

KHEIRON: Uma Aplicação Web para Acompanhamento Pedagógico Focado em Treinamentos de Programação Competitiva

Lucas Santana de Oliveira¹
Alexandre Douglas Leite Bezerra Filho²
Rafael Oliveira Vasconcelos³

RESUMO

A programação competitiva é um esporte mental em que os competidores resolvem problemas de Ciência da Computação dentro de um tempo limitado, sem acesso a materiais externos. Ganhou popularidade nas instituições de ensino fundamental e médio com o crescimento da internet e das competições, como a Olimpíada Brasileira de Informática e a Maratona Brasileira de Programação. Professores têm treinado seus alunos para participarem desses eventos, mas a falta de plataformas adequadas dificulta o sucesso no preparo dos competidores. Para suprir essa demanda, foi desenvolvida a aplicação *web* Kheiron para o auxílio pedagógico, aplicação que facilita o gerenciamento e acompanhamento dos estudos dos alunos por parte do docente e fornece ferramentas de análise e de repasse de conteúdo, além de permitir que o aluno seja atendido diretamente pelo docente/treinador ou por um robô (*bot*) dotado de Inteligência Artificial Generativa. Para o desenvolvimento deste trabalho, aplicou-se uma pesquisa exploratória, seguida de uma revisão bibliográfica.

Palavras-chave: aprendizagem; programação; computação; educação.

KHEIRON: A WEB APPLICATION FOR PEDAGOGICAL SUPPORT FOCUSED ON COMPETITIVE PROGRAMMING TRAINING

ABSTRACT

Competitive programming is a mental sport in which competitors solve Computer Science problems within a limited time without access to external materials. It gained popularity in primary and secondary education institutions with the growth of the internet and competitions such as the Brazilian Computer Olympiad and the Brazilian Programming Marathon. Teachers have trained their students to participate in these events, but the lack of suitable platforms makes it difficult to successfully prepare competitors. To meet this demand, the Kheiron web application was developed for pedagogical assistance, an application that facilitates the management and monitoring of students' studies by the teacher and provides analysis and content transfer tools, in addition to allowing the student to be served directly by the teacher/trainer or by a robot (*bot*) equipped with Generative Artificial Intelligence. To develop this work, exploratory research was carried out, followed by a bibliographic review.

Keywords: learning; programming; computing; education.

Submetido em: 6/2/2024

Aceito em: 28/6/2024

Publicado em: 15/8/2024

¹ Universidade Federal de Sergipe. São Cristóvão/SE, Brasil. <https://orcid.org/0009-0009-7228-1394>

² Universidade Federal de Sergipe. São Cristóvão/SE, Brasil. <https://orcid.org/0009-0005-0411-6072>

³ Universidade Federal de Sergipe. São Cristóvão/SE, Brasil. <https://orcid.org/0000-0001-7974-304X>

INTRODUÇÃO

Com a popularização dos *smartphones* e o crescimento do número de usuários na internet durante a década de 2010, a sociedade vem se mostrando cada vez mais inserida no ambiente cibernético (Machado; Cleophas, 2023). Por conta disso, tarefas cotidianas estão sendo aceleradas e aprimoradas com o uso de ferramentas digitais (Custódio; Rodrigues, 2023). Graças à grande variedade de programas, sites e aplicativos que oferecem diversos serviços para o ambiente escolar e universitário, os alunos têm tido seus caracteres sociais remodelados. Ao mesmo tempo, o corpo docente escolar é provocado a buscar um maior entendimento e domínio dos recursos tecnológicos (Bento; Belchior, 2016).

A familiarização dos alunos com o ambiente digital gerou um aumento de interesse nos discentes por assuntos que envolvam o uso do computador, por exemplo, o ensino de programação (Ferreira; Braga, 2023). O processo acelerado de digitalização do ambiente escolar e a popularização do ensino de programação para crianças e adolescentes despertou o interesse das escolas em acrescentarem aos seus currículos temas relacionados à Olimpíada Brasileira de Informática (OBI)⁴. A Olimpíada Brasileira de Informática (OBI) é um evento anual realizado, visando despertar o interesse dos alunos pela ciência da computação. Uma de suas modalidades, chamada “programação”, envolve uma habilidade chamada “Programação Competitiva”, no qual, por meio de uma prova, os competidores devem resolver problemas de computação, seguindo as especificações da competição. O treinamento para a OBI geralmente envolve a aprendizagem teórica e a resolução prática de problemas. No âmbito do nível superior, há a Maratona Brasileira de Programação, um evento no qual os alunos formam trios e competem em busca de uma vaga no Concurso Internacional de Programação Colegial (ICPC). O treinamento para essa competição segue uma abordagem similar ao da OBI, mas com ênfase em uma ementa teórica mais avançada e complexa.

Coordenar o ensino de programação, seja ela competitiva ou não, pode ser desafiador, especialmente quando há uma diversidade de níveis de conhecimento dos alunos (Silveira; Tonini, 2023). Grandes turmas e diferentes estilos de aprendizagem podem dificultar a mediação entre os conteúdos programáticos e os alunos. É importante haver um acompanhamento individual e meticuloso de cada aluno para garantir que suas necessidades sejam atendidas. Alguns professores, também chamados de treinadores na programação competitiva, adotam abordagens, como calendários de treinamento e rotinas personalizadas, para lidar com a diversidade da turma. No entanto, muitas plataformas on-line limitam o treinamento às questões disponíveis em suas próprias plataformas. Além disso, a falta de comunicação direta entre o professor e o aluno pode gerar dúvidas e dificuldades no progresso do estudante. Quando o aluno não tem sentimentos positivos pelo ambiente escolar, seu desenvolvimento afetivo e cognitivo estará comprometido, visto que ambos os aspectos estão intimamente ligados. Assim sendo, “é essencial que haja no ambiente de estudo a presença de sentimentos afetivos entre professor e alunos”, e para alcançar esse objetivo, é crucial

⁴ OBI – <https://olimpiada.ic.unicamp.br/>

que o acompanhamento pedagógico de cada discente seja realizado de forma individual e meticulosa (Queiroz, 2016).

Por conta do grande montante de questões a serem resolvidas e por utilizarem do computador para acessar e resolvê-las, muitos alunos optam por continuar o treinamento extraclasse, usando seus próprios meios digitais. Tal decisão, quando deliberada pelo aluno, demonstra grande motivação, empenho e compromisso desse com seu treinamento. Entretanto, ao mesmo tempo que resolver questões em casa encoraja o aluno a construir suas soluções sozinho, a ausência do acompanhamento pessoal do professor pode ocasionar problemas na linearidade do aprendizado e gerar dúvidas que, caso tardem a serem sanadas, impedem o progresso do discente em seus estudos.

Algumas abordagens utilizadas por treinadores para solucionar o problema da pluralidade da turma são a adoção de calendários de treinamento e a criação de rotinas personalizadas para cada aluno ou grupo de alunos. Há ferramentas disponíveis no mercado que possibilitam a organização da turma e a delegação de tarefas para os alunos individualmente. No entanto, como será apresentado na Fundamentação Teórica, estes trabalhos costumam limitar os treinamentos às questões disponíveis nas suas próprias plataformas ou não foram construídos para um treinamento contínuo, mas sim para simulações de provas ou eventos de programação. Além desses, existem *softwares* de criação de comunidades on-line e bate-papos utilizados por professores para contatar seus alunos e viabilizar a discussão do conteúdo e a resolução de problemas. Essas soluções são bem-sucedidas quando aplicadas para promover a consonância de uma turma ou equipe. Todavia, não é seguro afirmar que todos os alunos se sentem à vontade para debater em grupo sobre suas ideias, preferindo uma comunicação individual com o professor. A aplicação proposta neste trabalho, o Kheiron, foi pensada para mitigar estes dois problemas: restrições no planejamento de treinamentos e dificuldade no atendimento pessoal do aluno. Por meio de um sistema de cadastro de problemas de outros sites e do próprio Kheiron, a aplicação dará liberdade ao professor para customizar o treinamento de seus alunos. Além disso, por meio da funcionalidade de bate-papo individual, cada aluno terá acesso direto ao professor para esclarecer suas dúvidas sobre cada questão do seu treinamento.

No contexto educacional, onde o tempo dos professores é frequentemente um recurso escasso (Piovezan; Ri, 2019), e os alunos requerem suporte constante, as Inteligências Artificiais (IAs) Generativas (*GPT – Generative Pre-trained Transformer*)⁵, como o *ChatGPT*⁶, emergem como aliadas importantes, desempenhando o papel de assistentes virtuais inteligentes. Esse modelo avançado de IA possui a capacidade de assimilar vastos volumes de informações, processar consultas complexas e fornecer respostas contextualmente ricas em tempo real. Assim, em cenários onde o professor está sobrecarregado ou o aluno precisa ter um suporte imediato, o uso de robôs dotados de IAs Generativas tem o potencial para reduzir o tempo de espera do aluno e permitir

⁵ Uma IA generativa é um tipo de inteligência artificial projetada para criar novos conteúdos, como texto, imagens, música e vídeos, com base em padrões aprendidos a partir de dados existentes.

⁶ O ChatGPT é um modelo de inteligência artificial generativa que gera respostas em linguagem natural.

ao professor se concentrar em tarefas críticas, mantendo, ao mesmo tempo, um canal de comunicação direta e eficaz aluno-professor.

O desenvolvimento do Kheiron representa uma contribuição promissora para aprimorar as metodologias de ensino em programação. Ao facilitar o acesso do aluno ao professor, a inclusão do robô aprimora ainda mais a experiência educacional, permitindo que os estudantes recebam assistência instantânea para suas dúvidas, ampliando, assim, o laço afetivo entre ambos. Esse ambiente de treinamento enriquecido pela IA não apenas desperta sentimentos positivos nos alunos, mas também oferece uma plataforma para uma análise pedagógica mais acurada por parte do instrutor, contribuindo para um ensino mais personalizado e eficaz.

Deste modo, o objetivo deste trabalho é desenvolver um sistema de preparação de rotinas de treinamento de programação, com foco em treinamento para competições de programação, visando o aumento da eficiência do ensino e da produtividade dos alunos por meio de uma melhor gestão do ritmo de aprendizagem de cada aluno. Para o desenvolvimento deste trabalho, aplicou-se uma pesquisa exploratória para trazer mais familiaridade com o assunto abordado (Gil, 2002). Com o objetivo de se inteirar no domínio do problema explorado neste trabalho, realizou-se uma revisão bibliográfica a partir de materiais já elaborados, como livros e artigos científicos (Gil, 2002). Os outros passos envolveram a pesquisa exploratória qualitativa de serviços existentes no mercado que atendessem demandas semelhantes à atendida neste trabalho. Por meio desse levantamento, foi possível identificar os pontos de melhorias e as principais funcionalidades de cada aplicação encontrada, além de levantar os requisitos necessários para a construção da aplicação proposta neste trabalho. Também foram elaborados os diagramas que descrevem as funcionalidades da aplicação e as relações entre as entidades que o compõem. Foi construída uma versão funcional do *software* e realizados testes de usabilidade por usuários selecionados a fim de assegurar a qualidade do produto.

1. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, são apresentados os assuntos que servem de base para o trabalho.

1.1 Programação competitiva

O ensino de programação é considerado um dos maiores desafios para a ciência da computação (Mcgettrick *et al.*, 2005). A superação desse desafio, entretanto, desenvolve nos educandos uma habilidade que se correlaciona fortemente com outras habilidades de abstração, conceitualização, planejamento e resolução. No entanto, existem grandes preocupações entre a comunidade acadêmica internacional de que quando nos propomos ensinar competências de programação aos estudantes, somos menos bem-sucedidos do que necessitamos e devemos ser. Numa fase inicial, a visão da programação, como sendo seca e sem inspiração, em vez de uma excitante busca criativa, desencoraja os potenciais candidatos à computação (Mcgettrick *et al.*, 2005).

Neste contexto, urge a adoção de uma metodologia que familiarize o estudante com a ciência da computação de maneira mais eficiente. O uso de jogos eletrônicos

no ambiente escolar se provou uma preponderante ferramenta de potencialização de aprendizagem em diversas áreas (Bomfoco; Azevedo, 2012). Com isso, cunhou-se o termo *gamificação* para se referir às metodologias que se apropriam de mecânicas, projetos e estratégias provenientes de jogos para resolver problemas e promover o aprendizado (Kapp, 2012).

A programação competitiva é um exemplo do sucesso obtido ao adotar a gamificação como estratégia de ensino. A utilização de ferramentas que estimulem o aluno a construir sua solução de forma autônoma, criando um ambiente competitivo entre os discentes e os recompensem por seus acertos vem gerando uma comunidade global em torno da programação competitiva. Os alunos lidam com os problemas de forma análoga a um jogo, transportando-os para um ambiente familiar, onde se sentem mais confortáveis para desenvolver suas ideias de traçar suas próprias soluções.

Há, hoje, diversas competições de programação com um crescente número de participantes. Os competidores utilizam fóruns, sites e aplicativos de discussões para compartilhar informações, tirar dúvidas e se manter bem-informados sobre o cenário competitivo.

1.2 Competições de programação

As competições de programação são eventos onde os alunos devem desenvolver, no tempo estabelecido pelas regras da competição, algoritmos eficientes que resolvam problemas conhecidos da ciência da computação (Halim; Halim, 2013). Tais problemas giram em torno de linguagens de programação imperativas e problemas de entrada e saída, por exemplo. Os algoritmos criados pelos competidores são submetidos a um juiz, software que testa e avalia a solução proposta pelo competidor. No decorrer da competição, são gerados relatórios classificando os competidores por número de problemas resolvidos corretamente e o tempo demandado pelo competidor para resolver de cada um deles (Ribeiro; Guerreiro, 2008). A essa modalidade de competição dá-se o nome de “programação competitiva”.

No Brasil, a Maratona Brasileira de Programação é um evento da SBC que ocorre há mais de duas décadas. Ela é a fase regional de um campeonato maior, o *Internacional Collegiate Programming Contest* (ICPC – Competição Internacional Universitária de Programação). A maratona é destinada para alunos de graduação e pós-graduação em áreas afins da computação. Neste campeonato, os competidores se inscrevem em trios para participarem da primeira fase da competição e os times que obtiverem melhor pontuação são classificados para a final. O trio que melhor pontuar na etapa final sagra-se campeão, e os melhores classificados garantem suas vagas para a Olimpíada Internacional de Informática (UPF, 2019).

Ao buscar familiarizar os alunos dos níveis fundamental e médio com a ciência da computação, além de despertar novos talentos, a SBC organiza anualmente a OBI. A olimpíada é composta pelas categorias de programação e iniciação onde, na primeira, os alunos resolvem problemas práticos de programação competitiva em um tempo limitado, implementando as soluções utilizando uma linguagem de programação, ao passo que na segunda, os atletas precisam responder questões teóricas envolvendo lógica computacional (OBI, 2022).

1.3 Juízes on-line

Durante os treinamentos, para julgar as soluções dos competidores, são utilizados Juízes On-line (JO), que testam e avaliam os algoritmos submetidos. Juízes On-line são plataformas web que permitem aos usuários submeterem seus algoritmos e receberem uma avaliação automática sobre sua exatidão e eficiência. Essas plataformas são utilizadas principalmente para treinamento e prática de programação competitiva, oferecendo uma variedade de problemas para resolver. Os JOs são essenciais para os competidores testarem suas habilidades, compararem seu desempenho com outros competidores e se prepararem para competições reais.

Figura 1 – Problema de programação competitiva

Detectando Colisões

Nome do arquivo fonte: colisoes.c, colisoes.cpp, ou colisoes.pas

Deteção de colisão é uma das operações mais comuns (e importantes) em jogos eletrônicos. O objetivo, basicamente, é verificar se dois objetos quaisquer colidiram, ou seja, se a interseção entre eles é diferente de vazio. Isso pode ser usado para saber se duas naves colidiram, se um monstro bateu numa parede, se um personagem pegou um item, etc.

Para facilitar as coisas, muitas vezes os objetos são aproximados por figuras geométricas simples (esferas, paralelepípedos, triângulos etc). Neste problema, os objetos são aproximados por retângulos num plano 2D.

Tarefa

Escreva um programa que, dados dois retângulos, determine se eles se interceptam ou não.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). Cada caso de teste contém duas linhas. Cada linha contém quatro inteiros (x_0, y_0, x_1, y_1 , sendo $0 \leq x_0 < x_1 \leq 1.000.000$ e $0 \leq y_0 < y_1 \leq 1.000.000$) separados por um espaço em branco representando um retângulo. Os lados do retângulo são sempre paralelos aos eixos x e y .

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha para cada caso de teste, contendo o número 0 (zero) caso não haja interseção ou o número 1 (um) caso haja.

Entrada	Entrada	Entrada
0 0 1 1	0 0 2 2	0 0 1 1
0 0 1 1	1 1 3 3	2 2 3 3
Saída	Saída	Saída
1	1	0

Fonte: OBI, 2007.

Problemas de programação competitiva são geralmente formulados por competidores experientes, chamados *problem setters*. Todo problema é composto por uma descrição geral que detalha a problemática e uma descrição das entradas e das saídas esperadas, podendo, também, conter exemplos, assim como exemplificado na Figura 1. Além disso, o *problem setter* também é responsável por fornecer todos

os arquivos de texto que servirão de entrada para os algoritmos submetidos pelos competidores como resposta àquele problema, além dos arquivos das saídas esperadas que serão comparados com as saídas produzidas pelos algoritmos dos competidores.

Após desenvolver um algoritmo para solucionar um determinado problema, o competidor deve, então, submetê-lo ao julgamento para obter sua pontuação naquele exercício. O processo de julgamento envolve executar o algoritmo para os diferentes arquivos de entrada definidos pelo *problem setter*, chamados “casos de teste”. Cada caso de teste é um arquivo de texto que alimentará o algoritmo do competidor via entrada de dados padrão da linguagem na qual o programa foi escrito. Por fim, são comparadas as saídas geradas pelo algoritmo submetido e as saídas esperadas definidas pelo *problem setter*. A pontuação do competidor no problema será proporcional à quantidade de casos de testes executados corretamente, ou seja, aqueles que produziram a saída esperada.

Para auxiliar o treinamento para as competições de programação, foram criados sites que disponibilizam problemas e realizam o julgamento de algoritmo dos usuários. Tais sites, chamados Juízes On-line (JOs), são os principais recursos utilizados por estudantes de programação competitiva para validar as suas soluções.

Uma rotina comum de treinamento para competições de programação envolve a busca e seleção de problemas relevantes nas diversas plataformas de JO. Os problemas elencados são, então, passados para os alunos a fim de testar o progresso de suas habilidades e identificar pontos de melhoria. Cada discente recebe uma lista de problemas para ser resolvida, que contém os endereços eletrônicos que direcionam o aluno para cada problema no JO específico. Lá, o aluno é responsável por ler e entender o problema, pensar e construir uma solução e, por fim, submetê-la a julgamento.

1.4 Trabalhos relacionados

O esforço colaborativo da comunidade de programação competitiva em garantir que novos estudantes se interessem pela modalidade e que os competidores aperfeiçoem suas habilidades levou à criação de várias plataformas de treinamento. Muitas dessas iniciativas contam com seus próprios Juízes On-line integrados e problemas originais desenvolvidos pela comunidade.

Apesar de serem bastante diversas, muitas não possuem a completude necessária para resolver as demandas que este trabalho visa atender. O treinamento para os torneios de programação competitiva requer uma relação forte entre professor e aluno para que este possa esclarecer suas dúvidas de forma célere e clara. Apesar de alguns JO ainda possuírem funcionalidades parecidas, muitos possuem um conjunto de problemas reduzidos, limitando o professor a selecionar apenas os disponíveis na própria plataforma.

1.4.2 Beecrowd

Uma dessas soluções é o site Beecrowd⁷. Essa plataforma conta com seu próprio JO e, em 2022, possuía mais de 2.000 problemas agrupados em nove categorias diferentes que abordam o conteúdo de programação competitiva e de outros ramos

⁷ Beecrowd – <https://beecrowd.com/pt/>

da tecnologia da informação. O site também possui uma funcionalidade para criação de turmas onde o professor pode adicionar seus alunos e criar competições internas e visualizar, em tempo real, o desempenho de seus alunos.

1.4.3 Neps Academy

No ar desde 2016, a plataforma do Neps Academy⁸ surgiu como uma iniciativa para compartilhar o conhecimento necessário para um estudante participar de competições de programação. Desde então, o site é alimentado com problemas de diversas competições de todo o mundo, além de questões elaboradas pelos próprios usuários. A plataforma também possui um vasto repositório de conteúdo em texto e vídeo para auxiliar os alunos a assimilarem os assuntos listados nas ementas das principais competições de programação.

1.4.4 Virtual Judge

O Virtual Judge⁹ é um site para organização de torneios de treinamento. Por ele, o docente é capaz de criar um torneio privado e selecionar problemas de diversos JO suportados pelo Virtual Judge. A plataforma não conta com um JO integrado, antes, porém, submete os algoritmos dos competidores diretamente no JO do problema em questão.

Por utilizar o JO das próprias plataformas que suporta, o Virtual Judge consegue expandir seu catálogo de problemas com muita facilidade. Em troca, abdica da capacidade de detalhar o progresso dos competidores, visto que o histórico e os detalhes do progresso de cada aluno ficam retidos nas próprias plataformas de JO onde o Virtual Judge submete.

O Virtual Judge é o trabalho cujo objetivo é possuir a maior similaridade com o do Kheiron. Todavia, não há, no Virtual Judge, um meio otimizado onde os alunos possam esclarecer suas dúvidas remotamente e através da própria plataforma. Além disso, é focado principalmente em torneios com tempos limitados e não em treinamentos a longo prazo.

1.4.5 The Huxley

O The Huxley¹⁰ é amplamente conhecido no meio acadêmico, sendo muito utilizado por docentes do nível superior para consolidar, de forma prática, o ensino das linguagens básicas de programação. Por conhecer seu principal uso, o The Huxley funciona por meio da criação de grupos, em que o professor pode cadastrar seus alunos ou gerar uma chave que os estudantes podem utilizar para ingressar de forma automática no grupo. A partir daí, o professor delega para os alunos algumas questões selecionadas que estejam dentro do assunto ministrado em sala de aula. A plataforma também permite a criação de ritmos de ensino, onde o professor definirá o prazo para que um conjunto de questões seja resolvido.

⁸ Neps Academy – <https://neps.academy/br>

⁹ Virtual Judge – <https://vjudge.net/>

¹⁰ The Huxley – <https://thehuxley.com/>

O The Huxley conta, em 2022, com cerca de 2.000 questões de diversos níveis de conhecimento cadastradas no próprio site. Por possuir seu próprio JO, o site valida as soluções dos alunos e retorna, em tempo real, a nota obtida naquele problema.

1.5 Análise de diferenciais

O Kheiron oferece uma abordagem diferenciada, focada inteiramente no auxílio pedagógico. Ele estabelece um contato direto e individualizado entre instrutor e aluno, proporcionando um ambiente de aprendizado colaborativo e personalizado. A aplicação não se propõe a ser um Juiz On-line ou um mero banco de questões, mas sim a fornecer suporte aos alunos e facilitar a interação com o instrutor. Comparado aos sistemas existentes, o Kheiron se destaca por sua abordagem centrada no suporte pedagógico e pela ênfase no contato direto entre instrutor e aluno, além de prover ferramentas de análise e de repasse de conteúdo. Ele busca preencher a lacuna presente nos recursos disponíveis atualmente, oferecendo uma solução mais completa e personalizada para o treinamento em programação competitiva.

Uma das principais funcionalidades do Kheiron é prover um sistema de cadastro de questões que não restrinja o rol de problemas selecionados pelo instrutor a um banco específico de questões ou a uma plataforma de JO. Dos trabalhos relacionados supracitados, somente o Virtual Judge oferece esta funcionalidade, muito embora seu objetivo não seja no treinamento de longo prazo, mas sim com foco em torneios e simulações de provas. A ideia cerne dessa funcionalidade é trabalhar de forma cooperativa com os JO já existentes. O aluno ainda é obrigado a visitar a plataforma do JO para ler a questão e submeter sua resposta, fazendo do Kheiron apenas um ambiente de comunicação entre o docente e o discente no qual este informa ao primeiro sobre suas dúvidas e sucesso na resolução das questões.

De forma semelhante a como é implementado no Neps Academy, no The Huxley e mais recentemente no Beecrowd, o Kheiron também conta com turmas personalizadas nas quais o professor pode montar os treinamentos de seus alunos. Ademais, o Kheiron conta com uma visualização de tabelas relatando o progresso individual dos alunos em cada turma. Tal visualização é de suma importância para compreender a turma como um todo e comparar o progresso dos estudos de cada competidor. Por meio desta tabela, também é possível identificar a taxa de acertos para cada questão, viabilizando o replanejamento do treinamento para adequar o nível de dificuldade para o grau de conhecimento dos alunos acerca dos assuntos abordados nas questões. Essa funcionalidade não está presente atualmente em nenhum dos trabalhos relacionados listados.

Além disso, o tempo do professor é um elemento crucial no contexto educacional, muitas vezes subestimado e sobrecarregado (Viegas, 2022). A multiplicidade de tarefas, que vão desde o planejamento de aulas até atividades administrativas, coloca uma pressão significativa sobre o tempo disponível para a reflexão e o desenvolvimento profissional. A urgência em cumprir com diversas tarefas acadêmicas, turmas demasiadamente grandes e lidar com demandas administrativas pode comprometer a qualidade do ensino, deixando pouco espaço para a inovação pedagógica e o aprimoramento contínuo. Nesse cenário, é essencial repensar como o tempo do professor é gerenciado, buscando equilibrar as demandas burocráticas com a necessidade vital de promover um ambiente educacional dinâmico e enriquecedor (Viegas, 2022).

Por outro lado, os alunos requerem constante e rápido suporte, mas muitas vezes ficam receosos e/ou reticentes em entrar em contato com o professor para solicitar auxílio. Assim, uma funcionalidade que separa o Kheiron das demais soluções é a disponibilização de um assistente virtual dotado de um modelo avançado de IA Generativa, o qual permite personalizar a experiência de aprendizagem em programação, adaptando-se às necessidades individuais dos alunos. Tal funcionalidade não exclui a possibilidade do aluno, a qualquer tempo, entrar em contato direto com o professor para solicitar apoio pedagógico individualizado e humanizado, mantendo, assim, o laço afetivo entre aluno e professor.

2 DESENVOLVIMENTO

Antes de iniciar o desenvolvimento técnico do sistema, foi realizado um levantamento detalhado dos requisitos funcionais e não-funcionais, visando garantir o atendimento das principais demandas do público-alvo do Kheiron. Para isso, foram conduzidas pesquisas e entrevistas com competidores e treinadores da Olimpíada Brasileira de Informática, a fim de identificar e listar as funcionalidades-chave do sistema. Alguns dos principais requisitos funcionais estão representados no Quadro 1.

O sistema permite que os professores criem turmas personalizadas e selecionem questões de qualquer banco de questões ou JO para seus alunos. Ao criar uma turma, o professor informa algumas informações básicas, como nome, descrição e uma senha de acesso à turma. Após a criação, é disponibilizado um código exclusivo para a turma, o qual é compartilhado com os alunos que, com a senha criada pelo treinador, são utilizados para ingressar na turma.

Quadro 1 – Principais Requisitos Funcionais

Código	Identificação	Classificação	Objetivo
RF 006	Cadastrar turma	Essencial	O professor deve ser capaz de cadastrar uma turma.
RF 007	Ingressar em uma turma	Essencial	O aluno deve ser capaz de ingressar em uma turma.
RF 009	Criar assunto	Secundário	O professor deve ser capaz de criar um tópico.
RF 012	Criar questão	Essencial	O professor deve ser capaz de cadastrar uma questão.
RF 017	Responder questão	Essencial	O aluno deve ser capaz de marcar uma questão como “respondida”.
RF 018	Marcar questão com dúvida	Secundário	O aluno deve ser capaz de marcar uma questão como “em dúvida”.
RF 019	Criar dúvida	Essencial	O aluno deve ser capaz de cadastrar uma dúvida sobre uma questão.
RF 020	Responder dúvida	Essencial	O professor deve ser capaz de responder uma dúvida do aluno em uma questão.
RF 021	Responder dúvida	Essencial	O robô deve ser capaz de responder uma dúvida do aluno em uma questão.

Fonte: Elaborada pelos autores, 2024.

No painel da turma, o professor tem a opção de cadastrar novos tópicos e, dentro de cada tópico, suas questões. Os tópicos servem como agrupadores de questões, visando

facilitar a divisão e acompanhamento do conteúdo. Ao criar as questões, o treinador deve informar um nome para ela, o endereço eletrônico do site onde aquela questão está cadastrada (aqui podendo ser um banco de questões ou um JO) e um peso para a questão.

É importante pontuar que um diferencial do Kheiron é não se ater a plataformas ou a JO específicos, possibilitando o cadastro de questões com qualquer plataforma. Isso configura como um agregador de questões, facilitando a passagem de conteúdo do professor para o aluno e o seu acompanhamento pedagógico.

Para auxiliar na construção das telas do sistema, foi criado antecipadamente um protótipo das interfaces. Dessa forma, foi possível validar a usabilidade do sistema com os potenciais usuários e realizar ajustes tanto em termos de usabilidade quanto de design antes de iniciar o desenvolvimento das telas propriamente dito.

Um exemplo de funcionalidade do sistema é a página de progresso (Figura 2), na qual os professores podem visualizar uma tabela que relaciona os alunos da turma e as questões cadastradas por tópicos. Cada célula da tabela exibe um ícone indicando a situação de cada aluno em relação àquela questão específica: um sinal de “check” verde representa que o aluno marcou a questão como concluída, uma bandeira amarela indica que o aluno tem dúvidas e um “x” cinza indica que o aluno ainda não tentou solucionar aquela questão. Ao analisar as questões e o progresso individual de cada aluno, o professor pode identificar quais questões geram mais dificuldades e fornecer um atendimento personalizado para sanar suas dúvidas.

Algumas células da tabela são marcadas com um pequeno círculo vermelho no canto superior direito, indicando que o aluno utilizou o recurso de chat para aquela questão. O módulo de chat fornece uma ferramenta de comunicação individual e exclusiva para cada problema. Essa funcionalidade é de extrema importância para permitir que o aluno tenha acesso fácil e direto ao instrutor, sem a necessidade de recorrer a outras ferramentas de chat ou esperar por encontros presenciais da turma (Figura 3). Isso possibilita que o aluno tire suas dúvidas remotamente e fortaleça seu vínculo com o professor.

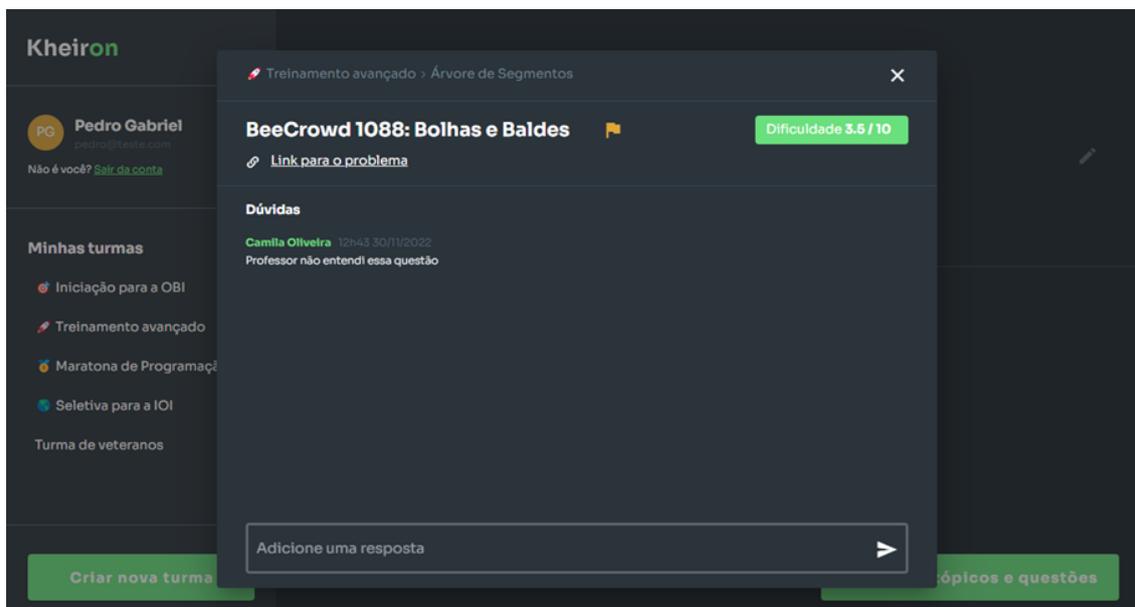
Figura 2 – Funcionalidade de visualização de progresso dos alunos da turma vista pelo professor

The screenshot shows the Kheiron web application interface. On the left, there is a sidebar with the user's name 'Pedro Gabriel' and a list of courses: 'Iniciação para a OBI', 'Treinamento avançado', 'Maratona de Programação', 'Seletiva para a IOI', and 'Turma de veteranos'. The main area displays the 'Treinamento avançado' course details, including the code '6AEURCPD' and the professor's name 'Pedro Gabriel'. Below this, there is a table titled 'Árvore de Segmentos' showing progress for three students: Maria Gabriela, Eduardo Santana, and Camila Oliveira. The table has columns for topics A, B, C, D, and E. Maria Gabriela has a red circle with '0' in the A column and 'x' in B, C, D, and E. Eduardo Santana has a yellow circle with '2' in the A column, green checkmarks in B and C, and a yellow flag in D. Camila Oliveira has a yellow circle with '3' in the A column, green checkmarks in B, C, and D, and a yellow flag in E. At the bottom, there are two buttons: 'Criar nova turma' and 'Visualizar tópicos e questões'.

Aluno	A	B	C	D	E
Maria Gabriela	0	x	x	x	x
Eduardo Santana	2	✓	✓	🚩	x
Camila Oliveira	3	✓	✓	✓	🚩

Fonte: Elaborada pelos autores, 2024.

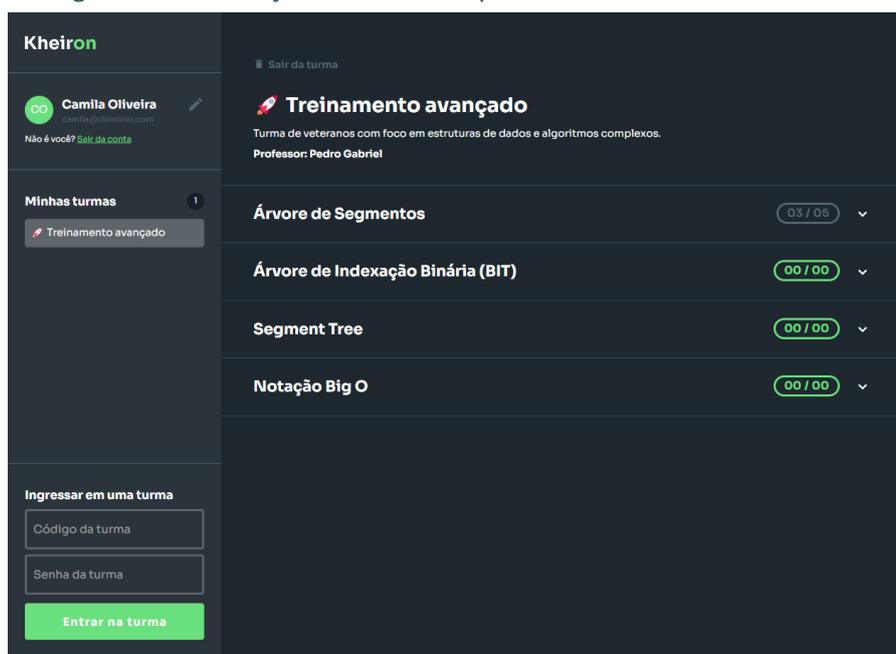
Figura 3 – Funcionalidade de chat entre o professor e o aluno.



Fonte: Elaborada pelos autores, 2024.

Para o aluno, a tela de turma, exibida na Figura 4, conta somente com recursos de visualização, não sendo possível editar turmas existentes ou criar novas turmas. Cada tópico possui um contador, sinalizando ao aluno seu progresso naquele assunto. Na Figura 5, é possível ver as questões após expandir algum tópico da turma. Cada questão traz consigo o nome, endereço eletrônico e a dificuldade do problema cadastrado pelo professor, além disso, há dois botões de ação para o aluno: o botão à esquerda com um ícone de afirmação marca a questão como resolvida, ao passo que o ícone de uma bandeira à direita sinaliza que o usuário tem uma dúvida na questão.

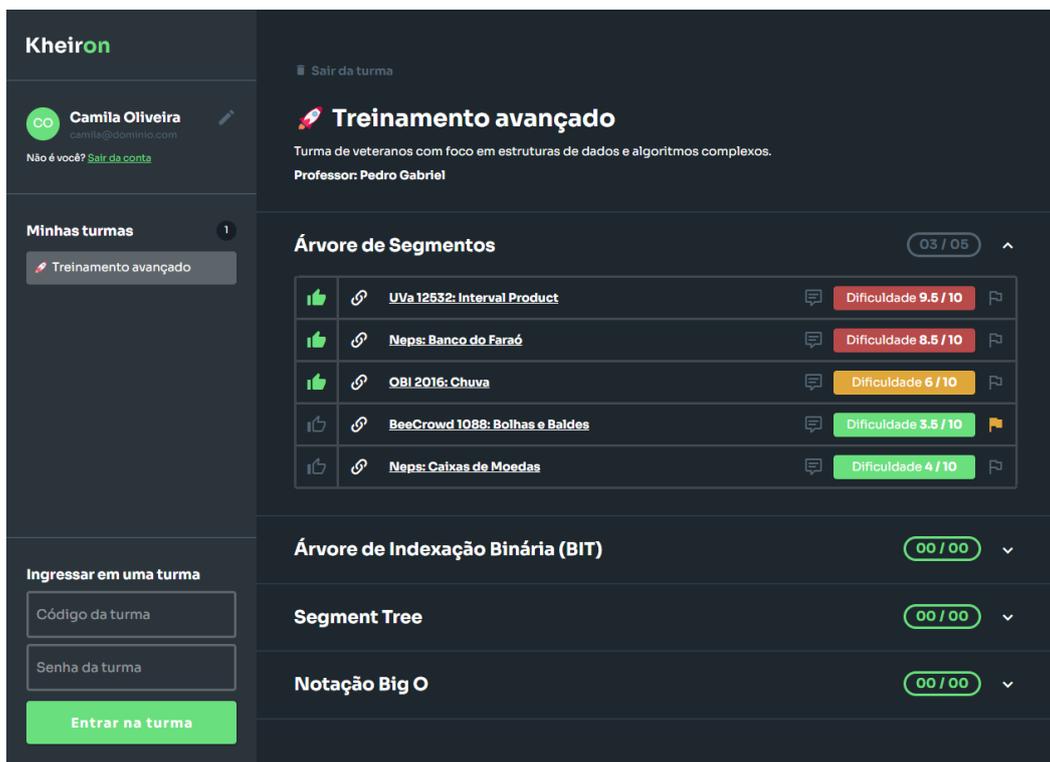
Figura 4 – Informações visualizadas pelo aluno com dados da turma.



Fonte: Elaborada pelos autores, 2024.

O sistema foi desenvolvido utilizando o padrão Model-View-Controller (MVC), no qual o modelo encapsula os dados e as funcionalidades. A visão é responsável pela renderização dos dados e o controle traduz as requisições dos usuários em funcionalidades da aplicação.

Figura 5 – Informações visualizadas pelo aluno, incluindo os detalhes da turma, os tópicos e suas respectivas questões



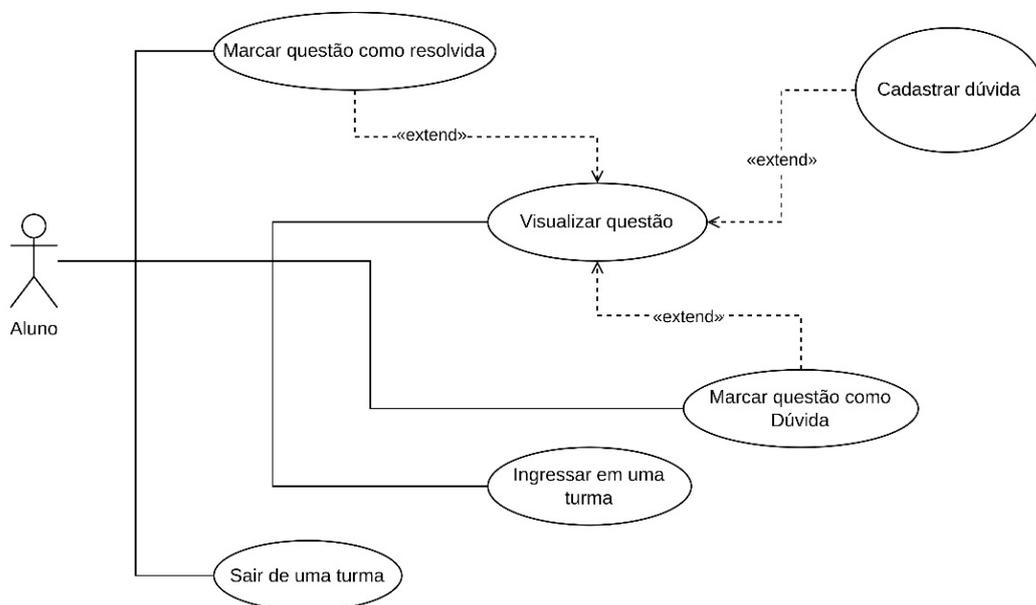
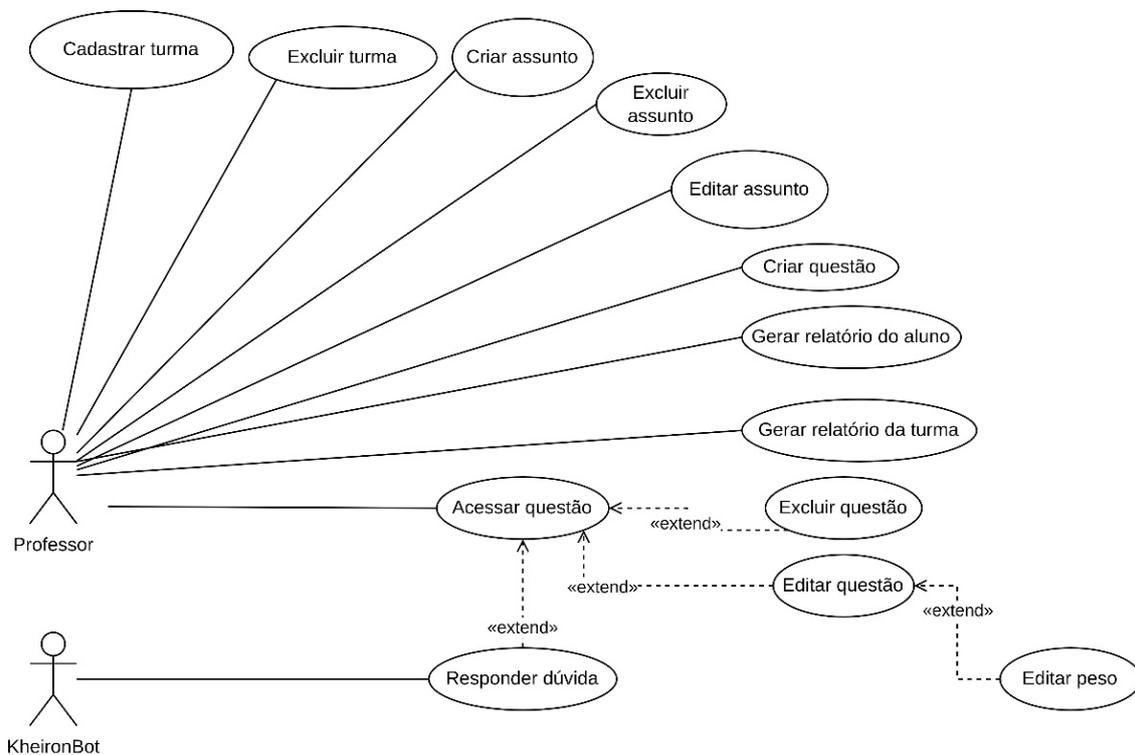
Fonte: Elaborada pelos autores, 2024.

Figura 6 – Tela de chat vista pelo aluno com interação com o robô



Fonte: Elaborada pelos autores, 2024.

Figura 7 – Diagrama de casos de uso do sistema

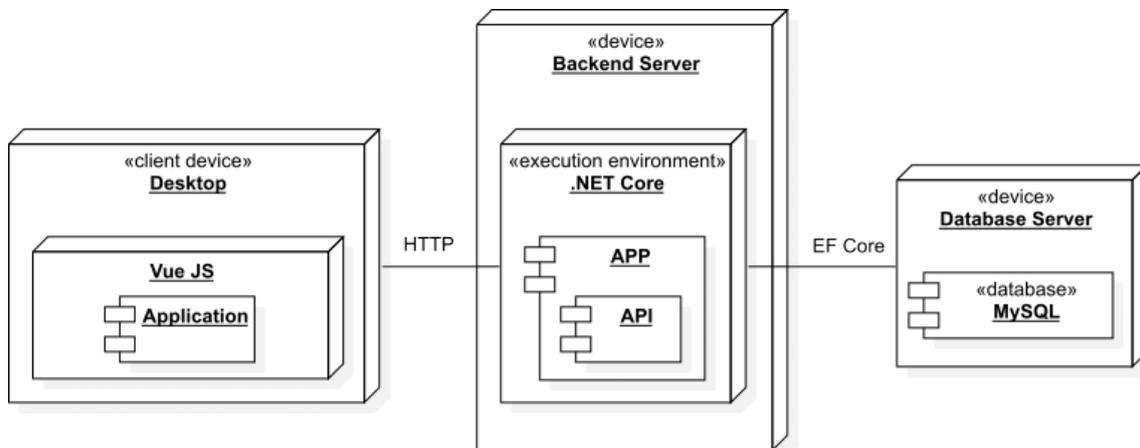


Fonte: Elaborada pelos autores, 2024.

Foram utilizados o Vue.js¹¹, como framework de JavaScript¹² para o frontend¹³, o .NET¹⁴, como framework¹⁵ de desenvolvimento para o backend¹⁶, o MySQL¹⁷, como sistema de gerenciamento de banco de dados, o ChatGPT, como IA Generativa, e Selenium WebDriver¹⁸ para extração de dados de JOs e sites educacionais. Para descrever o funcionamento do sistema de forma unificada, foram utilizados diagramas da Unified Modeling Language (UML)¹⁹, como o diagrama de casos de uso (Figura 7) e o diagrama de classes. Além disso, é possível analisar os diagramas de implementação do sistema (Figura 8) e o diagrama de pacotes (Figura 9), que fornecem uma compreensão sobre aspectos do desenvolvimento, como a estrutura dos diretórios e a arquitetura do sistema, bem como as relações entre os serviços que o compõem.

Ademais, a integração do *Selenium* desempenha um papel crucial ao extrair informações pertinentes dos sites educacionais e JOs, permitindo a automatização no cadastramento de questões. Esses dados, uma vez coletados, são armazenados de maneira estruturada no banco de dados. O uso de um robô, o *Kheiron Bot*, por meio da API (*Application Programming Interface*) fornecida pelo ChatGPT, se torna a peça central para melhorar o suporte dado ao aluno, como mostrado na Figura 6. Por meio de solicitações, a API é acionada para fornecer respostas relevantes e instrutivas. Essa abordagem automatizada não apenas otimiza a eficiência no atendimento às consultas dos alunos, mas também cria uma experiência educacional mais dinâmica e interativa, em que a tecnologia é empregada para aprimorar a qualidade do aprendizado.

Figura 8 – Diagrama de implementação do sistema



Fonte: Elaborada pelos autores, 2024.

¹¹ Vue.js é um framework JavaScript progressivo usado para construir interfaces de usuário.

¹² JavaScript é uma linguagem de programação para criar conteúdo dinâmico em páginas web.

¹³ Frontend refere-se à parte visual e interativa de um site ou aplicação web que os usuários veem e interagem diretamente.

¹⁴ .NET é uma plataforma de desenvolvimento da Microsoft usada para construir aplicativos.

¹⁵ Framework é uma estrutura de software reutilizável que fornece suporte e funcionalidade padrão para o desenvolvimento de aplicações.

¹⁶ De modo oposto ao frontend, não possui interface e está localizado em servidores remotos.

¹⁷ MySQL – <https://www.mysql.com/>

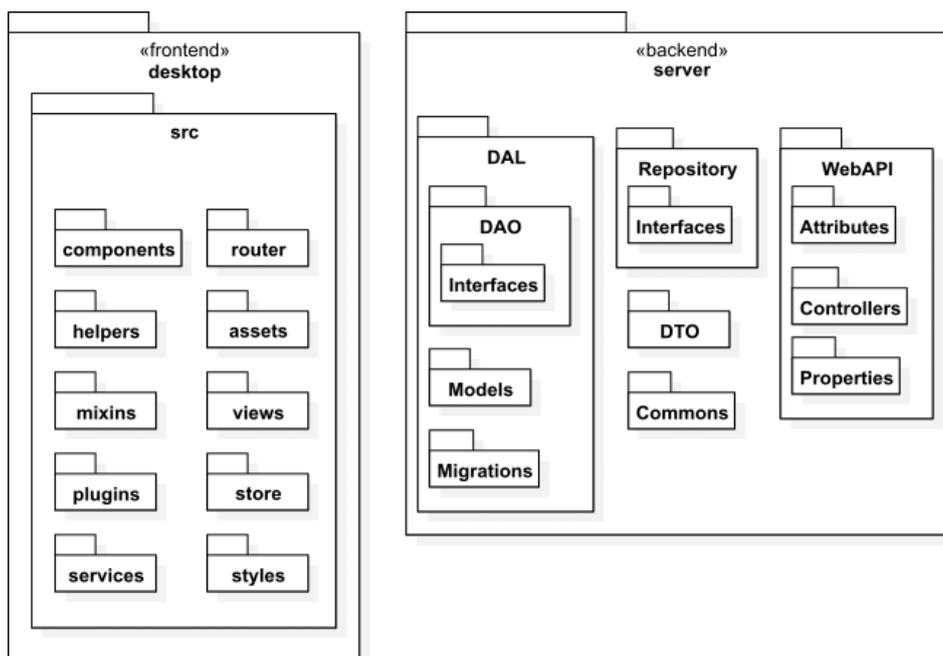
¹⁸ É uma ferramenta de automação para testar e manipular aplicativos web.

¹⁹ UML é uma linguagem visual usada para modelar sistemas de software.

Além disso, a evolução constante de tecnologias, como o ChatGPT, promete abrir novos horizontes para aprimorar a interação aluno-professor em ambientes educacionais. A capacidade do modelo de linguagem em entender contextos complexos e gerar respostas coesas proporciona respostas precisas e personalizadas às dúvidas dos alunos. Com a contínua atualização e refinamento dos modelos, as respostas geradas pelo robô serão mais contextualizadas e precisas, contribuindo para uma experiência de aprendizado mais enriquecedora. Ademais, à medida que novos recursos são incorporados ao ChatGPT, o Kheiron estará posicionado para aproveitar tais inovações, potencializando ainda mais a automação das respostas e proporcionando uma assistência mais eficaz e personalizada aos alunos, promovendo um ambiente educacional cada vez mais dinâmico e eficiente (Impacta, 2023).

Durante o desenvolvimento dos fluxos de criação, edição, remoção e leitura das entidades do sistema, uma atenção especial foi dada para garantir a proteção dos dados dos usuários e permitir o compartilhamento apenas mediante autorização. Para isso, foi implementada uma codificação de mão única. É comum encontrar sistemas que armazenam senhas de forma inadequada. Nesse contexto, a técnica de “SALT” em conjunto com um algoritmo de criptografia de mão única é considerada a maneira correta de armazenar informações sigilosas (Anderson, 2020). No caso específico, as senhas são armazenadas usando o algoritmo de sintetização de mensagem MD5 (*Message Digest Algorithm 5*), que converte um texto em um *hash* de 32 caracteres, tornando inviável a sua reversão para a mensagem original em tempo hábil. No entanto, o uso exclusivo desse algoritmo pode ser insuficiente em muitos casos, uma vez que existem ferramentas de dicionário que catalogam milhões de *hashes* gerados pelo algoritmo e os associam às mensagens correspondentes, facilitando a reversão do *hash* original.

Figura 9 – Diagrama de pacotes do sistema



Fonte: Elaborada pelos autores, 2024.

A utilização do “SALT” envolve a adição de informações aleatórias ao final dos dados antes da aplicação da função de *hash*. Essa adição garante que mesmo que dois usuários insiram a mesma senha, o valor gerado pela criptografia nunca será o mesmo (Joshi; Saini; Rathee, 2020). Portanto, cada vez que um novo usuário é criado, a senha informada é concatenada com um “SALT” de 16 caracteres aleatórios antes de ser aplicado o algoritmo de criptografia MD5. O *hash* resultante e o “SALT” são armazenados em colunas separadas no banco de dados, garantindo que nenhuma informação sensível do usuário seja armazenada no sistema. Essa abordagem fornece uma camada adicional de segurança aos dados sintetizados pelo algoritmo MD5. Dois usuários podem ter a mesma senha de base, mas os *hashes* armazenados na coluna de senha são completamente diferentes devido aos “SALTs” gerados aleatoriamente. Para fornecer mais detalhes sobre sua eficácia, foram cadastrados dois usuários com a mesma senha. Como pode ser observado na Figura 10, os *hashes* armazenados são completamente diferentes. Essa medida tem como principal objetivo evitar o uso de “Rainbow Tables”, que são bancos de senhas contendo milhões de *hashes* previamente calculados. Dessa forma, mesmo em caso de vazamento do conteúdo do banco de dados do Kheiron, não é possível obter o conteúdo original das senhas em tempo hábil.

Figura 10 – Eficiência da técnica SALT comparando os registros de dois usuários que cadastraram a mesma senha

Id	Name	Email	Salt	Password	IsInstructor
18	Usuário 1	usuario1@email.com	GCoVlhZlneNbqPYOxt7SwA==	K+XMd8eLaX8EChHirLNnV0PFSa0HT6xXez5jtKjBW1dkKgG3mIEU...	0
19	Usuário 2	usuario2@email.com	NEDH1mMY9+j1p5JKv91jdA==	EFnaAnoWfRb8zhH1xSPepSG47JKpeu8ebPIFYp8frW9x0ZxWZt...	0

Fonte: Elaborada pelos autores, 2024.

Outra restrição imposta ao sistema é a garantia de um controle de acesso da aplicação. Para isso, foi desenvolvido um sistema de sessões e permissões para garantir que os conteúdos criados por um usuário sejam disponibilizados apenas para ele mesmo e para aqueles aos quais o acesso foi concedido. Ao entrar no sistema, o navegador utilizado pelo usuário recebe e armazena um *token*, um código único, criptografado e intransferível, referente àquela sessão específica. Na plataforma, todas as requisições feitas pelo usuário para visualizar, criar, editar ou excluir algum conteúdo ou informação devem conter o *token*. Dessa forma, o *backend* valida se o usuário tem permissão para realizar a operação solicitada. Essa funcionalidade cria uma hierarquia de permissões e restringe o usuário a realizar requisições somente aos conteúdos para os quais ele possui privilégio de leitura ou gravação.

3. IMPLANTAÇÃO

Após a conclusão do desenvolvimento do sistema, foi necessário publicá-lo para viabilizar a realização de testes de usabilidade com os usuários. Inicialmente, os testes eram realizados pelo desenvolvedor no mesmo computador onde a aplicação era construída. À medida que o projeto progredia em escopo e funcionalidades, tornou-se necessário validar o comportamento com os potenciais usuários, como treinadores e competidores.

Foram consideradas várias opções para disponibilizar a aplicação em um ambiente on-line, a fim de permitir o acesso dos alunos e dos professores. No entanto, devido aos custos de manutenção da computação em nuvem e ao tamanho reduzido do grupo que testaria o programa, não parecia viável contratar serviços de IaaS (Infraestrutura como Serviço)²⁰. Em vez disso, optou-se por disponibilizar o acesso à plataforma diretamente pelo computador, executando uma versão local do sistema.

Após a conclusão do desenvolvimento do sistema, foram realizadas sessões de testes, principalmente com treinadores, a fim de obter informações sobre os casos de uso e funcionalidades desenvolvidas. Durante esses testes de usabilidade, foram identificados problemas de comportamento, bugs e oportunidades de melhoria para aumentar a produtividade dos usuários.

Durante algumas semanas, foram coletados pontos que necessitavam de correções, os quais foram prontamente resolvidos e publicados para retestes. Embora o sistema esteja completo em termos de requisitos, ainda é necessário realizar melhorias de desempenho e outras correções antes de ser devidamente publicado na internet para uso da comunidade como um produto. Essas melhorias e correções são essenciais para garantir uma experiência de uso mais satisfatória e um desempenho otimizado do sistema antes de disponibilizá-lo ao público em geral.

CONSIDERAÇÕES FINAIS

A popularização do ensino de computação para crianças e jovens se correlaciona fortemente com o aumento do interesse das escolas e das instituições de ensino pela programação competitiva como uma forma eficaz de educar seus discentes. O crescimento do número de competidores revelou a necessidade de uma ferramenta capaz de monitorar o progresso de seus estudos. A aplicação *web* Kheiron emerge como uma resposta aos desafios prementes enfrentados pelos professores em relação à gestão do tempo. Diante da sobrecarga de tarefas e das demandas burocráticas que frequentemente comprometem a qualidade do ensino, o Kheiron se destaca como uma ferramenta inovadora, proporcionando não apenas alívio na criação e administração de questões, mas também oferecendo um auxílio valioso aos alunos por meio de um assistente dotado de IA.

A fim de otimizar o processo de treinamento de alunos para a OBI e outras competições de programação, bem como providenciar ao docente ferramentas que possibilitem a análise do desempenho de seus educandos, foi construído o sistema Kheiron. A aplicação proposta neste trabalho visa cadastrar tarefas individuais para alunos de uma turma, juntamente com ferramentas por onde eles serão capazes de sanar suas dúvidas de forma individual. Por meio do Kheiron, os professores terão controle sobre os problemas passados para seus alunos, possibilitando analisar a taxa de acerto por questão e identificar para cada aluno quais assuntos geram mais dificuldade. Desse modo, o treinador é capaz de aprimorar seu método de seleção de problemas e montar rotinas de treinos individuais que correspondam aos ritmos de aprendizagem de cada aluno.

²⁰ IaaS é um modelo de computação em nuvem que fornece recursos de infraestrutura virtualizados pela internet.

Além de aprimorar a produtividade e a eficiência tanto dos docentes quanto dos alunos, o Kheiron também serve como um canal de comunicação direta entre os alunos e professores. Por meio da plataforma, o discente é capaz de sanar suas dúvidas diretamente com o treinador, sem a necessidade de aguardar até o próximo encontro síncrono. Para alcançar esse objetivo, foram coletadas informações sobre os torneios de programação competitiva e metodologias de ensino empregadas no treinamento preparatório para esses eventos. É importante destacar também que o Kheiron fornece um robô, ou seja, um assistente pessoal que utiliza IA generativa para fornecer suporte aos alunos de modo ágil e eficiente.

Após isso, buscou-se avaliar outras soluções existentes no mercado, investigando pontos de melhorias para garantir o diferencial deste trabalho. Após o levantamento de requisitos, a fundamentação teórica foi escrita, no qual buscou elucidar os conceitos teóricos cuja assimilação era necessária para o entendimento deste trabalho. Além disso, também foram elencadas as principais soluções para problemas similares ao abordado neste trabalho, pontuando as funcionalidades de cada uma.

O desenvolvimento da aplicação *web* Kheiron foi pensado de acordo com a arquitetura de *software* MVC. A construção das interfaces gráficas foi guiada pelo protótipo de telas desenvolvido neste trabalho. A utilização de ferramentas e tecnologias gratuitas embaratece os custos necessários para desenvolver e manter o programa. É importante pontuar também que, apesar dos requisitos levantados para o Kheiron dizerem respeito ao ecossistema de programação competitiva, as funcionalidades implementadas são genéricas o suficiente para permitir que o professor as aplique em outros âmbitos. O sistema também pode ser utilizado em ensinamentos de programação competitiva, algoritmos e técnicas de programação em universidades e escolas. Também é possível que sua utilização vá além do uso na ciência da computação, podendo ser aplicado em outros tipos de atividades que envolvam resolução de problemas na internet.

A possibilidade futura de utilizar a própria inteligência artificial para gerar questões e elaborar relatórios representa um avanço significativo na otimização do tempo do professor (Li; Zhou, 2018), permitindo um enfoque mais concentrado no desenvolvimento profissional e na inovação pedagógica, de tal modo que o centro da atenção do professor seja o aluno.

REFERÊNCIAS

- ANDERSON, R. *Security engineering: a guide to building dependable distributed systems*. 3rd ed. [S. l.]: Wiley, 2020.
- BENTO, L.; BELCHIOR, G. Mídia e educação: o uso das tecnologias em sala de aula. *Revista de Pesquisa Interdisciplinar*, v. 1, 2016.
- BOMFOCO, M. A.; AZEVEDO, V. A. Os jogos eletrônicos e suas contribuições para a aprendizagem na visão de J. P. Gee. *Renote – Novas Tecnologias na Educação*, no. 3, p. 8, 2012.
- CUSTÓDIO, N. M.; RODRIGUES, A. Tecnologias e formação inicial docente: O papel do professor formador na construção do pensamento crítico e da cidadania digital. *Revista Contexto & Educação*, v. 38, n. 120, 14 nov. 2023, p. e12765.
- FERREIRA, R.; BRAGA, M. Ensino de programação e as estratégias pedagógicas utilizadas no Brasil. *Revista Contexto & Educação*, v. 38, n. 120, 20 jun. 2023, p. e11377.
- GIL, A. C. *Como elaborar projetos de pesquisa*. 4a. [S. l.]: Atlas, 2002.

- HALIM, S.; HALIM, F. *Competitive Programming 3*. [S. l.]: lulu, 2013.
- IMPACTA. *Chat GPT x Educação: enxergando a tecnologia a favor do ensino*. 2023. Disponível em: <https://www.impacta.com.br/blog/chat-gpt-educacao-enxergando-a-tecnologia-a-favor-do-ensino/>. Acesso em: 15 out. 2023.
- JOSHI, S.; SAINI, H.; RATHEE, G. SALT cryptography for privacy in mobile crowdsourcing. *International Journal of Information Technology*, v. 12, n. 2, 14 jun. 2020, p. 585–591.
- KAPP, K. M. *The gamification of learning and instruction: game-based methods and strategies for training and education*. [S. l.]: Pfeiffer & Company, 2012.
- LI, C.; ZHOU, H. Enhancing the Efficiency of Massive Online Learning by Integrating Intelligent Analysis into MOOCs with an Application to Education of Sustainability. *Sustainability*, v. 10, n. 2, 9 fev. 2018, p. 468.
- MACHADO, M. S. M.; CLEOPHAS, M. G. Estratégia metacognitiva no ensino de ciências: Fornecendo o feedback com apoio das tecnologias digitais. *Revista Contexto & Educação*, v. 38, n. 120, 14 nov. 2023, p. e13447.
- MCGETTRICK, A.; BOYLE, R.; IBBETT, R.; LLOYD, J.; LOVEGROVE, G.; MANDER, Keith. Grand Challenges in Computing: Education – A Summary. *The Computer Journal*, v. 48, n. 1, 1 jan. 2005, p. 42–48.
- OBI. *Caderno de Tarefas*, 2007. Disponível em: https://olimpiada.ic.unicamp.br/static/extras/obi2007/provas/ProvaOBI2007_f1p1.pdf. Acesso em: 10 dez. 2023.
- OBI. *Sobre a OBI*, 2022. Disponível em: <https://olimpiada.ic.unicamp.br/info>. Acesso em: 5 dez. 2023.
- PIOVEZAN, P. R.; RI, N. M. Dal. Flexibilização e Intensificação do Trabalho Docente no Brasil e em Portugal. *Educação & Realidade*, v. 44, n. 2, 2019, p. 9–17.
- QUEIROZ, F. A. *A relação professor/aluno: importância dos vínculos afetivos ao processo de ensino aprendizagem*. 2016. Disponível em: <https://monografias.brasilecola.uol.com.br/educacao/relacao-professor-aluno-importancia-dos-vinculos-afetivos-ao-processo-de-ensino-aprendizagem.htm>. Acesso em: 5 dez. 2023.
- RIBEIRO, P.; GUERREIRO, P. *Early introduction of competitive programming. Olympiads in Informatics*, v. 2, 2008, p. 149–162.
- SILVEIRA, A. C. J.; TONINI, A. M. Evasão discente em cursos de engenharia de computação e o trabalho em TIC – Tecnologia da Informação e Comunicação. *Revista Contexto & Educação*, v. 38, n. 120, 27 nov. 2023, p. e10770.
- UPF. *Maratona de Programação SBC*. 2019. Disponível em: <https://www.upf.br/maratonasbc>. Acesso em: 10 dez. 2023.
- VIEGAS, M. F. Trabalhando Todo O Tempo: Sobrecarga E Intensificação No Trabalho De Professoras Da Educação Básica. *Educação E Pesquisa: Revista Da Faculdade De Educação Da Universidade De São Paulo*, v. 48, n. 1, 2022.

Autor correspondente:

Rafael Oliveira Vasconcelos

Universidade Federal de Sergipe

Cidade Univ. Prof. José Aloísio de Campos - Av. Marcelo Deda Chagas, s/n, Bairro Rosa Elze. São Cristóvão/SE, Brasil. CEP 49107-230.

rafael@dcomp.ufs.br

Este é um artigo de acesso aberto distribuído
sob os termos da licença Creative Commons.

