

# Expert Bee: Um Sistema Especialista de Apoio à Aprendizagem Integrada ao Moodle e Beecrowd

Raquel Cristina Schaly Behrens<sup>1</sup>, Maicon Rafael Zatelli<sup>1</sup>

<sup>1</sup>Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina (UFSC)

**Abstract.** *The use of technology in education has the potential to strengthen the teaching and learning relationship between educators and students, allowing access to materials, timetables and forums through virtual environments. Moodle is an example of a virtual platform that offers pedagogical support to educational institutions, and is widely used among universities. As well as simulating an online classroom, it has tools to support Information Technology and Computing courses, such as VPL. VPL is a Moodle-integrated module and programming task manager, but it has interface and usability problems. On the other hand, there are platforms not linked to Moodle that have an online judge for programming problems, such as Beecrowd. Beecrowd, by having ready-made programming questions, each with test batteries, which provide real-time feedback, allows users to develop programming and problem-solving skills. In addition, this platform is attracting growing interest in educational environments. This is partly due to its vast collection of questions and also because it is available in Brazilian Portuguese. This makes the platform accessible to a wider audience, especially those who are taking their first steps in the field and are not yet proficient in reading English. Therefore, the aim of this study is to favor the use of the Beecrowd platform among educators, leveraging the existing LTI integration between Moodle and Beecrowd, developed by the platform's team. Additionally, it aims to provide a tool based on an expert system to assist both teachers and students in clarifying recurring doubts about Beecrowd problems, facilitating their adaptation to the platform.*

**Resumo.** *A utilização da tecnologia na educação tem o potencial de fortalecer a relação de ensino e aprendizagem entre educadores e alunos, permitindo o acesso a materiais, cronogramas e fóruns por meio de ambientes virtuais. O Moodle é um exemplo de plataforma virtual que oferece suporte pedagógico a instituições de ensino, e é amplamente utilizado dentre as universidades. Ele, além de simular uma sala de aula online, possui ferramentas de suporte a cursos da área de Tecnologia da Informação e da Computação, como o VPL. O VPL é um módulo integrado ao Moodle e gerenciador de tarefas de programação, mas que apresenta problemas de interface e usabilidade. Em contrapartida, existem plataformas não vinculadas ao Moodle que possuem um juiz online para problemas de programação, como o Beecrowd. O Beecrowd, ao possuir questões prontas de programação, cada uma com baterias de testes, que fornecem feedback em tempo real, permite aos usuários desenvolver habilidades de programação e resolução de problemas. Além disso, essa plataforma desperta um interesse crescente nos ambientes pedagógicos. Isso ocorre, em parte, devido à sua vasta coleção de questões e também por estar disponível*

*em português brasileiro. Isso torna a plataforma acessível a um público mais amplo, especialmente àqueles que estão dando os primeiros passos na área e ainda não possuem proficiência em inglês. Dessa forma, o objetivo deste trabalho consiste em favorecer o uso da plataforma Beecrowd entre docentes, aproveitando a integração LTI já existente entre o Moodle e o Beecrowd, desenvolvida pela equipe da plataforma. Além disso, busca-se disponibilizar uma ferramenta baseada em sistema especialista para auxiliar professores e alunos no esclarecimento de dúvidas recorrentes sobre questões do Beecrowd, facilitando a adaptação de ambos ao uso da plataforma.*

## **1. Introdução**

A utilização da tecnologia no âmbito educacional aumenta o potencial de um vínculo proveitoso entre educadores e alunos, possibilitando a disponibilização de materiais, cronogramas, fóruns, entre outras atividades, por ambientes virtuais, e motivando, assim, os educandos [Francisco et al. 2016]. Isto posto, há ambientes virtuais de aprendizado que servem de apoio às instituições de ensino no gerenciamento pedagógico dos cursos, como o software livre Moodle – (*Modular Object-Oriented Dynamic Learning Environment*) – Ambiente de Aprendizagem Dinâmico Orientado a Objetos Modulares. O Moodle disponibiliza diversas ferramentas computacionais que proporcionam acesso a materiais didáticos, tarefas interativas e integração entre os membros do curso em que estão matriculados, reproduzindo, dessa forma, uma sala de aula [Lima 2021].

Por conseguinte, a fim de proporcionar uma maior assistência à cursos da área de tecnologia da informação, algumas iniciativas de integração de recursos de suporte à disciplinas de programação no ambiente Moodle foram elaboradas, como o VPL (*Virtual Programming Lab*) [França and Soares 2011].

O VPL é um módulo integrável ao Moodle que possibilita o desenvolvimento remoto de programas, permitindo realizar as seguintes atividades no navegador: editar o código-fonte dos programas, executá-los interativamente, realizar testes para revisá-los, definir restrições de edição e evitar colagem de texto externo [Rodríguez-Del-Pino 2023].

Contudo, [de Freitas 2016] constata que, em sua pesquisa de avaliação do uso do VPL em disciplinas de programação no curso de graduação de Tecnologias de Informação e Comunicação da UFSC, apesar da ferramenta cumprir seu papel de auxílio virtual à prática de programação, ela possui diversas limitações relacionadas à interface gráfica e à usabilidade, como a falta de mensagens de ajuda para erros, a necessidade de inserção manual de casos de teste e botões pouco intuitivos.

A plataforma Beecrowd destaca-se como uma ferramenta que auxilia alunos na resolução de problemas e desenvolvimento de algoritmos, além de automatizar a correção para educadores e incentivar maratonas de programação [da Cruz et al. 2022]. [Bez and Tonin 2014] ressaltam que sua construção foi voltada às necessidades pedagógicas, oferecendo um módulo acadêmico para acompanhar o desempenho dos alunos e recursos como listas temáticas e gamificação, com *badges* e *rankings*. [Ferreira 2022] enfatiza a base de problemas organizada por categorias e níveis, além de soluções modelo que ajudam na correção, enquanto [Lima 2022] destaca a acessibilidade do Beecrowd com questões em português, tornando-a uma escolha preferida por professores.

Alternativas como Neps Academy e CodeBench oferecem menos recursos, sendo limitadas quanto à gestão de turmas e acesso a problemas, conforme [Lima 2022]. Juízes online como BOCA e UVa apresentam limitações técnicas e operacionais que restringem sua eficiência em sala de aula [Bez and Tonin 2012].

Com sua popularidade crescente nacional e internacionalmente, o Beecrowd atende às demandas de ensino, promovendo aprendizado ativo e acessível. Dessa forma, sua adoção em ambientes acadêmicos é uma estratégia eficaz para aprimorar o ensino de programação e a resolução de problemas [Bez and Tonin 2012], [Ferreira 2022].

### **1.1. Objetivos**

Favorecer o uso da plataforma Beecrowd entre os docentes, para facilitar e aprimorar o ensino da prática de programação. Será aproveitada a integração LTI do Beecrowd com o ambiente Moodle já implementada, a qual facilita o acesso à plataforma e otimiza o aproveitamento de seu extenso banco de questões, oferecendo uma ampla variedade de exercícios de forma mais eficiente. Além disso, será disponibilizada uma ferramenta baseada em sistema especialista para auxiliar tanto professores quanto alunos no esclarecimento de dúvidas recorrentes sobre as questões do Beecrowd, facilitando a adaptação de ambos ao uso da plataforma.

## **2. Fundamentação Teórica**

A fundamentação teórica apresenta os conceitos essenciais para compreender a proposta, destacando tecnologias e ferramentas relacionadas ao tema. Examina suas funcionalidades, limitações e aplicações, além de explorar integrações e interoperabilidade entre sistemas. Por fim, aborda fundamentos técnicos necessários para o entendimento do trabalho.

### **2.1. Juízes online**

Os autores [Wasik et al. 2018] definem os juízes online como “sistemas projetados para a avaliação confiável do código-fonte do algoritmo enviado pelos usuários, que é compilado e testado em seguida em um ambiente homogêneo”. Da mesma forma, [Santos and Ribeiro 2011] explicam a principal função dos juízes online: “avaliar códigos fonte que foram enviados em uma determinada linguagem de programação”. Essa avaliação automática é feita a partir de casos de teste, ou seja, cada caso de teste possui um conjunto de entradas e saídas, e verifica-se as respostas aos casos de teste da solução do usuário, com as respostas cadastradas para aquela questão no *site* [Francisco et al. 2016].

Diversos juízes online podem ser encontrados na Internet, dentre eles estão o Beecrowd [da Cruz et al. 2022] e o Codebench [Ribeiro et al. 2018], plataformas online com problemas de programação competitiva, e que aceitam soluções para esses desafios, avaliando-as quanto à sua eficácia e eficiência. Também há o *BOCA Online Contest Administrator*, um sistema que visa apoiar as competições de programação na correção de soluções apresentadas pelos competidores [de Campos and Ferreira 2004].

Semelhantemente, [Francisco et al. 2016] ressaltam os benefícios da utilização de um juiz online no ensino de matérias iniciais de programação: “[...] Aprendizagem no ritmo do aluno, auto-aprendizagem e redução da carga de trabalho do professor, são alguns dos benefícios apontados que contribuem não só em ambientes tradicionais de ensino, mas em ambientes de Educação a Distância (EAD) e em MOOC’s. A liberdade de

definir listas de exercícios e a disponibilidade de instrumentos para acompanhar os alunos são questões importantes para o professor”.

## **2.2. Beecrowd**

O Beecrowd, anteriormente conhecido como URI Online Judge de janeiro de 2013 a outubro de 2021 [Piekarski et al. 2023], é um portal online criado para tornar a prática de programação mais dinâmica, interessante e acessível aos iniciantes na área [Bez and Tonin 2012]. Desenvolvido pela Universidade Regional Integrada (URI) - Campus de Erechim, o portal foi inicialmente concebido com a intenção de oferecer um juiz automático que atendesse às necessidades tanto de professores quanto de alunos, com foco em interatividade, flexibilidade e usabilidade [Bez and Tonin 2012]. O ambiente foi estruturado para ser didaticamente organizado e está disponível 24 horas por dia [Bez and Tonin 2014].

A construção do Beecrowd foi orientada pelas necessidades dos educadores e, principalmente, pelos alunos [Bez and Tonin 2014]. Integrando recursos de renomados portais internacionais de programação, a plataforma oferece funcionalidades exclusivas, como problemas para iniciantes, um sistema de recompensas com *badges*, e um módulo acadêmico completo para o acompanhamento do desempenho dos estudantes, incluindo o *ranking* por universidade [Bez and Tonin 2014].

Além disso, o Beecrowd disponibiliza problemas no estilo do ICPC (International Collegiate Programming Contest), com um juiz online que avalia as submissões dos usuários em tempo real [Berssanette and de Francisco 2018]. Os problemas são organizados por assunto e nível de dificuldade, e há flexibilidade quanto à escolha da linguagem de programação para resolver os exercícios [Bez and Tonin 2012]. Entre seus diferenciais, destacam-se materiais de estudo, tutoriais sobre algoritmos e programação, e um fórum colaborativo [Bez and Tonin 2012].

A plataforma oferece também recursos importantes para professores de Tecnologia da Informação, permitindo a criação de disciplinas com alunos, listas de exercícios organizadas por assunto e prazos, além do acompanhamento da evolução dos estudantes [Bez and Tonin 2014]. Por fim, o Beecrowd suporta diversas linguagens de programação, oferecendo ao usuário a liberdade de escolher a mais adequada para o desenvolvimento das suas soluções [Bez and Tonin 2014].

## **2.3. BOCA Online Contest Administrator**

O *BOCA Online Contest Administrator* é um sistema desenvolvido para gerenciar competições de programação, como a Maratona de Programação da Sociedade Brasileira de Computação [de Campos and Ferreira 2004]. Por ser um juiz online [Bez and Tonin 2012], avalia automaticamente as soluções enviadas pelos participantes, classificando-as como corretas ou incorretas, com feedback imediato para correção [Galasso and Moreira 2014]. Sua interface web permite a organização de competições distribuídas, com equipes localizadas remotamente, e possibilita competições com múltiplas sedes, garantindo a centralização da correção e a igualdade de condições entre os participantes [de Campos and Ferreira 2004].

Embora tenha sido utilizado em sala de aula para apoiar disciplinas de programação, o BOCA não é ideal para esse contexto, devido à necessidade de limpeza

constante do sistema e registro de novos problemas [Bez and Tonin 2012]. Isso se deve ao fato de o sistema não ter sido projetado para esse uso, o que acaba fazendo com que se perca a classificação dos problemas e dos usuários. Apesar disso, é uma excelente opção para competições de programação, incentivando os alunos a estudarem conceitos fundamentais da computação, como estruturas de dados e análise de algoritmos [de Campos and Ferreira 2004].

## **2.4. VPL**

O *Virtual Programming Lab* (VPL) é um módulo do Moodle para gerenciar tarefas de programação, permitindo a edição e execução de código diretamente no navegador. Ele oferece recursos como testes interativos, detecção de plágio e restrições de edição para um ambiente acadêmico controlado [VPL 2021].

O VPL funciona como uma ferramenta de avaliação automática para alunos, e como sistema de gerenciamento para instrutores, facilitando a criação de tarefas, verificação de plágio e condução de avaliações com flexibilidade e segurança [Rodríguez-Del-Pino et al. 2012]. Ele é independente da linguagem de programação e permite a personalização de cada tarefa com informações como prazos, tamanho de arquivos e casos de teste [?].

Para criar novas tarefas, o professor deve preencher um formulário de configuração e incluir manualmente os casos de teste. Isso exige a criação de uma nova atividade a cada semestre, garantindo que cada exercício seja adequado às necessidades da turma [?].

## **2.5. CodeRunner**

O CodeRunner é um plugin gratuito e de código aberto para o Moodle, projetado para avaliar respostas de programação em várias linguagens. Ele permite que os alunos enviem código, visualizem os resultados dos testes imediatamente e corrijam suas respostas, com avaliação adaptativa baseada nos testes definidos [CodeRunner 2023]. Ao contrário do VPL, oferece um sistema de penalidades ajustável, permitindo reenvios com penalidades graduais ou até 100% de penalização para tentativas erradas [Lobb and Harlow 2016].

O CodeRunner integra-se facilmente com outros tipos de perguntas do Moodle e é amplamente utilizado em cursos de introdução à programação e em cursos mais avançados de Ciência da Computação, como inteligência artificial e programação na web [Lobb and Harlow 2016]. Assim como no VPL, o criador da tarefa deve configurar os detalhes e casos de teste [CodeRunner 2023].

## **2.6. LTI**

*LTI Learning Tools Interoperability* é um padrão que permite a troca segura de dados entre o sistema de gestão de aprendizado (LMS) e ferramentas de aprendizagem externas, centralizando o acesso a conteúdo interno e externo, oferecendo uma experiência de aprendizado consistente, eliminando a necessidade de múltiplos logins para diferentes plataformas [Verdaguer 2021].

A especificação *Learning Tools Interoperability* foi inicialmente introduzida como Basic LTI em maio de 2010 pelo IMS Global Learning Consortium (agora conhecida como LTI 1.0). Desde então, essa especificação ganhou ampla aceitação como um

método simples, mas eficaz, para integrar conteúdos e produtos de terceiros aos Ambientes Virtuais de Aprendizagem (VLEs). Hoje em dia, ela faz parte integral dos principais VLEs, como o Moodle, Blackboard Learn 9, Desire2Learn e o Canvas da Instructure [Vickers and Booth 2014].

## **2.7. Moodle**

O Moodle é uma plataforma de aprendizagem robusta e segura, projetada para educadores, administradores e alunos, oferecendo um sistema integrado para ambientes de aprendizagem personalizados [Moodle 2023]. Como um Sistema de Gerenciamento de Aprendizagem (LMS) altamente extensível, permite acesso remoto por estudantes, tutores e administradores, sendo utilizado por instituições renomadas como o Open Polytechnic da Nova Zelândia [Moodle 2023].

Com foco em segurança, o Moodle oferece controles rigorosos contra acesso não autorizado e perda de dados, podendo ser implantado em nuvem privada ou servidor. Disponível em mais de 60 idiomas, é usado por organizações como London School of Economics e Microsoft, com mais de 213 milhões de usuários [Moodle 2023].

Como software de código aberto, o Moodle oferece ferramentas centradas no aluno, integração com diversas ferramentas colaborativas e escalabilidade para diferentes tamanhos de turmas. Seu design modular permite personalização e adaptação para necessidades específicas, e a comunidade ativa de desenvolvedores garante atualizações constantes e melhorias [Moodle 2023].

Adotando o conceito construcionista social de educação [Galasso and Moreira 2014], o Moodle se destaca como uma escolha popular entre instituições educacionais que buscam modernizar seus sistemas de aprendizagem.

### **2.7.1. LTI External Tools**

As Ferramentas Externas LTI do Moodle permitem integrar aplicativos ao curso, como conteúdos interativos ou avaliações, acessíveis diretamente pelos alunos no Moodle sem necessidade de login em outro sistema. Dependendo da ferramenta, as notas podem ser enviadas de volta ao Moodle [Moodle 2023].

A integração entre Moodle e essas ferramentas é feita pelo padrão Learning Tools Interoperability (LTI), permitindo uma troca segura de informações e criando uma experiência de aprendizado contínua [Moodle 2023].

Administradores podem gerenciar essas ferramentas, adicionando-as aos cursos por meio do botão ‘Adicionar ferramenta’ e configurando-as nas “Configurações da ferramenta” com informações fornecidas pelo fornecedor, podendo deixar alguns campos em branco se necessário [Moodle 2023].

## **2.8. Sistema Especialista**

De acordo com [Singla 2013], um sistema especialista é um conjunto de programas que manipula conhecimento para resolver problemas em um domínio especializado que exige a *expertise* humana. Já [Bohanec et al. 1990] definem sistemas especialistas como

sistemas inteligentes de informação que imitam, em certo grau, o comportamento de um especialista humano no domínio em questão.

Os principais componentes de um sistema especialista são a base de conhecimento e o motor de inferência. A base de conhecimento armazena informações sobre um domínio específico, geralmente em forma de regras, enquanto o motor de inferência utiliza esse conhecimento para resolver problemas e gerar explicações [Bohanec et al. 1990]. O motor é responsável por gerar conclusões através de raciocínio [Singla 2013].

Embora esses sistemas sejam eficazes, apresentam limitações, como a falta de senso comum, dificuldades em lidar com casos excepcionais e desafios em se adaptar a ambientes em constante mudança [Singla 2013].

### **2.8.1. Prolog**

O Prolog, desenvolvido por Alain Colmerauer e baseado na lógica de predicados de Kowalski, é amplamente utilizado em sistemas especialistas, especialmente em áreas como farmacologia e diagnóstico de falhas [Clark et al. 1980]. Popular para prototipação rápida e análise de linguagem natural, é eficaz em áreas como direito e medicina [Russell and Norvig 2013]. Sua semântica segura e gerenciamento automático de memória o tornam ideal para serviços Web [WIELEMAKER et al. 2008].

### **2.8.2. SWI-Prolog**

O SWI-Prolog, baseado na WAM, é um ambiente de desenvolvimento robusto e versátil, usado principalmente para protótipos de pesquisa e aplicações que requerem integração e lógica sofisticada [WIELEMAKER 2012]. Com bibliotecas que suportam integração com gráficos, bancos de dados e servidores HTTP, é uma excelente ferramenta para o desenvolvimento de sistemas interativos e Web [Wielemaker 2014].

## **3. Trabalhos Relacionados**

Este capítulo apresenta uma revisão de artigos relevantes sobre o uso de plataformas de programação, integração de juízes online com o Moodle e a criação de sistemas especialistas para a web. A seção é estruturada em três partes: um estudo sobre o uso da plataforma Beecrowd em um curso de algoritmos, a análise de ferramentas de integração de juízes online com o Moodle, e a exploração de sistemas especialistas aplicados à educação.

### **3.1. Uso da Plataforma Beecrowd no Ensino de Algoritmos**

O estudo de [da Cruz et al. 2022] investigou o impacto do uso da plataforma Beecrowd em seis semestres da disciplina de algoritmos na Universidade Federal do Maranhão, envolvendo alunos que, em sua maioria, não possuíam conhecimento prévio em programação. A disciplina foi aplicada de forma intercalada: em três semestres com o uso do Beecrowd e em outros três sem a utilização da plataforma. Os resultados demonstraram que, nos semestres em que o Beecrowd foi utilizado, o desempenho dos estudantes foi significativamente superior, sugerindo que o uso de plataformas de programação online pode melhorar os resultados no ensino de algoritmos.

### 3.2. Integração de Juízes Online com o Moodle

Os autores [Galasso and Moreira 2014], [Lobb and Harlow 2016], e [Rodríguez-Del-Pino et al. 2012] detalharam outras alternativas de juízes online já integrados ao Moodle, como o BOCA, CodeRunner e VPL, respectivamente. mas que possuem desvantagens em relação ao Beecrowd.

Vale destacar que, enquanto o Beecrowd, o CodeRunner e o VPL foram desenvolvidos com foco no apoio ao ensino, o BOCA foi projetado especificamente para avaliar submissões de código em maratonas de programação, embora também possa ser utilizado como ferramenta de suporte no ensino de programação.

Uma característica distintiva do Beecrowd é a disponibilização de questões pré-existentes, eliminando a necessidade de cadastrar manualmente perguntas e testes, algo obrigatório em BOCA, CodeRunner e VPL. Como destacado por [Lobb and Harlow 2016], a criação de questões no CodeRunner é trabalhosa, especialmente em atividades complexas ou turmas grandes, sendo mais apropriado para tarefas simples e bem definidas.

A prevenção contra plágio é um diferencial crucial no Beecrowd e no VPL. Segundo [da Cruz et al. 2022], o Beecrowd emprega análise avançada de código-fonte para identificar padrões suspeitos e similaridades, assegurando a originalidade dos trabalhos. Já o VPL, conforme [Rodríguez-Del-Pino et al. 2012], realiza comparações automáticas de códigos e detecta características indicativas de plágio, preservando a integridade acadêmica.

Outro ponto relevante é que os sistemas BOCA, VPL e CodeRunner adotam servidores independentes para o Moodle e o juiz online, o que assegura maior desempenho, segurança e escalabilidade. Essa arquitetura distribuída facilita a gestão da carga de trabalho, prevenindo sobrecargas e permitindo um dimensionamento mais eficiente de acordo com as demandas específicas de cada sistema. Em relação à criação e execução de código, [Rodríguez-Del-Pino et al. 2012] destacam que o VPL integra um editor Java para a execução e avaliação de códigos, enquanto [Lobb and Harlow 2016] e [Galasso and Moreira 2014] apontam que tanto o CodeRunner quanto o BOCA utilizam caixas de texto para a edição e submissão de código.

Outro trabalho relacionado foi o de [Chaves et al. 2013], onde foi apresentado o Módulo de Integração com Juízes Online (MOJO), ferramenta que integra plataformas com juízes online, como o Beecrowd, ao Moodle, delegando ao Moodle o controle das atividades e centralizando a comunicação com os juízes. Devido à ausência de APIs na época, implementações específicas foram feitas em PHP e JavaScript, com PostgreSQL para armazenamento de dados dos alunos e feedback preciso.

Neste trabalho, contudo, será disponibilizado um manual de uso da integração LTI fornecida pelo Beecrowd, simplificando a interação com a plataforma. Essa abordagem permite que o usuário desenvolva seu código diretamente na plataforma, enquanto ela realiza a correção das resoluções automaticamente.

### 3.3. Sistemas Especialistas Desenvolvidos para o Ambiente Acadêmico

Por fim, os projetos de [NASR et al. 2018] e [Dunstan 2013] estão relacionados ao desenvolvimento de sistemas especialistas para aconselhamento e planejamento acadêmico,



porém, com abordagens distintas.

[NASR et al. 2018] desenvolveram um sistema baseado em regras para replicar decisões de conselheiros acadêmicos, usando árvores de decisão no ES Builder, com lógica "se-então" e interface web simples. Já [Dunstan 2013] utiliza XML e Prolog em um planejador acadêmico híbrido, estruturando regras em XML e convertendo-as para Prolog. O sistema distribui funções entre cliente e servidor, permitindo consultas personalizadas e uso em dispositivos móveis.

Este artigo adota uma abordagem similar à de [Dunstan 2013], utilizando a linguagem de programação Prolog para construir o sistema especialista. O Prolog será responsável por gerenciar requisições HTTP, armazenar dados de sessões, possibilitar uma comunicação contínua com o front-end através dessas requisições, permitindo perguntas dinâmicas e respostas personalizadas, além de configurar permissões CORS para integração com a interface web.

#### **4. Manuais de Configuração e Uso da LTI Beecrowd**

Com a disponibilização de uma ferramenta LTI pelo Beecrowd, visando facilitar sua integração com o Moodle, foi elaborado um manual detalhado para orientar o administrador do Moodle na configuração dessa LTI.

Para configurar uma LTI externa, o administrador deve acessar "Administração do site", selecionar "Plugins" e, em "Módulos de atividade", escolher "Ferramenta externa" e clicar em "Gerenciar ferramentas". Em seguida, deve clicar em "Configurar uma ferramenta manualmente" e preencher os campos conforme as instruções especificadas no Manual de Configuração da LTI Beecrowd no Moodle. Após a configuração, será possível visualizar os detalhes da ferramenta, os quais deverão ser enviados ao Beecrowd para estabelecer a comunicação entre as plataformas.

Além disso, foi desenvolvido um manual de uso da LTI Beecrowd, direcionado aos professores que desejam utilizar o Beecrowd como suporte educacional em sala de aula. Esse manual instrui os docentes sobre como criar uma atividade do Beecrowd no Moodle, orientar o acesso dos estudantes e transferir as notas obtidas no Beecrowd para o Moodle.

Após seguir o guia do Manual de Uso da LTI Beecrowd, duas atividades ficam disponíveis na página do curso, apresentadas como "botões" visuais. A primeira atividade, que é ocultada para os estudantes, permite ao professor acessar o Beecrowd Academic para configurar disciplinas, criar listas de exercícios para os alunos e enviar as notas para o Moodle. A segunda atividade, acessível aos estudantes, redireciona-os diretamente para a página no Beecrowd da lista de exercícios criada pelo professor, sem necessidade de cadastro ou login. Assim, os estudantes podem resolver as atividades diretamente a partir dessa página. Esse botão também permite ao professor acessar a lista de exercícios criada, podendo verificar o progresso dos alunos, as tentativas realizadas e, quando desejado, enviar as notas para o Moodle.

#### **5. Expert Bee: Sistema Especialista Web**

O objetivo do sistema especialista desenvolvido, o Expert Bee, é oferecer suporte na resolução de dúvidas recorrentes dos estudantes em relação a questões do Beecrowd. A

proposta é uma aplicação acessível aos alunos via navegador, com um sistema de *chat* onde o aluno pode informar a questão sobre a qual necessita orientação. O sistema, então, conduz o aluno por meio de perguntas binárias (sim ou não) e, conforme as respostas, fornece dicas relevantes relacionadas ao exercício em questão.

Para viabilizar essa funcionalidade, foi estruturado com um conjunto de dados contendo perguntas direcionadas aos alunos e respostas predefinidas que orientam o aluno na resolução das atividades específicas da disciplina. Em disciplinas como Programação Orientada a Objetos I, onde as listas de exercícios frequentemente incluem questões já utilizadas em semestres anteriores, observa-se uma repetição de problemas e, consequentemente, de dúvidas recorrentes dos alunos.

A primeira etapa deste trabalho foi dedicada à coleta de dados sobre as questões, com o objetivo de identificar e catalogar as dúvidas mais recorrentes. Na segunda etapa, foram definidos os requisitos funcionais, os requisitos não funcionais e as regras de negócio da aplicação. Em seguida, na terceira etapa, realizou-se uma pesquisa sobre as tecnologias que seriam utilizadas no desenvolvimento. Na quarta etapa, foram elaborados os diagramas que definiram a estrutura e o funcionamento da aplicação. Por fim, a partir da quinta etapa, deu-se início ao desenvolvimento propriamente dito.

## 5.1. Coleta de Dados

Nesta fase, foram realizadas duas coletas de dados para integrar a base de conhecimento do sistema especialista. Na primeira, o professor de Programação Orientada a Objetos I da UFSC forneceu dicas de resolução para duas listas de exercícios do Beecrowd que ainda não haviam sido entregues aos alunos, com o objetivo de avaliar a eficácia da aplicação posteriormente. Na segunda coleta, foram registradas as dúvidas dos alunos enquanto resolviam exercícios sobre o uso de listas em Python, em sala de aula.

## 5.2. Requisitos da Aplicação

Esta seção descreve os requisitos essenciais para o desenvolvimento da aplicação, divididos em Regras de Negócio, Requisitos Funcionais e Não Funcionais.

### 5.2.1. Regras de Negócio

As Regras de Negócio definem os processos centrais e orientam o comportamento do sistema. São elas:

- **RN1 – Identificação da Questão:** O usuário deve informar o código da questão do Beecrowd que está tentando resolver para que o sistema possa fornecer dicas específicas.
- **RN2 – Perguntas Binárias:** O sistema deve formular perguntas binárias (sim/não) para guiar o usuário, baseando-se em possíveis erros ou dificuldades comuns para cada questão.
- **RN3 – Personalização de Orientações:** Com base nas respostas do usuário, o sistema deve ajustar as perguntas para fornecer a dica mais apropriada.
- **RN4 – Encerramento do Fluxo:** Ao atingir um número máximo de perguntas, o sistema deve fornecer uma dica final ao usuário.

### 5.2.2. Requisitos Funcionais

Os Requisitos Funcionais descrevem as funcionalidades que o sistema deve possuir para cumprir os objetivos propostos. São eles:

- **RF1 – Comunicação por Chat:** O usuário deve interagir com o sistema por uma interface de *chat* intuitiva, que apresente respostas sequenciais.
- **RF2 – Busca e Seleção de Questão:** O sistema deve permitir que o usuário informe o código da questão do Beecrowd que deseja resolver.
- **RF3 – Fluxo de Perguntas Binárias:** O sistema apresenta perguntas binárias adaptadas conforme as respostas do usuário.
- **RF4 – Sistema de Dicas:** Após a análise das respostas, o sistema deve oferecer dicas relevantes para o problema informado.
- **RF5 – Instruções sobre Erros Comuns do Beecrowd:** O sistema deve ter uma seção que informa ao usuário descrições dos principais erros retornados pelo Beecrowd, contidos na seção ??.

### 5.2.3. Requisitos Não Funcionais

Os Requisitos Não Funcionais especificam características de qualidade e restrições que o sistema deve atender. Eles incluem aspectos como desempenho, usabilidade, segurança e arquitetura. São eles:

- **RNF1 – Desempenho:**
  - O sistema deve responder em tempo real a cada interação do usuário, mantendo a fluidez do *chat*.
- **RNF2 – Usabilidade:**
  - O sistema deve ser intuitivo e fácil de usar, com uma interface de *chat* amigável.
  - Deve ser compatível com os principais navegadores modernos.
- **RNF3 – Escalabilidade:**
  - O sistema deve suportar o uso simultâneo de múltiplos usuários.
- **RNF4 – Segurança:**
  - As sessões de *chat* devem ser isoladas, para evitar vazamento de informações entre os usuários.
- **RNF5 – Manutenibilidade:**
  - O sistema deve ter um código modular, facilitando a inclusão de novas perguntas e dicas para futuras questões do Beecrowd.
- **RNF6 – Arquitetura Backend e Frontend:**
  - O sistema deve ter um *backend* desenvolvido em Prolog, utilizando SWI-Prolog para lidar com requisições HTTP.
  - O *backend* deve definir configurações de CORS apropriadas para permitir a comunicação entre cliente e servidor.
  - O *backend* deve seguir o padrão de API REST e devolver respostas no formato JSON.
  - O *backend* deve conter o sistema especialista, que processa as perguntas e respostas para oferecer suporte aos estudantes.
  - O *frontend* deve ser desenvolvido em React e TypeScript, proporcionando uma interface interativa e amigável para os alunos.

### 5.3. Tecnologias Usadas

#### 5.3.1. Prolog e SWI-Prolog

Para o desenvolvimento do *backend* da aplicação, utilizou-se a linguagem Prolog, com o auxílio da implementação SWI-Prolog. O SWI-Prolog é uma versão amplamente utilizada do Prolog, que inclui uma série de bibliotecas essenciais para a manipulação de requisições HTTP e dados JSON, como `http/thread_httpd`, `http/http_dispatch`, `http/json`, `http/http_session`, entre outras. Essas bibliotecas permitem a configuração de um servidor HTTP para o processamento de requisições web e a manipulação de dados JSON. Além disso, elas possibilitam o gerenciamento de sessões, o que viabiliza a integração com aplicações externas e permite o uso simultâneo da aplicação por diferentes usuários [Wielemaker 2024].

#### 5.3.2. API REST

O servidor do backend desse trabalho definiu os *endpoints* HTTP (`/server` e `/questions`) para o recebimento de requisições POST e GET, permitindo a interação com o *backend* por meio de uma arquitetura baseada em REST.

O termo REST (Representational State Transfer) é um estilo de comunicação baseado em padrões e protocolos da web, como HTTP, que permite a troca de dados entre sistemas de forma escalável e eficiente. Nesse modelo, as requisições são realizadas utilizando os métodos HTTP padrão (como GET, POST, PUT e DELETE), facilitando a comunicação entre sistemas de maneira eficiente e escalável [Masse 2011].

#### 5.3.3. JSON

JSON (JavaScript Object Notation) é um formato leve e baseado em texto para intercâmbio de dados, independente de linguagem de programação. Derivado do padrão ECMAScript, o JSON define um conjunto simples de regras de formatação para a representação portátil de dados estruturados. Ele busca remover inconsistências com outras especificações e oferece orientações para melhorar a interoperabilidade entre sistemas [Bray 2017].

A biblioteca `http/http_json` do SWI-Prolog é utilizada para formatar as respostas no formato JSON, possibilitando que o *backend* envie dados estruturados para o *frontend* em um formato amplamente compatível com diversos frameworks web, incluindo o React. Essa abordagem facilita a troca de informações entre o servidor e a interface do usuário, garantindo a interoperabilidade e a eficiência na comunicação entre as camadas da aplicação.

#### 5.3.4. HTTP e CORS

CORS (Cross-Origin Resource Sharing) é um mecanismo que permite que requisições do lado do cliente acessem recursos de origens diferentes. Ele define algoritmos que possibilitam que uma API faça requisições a recursos de outra origem, controlando

o acesso por meio de cabeçalhos de resposta, como o *Access-Control-Allow-Origin* [van Kesteren 2010].

HTTP (Hypertext Transfer Protocol) é um protocolo de aplicação para sistemas distribuídos e colaborativos de informações hipermídia. É um protocolo genérico e sem estado, utilizado em diversas tarefas além do hipertexto, como servidores de nomes e sistemas de gerenciamento de objetos distribuídos. O HTTP permite a negociação e tipificação da representação dos dados, possibilitando a construção de sistemas independentes dos dados transferidos [Fielding 1999].

As bibliotecas `http/thread_httpd`, `http/http_dispatch` e `http/http_cors`, do SWI-Prolog, configuram um servidor HTTP básico, sendo responsáveis pelo tratamento de requisições e respostas. Elas também gerenciam as permissões de CORS, permitindo que o *frontend* (React) acesse o *backend* Prolog, mesmo quando este está hospedado em uma origem diferente.

### 5.3.5. Sessões HTTP

Uma sessão HTTP é mantida por meio do uso de cookies, que são pequenos arquivos de dados armazenados nos agentes de usuário (como navegadores) pelos servidores HTTP. Os cabeçalhos `Cookie` e `Set-Cookie` permitem que os servidores mantenham o estado de uma sessão, mesmo em um protocolo essencialmente sem estado como o HTTP. Isso significa que, através das sessões, os servidores podem armazenar informações sobre o usuário, como preferências ou dados de autenticação, e usá-las em requisições subsequentes, garantindo uma experiência contínua. Embora os cookies tenham questões históricas relacionadas à segurança e privacidade, os cabeçalhos `Cookie` e `Set-Cookie` são amplamente utilizados para gerenciar sessões de usuário, permitindo, por exemplo, o login persistente em *sites* ou a manutenção de um carrinho de compras em uma loja online [Barth 2011].

A biblioteca `http/http_session` do SWI-Prolog possibilita o gerenciamento de sessões de usuário, permitindo o armazenamento de variáveis de sessão, como `question-Number` e `answers`. Esse recurso viabiliza a persistência de informações entre requisições subsequentes de um mesmo usuário, assegurando a continuidade do estado da aplicação ao longo da interação.

### 5.3.6. React e Typescript (Frontend)

O *frontend* foi feito em React<sup>1</sup> e Typescript<sup>2</sup>, interagindo com o *backend* por meio de requisições HTTP, acessando *endpoints* para enviar as respostas do usuário e buscar o resultado final da questão.

React é uma biblioteca para a construção de interfaces de usuário dinâmicas e reutilizáveis. Ela permite criar interfaces a partir de unidades individuais chamadas componentes, que podem ser reutilizados e combinados para formar interfaces complexas e

---

<sup>1</sup><https://react.dev/>

<sup>2</sup><https://www.typescriptlang.org/>

interativas. Cada componente em React gerencia seu próprio estado e pode ser renderizado de forma eficiente conforme as mudanças nos dados, facilitando o desenvolvimento de aplicações dinâmicas e de fácil manutenção. React pode ser utilizado com outras linguagens que compilam para JavaScript<sup>3</sup>, como TypeScript.

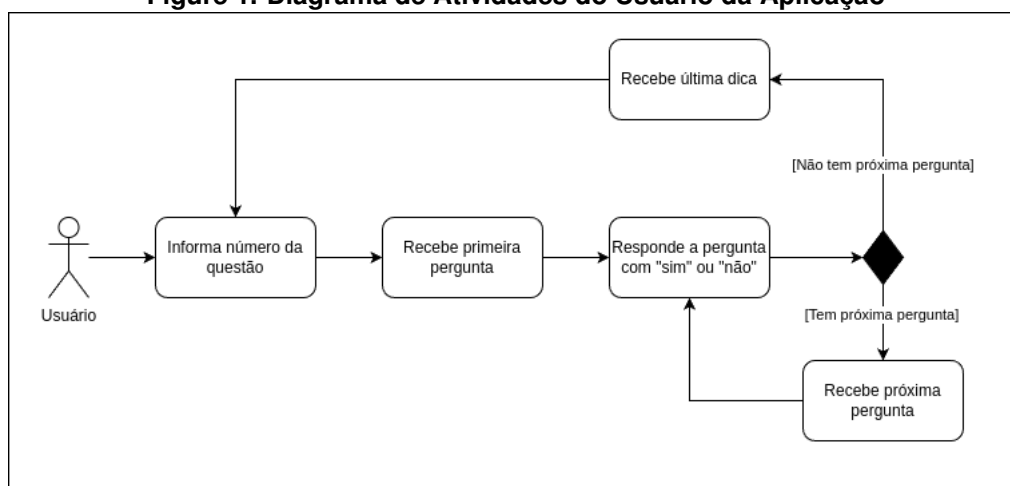
TypeScript é uma linguagem de programação que é um superconjunto do JavaScript, adicionando tipagem estática. Isso permite que os desenvolvedores definam tipos para variáveis e funções, ajudando a identificar erros antes da execução do código. TypeScript melhora a manutenção e segurança do código, especialmente em projetos grandes, e é amplamente utilizado com frameworks como React para criar aplicações mais escaláveis e robustas.

#### 5.4. Arquitetura do Software

A aplicação propõe uma interação dinâmica em que o usuário, ao informar o número da questão sobre a qual deseja obter orientações, inicia um diálogo com o sistema de *chat*. O *chat* responde com perguntas específicas para compreender melhor as necessidades do usuário, permitindo direcionar as orientações de forma mais precisa. Conforme o usuário responde com "sim" ou "não" a cada pergunta, o sistema adapta as próximas questões e sugestões, guiando-o progressivamente até fornecer uma orientação final ajustada ao contexto das respostas acumuladas.

A Figura 1 apresenta o diagrama de atividades que detalha o fluxo de interação entre o usuário e o *chat* da aplicação. Após informar o número da questão sobre a qual deseja obter orientações, o usuário recebe uma pergunta inicial, respondendo com "sim" ou "não". Se houver mais perguntas, o *chat* continua a apresentá-las até que todas sejam respondidas, conduzindo o usuário a uma orientação final baseada nas respostas acumuladas. Ao final, o usuário pode iniciar o processo novamente com outro número de questão ou com o mesmo, caso deseje explorar outras sugestões e perguntas relacionadas.

Figure 1. Diagrama de Atividades do Usuário da Aplicação

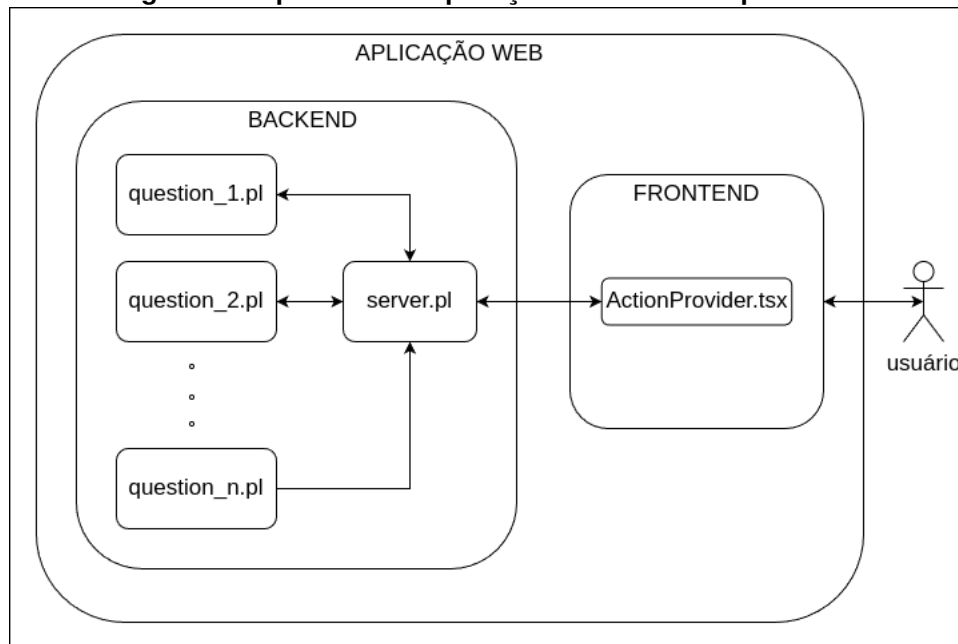


A Figura 2 representa a estrutura da aplicação, mostrando a interação entre o *frontend* e o *backend* através de requisições. No *frontend*, o arquivo *ActionProvider.tsx* gerencia as mensagens enviadas pelo usuário no *chat*, processando tanto o número da questão

<sup>3</sup><https://www.javascript.com/>

informada, quanto as respostas de "sim" ou "não". Por meio dessas requisições, o *frontend* envia esses dados ao *backend*, que, em resposta, fornece perguntas e dicas relacionadas à questão solicitada. Caso o usuário peça uma nova dica e o *backend* não tenha mais sugestões para aquela questão, ele envia a orientação final da questão, que é então apresentada ao usuário.

**Figure 2. Arquitetura da Aplicação do Sistema Especialista**



O *frontend* da aplicação foi desenvolvido utilizando as tecnologias React e TypeScript, conforme especificado no requisito não funcional **RNF6**, descrito na seção 5.2. Essas tecnologias foram escolhidas por sua capacidade de criar interfaces dinâmicas, escaláveis e de fácil manutenção, proporcionando uma experiência de usuário fluida e interativa (**RNF2**).

Conforme o requisito funcional **RF1** e a regra de negócio **RN1**, foi implementado um sistema de *chat* como interface principal de interação com o usuário. Por meio deste *chat*, o usuário pode informar o código da questão do Beecrowd que está tentando resolver (**RF2**). Além disso, conforme a regra de negócio **RN2**, o sistema especialista se comunica com o usuário por meio do *chat*, exibindo mensagens sequenciais baseadas nas respostas do sistema (**RF3**) em tempo real (**RNF1**), e dicas finais (**RF4**). O *chat* também permite que o usuário forneça respostas binárias (sim/não), conforme o fluxo de perguntas definido para personalizar a orientação fornecida, conforme as regras de negócios **RN2** e **RN3**.

O desenvolvimento do *frontend* foi baseado em bibliotecas e repositórios existentes que facilitaram a criação e configuração do sistema de *chatbot*. Especificamente, foi utilizado o repositório *react-chatbot-kit* de FredrikOseberg<sup>4</sup> como estrutura principal para a criação do *chatbot*, combinado com o repositório *Chatbot-using-React-Chatbot-Kit* de Subid Das<sup>5</sup>, que forneceu recursos adicionais para personalização da interface e

<sup>4</sup><https://github.com/FredrikOseberg/react-chatbot-kit>

<sup>5</sup><https://github.com/devsubid/Chatbot-using-React-Chatbot-Kit>

integração do *chatbot* com o fluxo de mensagens interativo. Esses repositórios foram adaptados para atender às necessidades específicas da aplicação, garantindo que o *frontend* fosse capaz de processar interações dinâmicas e personalizadas, conforme os requisitos estabelecidos.

Para atender ao Requisito Funcional **RF5**, descrito na seção 5.2, foi criado um componente *Details* para a rota *details*, a fim de mostrar as descrições dos possíveis resultados do juiz online do Beecrowd.

O *backend* foi desenvolvido em Prolog, utilizando o SWI-Prolog para possibilitar o tratamento de requisições HTTP, conforme especificado no Requisito Não Funcional **RNF6**, na seção 5.2.

Além disso, conforme ilustrado na Figura 2, o *backend* é composto pelo arquivo principal *server.pl* e por arquivos adicionais, cada um representando um sistema especialista específico para cada questão, seguindo, também, o Requisito Não Funcional **RNF6**. Esses arquivos são nomeados no formato *questao- $\{numeroDaQuestao\}.pl$*  e contêm dicas específicas para as respectivas questões. Por exemplo, o arquivo *questao\_1234.pl* armazena as orientações para a questão 1234 do Beecrowd.

Dessa forma, arquivos do formato *questao- $\{numeroDaQuestao\}.pl$*  devem a seguinte estrutura: conter o predicado *questao/3* para fornecer orientações sequenciais, e o predicado *diagnostico/3*, para mensagem final.

Além disso, o arquivo *server.pl* configura e inicializa um servidor HTTP, definindo rotas para processar requisições e habilitando o CORS, permitindo conexões da porta do *frontend*. Ele recebe requisições *POST* para iniciar ou continuar uma questão, associando cada sessão ao usuário correspondente, e também aceita requisições *GET* para fornecer o diagnóstico final com base nas respostas acumuladas durante a interação.

Os arquivos de questões são carregados dinamicamente no *server.pl* com base no número da questão recebido nas requisições *POST*. Se o usuário alterna para uma nova questão, por exemplo, a questão Y, o *backend* carrega o arquivo *questao\_Y.pl* e passa a conduzir a interação de acordo com suas informações.

O arquivo *server.pl* também armazena na sessão do usuário as informações fornecidas, como número da questão e respostas, para garantir a continuidade da interação.

## 6. Manuais de Como Adicionar Novas Questões no Expert Bee

Foi criado um manual que auxilia o professor, passo a passo, em como adicionar dicas para uma nova questão no Bee Expert. Este manual proporciona uma abordagem prática para garantir que as questões sejam adicionadas de forma correta e eficiente, facilitando a inclusão de novas questões e o fornecimento de orientações adequadas aos usuários do sistema.

Para adicionar novas questões ao sistema, deve-se criar um arquivo Prolog no diretório */backend*, nomeado conforme o formato *questao.numeroDaQuestao.pl*. Cada arquivo deve seguir uma estrutura padrão contendo dois predicados principais: *questao/3* e *diagnostico/3*.

O predicado *questao/3* associa uma sequência de respostas do usuário a uma



nova pergunta, enquanto o predicado `diagnostico/3` fornece um diagnóstico ou *feedback* final com base nas respostas completas. O sistema utiliza essas definições para iterar sobre as perguntas e fornecer respostas progressivas ou um diagnóstico final, de acordo com as respostas fornecidas até o momento. A consistência das respostas e do número da questão é essencial para o funcionamento adequado do sistema.

Ao adicionar uma nova questão, é fundamental que o arquivo siga corretamente essa estrutura, permitindo que o sistema integre automaticamente a questão ao fluxo de interações. As respostas parciais devem ser tratadas de forma coerente, e qualquer sequência de respostas não mapeada deve ser considerada com um curinga no predicado `diagnostico/3`.

## 7. Validação

Nesta seção, será feita uma análise crítica da integração entre o Moodle e o Beecrowd via LTI (*Learning Tools Interoperability*) e da aplicação com sistema especialista desenvolvida. Também será apresentada uma avaliação prática do uso da aplicação pelos alunos da disciplina *Programação Orientada a Objetos I* da UFSC, no segundo semestre de 2024, para verificar a efetividade do sistema na resolução das dúvidas dos estudantes.

### 7.1. Integração LTI entre o Moodle e o Beecrowd

#### 7.1.1. Vantagens

- **Vantagens do Beecrowd:** O Beecrowd oferece uma plataforma robusta para a resolução de problemas de programação, com uma grande variedade de questões que permitem aos alunos praticar e desenvolver habilidades de programação em diversos níveis de dificuldade. A plataforma também facilita o acompanhamento do progresso dos alunos, oferecendo métricas e *feedback* instantâneo sobre o desempenho nas questões.
- **Facilidade de acesso através da integração LTI:** A integração LTI simplifica o acesso ao Beecrowd, pois tanto o professor quanto o aluno podem acessar a plataforma diretamente do Moodle com um único clique.

#### 7.1.2. Desvantagens

- **Desenvolvimento externo e curva de aprendizado:** O desenvolvimento dos códigos e a criação das listas de exercícios ainda ocorre fora do Moodle, no Beecrowd, o que significa que o professor precisa aprender a usar o Beecrowd Academic, embora o processo seja relativamente simples. Isso pode exigir um tempo adicional de adaptação, especialmente para aqueles que não estão familiarizados com a plataforma.

### 7.2. Sistema Especialista para Resolução de Dúvidas no Beecrowd

#### 7.2.1. Vantagens

- **Atendimento automatizado:** O sistema especialista pode fornecer respostas rápidas e precisas para questões frequentes ou comuns, permitindo que os alunos resolvam suas dúvidas sem a necessidade de intervenção humana constante.

- **Apoio no aprendizado:** O sistema pode ser configurado para fornecer explicações detalhadas sobre questões específicas do Beecrowd, ajudando os alunos a entender melhor os problemas e as soluções.
- **Escalabilidade:** Como o sistema é automatizado, ele pode lidar com um grande volume de dúvidas de maneira eficiente.
- **Redução da sobrecarga para os professores:** Ao automatizar o atendimento às dúvidas mais comuns, o sistema especialista permite que os professores se concentrem em questões mais complexas e em atividades pedagógicas, sem se sobrecarregar com um grande volume de perguntas repetitivas.
- **Auxílio para professores não familiarizados com o Beecrowd:** Para professores que ainda não estão familiarizados com as questões do Beecrowd, o sistema especialista oferece uma ferramenta valiosa para auxiliá-los a responder dúvidas dos alunos de maneira eficiente, sem a necessidade de um profundo conhecimento prévio sobre a plataforma ou as questões.

### 7.2.2. Desvantagens

- **Limitação nas respostas para questões complexas:** O sistema especialista é eficiente para lidar com questões simples ou frequentes, mas pode encontrar dificuldades ao lidar com questões mais complexas ou específicas. Para adicionar suporte a novas questões, o professor precisa criar uma matriz de decisões detalhada, o que pode tornar o processo demorado e propenso a erros, especialmente quando se trata de questões que exigem uma análise mais aprofundada. Sem a devida cautela, o sistema pode acabar gerando respostas imprecisas ou inadequadas, comprometendo a qualidade das interações e a efetividade do suporte oferecido.
- **Dependência da qualidade da base de conhecimento:** O desempenho do sistema especialista depende diretamente da qualidade das regras e conhecimentos que foram programados nele. Se a base de dados for limitada ou desatualizada, o sistema pode fornecer respostas inadequadas.
- **Falta de interação humana:** Em casos onde o aluno necessita de uma explicação mais personalizada ou de orientação emocional, o sistema especialista pode não ser suficiente, criando uma experiência impessoal que pode não atender completamente às necessidades do aluno.

### 7.3. Avaliação Prática

A avaliação prática foi realizada durante o segundo semestre de 2024, com os alunos da disciplina *Programação Orientada a Objetos I* da UFSC, que utilizaram o sistema especialista Expert Bee enquanto resolviam a lista de exercícios do Beecrowd. À medida que surgiam dúvidas, os alunos foram incentivados a usar a ferramenta para buscar soluções. O uso do Expert Bee foi monitorado para avaliar sua eficácia na resolução das dúvidas. Com base na análise dos resultados, as dúvidas não resolvidas pelo sistema foram incorporadas ao banco de conhecimento para futuras turmas.

É importante destacar que, até o momento, o Expert Bee tinha apenas dicas de resoluções para essas questões da lista de exercícios fornecida, e não dicas para dúvidas reais coletadas de alunos. Essas dicas de resoluções tinham sido disponibilizadas pelo

professor durante a primeira coleta de dados, descrita na seção ??, para que fosse possível testar a eficácia da ferramenta na hora em que as dúvidas dos alunos surgiriam.

Assim, um dos objetivos da avaliação prática também era identificar dúvidas não tratadas pelo Expert Bee, para que elas pudessem ser adicionadas à ferramenta.

### 7.3.1. Resultados

Na Avaliação Prática realizada em 18 de novembro de 2024, 12 alunos estiveram presente na sala de aula e, sempre que um aluno apresentava uma dúvida, ele utilizava primeiro o Expert Bee para tentar resolvê-la, e, caso não conseguisse, recorria ao professor.

No total, de 11 dúvidas relatadas, o Bee Expert conseguiu resolver 9, resultando em uma taxa de sucesso de aproximadamente 81%. Além disso, 4 pontos de melhoria foram coletados.

## 8. Conclusão

Em conclusão, este trabalho abordou a relevância da integração de ferramentas tecnológicas no ensino de programação, com foco especial no uso de plataformas como Beecrowd e Moodle. A análise dos juízes online e outras plataformas educacionais, como o VPL, evidenciou desafios na usabilidade e na integração, mas também destacou o Beecrowd como uma solução eficaz para o ensino de algoritmos e programação, com recursos que incentivam o aprendizado ativo, como gamificação, *feedback* em tempo real e uma gama diversificada de questões.

Além disso, este estudo aprofundou o impacto dos juízes online no ensino de programação, enfatizando a relevância da integração entre plataformas como o Beecrowd e o Moodle. A integração do Beecrowd ao Moodle, por meio da tecnologia LTI, facilita o acesso de professores e alunos a uma vasta gama de questões, otimizando o uso da plataforma no ambiente acadêmico. Além disso, oferece uma interface amigável para a criação de listas de exercícios pelos professores e a resolução de problemas de programação pelos alunos. Com o objetivo de apoiar essa integração, foi desenvolvido um manual detalhado sobre como integrar o Moodle ao Beecrowd via LTI, bem como um guia para orientar os professores no uso da plataforma.

Para estimular o uso do Beecrowd no ambiente acadêmico, também foi criada uma ferramenta com um sistema especialista que auxilia os alunos ao esclarecer dúvidas recorrentes nas questões do Beecrowd. Esse sistema, testado em sala de aula, não só oferece suporte aos estudantes, mas também facilita a adaptação dos professores, promovendo um ambiente de aprendizado mais fluido e eficiente para todos os envolvidos.

### 8.1. Trabalhos Futuros

A seção a seguir apresenta algumas sugestões de melhorias e direções futuras para a aplicação de sistema especialista desenvolvida, a Expert Bee.

- **Adição de dicas de novas questões pela interface web:** Uma melhoria importante seria adaptar o sistema para permitir que os professores possam adicionar dicas para novas questões diretamente pela interface da aplicação web, sem a necessidade de modificar o código-fonte. Essa melhoria tornaria o sistema mais

acessível e eficiente, permitindo que os educadores personalizem o conteúdo de maneira prática e intuitiva, sem a dependência de habilidades técnicas.

- **Sistema autoalimentado (aprendizado com as interações dos alunos):** Uma possível evolução seria adaptar o sistema para se autoalimentar com base nas interações dos alunos. O sistema poderia aprender com as dúvidas mais frequentes e se ajustar automaticamente, oferecendo respostas e dicas mais precisas com o tempo. Essa melhoria contribuiria para uma experiência de aprendizagem mais personalizada, permitindo que os alunos superem suas dificuldades de forma mais eficiente.
- **Integração com ferramentas de processamento de linguagem natural (Chat-GPT):** Uma adaptação interessante seria integrar o sistema especialista com ferramentas de processamento de linguagem natural, como o ChatGPT. Isso permitiria uma interação mais fluida e natural entre os alunos e o sistema, além de melhorar a precisão, contextualização e dinamismo das respostas. Com essa integração, seria possível oferecer um nível maior de personalização nas interações, aprimorando a qualidade do suporte oferecido.
- **Adição de dicas de novas questões de outras plataformas:** O Expert Bee também pode ser utilizado para fornecer dicas de questões de outras plataformas além do Beecrowd. Um possível aprimoramento seria adaptar o Expert Bee para identificar de qual plataforma as questões estão vindo e oferecer dicas específicas para essas questões. Dessa forma, seria possível ampliar a utilidade da ferramenta para diferentes plataformas de ensino, tornando-a mais flexível e adaptada a diferentes contextos de aprendizagem.

## References

- Barth, A. (2011). Http state management mechanism.
- Berssanette, J. H. and de Francisco, A. C. (2018). Uma proposta de ensino de programação de computadores com base na pbl utilizando o portal uri online judge. In *II SIMPÓSIO IBERO-AMERICANO DE TECNOLOGIAS EDUCACIONAIS*, pages 348–354, Araranguá.
- Bez, J. L. and Tonin, N. A. (2012). Uri online judge: A new classroom tool for interactive learning. In *The Steering Committee Of The World Congress In Computer Science, Computer Engineering And Applied Computing (Worldcomp)*, pages 1–5, Athens.
- Bez, J. L. and Tonin, N. A. (2014). Uri online judge e a internacionalização da universidade. *Revista Eletrônica de Extensão da Uri*, 10(18):237–249.
- Bohanec, M., Bratko, I., and Rajkovic, V. (1990). Expert system for decision making. *Sistemica*, 1(1):145–157.
- Bray, T. (2017). The javascript object notation (json) data interchange format.
- Chaves, J. O., Castro, A., Lima, R., Lima, M. V., and Ferreira, K. H. A. (2013). Uma ferramenta baseada em juízes online para apoio às atividades de programação de computadores no moodle. *Revista Novas Tecnologias na Educação*, 11(3).
- Clark, K. L., McCabe, F., and McCabe, F. G. (1980). Prolog: a language for implementing expert systems. Master’s thesis, Imperial College of Science and Technology. Department of Computing.

CodeRunner (2023). Coderunner.

da Cruz, A. K. B. S., de Salles Soares Neto, C., da Cruz, P. T. M. B., and Teixeira, M. A. M. (2022). Utilização da plataforma beecrowd de maratona de programação como estratégia para o ensino de algoritmos. In *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital (Sbgames Estendido 2022)*, pages 754–764. Sociedade Brasileira de Computação.

de Campos, C. P. and Ferreira, C. E. (2004). Boca: um sistema de apoio a competições de programação. In *Sociedade Brasileira de Computação*, São Paulo.

de Freitas, L. M. (2016). Análise de usabilidade do módulo laboratório virtual de programação do moodle. Master's thesis, Universidade Federal de Santa Catarina, Araranguá.

Dunstan, N. (2013). An interactive webbased expert system degree planner. *The Second International Conference on Informatics Engineering Information Science (ICIEIS2013). The Society of Digital Information and Wireless Communication*, pages 302–308.

Ferreira, R. M. G. (2022). Brooms: Brazilian online marathon scoreboard. Master's thesis, Faculdade Unb Gama, Universidade de Brasília, Brasília.

Fielding, R. (1999). Hypertext transfer protocol – http/1.1.

Francisco, R., Júnior, C. P., and Ambrósio, A. P. (2016). Juiz online no ensino de programação introdutória - uma revisao sistemática da literatura. In *Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016)*, pages 11–20.

França, A. B. and Soares, J. M. (2011). Sistema de apoio a atividades de laboratório de programação via moodle com suporte ao balanceamento de carga. In *Anais do XXII SBIE - XVII WIE*, pages 710–719. SBIE.

Galasso, R. H. and Moreira, B. G. (2014). Integração do ambiente boca com o ambiente moodle para avaliação automática de algoritmos. In *COMPUTER ON THE BEACH*, pages 22–31, Ponta Grossa.

Lima, G. M. (2022). Gamma online judge: projeto de criação de uma plataforma para o armazenamento das questões das maratonas unb de programação. Master's thesis, Faculdade Unb Gama, Universidade de Brasília, Brasília.

Lima, J. M. M. (2021). Plataforma moodle: a educação por mediação tecnológica. *Revista Científica Multidisciplinar Núcleo Do Conhecimento*, 7(6):17–37.

Lobb, R. and Harlow, J. (2016). Coderunner: a tool for assessing computer programming skills. In *ACM Inroads*, volume 7, pages 47–51, Ponta Grossa.

Masse, M. (2011). Rest api design rulebook.

Moodle (2023). Moodle developer resource centre.

NASR, O. A., TALAB, A., and AL-GAHTANI, S. S. (2018). Building of a rule-based expert system for academic advising via web expert system tools. Master's thesis, FKing Khalid University - College of Business - Dept. of MIS.

- Piekarski, A. E. T., Miazaki, M., da Rocha Junior, A. L., Militão, E. P., and da Silva, J. V. P. (2023). ProgramaÇÃO competitiva em um projeto de extensão para o ensino técnico em informática. *Revista Conexão UEPG*, 19(1):1–15.
- Ribeiro, R. B., Fernandes, D., de Carvalho, L. S. G., and Oliveira, E. (2018). Gamificação de um sistema de juiz online para motivar alunos em disciplina de programação introdutória. In *Anais do XXIX Simpósio Brasileiro de Informática na Educação (SBIE 2018)*, pages 805–814.
- Rodríguez-Del-Pino, J. C. (2023). Moodle plugins directory: Virtual programming lab.
- Rodríguez-Del-Pino, J. C., Royo, E. R., and Figueroa, Z. H. (2012). A virtual programming lab for moodle with automatic assessment and anti-plagiarism features. In *THE 2012 INTERNATIONAL CONFERENCE ON E-LEARNING, E-BUSINESS, ENTERPRISE INFORMATION SYSTEMS, E-GOVERNMENT*, Hong Kong.
- Russell, S. and Norvig, P. (2013). Inteligência artificial: tradução da terceira edição. Elsevier.
- Santos, J. C. S. and Ribeiro, A. R. L. (2011). Jonline: proposta preliminar de um juiz online didático para o ensino de programação. In *Anais do Simpósio Brasileiro de Informática na Educação (SBIE 2011)*, pages 964–967.
- Singla, J. (2013). The diagnosis of some lung diseases in a prolog expert system. *International Journal of Computer Applications*, 78(15):37–40.
- van Kesteren, A. (2010). Cross-origin resource sharing, w3c working draft wd-cors-20100727.
- Verdaguer, J. (2021). O que é lti e como ele pode melhorar seu ecossistema de aprendizagem.
- Vickers, S. P. and Booth, S. (2014). Creating environments for learning through instigating a community of developers: A jisc-funded project. *Learning Tools Interoperability (LTI): A Best Practice Guide*.
- VPL (2021). Virtual programming lab for moodle (vpl).
- Wasik, S., Antczak, M., Badura, J., Laskowski, A., and Sternal, T. (2018). A survey on online judge systems and their applications. *Acm Computing Surveys*, 51(1):1–34.
- WIELEMAKER, J. (2012). In *SWI-Prolog 6.1: Reference Manual*.
- Wielemaker, J. (2014). Swi-prolog version 7 extensions. In *Workshop on Implementation of Constraint and Logic Programming Systems and Logic-based Methods in Programming Environments*.
- Wielemaker, J. (2024). Swi-prolog http support.
- WIELEMAKER, J., HUANG, Z., and MEIJ, L. V. D. (2008). Swi-prolog and the web. In *Theory and Practice of Logic Programming*, pages 363–392.