

SISTEMA WEB PARA GERAÇÃO AUTOMÁTICA DO TESTE DE PROGRESSO – FASE III

WEB SYSTEM FOR AUTOMATIC GENERATION OF PROGRESS TEST – PHASE III

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

Gabriel Resende Machado - gabriel.machado@ecoa.puc-rio.br, pesquisador, ECOA, PUC-Rio.

Kauã Landi Fernando, discente, Ciência da Computação, UNIFESO.

Pedro Pereira Dias, discente, Ciência da Computação, UNIFESO.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema web que proporcione uma solução mais amigável, prática e confiável para a elaboração e a geração da prova do Teste de Progresso. O sistema foi desenvolvido no âmbito do PICPq e estruturado em três fases. Nas Fases I e II foram implementados os módulos de cadastro e validação de questões, geração de gráficos de distribuição de questões e de administração, estabelecendo a base funcional da aplicação. Na Fase III, foi realizada uma reformulação da arquitetura do sistema e a atualização das tecnologias de desenvolvimento, com o objetivo de torná-lo mais robusto e alinhado às boas práticas mais modernas de engenharia de software. Ainda nessa fase, foram desenvolvidas as funcionalidades de parametrização e geração manual de provas, bem como a modelagem do problema de otimização associado à geração automática de provas. Para o modelo de otimização, baseado na técnica de Algoritmos Genéticos, foram definidos a representação cromossômica, os operadores genéticos de cruzamento e de mutação, bem como a função de aptidão. Com isso, o sistema se encontra em um estágio avançado de maturidade, reunindo os principais recursos necessários para incorporar a funcionalidade de geração automática de provas do Teste de Progresso.

Palavras-chave: Teste de Progresso; modelo de otimização; Algoritmos Genéticos.

ABSTRACT

This paper presents the development of a web-based system that provides a more user-friendly, practical, and reliable solution for the preparation and generation of Progress Test exams. The system was developed within the scope of PICPq and structured into three phases. In Phases I and II, the modules for question registration and validation, generation of question distribution charts, and administration were implemented, establishing the functional foundation of the application. In Phase III, the system architecture was redesigned and the development technologies were updated in order to make the system more robust and aligned with modern software engineering best practices. Also in this phase, functionalities for exam parameterization and manual exam generation were developed, as well as the modeling of the optimization problem associated with automatic exam generation. For the optimization model, based on Genetic Algorithms, the chromosome representation, the genetic operators of crossover and mutation, and the fitness function were defined. As a result, the system is now at an advanced stage of maturity, bringing together the main resources required to incorporate the automatic generation of Progress Test exams.

Keywords: Progress Test; optimization model; Genetic Algorithms.

1. INTRODUÇÃO

Desde sua implantação no UNIFESO, iniciada em 2007 e consolidada em 2009, o Teste de Progresso vem se destacando como um importante instrumento do processo de avaliação institucional dos cursos de graduação. Por meio dos indicadores gerados, gestores e docentes são apoiados na definição de estratégias para o aprimoramento do processo de ensino-aprendizagem. Paralelamente, os estudantes também têm no Teste de Progresso uma forma de monitorar o próprio crescimento acadêmico (MORGADO, 2014) e (MIRANDA, MORAES, 2014).

Contudo, ao longo do tempo, o processo tradicional de elaboração da prova se mostrou moroso e suscetível a erros. Diante dessa realidade, na edição 2020–2021 do Plano de Iniciação Científica e Pesquisa (PICPq), foi proposta a construção de um sistema web voltado a oferecer uma solução mais amigável, prática e confiável para a elaboração e a geração da prova do Teste de Progresso (SILVA, GEONIZELI, FONTE, MELLO JÚNIOR, 2020). Para isso, os esforços foram concentrados no desenvolvimento de um sistema web capaz de oferecer as seguintes funcionalidades essenciais:

1. Cadastro de questões: permitir que docentes cadastrados no sistema elaborem questões contemplando não apenas seus componentes fundamentais (enunciado, suporte, alternativa correta e distratores), mas também um conjunto ampliado de características relevantes para a parametrização e a geração da prova. Dentre essas características, destacam-se o assunto, o tipo de questão objetiva, o nível de dificuldade e a habilidade cognitiva, segundo a Taxonomia de Bloom, avaliada pela questão.

2. Validação por pares: como parte do processo de elaboração de uma questão, possibilitar ao autor a indicação do docente responsável por sua revisão. Essa funcionalidade se mostra especialmente relevante, pois contribui para reduzir a probabilidade de inclusão de questões com inconsistências na prova, prevenindo, assim, a ocorrência de anulações posteriores.

3. Exibição de status de questões: disponibilizar ao usuário a visualização do status de cada uma das questões associadas ao seu perfil. Esses status correspondem às diferentes etapas do processo de elaboração da questão. Como exemplo, a questão pode estar em modo rascunho, aguardando a avaliação do revisor ou ainda com ajustes pendentes.

4. Geração de gráficos: apresentar ao usuário um painel com gráficos que ilustram a distribuição das questões cadastradas por ele de acordo com diferentes atributos, tais como o assunto, o tipo de questão, o nível de dificuldade e a habilidade do domínio cognitivo avaliada na questão.

5. Administração: possibilitar a criação, a remoção e a definição dos perfis de usuários do sistema. Esses perfis determinam tanto os níveis de acesso às questões cadastradas, quanto as permissões para a execução das atividades de elaboração de questões e de geração da prova.

6. Parametrização da prova: possibilitar a definição dos parâmetros da prova por meio da atribuição de percentuais aos diferentes atributos das questões, de modo a orientar o processo de seleção das questões que irão compor a prova.

7. Geração da prova: possibilitar a geração do arquivo da prova tanto de forma manual quanto automática, assegurando o atendimento aos parâmetros previamente definidos para a seleção das questões.

Nas edições 2020–2021 e 2022–2023 do PICPq, correspondentes, respectivamente, às Fases I e II do projeto, foram concluídas as implementações das cinco primeiras funcionalidades previstas (SILVA, GEONIZELI, FONTE, MELLO JÚNIOR, 2020), (SILVA, GEONIZELI, FONTE, MELLO JÚNIOR, 2021) e (SILVA, GEONIZELI, MACHADO, RUFINO, 2022). Na Fase II, também foram discutidos o problema de otimização subjacente à geração automática da prova e sua respectiva modelagem. Nesse contexto, foi definido um modelo de solução, fundamentado em Algoritmos Genéticos (MICHALEWICZ, 1996) e (LINDEN, 2008), e foram apresentadas a representação cromossômica

e os operadores genéticos de cruzamento e mutação, permanecendo pendente a formalização da função de aptidão (SILVA, GEONIZELI, CORRÊA, ROCHA, 2023). As implementações das funcionalidades de parametrização e de geração manual e automática da prova foram transferidas para a Fase III, desenvolvida na edição 2024–2025 do PICPq. Contudo, antes do início dessas implementações, houve a necessidade de realizar melhorias e ajustes no código existente (SILVA, MACHADO, FERNANDO, CARREIRO, 2024).

Diante disso, este trabalho apresenta, inicialmente, as alterações realizadas na interface do sistema, bem como na sua arquitetura e nas tecnologias adotadas no desenvolvimento. Em seguida, descreve de maneira detalhada o desenvolvimento das funcionalidades relacionadas à parametrização e à geração manual da prova. Por fim, a modelagem do problema de otimização é retomada, com foco na formalização da função de aptidão.

2. JUSTIFICATIVA

Atualmente, o UNIFESO utiliza o sistema comercial Qstione¹ para a criação de bancos de questões e para a elaboração de avaliações aplicadas tanto nos cursos de graduação quanto no Teste de Progresso. Embora o sistema proposto compartilhe algumas funcionalidades com o Qstione, há diferenças relevantes que justificam os esforços empreendidos no seu desenvolvimento. Uma delas é o processo de validação por pares, que se caracteriza por um critério bidirecional que permite múltiplas interações entre o autor e o revisor da questão, até que seja alcançado um consenso para sua aprovação. Nesse contexto, cabe ao elaborador indicar o revisor, o que possibilita a escolha de um docente com domínio sobre o conteúdo avaliado. Com isso, a avaliação pode ser mais abrangente, não se limitando apenas a aspectos de forma.

Em contraste, no sistema Qstione a validação ocorre de maneira unidirecional, sem que o parecer do revisor possa ser contestado pelo elaborador. Além disso, o autor da questão não possui autonomia para escolher o revisor. Dessa forma, não é possível garantir que o revisor tenha conhecimento específico sobre o tema abordado, restringindo, assim, a profundidade da análise.

Outro aspecto que distingue o sistema proposto diz respeito à flexibilidade no processo de geração da prova, que pode ser realizada de forma manual ou automática. Na modalidade manual, o usuário seleciona diretamente as questões que vão compor a prova. Na modalidade automática, por sua vez, a seleção é conduzida por um modelo de otimização baseado em Algoritmos Genéticos. Em ambas as abordagens, a parametrização previamente definida atua como um conjunto de diretrizes para a escolha das questões, contribuindo para que o resultado esteja alinhado às características desejadas da prova.

3. OBJETIVOS

O projeto no qual este trabalho se insere tem como propósito principal o desenvolvimento de um sistema *web* capaz de apoiar o trabalho colaborativo entre os docentes do UNIFESO, tornando mais ágil, simples e confiável o processo de elaboração e geração da prova do Teste de Progresso.

Como objetivos específicos deste trabalho, que compreende as atividades realizadas na Fase III, têm-se:

- atualizar os elementos da interface gráfica do sistema, com o objetivo de melhorar a usabilidade e a experiência do usuário;

¹ <https://www.qstione.com.br/novo/>

- revisar a arquitetura e as tecnologias empregadas no desenvolvimento do sistema, de modo a alinhá-las às boas práticas contemporâneas de engenharia de software;
- implementar as funcionalidades relacionadas à parametrização e à geração manual da prova;
- retomar a modelagem do problema de otimização associado à geração automática da prova, acrescentando a formalização da função de aptidão.

4. METODOLOGIA

4.1 Arquitetura e Tecnologias de Desenvolvimento

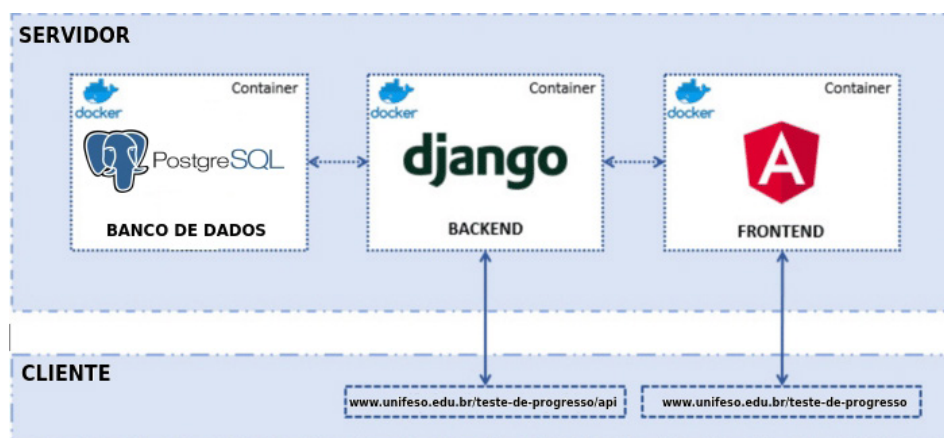
Após uma avaliação detalhada do código desenvolvido nas Fases I e II do projeto, concluiu-se que seria necessária a atualização das tecnologias da arquitetura existente, adotando-se a abordagem de *greenfield development* (GIARDINO *et al.*, 2015). Essa decisão foi motivada por três fatores principais: (i) a necessidade de alinhar as práticas de desenvolvimento ao estado da arte, considerando as evoluções ocorridas ao longo das diferentes edições do projeto; (ii) a possibilidade de acelerar a correção de falhas em funcionalidades associadas a requisitos funcionais e não funcionais, por meio da refatoração integral viabilizada pela abordagem *greenfield*; e (iii) o objetivo de tornar o produto final mais extensível e adequadamente documentado, facilitando futuras atualizações relacionadas às demandas do negócio.

O novo conjunto de ferramentas empregado no desenvolvimento do sistema passou a ser composto pelas seguintes tecnologias:

- **PostgreSQL:** sistema gerenciador de banco de dados relacional de código aberto, reconhecido por sua robustez, escalabilidade e extensibilidade. Permite o gerenciamento de diferentes tipos de dados, a execução de consultas complexas e é amplamente utilizado devido ao suporte contínuo da comunidade e à diversidade de ferramentas e extensões disponíveis.
- **Django:** trata-se de um *framework* de alto nível, desenvolvido em Python, para a construção do *backend* de aplicações de forma ágil e com arquitetura limpa e bem estruturada. Adota o padrão MVC (*Model-View-Controller*), que se caracteriza por uma organização modular que facilita a manutenção e a evolução das aplicações. O Django disponibiliza um conjunto abrangente de recursos integrados, como interface administrativa automática, mecanismos de autenticação e gerenciamento de rotas, permitindo o desenvolvimento eficiente de aplicações robustas e escaláveis. Sua ampla adoção decorre da combinação entre flexibilidade, segurança e facilidade de uso, além do suporte oferecido por uma comunidade ativa e consolidada.
- **Angular:** trata-se de um *framework* de código aberto, mantido pelo Google, voltado ao desenvolvimento *frontend* de aplicações *web* dinâmicas do tipo *Single Page Application* (SPA). Implementado em TypeScript, um superconjunto do JavaScript, adota uma arquitetura orientada a componentes, favorecendo a modularização e a reutilização de código. O Angular disponibiliza ainda recursos avançados, como mecanismos de roteamento, injeção de dependências e suporte integrado a testes. Sua ampla adoção está associada à capacidade de construir interfaces de usuário interativas e escaláveis, aliada ao suporte de uma comunidade ativa e consolidada.
- **Docker:** trata-se de uma plataforma de virtualização que possibilita a criação, a implantação e a execução de aplicações em contêineres. Os contêineres consistem em unidades leves e portáteis que encapsulam todos os elementos necessários para a execução de uma aplicação, incluindo código, dependências e componentes do sistema operacional. O Docker assegura a consistência do funcionamento das aplicações em diferentes ambientes, do desenvolvimento à produção, reduzindo problemas relacionados à compatibilidade.

A Figura 1 ilustra a arquitetura do sistema, indicando as tecnologias mencionadas anteriormente.

Figura 1: Desenho da nova arquitetura do sistema.



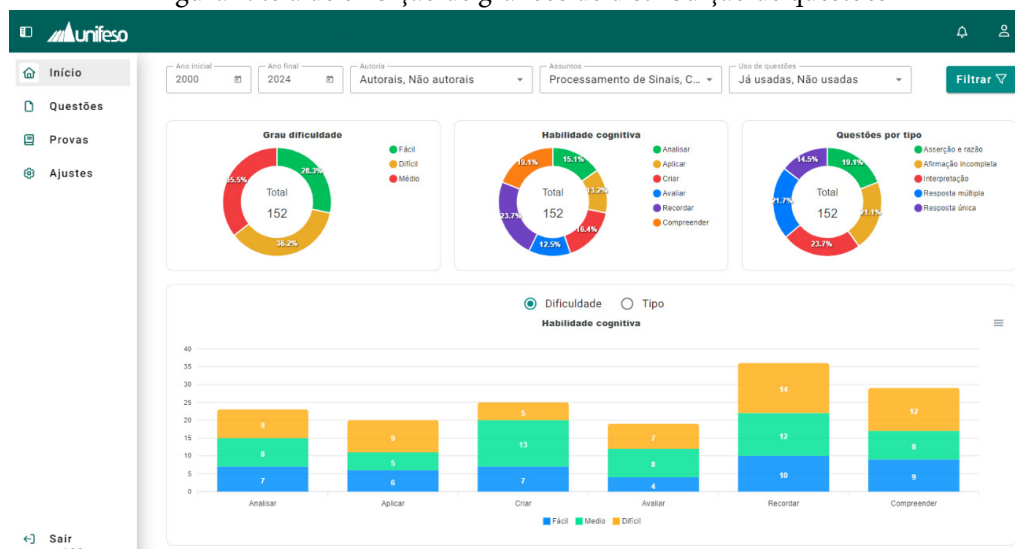
Fonte: Adaptado de MOREIRA *et al.* (2021)

4.2 Nova Experiência de Usuário

Após a refatoração do código da aplicação, que incluiu alterações na arquitetura e nas tecnologias utilizadas no desenvolvimento, foram realizadas mudanças na interface do sistema. Essas alterações tiveram como objetivo aprimorar a experiência do usuário, disponibilizando uma interface mais atual, intuitiva e funcional.

A Figura 2 ilustra o novo layout adotado para a visualização dos gráficos de distribuição das questões, considerando os atributos considerados mais relevantes para a análise de diversidade. Os gráficos em formato de rosca apresentam a distribuição das questões segundo os atributos grau de dificuldade, habilidade cognitiva e tipo de questão. Já o gráfico de barras possibilita a análise da distribuição a partir da combinação do atributo habilidade cognitiva com o grau de dificuldade ou com o tipo. Acima dos gráficos, encontra-se um conjunto de filtros associado a outros atributos das questões, que também contribuem para a análise da distribuição. Ressalta-se, ainda, que o menu lateral pode ser ocultado, permitindo maior aproveitamento do espaço destinado aos elementos de interface exibidos à direita.

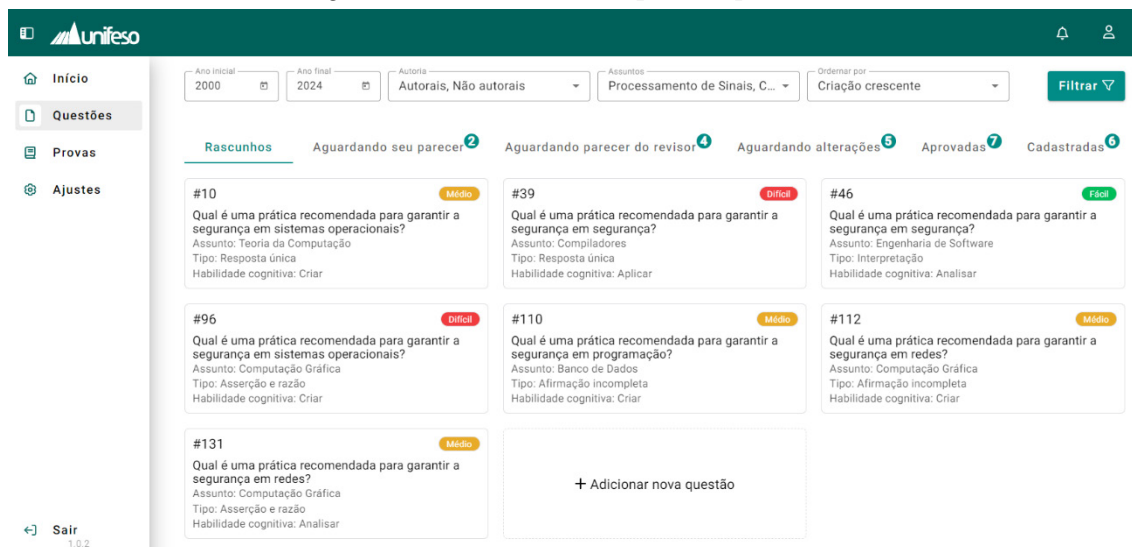
Figura 2: tela de exibição de gráficos de distribuição de questões



Fonte: elaborada pelos autores

A Figura 3 apresenta as questões organizadas em abas, sendo cada aba associada a um dos possíveis estados que uma questão pode assumir. Em cada uma delas, há uma notificação que sinaliza, de maneira simples e intuitiva, a necessidade de alguma ação por parte do usuário em relação às questões. Na listagem, para facilitar a identificação rápida das principais características de cada questão, são mostrados os valores correspondentes a alguns dos atributos mais relevantes (grau de dificuldade, assunto, tipo e habilidade cognitiva). Além disso, é exibido também o trecho inicial do enunciado de cada questão. Adicionalmente, os mesmos filtros disponíveis na tela de gráficos estão presentes na tela de questões, permitindo refinar a visualização das questões em todas as abas.

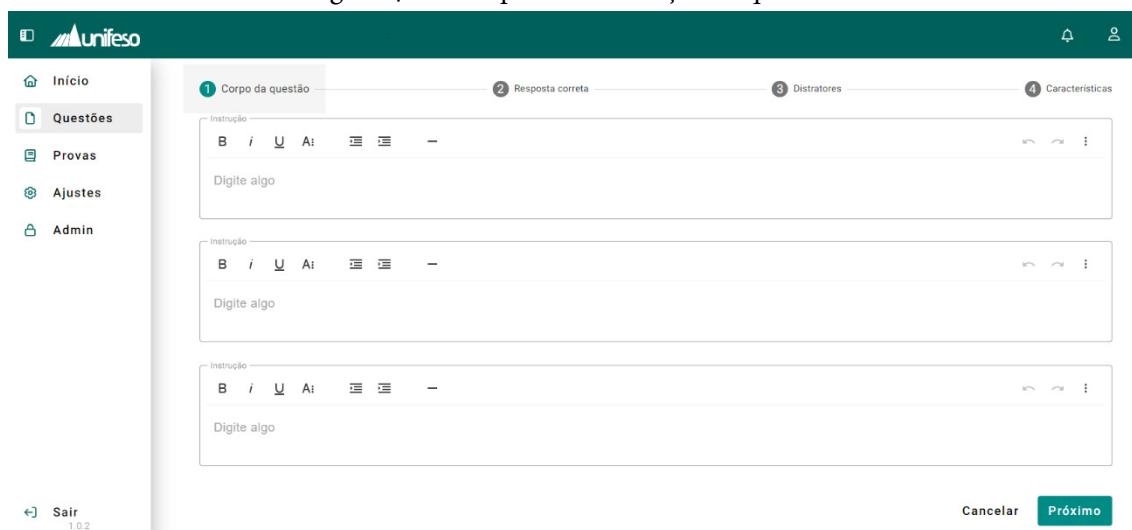
Figura 3: tela de exibição de questões por *status*



Fonte: elaborada pelos autores

A Figura 4 ilustra a primeira das quatro telas que integram o processo de criação de uma questão. Nessa etapa inicial, são inseridos os elementos referentes à instrução, ao suporte e ao enunciado. Na segunda etapa, são informadas a alternativa correta, a justificativa e a(s) referência(s) que fundamenta(m) a explicação apresentada, enquanto na terceira etapa são definidos os distratores.

Figura 4: tela do passo 1 de criação de questões



Fonte: elaborada pelos autores

No passo 4, são definidos os atributos associados à questão que são relevantes não apenas como critérios de filtragem nas telas anteriores, mas também como parâmetros que orientam o processo de geração da prova, tanto na modalidade manual quanto automática. Durante a parametrização da prova, o usuário deve escolher os atributos que serão considerados na seleção das questões e, para cada atributo selecionado, indicar o percentual de questões que devem apresentar determinada característica. Os atributos especificados nessa etapa incluem a autoria da questão, o nível de dificuldade, o tipo, a habilidade cognitiva, o assunto, o revisor e a intenção da questão. O eixo de formação é preenchido automaticamente a partir da seleção do assunto. A interface correspondente a essa etapa é apresentada na Figura 5.

Figura 5: tela do passo 4 de criação de questões

Fonte: elaborada pelos autores

4.3 Parametrização de Prova

Do ponto de vista da interface, o processo de parametrização da prova está estruturado em duas etapas. Na etapa inicial, o usuário deve fornecer a identificação da prova e, opcionalmente, uma descrição com informações complementares sobre a prova a ser gerada. A Figura 6 apresenta o layout da tela correspondente à identificação da prova.

Figura 6: tela de identificação da prova

Fonte: elaborada pelos autores

Na segunda etapa, o usuário define o número de questões e os parâmetros que caracterizam a prova. Para cada parâmetro selecionado, são especificados os percentuais associados aos diferentes valores que esse parâmetro pode assumir. A Figura 7, por exemplo, ilustra a tela de parametrização em que o parâmetro grau de dificuldade é escolhido, sendo então indicadas as proporções desejadas de questões de nível fácil, médio e difícil.

Figura 7: tela de parametrização de prova (grau de dificuldade)

Fonte: elaborada pelos autores

Os demais parâmetros seguem o mesmo procedimento: ao serem selecionados, passam a exibir os respectivos valores para que os percentuais correspondentes sejam definidos. Cabe destacar que, à medida que a parametrização é realizada, a expressão da função de aptidão utilizada na geração da prova é construída de forma paralela. Uma vez definida, essa função passa a orientar tanto a geração manual quanto a automática da prova, tendo como objetivo, em ambos os casos, a obtenção de uma prova que minimize o valor da função de aptidão.

4.4 Geração Manual de Prova

A interface de geração manual da prova, ilustrada na Figura 8, disponibiliza ao usuário uma forma prática de selecionar as questões que vão compor a prova. Para isso, há um conjunto de filtros que auxilia o processo de pesquisa e escolha das questões. Esses filtros consideram os atributos das questões que também são parâmetros utilizados na função de aptidão. As questões resultantes da filtragem são exibidas na coluna à esquerda, enquanto as questões selecionadas são apresentadas na coluna à direita.

À medida que as questões são incorporadas à prova, um indicador exibe o percentual de aderência da prova em relação à parametrização previamente definida. Assim, o usuário pode avaliar continuamente se as escolhas contribuem ou não para alcançar as características desejadas para a prova.

Figura 8: tela de geração manual de prova

Fonte: elaborada pelos autores

No caso da geração automática, ao final do processo de otimização, também deve ser apresentada a tela exibida na Figura 8, possibilitando ao usuário analisar a melhor solução obtida pelo modelo. A partir dessa visualização, é possível avaliar o grau de proximidade da solução em relação à configuração ótima, medido pelo percentual de conformidade com a parametrização definida. Essa interface permite ainda verificar a validade da solução apresentada, assegurando que a prova não contenha questões repetidas. Dessa forma, caso haja necessidade, o usuário pode realizar ajustes pontuais, como substituições de questões, de modo a aproximar ainda mais a prova da parametrização desejada.

4.5 Modelagem do Problema de Otimização

De modo geral, problemas de otimização envolvem um conjunto muito amplo de soluções possíveis, tendo como objetivo identificar aquela que seja a melhor. De modo mais formal, esses problemas consistem em explorar um espaço de soluções visando encontrar a alternativa que maximize (ou minimize) uma determinada métrica de adequação. Cabe ressaltar que a complexidade do problema cresce à medida que o espaço de busca se amplia. Em cenários práticos, é comum a quantidade de soluções ser grande o suficiente para inviabilizar a aplicação de abordagens baseadas em busca exaustiva para encontrar a solução ótima.

A geração automática da prova pode ser formulada como um problema de otimização. Nesse cenário, assume-se a existência de um banco com um grande volume de questões, de modo que cada questão é caracterizada por atributos como grau de dificuldade, tipo de questão objetiva, habilidade do domínio cognitivo, eixo de formação, entre outros. A partir desse conjunto, busca-se compor uma prova em que a seleção das questões minimize uma métrica definida com base nesses atributos, conforme os critérios estabelecidos na etapa de parametrização. Vale destacar que a complexidade desse problema cresce proporcionalmente ao aumento do número de questões disponíveis no banco.

Entre as diversas abordagens empregadas para encontrar soluções eficientes em problemas complexos de otimização, destacam-se os métodos heurísticos. Algumas das características desses métodos são: (i) a inspiração em processos naturais; (ii) a simplificação da modelagem do problema; e (iii) a capacidade de produzir, em um tempo computacional aceitável, soluções que não são necessariamente ótimas, mas que são de boa qualidade.

Neste trabalho, a técnica de Algoritmos Genéticos é utilizada como base para a construção do modelo de otimização responsável pela geração automática da prova do Teste de Progresso. Uma abordagem mais aprofundada sobre problemas de otimização de alta complexidade e sobre o funcionamento dos algoritmos genéticos pode ser consultada em Silva, Geonizelli, Corrêa e Rocha (2023).

Para que possa ser solucionado com o uso de um algoritmo genético, o processo de modelagem do problema em questão deve considerar as seguintes etapas:

- representação cromossômica;
- definição dos operadores genéticos de cruzamento e de mutação;
- definição da função de aptidão.

4.6 Representação Cromossômica

A representação cromossômica utilizada é composta por um vetor de inteiros com n posições, sendo n equivalente ao número de questões que vão compor a prova. Cada gene, correspondente a uma posição do vetor, armazena o identificador de uma questão específica. Assim, o cromossomo representa um indivíduo, isto é, uma determinada seleção de questões que compõem a prova. A Figura 9 ilustra um exemplo de um cromossomo.

Figura 9: exemplo de cromossomo que representa uma prova

45	32	17	28	95	74	23	15	94	67
----	----	----	----	----	----	----	----	----	----

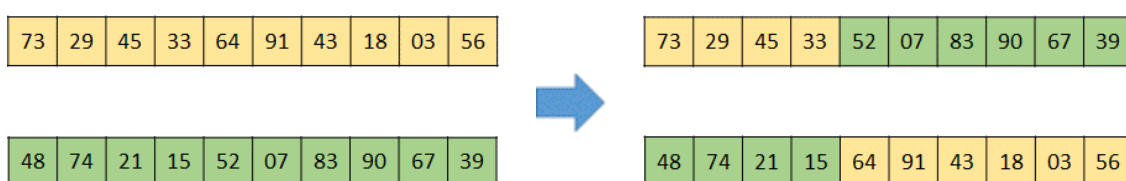
Fonte: elaborada pelos autores

No exemplo está representada uma prova com 10 questões, em que os valores correspondem aos identificadores de questões que estão armazenadas no banco.

4.7 Operadores Genéticos

Considerando a representação cromossômica definida, os operadores de cruzamento e mutação adotados se baseiam naqueles tradicionalmente aplicados a cromossomos com representação binária. O operador de cruzamento realiza a recombinação do material genético de dois cromossomos em um ou mais pontos selecionados aleatoriamente. O operador de mutação, por sua vez, requer uma adaptação. Diferentemente da representação binária, em que os genes apenas têm seus valores invertidos (de 0 para 1 ou vice-versa), neste caso a mutação consiste na substituição dos genes selecionados por identificadores de questões escolhidos aleatoriamente a partir do banco de questões. As Figuras 10 e 11 apresentam, respectivamente, exemplos da aplicação dos operadores de cruzamento e de mutação.

Figura 10: exemplo de aplicação do operador de cruzamento



Fonte: elaborada pelos autores

Figura 11 exemplo de aplicação do operador de mutação



Fonte: elaborada pelos autores

Cabe ressaltar que, apesar de serem compatíveis com a representação cromossômica adotada, os operadores genéticos apresentados não garantem a produção de indivíduos válidos. Isso se deve ao fato de que as alterações realizadas podem gerar cromossomos contendo identificadores de questões repetidos. Em termos práticos, tal situação corresponde à geração de uma prova com questões duplicadas, o que não é admissível.

Uma alternativa para minimizar a geração de indivíduos inválidos consiste em incorporar um critério de penalização à função de aptidão, atribuindo avaliações piores àqueles que violam as restrições estabelecidas. Dessa forma, o mecanismo de seleção do algoritmo genético tende a desfavorecer tais soluções. Entretanto, essa estratégia não assegura a eliminação dos indivíduos inválidos e pode ainda impactar negativamente o desempenho computacional do processo de otimização.

Na aplicação proposta, a prevenção de soluções inválidas é tratada por meio de uma abordagem de otimização semiautomática. Ao final da execução do algoritmo genético, o resultado obtido deve ser apresentado ao usuário por meio da tela da Figura 8, que permite avaliá-lo e, se necessário, corrigir eventuais inconsistências por meio da substituição manual de questões.

4.8 Função de Aptidão

Na formalização da função de aptidão são considerados todos os parâmetros que compõem as características da prova a ser gerada. Esses parâmetros envolvem os seguintes atributos das questões:

grau de dificuldade, tipo de questão objetiva, habilidade cognitiva e eixo de formação. A função permite ainda combinar questões novas com questões utilizadas em outras provas e também de estabelecer o intervalo de data de criação das questões a serem selecionadas.

Ao longo da parametrização da prova, o usuário pode escolher os atributos desejados para a configuração da função de aptidão e, para cada atributo escolhido, são indicados os percentuais de questões para cada valor daquele atributo. A Equação 1 apresenta a formalização da função de aptidão F :

$$\begin{aligned}
 F = & c_1 \left[\left| p_{11} - \frac{n_{11}}{n} \right| + \left| p_{12} - \frac{n_{12}}{n} \right| + \left| p_{13} - \frac{n_{13}}{n} \right| \right] + \\
 & c_2 \left[\left| p_{21} - \frac{n_{21}}{n} \right| + \left| p_{22} - \frac{n_{22}}{n} \right| + \left| p_{23} - \frac{n_{23}}{n} \right| + \left| p_{24} - \frac{n_{24}}{n} \right| + \left| p_{25} - \frac{n_{25}}{n} \right| \right] + \\
 & c_3 \left[\left| p_{31} - \frac{n_{31}}{n} \right| + \left| p_{32} - \frac{n_{32}}{n} \right| + \left| p_{33} - \frac{n_{33}}{n} \right| + \left| p_{34} - \frac{n_{34}}{n} \right| + \left| p_{35} - \frac{n_{35}}{n} \right| + \left| p_{36} - \frac{n_{36}}{n} \right| \right] + \\
 & c_4 \left[\left| p_{41} - \frac{n_{41}}{n} \right| + \left| p_{42} - \frac{n_{42}}{n} \right| + \left| p_{43} - \frac{n_{43}}{n} \right| + \left| p_{44} - \frac{n_{44}}{n} \right| + \left| p_{45} - \frac{n_{45}}{n} \right| \right] + \\
 & c_5 \left[\left| p_{51} - \frac{n_{51}}{n} \right| + \left| p_{52} - \frac{n_{52}}{n} \right| \right]
 \end{aligned} \quad (1)$$

Onde:

n : número de questões da prova

Grau de dificuldade:

$c_1 = 0$: atributo *grau de dificuldade* não selecionado pelo usuário

$c_1 = 1$: atributo *grau de dificuldade* selecionado pelo usuário

p_{11} : percentual desejado de questões de *nível fácil*

n_{11} : número de questões de *nível fácil* selecionadas

p_{12} : percentual desejado de questões de *nível médio*

n_{12} : número de questões de *nível médio* selecionadas

p_{13} : percentual desejado de questões de *nível difícil*

n_{13} : número de questões de *nível difícil* selecionadas

Tipo de questão objetiva:

$c_2 = 0$: atributo *tipo de questão objetiva* não selecionado pelo usuário

$c_2 = 1$: atributo *tipo de questão objetiva* selecionado pelo usuário

p_{21} : percentual desejado de questões de *resposta única*

n_{21} : número de questões de *resposta única* selecionadas

p_{22} : percentual desejado de questões de *afirmação incompleta*

n_{22} : número de questões de *afirmação incompleta* selecionadas

p_{23} : percentual desejado de questões de *resposta múltipla*

n_{23} : número de questões de *resposta múltipla* selecionadas

p_{24} : percentual desejado de questões de *interpretação*

n_{24} : número de questões de *interpretação* selecionadas

p_{25} : percentual desejado de questões de *asserção e razão*

n25: número de questões de *asserção e razão* selecionadas

Habilidade cognitiva

c3 = 0: atributo *habilidade cognitiva* não selecionado pelo usuário

c3 = 1: atributo *habilidade cognitiva* selecionado pelo usuário

p31: percentual desejado de questões de *recordar*

n31: número de questões de *recordar* selecionadas

p32: percentual desejado de questões de *compreender*

n32: número de questões de *compreender* selecionadas

p33: percentual desejado de questões de *aplicar*

n33: número de questões de *aplicar* selecionadas

p34: percentual desejado de questões de *analisar*

n34: número de questões de *analisar* selecionadas

p35: percentual desejado de questões de *avaliar*

n35: número de questões de *avaliar* selecionadas

p36: percentual desejado de questões de *criar*

n36: número de questões de *criar* selecionadas

Eixo de formação

c4 = 0: atributo *eixo de formação* não selecionado pelo usuário

c4 = 1: atributo *eixo de formação* selecionado pelo usuário

p41: percentual desejado de questões de *infraestrutura de sistemas computacionais*

n41: número de questões de *infraestrutura de sistemas computacionais* selecionadas

p42: percentual desejado de questões de *sistemas de software*

n42: número de questões de *sistemas de software* selecionadas

p43: percentual desejado de questões de *computação avançada*

n43: número de questões de *computação avançada* selecionadas

p44: percentual desejado de questões de *ciência, tecnologia e sociedade*

n44: número de questões de *ciência, tecnologia e sociedade* selecionadas

p45: percentual desejado de questões de *integração profissional e prática*

n45: número de questões de *integração profissional e prática* selecionadas

Reaproveitamento

c5 = 0: atributo *reaproveitamento* não selecionado pelo usuário

c5 = 1: atributo *reaproveitamento* selecionado pelo usuário

p51: percentual desejado de questões *novas* no intervalo de datas

n51: número de questões *novas* selecionadas no intervalo de datas

p52: percentual desejado de questões *usadas* no intervalo de datas

n52: número de questões *usadas* selecionadas no intervalo de datas

OBS.: ao informar os valores de *p51* e *p52*, devem ser informados também os intervalos de datas a serem considerados para a seleção de questões *novas* e *usadas*.

Após a parametrização, a função resultante deve retornar valores menores para as provas cuja distribuição de questões esteja mais próxima dos percentuais estabelecidos para os parâmetros selecionados. Com isso, tem-se o problema de otimização caracterizado como um problema de minimização, em que indivíduos de melhor avaliação (mais aptos) são aqueles com menor valor de aptidão.

5. CONSIDERAÇÕES FINAIS

Na edição 2020–2021 do PICPq, foi apresentada a proposta de desenvolvimento de um sistema com o objetivo de tornar o processo de elaboração da prova do Teste de Progresso mais simples e confiável. Nessa etapa inicial, denominada Fase I, parte das funcionalidades originalmente previstas foi implementada, enquanto as demais foram desenvolvidas na Fase II, correspondente à edição 2022–2023 do PICPq.

Nessas duas fases houve avanços significativos na implementação de funcionalidades fundamentais, como o gerenciamento de questões, abrangendo o cadastro, a validação e a organização de acordo com seus respectivos status. Também foram desenvolvidos os gráficos para análise da distribuição das questões cadastradas, bem como o módulo de administração, voltado ao gerenciamento de usuários e seus perfis de acesso. Na Fase II, também foi apresentada a modelagem do problema de otimização subjacente à geração automática da prova. No entanto, essa modelagem não incluiu a formalização da função de aptidão do algoritmo genético, utilizado como base para o modelo proposto.

A Fase III, correspondente à edição 2024–2025 do PICPq, teve início com uma revisão abrangente do código produzido nas etapas anteriores. Nesse contexto, foram corrigidas inconsistências identificadas no funcionamento do sistema e foi promovida uma modernização ampla, envolvendo a adoção de boas práticas arquiteturais, a atualização das tecnologias utilizadas e a melhoria da experiência do usuário.

Ainda na Fase III foram incorporadas as funcionalidades de parametrização e de geração manual da prova, que representam avanços importantes para a conclusão do desenvolvimento do sistema. Importante destacar que a parametrização oferece suporte tanto ao modo de geração manual quanto ao automático. Quanto à geração automática, o modelo de otimização foi completamente formalizado e está pronto para ser implementado e, em seguida, integrado ao sistema.

6. REFERÊNCIAS

- GIARDINO, C.; UNTERKALMSTEINER, M.; GORSCHKE, T. Software development in startup companies: the greenfield startup model. *IEEE Transactions on Software Engineering*, v. 42, n. 6, p. 585-604, 2015.
- LINDEN, R. *Algoritmos Genéticos: Uma Importante Ferramenta de Inteligência Computacional*. 2ª ed. Rio de Janeiro: Brasport, 2008.
- MICHALEWICZ, Z. *Genetic Algorithm + Data Structures = Evolution Programs*. Germany: Springer-Verlag, 1996.
- MIRANDA, J. F. A.; MORAES, M. B. V. B. PAAI: Programa de Autoavaliação Institucional. In: *Autoavaliação Institucional no UNIFESO – 15 Anos de Avaliação Transformadora*. Vol. 1. Teresópolis: UNIFESO, 2014.
- MOREIRA, P.; SEQUEIRA, A.; PEREIRA, S.; RODRIGUES, R.; ROCHA, M.; LOUSA, D. ViralFP: a web application of viral fusion proteins. *Frontiers in Medical Technology*, v. 3, 2021. doi: 10.3389/fmedt.2021.722392.
- MORGADO, F. E. F. O Teste de Progresso. In: *Autoavaliação Institucional no UNIFESO – 15 Anos de Avaliação Transformadora*. Vol. 1. Teresópolis: UNIFESO, 2014.
- SILVA, E.; GEONIZELI, J. V. S.; FONTE, R. C.; MELLO JÚNIOR, L. C. R. Sistema Web para Geração Automática do Teste de Progresso. In: *Anais do V Congresso Acadêmico Científico do UNIFESO – CONFESO*, Teresópolis: Editora UNIFESO, 2020.

SILVA, E.; GEONIZELI, J. V. S.; FONTE, R. C.; MELLO JÚNIOR, L. C. R. Sistema Web para Geração Automática do Teste de Progresso. In: *Anais do VI Congresso Acadêmico Científico do UNIFESO – CONFESO*, Teresópolis: Editora UNIFESO, 2021.

SILVA, E.; GEONIZELI, J. V. S.; MACHADO, J. V. S.; RUFINO, J. V. Sistema Web para Geração Automática do Teste de Progresso – Fase II. In: *Anais do VII Congresso Acadêmico Científico do UNIFESO – CONFESO*, Teresópolis: Editora UNIFESO, 2022.

SILVA, E.; GEONIZELI, J. V. S.; CORRÊA, F. L. C.; ROCHA, E. F. Sistema Web para Geração Automática do Teste de Progresso – Fase II. In: *Anais do VIII Congresso Acadêmico Científico do UNIFESO – CONFESO*, Teresópolis: Editora UNIFESO, 2023.

SILVA, E.; MACHADO, G. R.; FERNANDO, K., L.; CARREIRO, R. R. Sistema Web para Geração Automática do Teste de Progresso – Fase III. In: *Anais do IX Congresso Acadêmico Científico do UNIFESO – CONFESO*, Teresópolis: Editora UNIFESO, 2024.