

Quantidade de Questões Médias
2

Quantidade de Questões Difíceis
1

Gerar

Fonte: Elaborado pelo próprio autor

4.2.1 Algoritmo Gerador de Avaliações

A seguir será apresentado o *pseudocódigo* para o algoritmo utilizado e adaptado de Delfino (2010, p. 65). A principal mudança foi o controle baseado no tipo de avaliação, onde agora cada tipo de avaliação (Lista de Exercícios, Avaliação ou Exame Especial) tem um critério de escolha de questões, o que não era considerado em sua versão inicial.

Esse algoritmo recebe como parâmetros os dados passados no momento da geração da avaliação, verifica e lista todas as questões pertencentes àqueles assuntos e ao usuário que solicitou, as ordena por ordem de último uso (da mais antiga para a mais recente) e então as trata para finalmente gerar a avaliação.

Alguns aspectos desse algoritmo requerem certa atenção: caso o usuário não informe um valor para *tempo* ou para os números de questões para os três tipos, será considerado apenas aquele preenchido. Por exemplo, caso o usuário não informe uma quantidade de minutos esperada para a avaliação, mas informar uma quantidade de questões para os tipos, então o tempo será desconsiderado. O mesmo vale para o sentido contrário, ou seja, caso o usuário não informe o número de nenhum tipo de questão, então será considerado apenas o tempo. A inserção das questões acontecerá enquanto um desses (ou os dois) não for contemplado. Vale ressaltar que, caso não haja mais questões disponíveis, essas condições são desconsideradas, mesmo que não tenham sido contempladas.

```

1: função GERARAVALIACAO(tempo, nome, numeroQuestoesF, numeroQuestoesM, nu-
   numeroQuestoesD, assuntos, tipo)
2:   totalQuestoes ← numeroQuestoesF + numeroQuestoesM + numeroQuestoesD
3:   numeroQuestoes ← 0
4:   se tipo = lista de exercicio então
5:     continue com todas as questões
6:   fim se
7:   se tipo = avaliacao então
8:     apenas metade das questões podem ter sido usadas como lista de exercicio no
       último uso
9:   fim se
10:  se tipo = exame especial então
11:    nenhuma das questões pode ter sido usada em uma lista de exercicio em seu
       último uso
12:  fim se
13:  enquanto houver questoes faça
14:    se houver tempo e numeroQuestoes for menor que totalQuestoes então
15:      se dificuldade for facil então
16:        se numeroQuestoesF > 0 então
17:          adiciona questao
18:          tempo = tempo - questao.tempo
19:          numeroQuestoes ++
20:          numeroQuestoesF --
21:        fim se
22:      fim se
23:      Os passos descritos nas linhas 15 a 22 são repetidos para cada dificuldade
       (média e difícil).
24:    fim se
25:  fim enquanto
26: fim função
27: devolve lista de questoes selecionadas

```

Fonte: Elaborado pelo autor

O controle de tipos descrito nas linhas 4 a 12 permite que as avaliações dos tipos *Avaliação* e *Exame Especial* sejam mais restritivas, especialmente para o último caso, não permitindo que todas as questões tenham sido usadas em *Listas de Exercício* anteriormente.

Para fins de teste, duas questões foram geradas em sequência utilizando as mesmas configurações. Esses testes foram feitos utilizando um banco de dados que continha 33 questões, das quais 13 eram de nível fácil, 10 de nível médio e 10 de nível difícil, todas pertencentes ao mesmo usuário e aos mesmos assuntos. As configurações utilizadas foram:

- Assuntos: Álgebra, Matemática, Conjuntos
- Tipo: Avaliação
- Tempo: 60 minutos
- Quantidade de Questões Fáceis: 2
- Quantidade de Questões Médias: 1
- Quantidade de Questões Difíceis: 1

As avaliações foram geradas conforme as figuras 26 e 25, disponíveis no **Apêndice C**, com questões diferentes entre elas. Neste caso, ambas as condições (tempo e quantidade) foram contempladas.

5 Conclusão

Este trabalho apresentou o processo de desenvolvimento de um sistema *web*, o Avalog 2.0, usado para a geração de avaliações de forma automatizada, criando-as conforme solicitado pelo usuário. Esse sistema é gratuito, completamente em português, desenvolvido para funcionar na *web*, com uma interface intuitiva e que permita a formatação dos textos das questões e avaliações conforme a necessidade.

Baseado no levantamento de requisitos e nos resultados obtidos, é possível concluir que o sistema é capaz de gerar as avaliações automaticamente e com diferenciações, assim como o cadastro de assuntos e questões, suas configurações e respostas atendendo assim aos requisitos funcionais.

O sistema está disponível¹ em um servidor *web* gratuito, assim como seu código². Ele funciona em todos os sistemas operacionais, inclusive os de dispositivos móveis, desde que conectados à Internet e utiliza um banco de dados, também gratuito, para armazenar os dados do usuário, garantindo assim que um usuário consiga acessar suas informações independente do local de acesso.

O Avalog 2.0 permite que apenas o usuário dono da informação a acesse, garantindo sua segurança. Dessa maneira, o sistema só pode ser acessado através de autenticação (*login* e senha) e toda tentativa de acesso a dados de outros usuários será impedida, tratada e notificada através de alertas.

Tendo como requisito não-funcional essencial a capacidade de inserir questões estilizadas e formatadas, o Avalog 2.0 traz um editor de texto completo, fornecendo os principais mecanismos de edição: estilos de fonte, cor, tabelas, figuras, caracteres especiais e equações, fazendo assim com que o usuário não precise se preocupar em fazer grandes mudanças a cada avaliação gerada.

Em se tratando de possibilidades no momento de geração, o usuário fica preso às limitações do tipo de avaliação (Lista de Exercício, Avaliação ou Exame Especial), e no tipo de questão (Objetiva ou Discursiva). No entanto, para o caso das questões, é possível contornar a situação adaptando o enunciado para aquilo que o sistema pode oferecer. Ainda assim, mesmo com as mesmas configurações (e desde que haja uma base de dados capaz), o sistema consegue gerar avaliações diversificadas em gerações consecutivas.

Em relação ao trabalho baseado, o Avalog de Delfino (2010), o *Avalog 2.0* possui algumas melhorias: roda em ambiente *web* e não se limita a um único computador, possui um editor de texto com opções de formatação e estilização dos enunciados das questões,

¹ <https://avalog.herokuapp.com/>

² <https://github.com/BrunoCamarda/TCC-Geracao-Automatica-de-Avaliacao>

permite uma restrição em relação ao tipo das avaliações e ainda o reuso de avaliações.

Portanto, conclui-se que os objetivos deste trabalho foram atendidos e ele foi realizado com sucesso, ainda que com algumas limitações, ficando para realizações futuras a possibilidade de aprimorá-las.

5.1 Trabalhos futuros

Como possíveis trabalhos futuros pode-se apontar alguns aprimoramentos e mesmo a adição de novas funcionalidades.

O estudo da plataforma Moodle³ como trabalho relacionado, uma vez que ela possui a funcionalidade de gerar e gerir avaliações.

O editor de texto do Avalog 2.0 armazena as questões em um formato HTML de forma que, quando a avaliação é gerada, ela sempre estará com uma quebra de linha entre o número indicando a questão e o enunciado, sendo necessário que o usuário edite manualmente a avaliação antes de poder finalizá-la de fato. Uma possível melhora seria garantir que isso não aconteça e a avaliação já seja criada totalmente pronta para ser utilizada.

Outra limitação do próprio editor de texto, e consequentemente do próprio sistema, é o impedimento de inclusão de imagens diretamente do computador, sendo necessário que a imagem esteja hospedada e disponível na *web* para que seja introduzida no enunciado de uma questão. Atualmente o editor possui um *plug-in* pago para a adição dessa funcionalidade, mas desenvolver um próprio pode ser mais interessante.

Por se tratar de um serviço *web*, pode ser interessante que os usuários consigam compartilhar questões entre eles. Assim, a implementação de uma maneira de solicitar a outros usuários esse compartilhamento pode ser uma maneira de facilitar ainda mais esse processo. Atualmente cada usuário consegue acessar apenas dados pertencentes a ele.

Durante o cadastro de questões é possível apenas defini-las como objetivas ou discursivas, eliminando tipos comuns como "verdadeiro ou falso", "complete a lacuna" e ainda objetivas com mais de uma alternativa correta. Além disso, poder alternar a ordem das alternativas para garantir uma maior diferenciação entre avaliações.

Outro ponto importante é que, ao gerar uma avaliação, um usuário fica preso aos três tipos possíveis, Lista de Exercícios, Avaliação e Exame Especial e suas particularidades já definidas no algoritmo. Essas configurações podiam ser abertas para que o usuário definisse de acordo com a necessidade dele.

Como em alguns casos professores geram diversas avaliações diferentes para uma

³ <https://moodle.org/>

mesma turma, pode ser útil que haja a possibilidade de se utilizar um "esqueleto" de questão, onde a cada iteração, os valores desta seja alterado, sem que o enunciado seja alterado completamente ou sem que haja a necessidade de se cadastrar várias vezes a mesma questão, apenas alterando alguns dados.

Referências

- ALCÂNTARA, A. A. d. *Desenvolvimento de sistemas de informacção baseados na Web: uma abordagem utilizando o ciclo de vida espiral*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Recife, Dezembro 1998. Citado na página 17.
- ARAN, A. et al. *Evaluación como ayuda al aprendizaje*. Barcelona: EDITORIAL GRAO, 2000. (EDITORIAL POPULAR). ISBN 9788478275779. Citado na página 13.
- BEZERRA, E. *Princípios de Análise e Projeto de Sistema com UML*. Rio de Janeiro: Elsevier Editora Ltda., 2015. ISBN 9788535226270. Citado 2 vezes nas páginas 18 e 26.
- BLUMENSTEIN, K. et al. Livevis: Visualizing results of second screen surveys in real time at tv stages. Phoenix, USA, 10 2017. Citado na página 32.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. Rio de Janeiro: Elsevier Brasil, 2006. ISBN 9788535217841. Citado 3 vezes nas páginas 18, 27 e 29.
- CONVERSE, T.; PARK, J. *PHP: a bíblia*. Rio de Janeiro: Editora Campus, 2003. ISBN 9788535211306. Citado na página 21.
- DARSIE, M. M. P. Avaliação e aprendizagem. *Cadernos de pesquisa*, Fundação Carlos Chagas, São Paulo, n. 99, p. 47–59, 1996. Citado na página 21.
- DATE, C. *Introdução a sistemas de bancos de dados*. Rio de Janeiro: ELSEVIER EDITORA, 2004. ISBN 9788535212730. Citado 2 vezes nas páginas 19 e 20.
- DELFINO, T. A. *Avalog: Sistema de Geração Automática de Avaliações*. Monografia (TCC) — Universidade Federal de Ouro Preto, 2010. Citado 6 vezes nas páginas 24, 25, 26, 32, 36 e 39.
- EASYTESTMAKER. *Easy Test Maker*. USA: [s.n.], 2019. Disponível em: <<https://www.easytestmaker.com/>>. Citado na página 24.
- ELMASRI, R.; NAVATHE, S. *Sistemas de banco de dados*. São Paulo: PEARSON BRASIL, 2011. ISBN 9788579360855. Citado na página 20.
- FOWLER, M. *UML Essencial: Um Breve Guia para Linguagem Padrao*. Porto Alegre: Bookman Editora, 2014. ISBN 9788560031382. Citado 2 vezes nas páginas 27 e 28.
- GASPAR, M. L. F. Avaliação da aprendizagem escolar: pr aticas avaliativas e suas representacções pedagogicas na avaliaccão da aprendizagem. Programa de Desenvolvimento Educacional - PDE, Cornélio Procópio, 2009. Citado na página 13.
- GASPARIN, J. *Uma didática para a pedagogia histórico-crítica*. Campinas: Autores Associados, 2011. (Coleção Educação contemporânea). ISBN 9788574960548. Citado na página 13.
- GATTI, B. A. Testes e avaliações do ensino no brasil. *Educação e Seleção*, São Paulo, n. 16, p. 33–42, 2013. Citado na página 13.

- HE, R. Y. design and implementation of web based on laravel framework. In: *2014 International Conference on Computer Science and Electronic Technology (ICCSET 2014)*. Atlantis Press, 2015/01. p. 301 – 304. ISBN 978-94-62520-47-9. ISSN 2352-538X. Disponível em: <<https://doi.org/10.2991/iccset-14.2015.66>>. Citado na página 21.
- HEUSER, C. A. *Projeto de banco de dados : Volume 4 da Série Livros didáticos informática UFRGS*. Porto Alegre: Bookman Editora, 2009. ISBN 9788577804528. Citado na página 19.
- JUNIOR, L. A. Z.; VIDAL, A. G. d. R. Construção de sistemas de informação baseados na tecnologia web. *Revista de Administração-RAUSP*, Universidade de São Paulo, v. 41, n. 3, p. 232 – 244, 2006. Citado na página 17.
- LAUDON, K.; LAUDON, J. *Sistemas de informação gerenciais*. São Paulo: Pearson Prentice Hall, 2011. ISBN 9788576059233. Citado na página 16.
- LERDORF, R. et al. *Programming PHP*. USA: O'Reilly Media, Incorporated, 2002. (Creating Dynamic Web Pages). ISBN 9781565926103. Citado na página 21.
- LUCKESI, C. C. O que é mesmo o ato de avaliar a aprendizagem. *Revista Pátio*, ano, Salvador, BA, v. 12, p. 6–11, 2000. Citado na página 22.
- MCCOOL, S. *Laravel Starter*. Birmingham, UK: Packt Pub., 2012. (Community experience distilled). ISBN 9781782160915. Citado na página 21.
- NIEDERAUER, J. *Desenvolvendo Websites com PHP – 3ª Edição: Aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados*. São Paulo: Novatec Editora, 2016. ISBN 9788575225349. Citado na página 21.
- OLIVEIRA, J.; CHADWICK, C. *Aprender e ensinar*. 9. ed. Belo Horizonte: Instituto Alfa Educativca, 2001. ISBN 9788526006904. Citado na página 13.
- PACHECO, J. A. *Avaliação da aprendizagem*. Porto Editora, Minho, Portugal, 1998. Citado na página 22.
- PERRENOUD, P. *Avaliação: da excelência à regulação das aprendizagens : entre duas lógicas*. Porto Alegre: Artmed, 1999. ISBN 9788573075441. Citado na página 13.
- QUIZWORKS, B. V. *EASYLMS*. Netherlands: [s.n.], 2019. Disponível em: <<https://goo.gl/JF7Gdq>>. Citado 2 vezes nas páginas 22 e 23.
- SILBERSCHATZ, A.; SUNDARSHAN, S.; KORTH, H. *Sistema de banco de dados*. Rio de Janeiro: Elsevier Editora Ltda., 2016. ISBN 9788535251425. Citado 3 vezes nas páginas 19, 20 e 30.
- SMBRASIL. *SM Provas*. Brasil: [s.n.], 2019. Disponível em: <<http://simplificabrasil.smeducacao.com.br/>>. Citado na página 23.
- STAUFFER, M. *Laravel: Up & Running: A Framework for Building Modern PHP Apps*. USA: O'Reilly Media, 2019. ISBN 9781492041184. Citado na página 21.
- THALHEIM, B. *Web information systems: Analysis, design, development, and implementation of business sites, collaboration sites, edutainment (e-learning) sites, and infotainment (information) sites*. Alemanha: Christian-Albrechts-University Kiel, 2013. Citado na página 17.

ZANON, D.; FREITAS, A. V. A aula de ciências nas séries iniciais do ensino fundamental: ações que favorecem a sua aprendizagem. *Ciências Cognição*, Rio de Janeiro, v. 10, n. 0, 2007. ISSN 1806-5821. Citado na página [13](#).

Apêndices

APÊNDICE A – Casos de Uso

Figura 20 – Caso de Uso - Cadastrar Assunto

<p style="text-align: center;">Caso de Uso - Cadastrar Assunto</p> <p>Ator Primário: Usuário do Sistema</p> <p>Interesses: Usuário quer conseguir cadastrar assuntos para que ele consiga cadastrar questões referentes a eles.</p> <p>Garantia de Sucesso: Assunto é gravado. Banco de dados é atualizado.</p> <p>Cenário de Sucesso Principal</p> <ol style="list-style-type: none">1. Usuário faz login no sistema2. Usuário escolhe a opção de cadastrar assunto3. O sistema exibe uma janela solicitando um nome4. Usuário informa esse nome5. O assunto é cadastrado6. Usuário é redirecionado para a tela de seleção de assuntos
--

FONTE: Elaborado pelo próprio autor

Figura 21 – Caso de Uso - Gerar Gabarito

<p style="text-align: center;">Caso de Uso - Gerar Gabarito</p> <p>Pré-condições: é preciso haver pelo menos uma avaliação com pelo menos uma questão.</p> <p>Ator Primário: Usuário do Sistema</p> <p>Interesses: Usuário quer ter acesso a um gabarito para a avaliação gerada</p> <p>Garantia de Sucesso: Gabarito é gerado e retornado como um arquivo para o usuário.</p> <p>Cenário de Sucesso Principal</p> <ol style="list-style-type: none">1. Usuário gera uma avaliação2. Usuário solicita o gabarito daquela avaliação3. Sistema gera a avaliação4. Um arquivo <i>PDF</i> é gerado e entregue ao usuário
--

FONTE: Elaborado pelo próprio autor

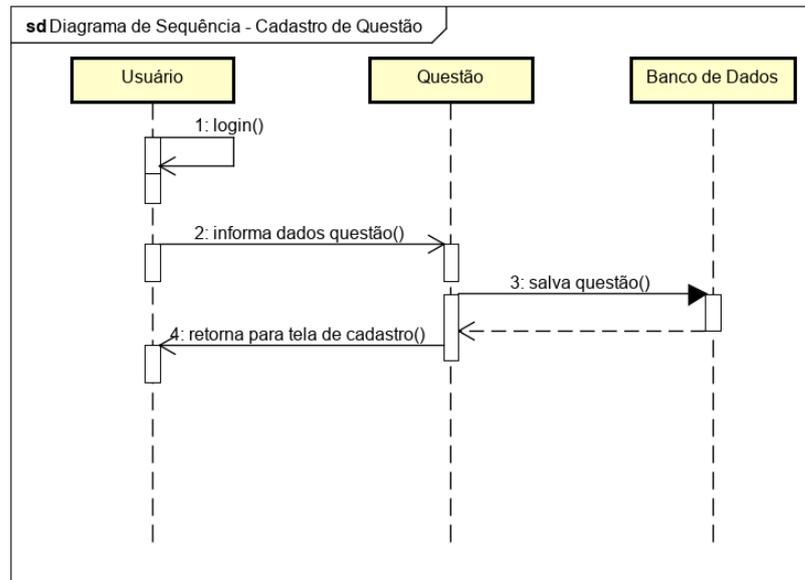
Figura 22 – Caso de Uso - Gerar Avaliação

<p style="text-align: center;">Caso de Uso - Gerar Avaliação</p> <p>Pré-condições: ter pelo menos um assunto e uma questão cadastrada.</p> <p>Ator Primário: Usuário do Sistema</p> <p>Interesses: Usuário quer conseguir gerar avaliações rápido e automaticamente, selecionando apenas algumas configurações.</p> <p>Garantia de Sucesso: Avaliação é gerada. Avaliação é exibida em forma de um editor de texto para o usuário.</p> <p>Cenário de Sucesso Principal</p> <ol style="list-style-type: none">1. Usuário faz login no sistema2. Usuário entra no menu de geração de avaliação3. Usuário preenche os dados e configurações4. Sistema encontra, utilizando seu algoritmo, a melhor seleção de questões, baseado nas configurações passadas no passo 3.5. Sistema apresenta um editor de texto contendo a avaliação gerada pronta para ser finalizada e/ou impressa. <p>Extensões:</p> <ul style="list-style-type: none">• 3a. Sistema informa que nem todos os dados obrigatórios foram inseridos corretamente.
--

FONTE: Elaborado pelo próprio autor

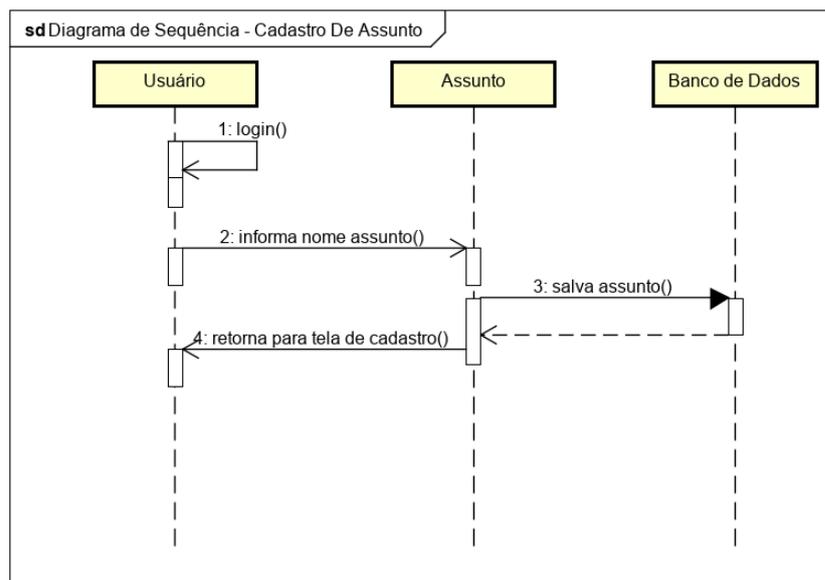
APÊNDICE B – Diagramas de Sequência

Figura 23 – Diagrama de Sequência - Cadastrar Questão



FONTE: Elaborado pelo próprio autor

Figura 24 – Diagrama de Sequência - Cadastrar Assunto



FONTE: Elaborado pelo próprio autor

APÊNDICE C – Avaliações Geradas

Figura 25 – Avaliação Gerada - Avalog 2

1 (Médio) Considere os dados abaixo.

Uma enquête com os 450 alunos de uma escola para saber os tipos de calçados mais usados apresentou o seguinte resultado:

48% dos alunos usavam sandália;
22% dos alunos usavam tênis;
30% dos alunos usavam sapato.

Esse resultado foi representado em um gráfico de setores:

O ângulo θ no gráfico acima mede:

- 95°
- 100°
- 105°
- 108°

2 (Fácil) Se X e Y são dois conjuntos não vazios, então $(X - Y) \cup (X \cap Y)$ é igual a

- \emptyset
- X
- Y
- $X \cap Y$
- $X \cup Y$

3 (Fácil) Considere dois conjuntos de números A e B com 12 e 15 elementos, respectivamente. Então, sempre se pode afirmar que

- $A \cap B$ terá, no mínimo, 12 elementos.
- $A \cup B$ terá, no mínimo, 15 elementos.
- o número máximo de elementos de $A \cap B$ é igual ao número máximo de elementos de $A \cup B$.
- o número mínimo de elementos de $A \cap B$ é igual ao número máximo de elementos de $A \cup B$.

4 (Difícil) Durante uma viagem choveu 5 vezes. A chuva caiu pela manhã ou à tarde, nunca o dia todo. Houve 6 manhãs e 3 tardes sem chuvas. Quantos dias durou a viagem:

- 6
- 7
- 8
- 9
- 10

FONTE: Elaborado pelo próprio autor

Figura 26 – Avaliação Gerada - Avalog 2

1 (**Fácil**) Em uma amostra de indivíduos, 40% foram afetados pela doença A, 20% foram afetados pela doença B e 5% foram afetados por ambas as doenças. Dos indivíduos da amostra que não foram afetados nem por A nem por B, 2% morreram. A porcentagem de indivíduos da amostra que morreram sem terem sido afetados por quaisquer das duas doenças analisadas é de:

- a) 0,7%.
- b) 0,8%.
- c) 0,9%.
- d) 1,0%.
- e) 1,1%

2 (**Fácil**) Em uma cidade com 40.000 habitantes há três clubes recreativos: Colina, Silvestre e Campestre. Feita uma pesquisa, foram obtidos os seguintes resultados: 20% da população frequenta o Colina; 16% o Silvestre; 14% o Campestre; 8% o Colina e o Silvestre; 5% o Colina e o Campestre; e 4% o Silvestre e o Campestre. Somente 2% frequêntam os três clubes. O número de habitantes que não frequêntam nenhum destes três clubes é:

- a) 26000
- b) 30000
- c) 28000
- d) 32000
- e) 34000

3 (**Médio**) Dado o conjunto $\{a, b, c, d, e, f, g\}$ o número máximo de subconjuntos distintos é:

- a) 21
- b) 128
- c) 64
- d) nenhuma dessas

4 (**Difícil**) Numa pesquisa sobre o consumo dos produtos A, B e C, obteve-se o seguinte resultado: 68% dos entrevistados consomem A, 56% consomem B, 66% consomem C e 15% não consomem nenhum dos produtos. Qual a porcentagem mínima de entrevistados que consomem A, B e C?

- a) 30%
- b) 28%
- c) 25%
- d) 27%
- e) 20%

FONTE: Elaborado pelo próprio autor

TERMO DE RESPONSABILIDADE

Eu, **Bruno Araujo Camarda** declaro que o texto do trabalho de conclusão de curso intitulado “*Avalog 2.0 - Sistema Web para Geração de Avaliações*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 10 de julho de 2019

Bruno Araujo Camarda

Bruno Araujo Camarda

DECLARAÇÃO DE CONFORMIDADE

Certifico que o(a) aluno(a) **Bruno Araujo Camarda**, autor do trabalho de conclusão de curso intitulado “*Avalog 2.0 - Sistema Web para Geração de Avaliações*” efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

João Monlevade, 25 de julho de 2019.



Bruno Rabello Monteiro