

SISTEMA WEB PARA GERAÇÃO AUTOMÁTICA DO TESTE DE PROGRESSO

Área temática: Metodologias e Abordagens de Ensino Aplicadas à Área de Ciência e Tecnologia

Eugênio Silva - eugeniosilva@unifeso.edu.br, docente, Ciência da Computação, UNIFESO.

João Victor de Souza Geonizeli, discente, Ciência da Computação, UNIFESO.

Ricardo Coutinho Fonte, discente, Ciência da Computação, UNIFESO.

Luiz Cláudio Ramos de Mello Júnior, discente, Ciência da Computação, UNIFESO.

PICPq 2020-2021.

RESUMO

Desde 2007, quando começou a ser implantado nos cursos do UNIFESO, o Teste de Progresso tem se mostrado uma importante ferramenta de auto avaliação institucional que permite tanto a alunos quanto professores identificar, além de virtudes, potenciais vulnerabilidades no processo de ensino-aprendizagem e, com isso, proporcionar à instituição a oportunidade de adotar alguma medida corretiva. Atualmente o UNIFESO tem à sua disposição ferramentas computacionais que auxiliam tanto na correção das questões objetivas do Teste de Progresso quanto na geração de gráficos e resultados estatísticos que facilitam enormemente a avaliação dos resultados. Contudo, o processo de elaboração da prova ainda é essencialmente manual e se caracteriza por ser muito moroso e propenso a erros. Diante disso, a proposta deste trabalho consiste em construir um sistema computacional *web* que permita que os vários atores envolvidos no processo de elaboração do Teste de Progresso possam trabalhar de forma colaborativa e centralizada, contribuindo assim para que essa tarefa possa ser realizada com mais rapidez, conforto e confiabilidade.

Palavras-chave: Teste de Progresso; sistema *web*; trabalho colaborativo.

INTRODUÇÃO

Segundo Miranda e Crisostomo (2014):

“A autoavaliação numa Instituição de Ensino Superior (IES) consiste numa oportunidade importante de realizar, por meio de pesquisa acurada, um levantamento diagnóstico e a consequente reflexão sobre a realidade de um contexto acadêmico em vistas a ir além do autoconhecimento e, assim, promover a superação das fragilidades e a otimização das potencialidades identificadas.”

Ciente de que a autoavaliação institucional representa um importante instrumento a ser usado na busca pela excelência acadêmica, em 2008 a gestão do UNIFESO incentivou e aprovou a elaboração do Programa de Autoavaliação Institucional (PAAI). O PAAI compreende um conjunto de mecanismos de autoavaliação institucional que se divide em oito projetos que são aplicados aos três segmentos da comunidade acadêmica: docentes, estudantes e técnicos-administrativos (MIRANDA; MORAES, 2014). Um desses projetos compreende a avaliação anual do desenvolvimento cognitivo dos estudantes, comumente denominado Teste de Progresso.

De acordo com Miranda e Moraes (2014), o Teste de Progresso:

“é uma avaliação formativa cujo conteúdo tem por objetivo avaliar o crescimento cognitivo do estudante, aplicado longitudinalmente em todos os períodos, sem caráter de premiação, punição ou promoção, traduzindo na prática a política de avaliação formativa preconizada no Projeto Político-Pedagógico (PPPI) do UNIFESO.”

O Teste de Progresso foi gradativamente implantado de 2007 a 2009, quando passou a ser aplicado a todos os cursos do UNIFESO (MORGADO, 2014). Desde então tem se

mostrado um importante instrumento que permite ao estudante acompanhar o seu processo de formação e identificar suas fragilidades e potencialidades. Além disso, é um indicador bastante útil para que docentes e gestores acadêmicos (diretores e coordenadores) identifiquem fortalezas ou lacunas no processo de ensino-aprendizagem e adotem medidas corretivas que se façam necessárias (MIRANDA; MORAES, 2014).

JUSTIFICATIVA

Atualmente, o UNIFESO conta com ferramentas computacionais que automatizam o processo de correção das questões objetivas do Teste de Progresso e também a elaboração de gráficos e cálculos estatísticos que são de grande utilidade para a consolidação e análise dos resultados. A despeito disso, o processo de construção da prova ainda é executado de forma essencialmente manual. Esse processo envolve não só a elaboração das questões, mas também a definição das características da prova e a montagem do arquivo (formato .DOCX) com as questões, sabendo que esse arquivo deve sempre respeitar uma formatação previamente estabelecida. Vale destacar que as características que norteiam a escolha das questões que vão compor a prova levam em consideração, principalmente, o grau de dificuldade desejado para a prova e a distribuição das questões entre as várias disciplinas que compõem um determinado curso de graduação. Em linhas gerais, o processo de construção da prova é guiado pelos seguintes passos:

1. o coordenador de curso solicita aos seus professores, via correio eletrônico, a elaboração de um conjunto de questões para cada disciplina sob sua responsabilidade;
2. o coordenador recebe, também via correio eletrônico, as questões elaboradas pelos professores;
3. o coordenador encaminha as questões elaboradas por todos os professores ao professor designado para a construção da prova;
4. o professor responsável pela construção da prova define as suas características e, com base nisso, escolhe as questões que vão compor a prova segundo critérios que levam em consideração, por exemplo, o grau de dificuldade e a temática das questões;
5. o professor responsável pela construção da prova monta o arquivo (formato .DOCX) com as questões escolhidas, seguindo a formatação exigida;
6. o professor responsável pela construção da prova encaminha a prova ao coordenador via correio eletrônico;
7. o coordenador de curso envia a prova por correio eletrônico ao setor da instituição responsável pela inclusão das questões de conhecimentos gerais (comuns a todos os cursos), inclusão do cartão resposta, personalização e impressão do Teste de Progresso.

Como se pode observar, trata-se de um processo bastante extenso em que os passos envolvidos estão sujeitos a atrasos, esquecimentos e extravios. Além disso, o trabalho do professor responsável pela construção da prova, que basicamente consiste em definir as características da prova, escolher as questões dentre todas aquelas elaboradas pelos professores e montar o arquivo correspondente, é bastante demorado e altamente propenso a falhas que são bastante comuns em tarefas do tipo “copia e cola”. Essas falhas vão desde a simples variação de formatação do texto quando um trecho é copiado de um arquivo e colado em outro, até falhas mais sérias como replicação ou subtração de trechos de texto.

Diante do exposto, entende-se que uma boa forma de tornar o trabalho de construção da prova mais amigável e confiável é automatizá-lo. Para isso, o que se propõe aqui é a construção de um sistema *web* que ofereça um ambiente que possa ser acessado por todos os atores envolvidos na elaboração do Teste de Progresso e que concentre todas as ações. Além disso, o fato de ser um sistema *web* proporciona ainda o conforto de poder ser acessado de

qualquer computador e de qualquer lugar, bastando para isso a disponibilidade de conexão com a Internet (ALVES, 2015; QUEIRÓS, PORTELA, 2018).

OBJETIVOS

Objetivo geral

O objetivo geral do projeto consiste em desenvolver um sistema *web* que permita automatizar o processo de elaboração da prova do Teste de Progresso de todos os cursos do UNIFESO, contribuindo assim para tornar esse processo mais simples e confiável.

Objetivos específicos

Os objetivos específicos do projeto são:

proporcionar aos professores responsáveis pela elaboração das questões do Teste de Progresso um ambiente amigável e centralizado para o cadastro dessas questões;

permitir que as questões cadastradas sejam validadas por pares, ou seja, por outro(s) professor(es) com conhecimentos aderentes àquelas questões;

permitir que as questões cadastradas sejam classificadas segundo critérios como: grau de dificuldade, eixo curricular, modalidade (objetiva ou discursiva), tipo (resposta única, afirmação completa, interpretação, etc.) no caso de questões objetivas e habilidade do domínio cognitivo segundo a Taxonomia de Bloom;

proporcionar aos professores responsáveis pela elaboração da prova que o seu conteúdo seja definido de forma simples, por meio da configuração dos critérios a serem considerados na seleção automática das questões.

REVISÃO BIBLIOGRÁFICA

Até a submissão da proposta do projeto apresentado aqui ao Plano de Iniciação Científica e Pesquisa (PICPq), não era de conhecimento dos autores nenhuma aplicação que se propusesse a oferecer funcionalidades semelhantes. Apenas recentemente, com a aquisição do sistema Qstione¹ por parte do UNIFESO, é que se tomou conhecimento de que há no mercado um sistema com características similares às daquele proposto aqui. Em termos gerais, o Qstione foi adquirido especificamente para a criação de bancos de questões e para a geração de avaliações para as disciplinas dos diversos cursos da instituição. Contudo, parece não haver limitações que o impeçam de ser empregado também no Teste de Progresso. Apesar disso, o trabalho proposto ainda se justifica devido a alguns diferenciais em relação ao Qstione.

Dentre os diferenciais, merece destaque a funcionalidade de *validação por pares*. No Qstione, o processo de validação de uma questão é unidirecional, ou seja, o revisor envia o seu parecer ao criador da questão, mas o criador não tem a oportunidade de, se necessário, questionar ou solicitar esclarecimentos sobre alguma observação emitida pelo revisor. No sistema proposto aqui, o processo de validação é bidirecional, permitindo a interação entre criador e revisor até que um consenso seja alcançado. Além disso, no Qstione o revisor pode ser um professor que eventualmente não tem conhecimentos aderentes ao tema da questão que está revisando. Com isso, sua avaliação fica limitada à forma da questão, não tendo condições de estender a sua avaliação ao conteúdo. Na aplicação proposta, como parte do processo de criação de uma questão, o criador deve indicar explicitamente quem será o revisor. Com isso, é possível escolher alguém com conhecimentos suficientes para avaliar a questão tanto do ponto de vista da forma quanto do conteúdo.

METODOLOGIA

Com o advento da Internet e a modernização das ferramentas de desenvolvimento de

¹ <https://www.qstione.com.br/novo/>

aplicações, a *web* deixou de ser um espaço apenas para a apresentação de conteúdo e passou a ser também um ambiente capaz de hospedar aplicações interativas para as mais variadas finalidades. Uma grande vantagem proporcionada pelo desenvolvimento *web* perante os sistemas *desktop* tradicionais, é a possibilidade de acesso global. Um sistema *web*, se não houver qualquer restrição, está disponível a qualquer hora e em qualquer lugar e, para que possa ser acessado, basta ter em mãos um computador (ou até mesmo um *tablet* ou *smartphone*) conectado à Internet. Essas características tornam o desenvolvimento *web* uma ótima opção para a construção de aplicações voltadas ao trabalho colaborativo, em que vários atores participam da execução de uma mesma tarefa, mas não necessariamente estão no mesmo ambiente físico (ALVES, 2015; QUEIRÓS, PORTELA, 2018).

O sistema proposto aqui se enquadra muito bem nas características de um sistema *web*, uma vez que envolve a participação de várias pessoas que precisam trabalhar de forma colaborativa para alcançar um objetivo comum, que é a elaboração do Teste de Progresso. Algumas das funcionalidades essenciais previstas para o sistema são as seguintes:

Cadastro de questões: permite que professores previamente designados possam cadastrar as questões que foram solicitados a elaborar. Nesse cadastro, além do enunciado da questão e da resposta esperada, outros atributos devem ser preenchidos. Dentre eles estão: o grau de dificuldade da questão, o eixo curricular e a disciplina aos quais está associada, a modalidade, o tipo (caso seja uma questão objetiva) e a habilidade cognitiva à qual está associada. Esses atributos são de extrema importância nas etapas de parametrização e geração da prova.

Validação por pares: permite que, ao cadastrar uma questão, o professor possa indicar outro professor para validá-la. Essa é uma medida importante para evitar que questões que apresentem eventuais falhas em sua formulação possam ser incluídas no Teste de Progresso e precisem ser anuladas posteriormente.

Parametrização da prova: permite definir as características da prova, levando em consideração a distribuição de questões de acordo com a configuração de percentuais associados aos diversos atributos associados às questões.

Geração da prova: permite a geração automática do arquivo da prova levando em consideração os parâmetros definidos anteriormente como critérios para a seleção das questões a serem incluídas no Teste de Progresso.

Características Técnicas do Sistema

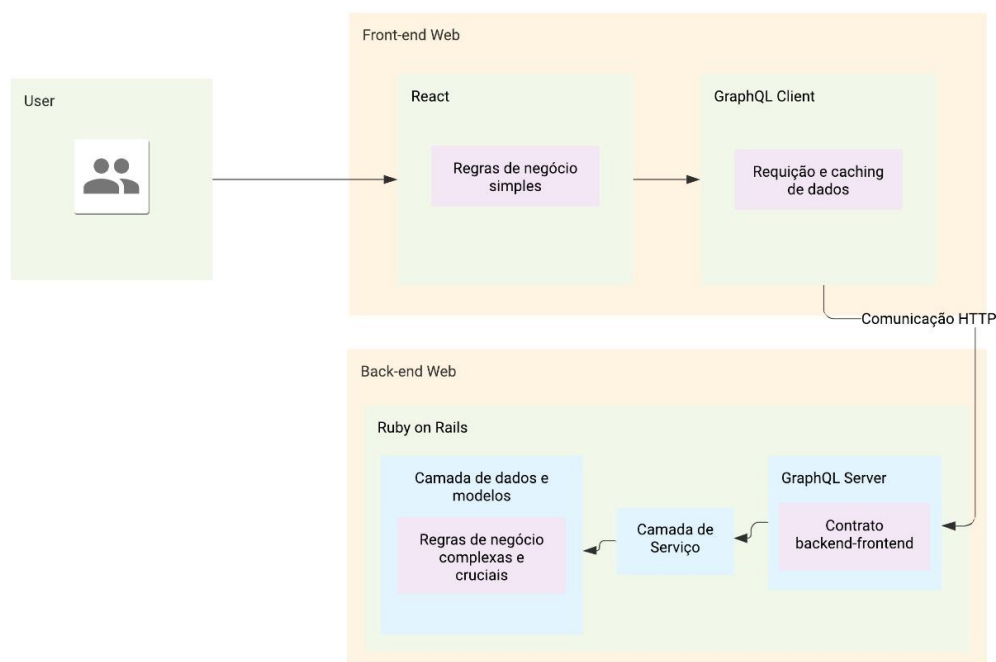
No desenvolvimento *web*, em geral as aplicações são compostas por dois módulos: o *front-end* e o *back-end*. O *front-end* pode ser entendido como um módulo que envolve toda a parte visível da aplicação, ou seja, a interface exibida em um navegador, instalado no dispositivo do usuário, e com a qual o usuário interage diretamente. Para o desenvolvimento desse módulo, obviamente são necessários conhecimentos de linguagens de programação que sejam mais voltadas para a criação de interfaces gráficas. Dentre as várias possibilidades, as linguagens mais comumente utilizadas para esse propósito são: HTML, CSS, JavaScript e TypeScript. Além disso, há também algumas bibliotecas que facilitam bastante o trabalho de desenvolvimento, uma vez que oferecem vários componentes reutilizáveis de interface. Dentre elas, as mais populares são: React, jQuery, Angular e Vue.js.

O *back-end*, por sua vez, diz respeito a tudo o que está por trás da interface e que é responsável pelo funcionamento da aplicação. Isso envolve não só o processamento das requisições vindas do *front-end* e a devolução de respostas a essas requisições, mas também o armazenamento de dados e informações. Esse módulo da aplicação não está hospedado no dispositivo do cliente (usuário) e sim em um servidor remoto e toda a comunicação entre o *front-end* e o *back-end* acontece por meio de uma conexão de Internet. Para o desenvolvimento do *back-end*, as principais linguagens de programação utilizadas são: Python, Java, PHP e Ruby. Além disso, há também *frameworks* como o Ruby on Rails que tornam o desenvolvimento do *back-end* mais fácil e rápido.

No desenvolvimento do *front-end*, a fim de entregar uma experiência fluida e prática ao usuário, optou-se por implementá-lo segundo o conceito *Single Page Application* (SPA). Com ele, ao acessar o sistema, o navegador carrega todas as informações necessárias para exibir qualquer de suas telas, evitando assim a necessidade de recarregar a aplicação conforme a navegação. Para isso, foi utilizada a biblioteca React do JavaScript que não apenas provê o SPA como também proporciona maior produtividade no desenvolvimento.

Para o desenvolvimento do *back-end* utilizou-se o *framework* Rails, que permite a criação de aplicações bastante robustas e em muito pouco tempo. Por meio do Rails, criou-se uma API (*Application Program Interface*) na arquitetura GraphQL (*Graph Query Language*), uma alternativa ao REST (*Representational State Transfer*). O REST consiste em um conjunto de princípios, regras e restrições na abstração de uma API que, quando seguidos, permite a criação de um projeto com interfaces bem definidas. O GraphQL, por sua vez, consiste em uma linguagem de consulta que permite ao *front-end* requisitar ao *back-end* exatamente o que precisa, tornando mais fácil a evolução da API ao longo do tempo. Além disso, possibilita o uso de ferramentas de desenvolvimento bastante poderosas. Enquanto o REST e suas restrições buscam entregar uma API padrão que poderia até mesmo ser utilizada por outros *back-ends*, o GraphQL se baseia em abstrações que atendem as requisições dos *front-ends* (CALISTRO, 2018). A arquitetura da aplicação proposta está ilustrada da Figura 1.

Figura 1: arquitetura do sistema proposto



Fonte: Elaborada pelos autores

Outro aspecto de grande relevância em desenvolvimento de software é o versionamento de código. Especialmente quando se tem uma equipe de desenvolvedores em que cada membro deve se encarregar de uma determinada parte da aplicação. Orquestrar as diversas versões de código é algo desafiador e, para facilitar esse processo, foi utilizado o versionador Git (AQUILES, 2014), que contribuiu decisivamente para a eficiência do processo de desenvolvimento.

RESULTADOS E DISCUSSÃO

Até o momento, de todas as funcionalidades originalmente propostas, o sistema contempla plenamente o *cadastro de questões* objetivas e a *validação por pares* considerando,

especificamente, o contexto do curso de Ciência da Computação do UNIFESO. A título de exemplo, a Figura 2 mostra as telas correspondentes aos quatro passos necessários para o cadastro de questões. Os passos de 1 a 3 envolvem os elementos que são parte do conteúdo da questão. O passo 4, por sua vez, associa à questão um conjunto de características, sendo que algumas delas serão úteis posteriormente para a parametrização da prova. É importante destacar que no passo 4, no último campo, está indicado explicitamente o revisor selecionado para avaliar a questão.

Figura 2: Sequência de telas de cadastro de questão

Instrução (opcional)

B I “ ” : = < ≥ ∑ ∏ ⌊ ⌋ ✓ ✗ ↶ ↷

Imagine um sistema nebuloso que tenha duas variáveis de entrada (V_1 e V_2) e uma variável de saída (S). Imagine ainda que para valores de entrada específicos de V_1 e V_2 sejam ativadas as seguintes regras nebulosas:

SupORTE (opcional)

B I “ ” : = < ≥ ∑ ∏ ⌊ ⌋ ✓ ✗ ↶ ↷

If V₁ is X₁(0,25) and V₂is Y₁(0,30) then S is Z₁
If V₁ is X₁(0,20) and V₂is Y₂(0,10) then S is Z₂
If V₁ is X₂(0,30) and V₂is Y₁(0,45) then S is Z₁
If V₁ is X₂(0,15) and V₂is Y₂(0,35) then S is Z₂

Onde: X, Y e Z são rótulos dos conjuntos nebulosos que compõem as variáveis V_1 , V_2 e S , respectivamente.

Enunciado

B I “ ” : = < ≥ ∑ ∏ ⌊ ⌋ ✓ ✗ ↶ ↷

Aplmando-se o método de Inferência MIN-MAX às regras nebulosas acima, tem-se como resposta:

Cancelar Prosseguir

(a) Passo 1

Resposta Correta

B I “ ” := : = [] < > { } ~ ^ _ ` ¨ ˇ ¢ ↶ ↷

Z1 = 0,30 e Z2 = 0,15

Explicação

B I “ ” := : = [] < > { } ~ ^ _ ` ¨ ˇ ¢ ↶ ↷

Em uma regra nebulosa onde os valores dos graus de inclusão presentes no antecedente são combinados por meio do conectivo **and**, o valor do grau de inclusão de saída é dado pelo **mínimo** dos valores que estão no antecedente. Durante a inferência, quando se tem duas ou mais regras ativas cujos antecedentes são diferentes, mas que mapeiam para o mesmo consequente, o resultado final é dado pelo **máximo** entre os valores desses consequentes.

Referências

B I “ ” := : = [] < > { } ~ ^ _ ` ¨ ˇ ¢ ↶ ↷

Oliveira Junior, H. A. Lógica Difusa: Aspectos práticos e aplicações. Rio de Janeiro: Interciência, 1999.

Cancelar

Retornar

Prosseguir

(b) Passo 2

Distratores

B I “ ” = : ≡ ∑ ∫ ∇ √ ∘ ↶ ↷

Z1 = 0,30 e Z2 = 0,20

B I “ ” = : ≡ ∑ ∫ ∇ √ ∘ ↶ ↷

Z1 = 0,25 e Z2 = 0,10

B I “ ” = : ≡ ∑ ∫ ∇ √ ∘ ↶ ↷

Z1 = 0,45 e Z2 = 0,35

B I “ ” = : ≡ ∑ ∫ ∇ √ ∘ ↶ ↷

Z1 = 0,55 e Z2 = 0,25

Cancelar Retornar Prosseguir

(c) Passo 3

Características

Autoria

☒ Própria ☐ Outro

Grau de Dificuldade

Fácil

Tipo

Afirmção Incompleta

Habilidade Cognitiva

Aplicar

Intenção

Verificar conhecimentos sobre a estratégia MIN-MAX de inferência nebulosa.

Revisor

Eugênio Testes

Fonte

UNIFESO

Assunto

Sistemas Inteligentes

Eixo de Formação

Algoritmos de Alto Desempenho

Categoria

Sistemas Inteligentes

Cancelar

Retornar

Finalizar

(d) Passo 4

Fonte: Elaboradas pelos autores

Algumas funcionalidades acessórias também estão com a implementação concluída. Uma delas é a tela de edição que exhibe ao usuário corrente as questões existentes em seu perfil e o status de cada uma. Cada status corresponde a uma seção da tela e os tipos de status considerados são os seguintes:

Aguardando seu Parecer: mostra as questões criadas por outros professores que foram endereçadas ao usuário corrente para revisão.

Aguardando Parecer do Revisor: mostra as questões do usuário corrente que estão esperando o parecer dos professores indicados como revisores.

Pendentes de Alterações: mostra as questões que já foram avaliadas pelos revisores, porém ainda não aprovadas, que precisam ser alteradas e submetidas novamente aos revisores.

Rascunhos: mostra as questões que ainda estão em processo de criação e que,

portanto, ainda não foram enviadas aos revisores.

Aprovadas: mostra as questões que foram avaliadas e aprovadas pelos revisores, mas que, mesmo aprovadas, eventualmente ainda precisam que alguns ajustes sugeridos pelos revisores sejam efetuados.

Registradas: mostra as questões que foram devidamente aprovadas pelos revisores, que já passaram pelos últimos ajustes e, por isso, não podem mais ser editadas.

Revisadas por Você: mostra o histórico de todas as questões que foram avaliadas pelo usuário corrente.

Ainda na tela de edição é possível definir diferentes filtros para a exibição das questões nas seções. É possível filtrar as questões por assunto, tipo, habilidade cognitiva, ano de autoria e grau de dificuldade. Vale destacar que os filtros não são excludentes e, assim, podem ser combinados de diversas formas conforme a conveniência do usuário.

Outra funcionalidade acessória, cuja implementação também está concluída, é um painel que exibe gráficos ao usuário corrente com a distribuição das questões registradas por ele segundo diferentes aspectos. Os aspectos considerados são: assunto, habilidade cognitiva, grau de dificuldade e tipo. A Figura 3 a seguir mostra um exemplo de distribuição de questões registradas:

Figura 3: painel com a distribuição das questões sob diferentes aspectos



Fonte: Elaborada pelos autores

O desenvolvimento das funcionalidades de *parametrização* e *geração da prova* depende da existência de uma quantidade mínima de questões válidas cadastradas no banco de dados para que possam ser efetivamente testadas. Para isso, é necessário que a aplicação esteja devidamente disponível para acesso e uso por parte dos professores. No entanto, por questões técnicas que envolvem, dentre outras, requisitos de conformidade para hospedagem e acesso ao

sistema no servidor do UNIFESO, ainda não foi possível disponibilizá-lo para uso dos professores.

CONSIDERAÇÕES FINAIS

Atualmente, todo o processo de elaboração da prova do Teste de Progresso é executado de forma totalmente manual, o que o torna penoso e sujeito a diversos tipos de falhas. Essas falhas vão desde simples problemas de formatação de texto e qualidade de figuras, gráficos e tabelas, até a ocorrência de questões com erros em sua formulação. As falhas apontadas devem ser evitadas ao máximo, uma vez que, em geral, resultam em anulação de questões, o que traz prejuízos ao processo avaliativo como um todo. Com a geração automática do texto final e com a validação de questões por pares propostas aqui, espera-se que tais problemas sejam minimizados ou até mesmo eliminados.

Até o momento, o sistema contempla apenas parte das funcionalidades originalmente propostas, especificamente o *cadastro de questões* objetivas e a *validação por pares*. As outras funcionalidades ainda não foram desenvolvidas devido a entraves técnicos que estão em processo de solução. Tão logo esses empecilhos sejam solucionados, o sistema deve ser disponibilizado para avaliação e crítica dos professores integrantes do Núcleo Docente Estruturante (NDE) do curso de Ciência da Computação do UNIFESO. Após os eventuais ajustes decorrentes dessa avaliação/crítica, o sistema deve ser posto em produção para que todos os professores do curso possam se familiarizar com o seu uso e inserir questões em seu banco de dados. Enquanto isso, em paralelo, as outras funcionalidades serão desenvolvidas. Como as funcionalidades seguintes dependem da existência de questões cadastradas no sistema, para o sucesso delas é fundamental o engajamento de todos os professores do curso.

Uma vez concluído o desenvolvimento de todo o sistema e realizada a sua validação no contexto do curso de Ciência da Computação, pretende-se estendê-lo aos outros cursos da instituição. Com isso, espera-se que todas as coordenações de curso, e consequentemente seus membros de NDE e professores, tenham em mãos uma ferramenta que possa oferecer a seus usuários maior conforto e confiabilidade no processo de elaboração da prova do Teste de Progresso.

Por último, e não menos importante, espera-se que, com o tempo, o sistema se torne um rico repositório de conhecimento para todos os cursos do UNIFESO. Esse conhecimento será de grande utilidade, por exemplo, para a elaboração de oficinas preparatórias para as provas do ENADE.

REFERÊNCIAS

Alves, WP. Projetos de Sistemas Web – conceitos, estruturas, criação de banco de dados e ferramentas de desenvolvimento. ed. 1. São Paulo: Érica; 2015.

Aquiles, A. Controlando Versões com Git e GitHub. São Paulo: Casa do Código; 2014.

Calistro, J. GraphQL e REST quais as diferenças? Medium - Idexo for Developers. 14 de jun. de 2018. Disponível em: <https://medium.com/idexo-developers/graphql-e-rest-quais-as-diferen%C3%A7as-99cae22871a7>. Acesso em: 30 de ago. de 2021.

Miranda, JFA, Crisostomo, RPG. Autoavaliação Institucional no UNIFESO. Autoavaliação Institucional no UNIFESO – 15 Anos de Avaliação Transformadora. Vol. 1. Teresópolis: UNIFESO, 2014.

Miranda, JFA, Moraes, MBVB. PAAI: Programa de Autoavaliação Institucional. Autoavaliação Institucional no UNIFESO – 15 Anos de Avaliação Transformadora. Vol. 1. Teresópolis: UNIFESO; 2014.

Morgado. FEF. O Teste de Progresso. Autoavaliação Institucional no UNIFESO – 15 Anos de

Avaliação Transformadora. Vol. 1. Teresópolis: UNIFESO; 2014.

Queirós, R, Portela, F. Introdução ao Desenvolvimento Moderno para a Web: do front-end ao back-end, um visão global! ed. 1. Lisboa: FCA; 2018.