



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS REITOR JOÃO DAVID FERREIRA LIMA
PROGRAMA DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Raquel Cristina Schaly Behrens

**EXPERT BEE: UM SISTEMA ESPECIALISTA DE APOIO À APRENDIZAGEM
INTEGRADA AO BEECROWD**

Florianópolis, Santa Catarina – Brasil
2024

Raquel Cristina Schaly Behrens

**EXPERT BEE: UM SISTEMA ESPECIALISTA DE APOIO À APRENDIZAGEM
INTEGRADA AO BEECROWD**

Trabalho de Conclusão de Curso submetido
ao Programa de Graduação em Ciências da
Computação da Universidade Federal de Santa
Catarina para a obtenção do Grau de Bacharel em
Ciências da Computação.

Orientador(a): Prof. Maicon Rafael Zatelli, Dr.

Coorientador(a): Prof. Antonio Carlos Mariani

Florianópolis, Santa Catarina – Brasil
2024

Raquel Cristina Schaly Behrens

**EXPERT BEE: UM SISTEMA ESPECIALISTA DE APOIO AO ENSINO INTEGRADO
AO BEECROWD**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharela em Ciências da Computação, e foi aprovado em sua forma final pelo Programa de Graduação em Ciências da Computação do INE – Departamento de Informática e Estatística, CTC – Centro Tecnológico da Universidade Federal de Santa Catarina.

Florianópolis, Santa Catarina – Brasil, 28 de novembro de 2024.

Nome do Coordenador(a), Dr.

Coordenador(a) do Programa de
Graduação em Ciências da Computação

Banca Examinadora:

Prof. Maicon Rafael Zatelli, Dr.

Orientador(a)
Universidade Federal de Santa
Catarina – UFSC

Prof. Antonio Carlos Mariani

Coorientador(a)
Universidade Federal de Santa
Catarina – UFSC

Prof. Álvaro Junio Pereira Franco, Dr.

Avaliador
Universidade Federal de Santa
Catarina – UFSC

Profa. Luciana de Oliveira Rech, Dra.

Avaliadora

Universidade Federal de Santa
Catarina – UFSC

*Ao meu querido irmão, Lucas Guilherme Schaly Behrens (in memoriam),
que nos deixou há pouco tempo e faz muita falta. Você marcou profundamente
a vida de todos que tiveram o privilégio de conhecê-lo.*

A saudade é eterna.

AGRADECIMENTOS

Agradeço a Deus pela força que me concedeu em toda essa caminhada, pois tudo posso naquele que me fortalece.

Agradeço à minha família, e em especial aos meus pais, Valter Germano Behrens e Ana Lidia Schaly Behrens, e à minha avó Denilce Jung, por possibilitarem toda a minha trajetória acadêmica e pelo amparo incondicional.

Agradeço ao meu namorado, Pedro Augusto Probst Bennemann, por estar ao meu lado desde o início dessa caminhada, me incentivando e apoiando.

Aos meus queridos amigos, Gabriella Schmid Scapini e Jose Daniel Alves do Prado, que sempre estiveram ao meu lado. Agradeço por todos os momentos compartilhados, e que venham muitos outros.

Ao meu orientador, o professor Maicon Rafael Zatelli, agradeço pela atenção, paciência e orientação, que foram essenciais para que o projeto fosse concluído.

À UFSC e a todos os seus professores que sempre proporcionaram um ensino de alta qualidade.

E a todos os meus amigos, que fizeram parte dessa trajetória e que a tornaram mais leve e divertida.

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 incentivar 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.

Palavras-chaves: beecrowd. moodle. sistema especialista. prolog. swi-prolog. LTI. API. programação.

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 encourage 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.

Keywords: beecrowd. moodle. expert system. prolog. swi-prolog. LTI. API. programming.

LISTA DE FIGURAS

Figura 1	-	Portal Beecrowd - Home	25
Figura 2	-	Portal Beecrowd - Categorias	26
Figura 3	-	Portal Beecrowd - Exercícios dentro da categoria Paradigmas	27
Figura 4	-	Portal Beecrowd - Fórum do Problema 1063	28
Figura 5	-	Portal Beecrowd - Academic	28
Figura 6	-	Portal Beecrowd - Exercício 1001	29
Figura 7	-	Portal Beecrowd - Exercício Submetido	30
Figura 8	-	Portal Beecrowd Academic - Progresso dos Alunos	30
Figura 9	-	Portal Beecrowd Academic – Linguagens e Versões	31
Figura 10	-	Integração BOCA-Moodle - Edição de tela de cadastro de questão	45
Figura 11	-	Integração BOCA-Moodle - Visualização do retorno da avaliação da questão	45
Figura 12	-	Integração BOCA-Moodle - Fluxo da interação das bases de dados do Moodle e do BOCA	45
Figura 13	-	Integração BOCA-Moodle - retorno da avaliação da questão como correta	46
Figura 14	-	CodeRunner - A simples pergunta sobre Python, respondida erroneamente	47
Figura 15	-	CodeRunner - A simples pergunta sobre Python, respondida corretamente	48
Figura 16	-	VPL - Componentes	49
Figura 17	-	VPL - Configuração básica de restrições	50
Figura 18	-	VPL - Exemplo de configuração de casos de teste	51
Figura 19	-	VPL - Arquivos de execução para testes avançados	51
Figura 20	-	Sistema especialista baseado em regras - Árvore de decisão	53
Figura 21	-	Planejador de grau acadêmico - Layout da página	55
Figura 22	-	Planejador de grau acadêmico - Interação cliente-servidor	55
Figura 23	-	Atividade para o professor acessar o Beecrowd Academic	60
Figura 24	-	Beecrowd Academic	61
Figura 25	-	Atividade para o aluno acessar o Beecrowd	61
Figura 26	-	Beecrowd do aluno	62
Figura 27	-	Diagrama de Atividades do Usuário da Aplicação	69
Figura 28	-	Arquitetura da Aplicação do Sistema Especialista	70
Figura 29	-	Expert Bee: Tela Inicial	81
Figura 30	-	Expert Bee: Detalhes das Respostas do Juiz Online do Beecrowd	82
Figura 31	-	Expert Bee: Chat	83
Figura 32	-	Expert Bee: Escolhendo a Questão	84
Figura 33	-	Expert Bee: Digitando o Número da Questão	85

Figura 34 – Expert Bee: Questão não encontrada	86
Figura 35 – Expert Bee: Primeira Pergunta	87
Figura 36 – Expert Bee: Sim para a Primeira Pergunta	88
Figura 37 – Expert Bee: Não para a Primeira Pergunta	89
Figura 38 – Expert Bee: Usuário conseguiu	90
Figura 39 – Expert Bee: Usuário não conseguiu	91

LISTA DE TABELAS

Tabela 1	– Comparação entre quatro integrações de sistemas, que auxiliam no ensino da programação, com o Moodle: Beecrowd, BOCA, CodeRunner e VPL	57
Tabela 2	– Matriz de Decisões para o Predicado <i>questao/3</i> na Questão 1181	76

LISTA DE CÓDIGOS

Código 1	-	Componente Chat	71
Código 2	-	Rotas do frontend	72
Código 3	-	Funções assíncronas para requisição HTTP	73
Código 4	-	Arquivo <i>questao_1181.pl</i>	74
Código 5	-	Arquivo <i>server.pl</i> - <i>Handlers</i> e Manipuladores de requisições . .	77
Código 6	-	Arquivo <i>server.pl</i> - <i>Handlers</i> e Manipuladores de requisições . .	79
Código 7	-	Arquivo <i>server.pl</i> - <i>Handlers</i> e Manipuladores de requisições . .	80

LISTA DE ABREVIATURAS E SIGLAS

IDE	<i>Integrated Development Environment</i>
URI	Universidade Regional Integrada
VPL	<i>Virtual Programming Lab</i>
API	<i>Application Programming Interface</i>
VLE	<i>Virtual Learning Environment</i>
LMS	<i>Learning Management System</i>
SGBD	Sistema Gerenciador de Banco de Dados
AVA	Ambiente Virtual de Aprendizagem
RG	Regra de Negócio
RNF	Requisito Não Funcional
RF	Requisito Funcional

SUMÁRIO

1	INTRODUÇÃO	17
1.1	OBJETIVOS	20
1.1.1	Objetivo Geral	20
1.1.2	Objetivos Específicos	21
1.2	MÉTODOS DE PESQUISA	21
1.3	ESTRUTURA DO TRABALHO	22
2	FUNDAMENTAÇÃO TEÓRICA	23
2.1	JUÍZES ONLINE	23
2.2	BEECROWD	25
2.3	BOCA ONLINE CONTEST ADMINISTRATOR	30
2.4	VPL	32
2.5	CODERUNNER	33
2.6	MOODLE	34
2.6.1	Características Gerais do Moodle	35
2.6.1.1	Administração do site	35
2.6.1.2	Administração dos usuários	36
2.6.1.3	Administração do Curso	36
2.6.2	LTI External tools	36
2.7	LTI	38
2.8	SISTEMA ESPECIALISTA	39
2.8.0.1	Prolog	39
2.8.0.2	SWI-Prolog	40
2.9	WEB API	41
3	TRABALHOS RELACIONADOS	42
3.1	UTILIZAÇÃO DA PLATAFORMA BEECROWD DE MARATONA DE PROGRAMAÇÃO COMO ESTRATÉGIA PARA O ENSINO DE ALGORITMOS	42
3.2	INTEGRAÇÃO DO AMBIENTE BOCA COM O AMBIENTE MOODLE PARA AVALIAÇÃO AUTOMÁTICA DE ALGORITMOS	44
3.3	CODERUNNER: A TOOL FOR ASSESSING COMPUTER PROGRAMMING SKILLS	46
3.4	A VIRTUAL PROGRAMMING LAB FOR MOODLE WITH AUTOMATIC ASSESSMENT AND ANTI-PLAGIARISM FEATURES	49
3.5	UMA FERRAMENTA BASEADA EM JUÍZES ONLINE PARA O APOIO ÀS ATIVIDADES DE PROGRAMAÇÃO DE COMPUTADORES NO MOODLE	52

3.6	BUILDING OF A RULE-BASED EXPERT SYSTEM FOR ACADEMIC ADVISING VIA WEB EXPERT SYSTEM TOOLS	53
3.7	AN INTERACTIVE WEB-BASED EXPERT SYSTEM DEGREE PLANNER	54
3.8	COMPARAÇÃO ENTRE OS TRABALHOS	56
4	DESENVOLVIMENTO	60
4.1	LTI BEECROWD	60
4.2	SISTEMA ESPECIALISTA WEB	62
4.2.1	Coleta de Dados	63
4.2.2	Requisitos da Aplicação	64
4.2.2.1	Regras de Negócio	65
4.2.2.2	Requisitos Funcionais	65
4.2.2.3	Requisitos Não Funcionais	65
4.2.3	Tecnologias Usadas	66
4.2.3.1	Prolog e SWI-Prolog	66
4.2.3.2	API REST	67
4.2.3.3	JSON	67
4.2.3.4	HTTP e CORS	67
4.2.3.5	Sessões HTTP	68
4.2.3.6	React e Typescript (Frontend)	68
4.2.4	Arquitetura de Software	69
4.2.5	Elementos do Frontend	70
4.2.6	Comunicação Frontend-Backend	72
4.2.7	Elementos do Backend	74
4.2.7.1	Arquivos no formato questao_{numeroDaQuestao}.pl	74
4.2.7.2	Arquivo server.pl	77
4.2.7.2.1	<i>Configurando servidor HTTP</i>	77
4.2.7.2.2	<i>Carregando os arquivos das questões</i>	79
4.2.7.2.3	<i>Recebe respostas</i>	80
4.2.8	Expert Bee: Sistema Especialista de Apoio à Resolução de Dúvidas nos Exercícios do Beecrowd	81
4.2.9	Adicionar novas questões no Expert Bee	91
5	VALIDAÇÃO	93
5.1	INTEGRAÇÃO LTI ENTRE O MOODLE E O BEECROWD	93
5.1.1	Vantagens	93
5.1.2	Desvantagens	93
5.2	SISTEMA ESPECIALISTA PARA RESOLUÇÃO DE DÚVIDAS NO BEECROWD	93

5.2.1	Vantagens	93
5.2.2	Desvantagens	94
5.3	AVALIAÇÃO PRÁTICA	95
5.3.1	Resultados	95
5.3.2	Pontos de Melhoria	96
6	CONSIDERAÇÕES FINAIS	97
6.1	TRABALHOS FUTUROS	97
	REFERÊNCIAS	99
	APÊNDICE A – MANUAL DE CONFIGURAÇÃO DA LTI BEE-CROWD NO MOODLE	108
	APÊNDICE B – MANUAL DE USO LTI BEECROWD NO MOODLE	111
B.1	UTILIZAR LTI DO BEECROWD JÁ CONFIGURADA NO MOODLE	111
B.1.1	Criando a atividade	111
B.1.2	Entrando como aluno	118
B.1.3	Notas	121
	C – COMO ADICIONAR RESPOSTAS PARA NOVAS QUESTÕES NO EXPERT BEE	123
C.1	ESTRUTURA DO ARQUIVO	123
C.1.1	Definição do módulo	123
C.1.2	Predicado questao/3	123
C.1.3	Predicado diagnostico/3	123
C.2	CONSIDERAÇÕES IMPORTANTES	124

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; JÚNIOR; AMBRÓSIO, 2016, p. 18-19). 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, J. M. M., 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; SOARES, 2011, p. 712).

O VPL é um módulo integrável ao Moodle que permite 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, Freitas (2016, p. 129) 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 diversos problemas de interface gráfica e de usabilidade. Destacou também falta de mensagens de ajuda apresentadas pela interface ao encontrar-se erros na submissão do código, a obrigatoriedade de casos de testes serem inseridos manualmente pelo criador da atividade, e a existência de botões não intuitivos para criação de atividades, como alguns dos defeitos encontrados, e sugeriu uma lista de melhorias necessárias nesse módulo. Além disso, Freitas (2016, p. 58) cita que o VPL não permite o balanceamento de carga, já que há apenas um servidor responsável pela compilação e execução do código, podendo gerar um possível gargalo se muitos alunos submeterem trabalhos simultaneamente em um mesmo ambiente do Moodle.

Em contrapartida, Cruz et al. (2022, p. 5) destaca a plataforma Beecrowd como uma ferramenta que possibilita a resolução de problemas e desenvolvimento de algoritmos pelos alunos, ao mesmo tempo em que incentiva a participação em maratonas de programação e auxilia o educador na correção automatizada das respostas às

questões. Do mesmo modo, [Bez e Tonin \(2014, p. 239\)](#) mostram que a construção do Beecrowd (antigo URI Online Judge) foi inteiramente focada nas necessidades dos professores e, sobretudo, nas necessidades dos alunos. Nos aspectos didáticos e pedagógicos, é possível que o professor utilize o módulo acadêmico do portal a fim de fornecer aos alunos listas de exercícios, acompanhando o desempenho individual de cada um, podendo categorizar as listas por tema e estabelecer prazos para sua conclusão. Ainda, o portal possui recursos extremamente importantes para uma vida estudantil acadêmica, como “um nível de problemas para iniciantes, sistema de recompensa por *badges*, um módulo acadêmico completo para acompanhamento de listas de exercícios pelos alunos, *ranking* dos alunos por Universidade, entre outros.” ([BEZ; TONIN, 2014, p. 239](#)).

À vista disso, e ressaltando que o maior problema no ensino de algoritmos e programação para novos estudantes é atender com eficiência à grande diversidade de alunos e seus diferentes modos e ritmos de aprendizado ([BEZ; TONIN, 2012, p. 1](#)), “a possibilidade de utilização de ferramentas online com grande disponibilidade para aprendizagem ativa pode ser uma forma de possibilitar que cada aluno aprenda em seu próprio ritmo e velocidade” ([CRUZ et al., 2022, p. 5](#)). Ademais, o uso de ferramentas online de auto aprendizado faz com que os educandos possam desenvolver suas habilidades de programação, desenvolvimento de código e solução de problemas com confiança e segurança, seguindo seu próprio ritmo de aprendizado ([BEZ; TONIN, 2014, p. 239-240](#)).

[Berssanette e Francisco \(2018, p. 248\)](#), salientou que o uso do Beecrowd contribui para o processo de aprendizagem de programação dos alunos ao proporcionar *feedback* em tempo real aos estudantes e estimular a competitividade entre os alunos, por meio da gamificação presente na ferramenta, com *badges* e *ranking* dos estudantes. Também evidenciou o crescimento do anseio dos alunos por aprofundarem seus conhecimentos na resolução de problemas, e o envolvimento mais intenso dos mesmos na disciplina. Equitativamente, [Ferreira \(2022, p. 31\)](#) informa que a plataforma Beecrowd possui estudantes de mais de 240 países registrados, e frisou alguns dos seus recursos interessantes, como: a existência de uma base de problemas classificados em categorias e níveis de dificuldade, e a presença de um modelo de solução para cada problema, o qual é comparado com os dados de saída do programa do usuário, informando se o programa possui algum percentual de erro.

De maneira geral, os alunos respondem de forma positiva ao *feedback* imediato. No ambiente de laboratório, a rápida avaliação de cada pergunta motiva os alunos a buscar notas positivas, sendo raros os casos em que avançam sem corrigir eventuais erros. Eles apreciam a capacidade de acompanhar seu progresso durante todo o laboratório. Em exames, o *feedback* imediato elimina a incerteza sobre o desempenho, permitindo que os alunos saiam da sala de exame com clareza sobre suas notas ([LOBB; HARLOW, 2016, p. 49](#)).

Ainda, de acordo com [Gustavo Marques Lima \(2022, p. 24\)](#), é notável que a plataforma Beecrowd seja uma das poucas no Brasil a disponibilizar questões em português brasileiro, o que a torna uma escolha significativa por parte dos professores para a utilização em sala de aula. Essa preferência amplia consideravelmente a acessibilidade do juiz online, beneficiando principalmente os iniciantes na área que ainda não possuem habilidades de leitura em inglês.

Vale ressaltar que o autor também identificou duas outras plataformas, a Neps Academy e a CodeBench, que possuem questões em português brasileiro. Entretanto, essas alternativas não oferecem a mesma gama de recursos encontrados no Beecrowd. A Neps Academy, por exemplo, não permite o registro de turmas e a inscrição de alunos, funcionalidades que o Beecrowd oferece. Por outro lado, a CodeBench, embora seja amigável para professores e alunos, disponibiliza problemas privados. Nessa plataforma, apenas é possível visualizar as questões por meio de turmas com acesso restrito criadas pelos professores, o que impede que os alunos tentem resolver exercícios fora do ambiente de sala de aula ([LIMA, G. M., 2022, p. 41](#)).

Além disso, os autores [Bez e Tonin \(2012, p. 41\)](#) testaram o uso dos juizes online BOCA e UVa Online Judge em sala de aula. Quanto ao BOCA, os professores explicaram que era necessário fazer uma limpeza no site a cada semestre, e novos problemas tinham de ser registrados. E, toda vez que isso tinha de ser feito, a classificação dos problemas e dos usuários eram perdidas. Já no uso do UVa Online Judge, apesar desse juiz conter uma enorme variedade de problemas em todos os assuntos de algoritmos, o período de manutenção do servidor geralmente coincidia com os horários das aulas, ficando off-line, e impossibilitando a dependência da ferramenta ([BEZ; TONIN, 2012, p. 1](#)).

[Bez e Tonin \(2014, p. 248\)](#) também registraram que, uma semana após o lançamento online da plataforma Beecrowd (conhecida como URI Online Judge na época), estudantes da área de Computação em todo o país já acessaram o portal, e, após um mês, os dados do Google Analytics revelaram a presença constante de usuários em todos os estados brasileiros e além-fronteiras. Após cerca de um ano e meio de operação, o Beecrowd mostrou-se sendo recebido, tanto nacional quanto internacionalmente, sendo regularmente utilizado por estudantes ao redor do globo, apresentando uma notável taxa de crescimento de usuários.

Dessa maneira, ao considerar as limitações do módulo VPL, incluindo problemas de interface gráfica e usabilidade, e as desvantagens dos diversos juízes online mencionados anteriormente, destaca-se a popularidade do Beecrowd dentro e fora do Brasil, bem como seus diversos recursos avançados. Esses recursos foram desenvolvidos para atender às necessidades de alunos e professores, promovendo efetivamente o aprendizado. O crescente interesse da comunidade acadêmica em adotar o Beecrowd reflete uma demanda crescente por essa plataforma, que tem demonstrado sucesso em alcançar seus objetivos ([FERREIRA, 2022, p. 31](#)). Portanto, deve-se estimular o

uso do portal em ambientes acadêmicos, a fim de melhorar a qualidade do ensino de programação e resolução de problemas nas salas de aula.

1.1 OBJETIVOS

Neste contexto, o presente trabalho visa incentivar o uso da plataforma Beecrowd entre os docentes, aproveitando a integração LTI (Learning Tools Interoperability) já estabelecida entre o Moodle e o Beecrowd, desenvolvida pela equipe do Beecrowd. A LTI é um padrão que facilita a troca segura de dados entre o Sistema de Gestão de Aprendizado (LMS) e ferramentas de aprendizagem externas, centralizando o acesso a conteúdos internos e externos e facilitando logins entre diferentes plataformas ([VERDAGUER, 2021](#)). Assim, a LTI do Beecrowd permite que os usuários acessem o Beecrowd diretamente do Moodle, de maneira rápida e intuitiva por meio de um botão, tornando o uso da plataforma mais acessível e ágil.

Para complementar essa integração e oferecer um suporte abrangente, será disponibilizada uma aplicação web de apoio, com um sistema especialista que atende tanto professores quanto alunos. Para os professores, a ferramenta ajudará a esclarecer dúvidas frequentes dos alunos. Já para os alunos, ela permitirá que resolvam suas dúvidas de forma independente, sem depender da disponibilidade do professor.

Como as questões do Beecrowd podem ser reutilizadas a cada semestre em atividades de ensino e prática de programação, é comum que os alunos enfrentem dúvidas semelhantes ao resolverem esses problemas, o que gera um padrão previsível de questionamentos. Dada a diversidade de questões e os critérios rigorosos do sistema de correção do Beecrowd — que exige, por exemplo, formatação exata das respostas e a aprovação em múltiplos casos de teste —, essa ferramenta de apoio pretende facilitar a adaptação dos professores, fornecendo respostas a problemas comuns e orientações sobre as expectativas de correção do juiz online do Beecrowd, além de contribuir para uma melhor experiência de ensino e aprendizado na plataforma.

1.1.1 Objetivo Geral

Incentivar o uso da plataforma Beecrowd pelos docentes, promovendo sua integração com o ambiente Moodle por meio da LTI já implementada. Essa integração 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, disponibilizar 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.

1.1.2 Objetivos Específicos

- (a) Desenvolver um manual de instruções para auxiliar os professores no uso da LTI do Beecrowd, permitindo que compreendam e utilizem a ferramenta de forma eficaz, com ênfase nos recursos robustos do Beecrowd e seu vasto banco de questões;
- (b) Criar uma plataforma com um sistema especialista voltado ao esclarecimento de dúvidas frequentes dos alunos sobre as questões do Beecrowd. Este sistema atenderá as dúvidas recorrentes que surgem ao longo do semestre, já que muitas questões e suas respectivas dificuldades permanecem semelhantes;
- (c) Elaborar um manual de instruções para orientar os docentes sobre como adicionar respostas a novas dúvidas no sistema especialista.

1.2 MÉTODOS DE PESQUISA

Na seção de fundamentação teórica, realiza-se uma análise abrangente sobre juízes online e suas aplicações no contexto acadêmico, com foco específico na plataforma Beecrowd e no BOCA Online Contest Administrator, investigando suas operações e funcionalidades. Nesta mesma seção, são relatadas e explicadas outras alternativas para o uso de ambientes auxiliares no ensino de programação, como o VPL e o Code-Runner, integrados ao AVA Moodle. Detalhes de funcionamento e execução de cada um são observados. Além disso, conduz-se uma análise aprofundada do ambiente AVA Moodle, explorando o conceito de LTI External Tools e suas relações com o Moodle, juntamente com os detalhes dos campos a serem preenchidos para a configuração de uma ferramenta externa LTI. Os conceitos LTI, Sistema Especialista e sua relação com a linguagem de programação Prolog e o projeto SWI-Prolog, e API também são expostos nessa seção, essenciais para o entendimento do desenvolvimento do sistema especialista.

Na seção de trabalhos relacionados, realiza-se uma pesquisa bibliográfica sobre a integração de juízes online e outros ambientes de ensino de programação com o Moodle. O objetivo é avaliar a eficácia do juiz online Beecrowd na aprendizagem dos alunos, além de analisar como outros projetos semelhantes foram desenvolvidos, comparando suas abordagens com o presente trabalho. Adicionalmente, são discutidos dois estudos focados na criação de sistemas especialistas web, explorando suas características e destacando as principais diferenças em relação à proposta deste trabalho.

Na seção de desenvolvimento, são explorados os aspectos relacionados à LTI do Beecrowd, que facilita a integração entre o Moodle e a plataforma Beecrowd. Nessa etapa, são apresentados manuais detalhados para a configuração e utilização da LTI

no Moodle. Além disso, a seção inclui diagramas, requisitos funcionais e não funcionais, bem como regras de negócio, que juntos definem a arquitetura geral do sistema especialista desenvolvido. Também são descritas as tecnologias selecionadas para o desenvolvimento, acompanhadas de uma explicação detalhada sobre a implementação da aplicação web do sistema especialista. Por fim, são exibidas imagens ilustrando o resultado final da aplicação.

Na seção de validação, será realizada uma análise da proposta, destacando as vantagens e desvantagens de utilizar a integração LTI entre o Moodle e o Beecrowd, bem como os benefícios e limitações de empregar um sistema especialista para resolver dúvidas relacionadas às questões do Beecrowd. Além disso, essa seção apresentará uma avaliação prática do uso da aplicação pelos alunos da disciplina *Programação Orientada a Objetos I* da UFSC, no segundo semestre de 2024, verificando se as dúvidas dos estudantes foram efetivamente resolvidas pelo sistema especialista.

Por fim, na seção de conclusão, será apresentado um resumo das principais ações realizadas ao longo do trabalho, com ênfase na integração do Moodle com o Beecrowd por meio de LTI e no desenvolvimento da ferramenta com sistema especialista para apoiar os alunos. Além disso, serão discutidas as possibilidades de trabalhos futuros, com sugestões de aprimoramentos e novas direções para a ferramenta.

1.3 ESTRUTURA DO TRABALHO

A estrutura deste trabalho é organizada da seguinte maneira: a seção de Fundamentação Teórica apresenta os conceitos essenciais para compreender o desenvolvimento do projeto. Em Trabalhos Relacionados, são comparadas abordagens de integração de juízes online com o Moodle, além de artigos que tratam do desenvolvimento de sistemas especialistas web. O capítulo de Desenvolvimento apresenta a arquitetura da integração LTI, as tecnologias utilizadas e a implementação do sistema especialista. Na seção de Validação, são avaliadas as vantagens e limitações do sistema. A Conclusão discute um resumo das ações realizadas no trabalho e apresenta sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos necessários para a compreensão da proposta. Serão mostrados tópicos como: juízes online, destacando o impacto dessa tecnologia na formação dos estudantes; Beecrowd, detalhando suas funcionalidades como plataforma de avaliação; BOCA Online Contest Administrator, VPL (Virtual Programming Lab) e CodeRunner, ferramentas de apoio ao aprendizado de programação, mas apresentam limitações para o uso em sala de aula. Em seguida, será apresentado o Moodle, com ênfase em suas características gerais e na administração de cursos, usuários e sites. A integração de ferramentas externas por meio de LTI também será abordada, destacando sua importância na interoperabilidade entre sistemas. Por fim, o capítulo abordará o conceito de web API, explicando seus fundamentos e funcionamento básico.

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 e Ribeiro \(2011, p. 1\)](#) 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; JÚNIOR; AMBRÓSIO, 2016, p. 12](#)).

Diversos juízes online podem ser encontrados na Internet, dentre eles estão o Beecrowd ([CRUZ et al., 2022](#)) e o Codebench ([RIBEIRO et al., 2018, p. 806](#)), 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 ([CAMPOS; FERREIRA, 2004](#)).

Ribeiro *et al.* (2018, p. 807-808) expôs que o Codebench é utilizado na UFAM (Universidade Federal do Amazonas) para dar suporte a professores e estudantes em disciplinas iniciais de programação. Assim, por meio desse sistema, professores e tutores das disciplinas podem disponibilizar exercícios de programação, listas e até mesmo provas para seus alunos, que, por sua vez, podem desenvolver soluções em uma IDE (*Integrated Development Environment*) acoplada ao próprio sistema. Consequentemente, após a adoção do Codebench como ferramenta de apoio a uma metodologia de ensino híbrido de programação na UFAM, houve um aumento no índice de aprova-

ções dos alunos da disciplina Introdução à Programação de Computadores ([GALVÃO; FERNANDES; GADELHA, 2016](#), p. 148-149).

Semelhantemente, [Francisco, Júnior e Ambrósio \(2016](#), p. 18-19) 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”.

Da mesma maneira, o desempenho dos estudantes do curso de Engenharia de Software da Universidade de Brasília foi influenciado diretamente pelo uso de problemas da maratona de programação e de juízes eletrônicos: observou-se que 50,3% dos alunos apresentaram um aumento no desempenho em disciplinas de programação ([SALES; JUNIOR; SALES, 2016](#), p. 218).

Sob outra perspectiva, o desenvolvimento dos juízes online também é voltado ao uso em competições de programação ([SANTOS; RIBEIRO, 2011](#), p. 964-965), já que são utilizados juízes eletrônicos, corretores automáticos de problemas, em maratonas de programação ([LIMA, G. M., 2022](#), p. 34).

É interessante observar que o formato das questões em maratonas de programação é muito semelhante ao das questões em Juízes Online, e muitos dos exercícios disponíveis em Juízes Online originaram-se de competições de programação. Por conseguinte, a introdução dos estudantes a essas plataformas cria um ambiente mais familiar para as maratonas de programação, promovendo um engajamento positivo dos alunos com essa atividade e gerando impactos favoráveis em seu desempenho em disciplinas gerais de programação ([LIMA, G. M., 2022](#), p. 33-34).

Outrossim, [Campos e Ferreira \(2004](#), p. 2) mostram as vantagens do incentivo à participação de alunos em competições de programação: “Competições de programação são uma excelente oportunidade para desenvolver nos alunos diversas habilidades que serão extremamente úteis no seu crescimento profissional. Nelas, os alunos aprendem de forma divertida conceitos importantes sobre estruturas de dados, algoritmos e desenvolvimento de software enquanto, ao mesmo tempo, aprendem a trabalhar em grupo e sob pressão” ([CAMPOS; FERREIRA, 2004](#), p. 2).

Portanto, além de seu papel essencial em competições de maratona de programação, os juízes online têm se destacado como ferramentas valiosas no ensino de programação básica. Como exposto acima, a integração dessas plataformas no ambiente educacional proporciona vantagens significativas, como a aprendizagem no ritmo do aluno, a promoção da participação em maratonas de programação e a facilidade dos professores em gerenciar uma turma. Portanto, essas ferramentas desempenham uma função fundamental na promoção do engajamento dos estudantes e no aprimoramento de seu desempenho em disciplinas gerais de programação, contribuindo

significativamente para o campo educacional.

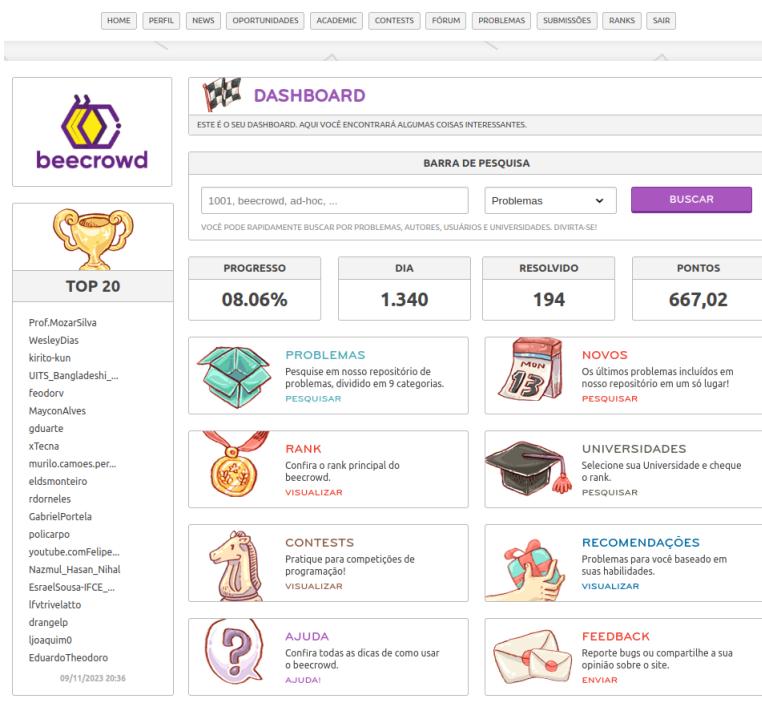
2.2 BEECROWD

O Beecrowd, desde janeiro de 2013 até outubro de 2021 conhecido como URI Online Judge ([PIEKARSKI et al., 2023](#), p. 5), é um portal online criado com o objetivo principal de tornar a prática de programação mais dinâmica, interessante e estimulante para aqueles que acabaram de ingressar na arte de programar ([BEZ; TONIN, 2012](#), p. 1).

A plataforma é um projeto desenvolvido na Universidade Regional Integrada - URI - Campus de Erechim, inicialmente com o intuito de criar um site que tivesse um juiz automático, capaz de atender às necessidades dos professores e dos alunos, visando principalmente a interatividade, flexibilidade e novas fontes de informação e usabilidade ([BEZ; TONIN, 2012](#), p. 1). O ambiente foi estruturado de forma a ser agradável, didaticamente organizado, e está disponível 24 horas por dia, oferecendo suporte tanto em inglês quanto em português ([BEZ; TONIN, 2014](#), p. 239).

A figura 1 mostra a página inicial do Beecrowd, após o usuário logar-se na plataforma.

Figura 1 – Portal Beecrowd - Home



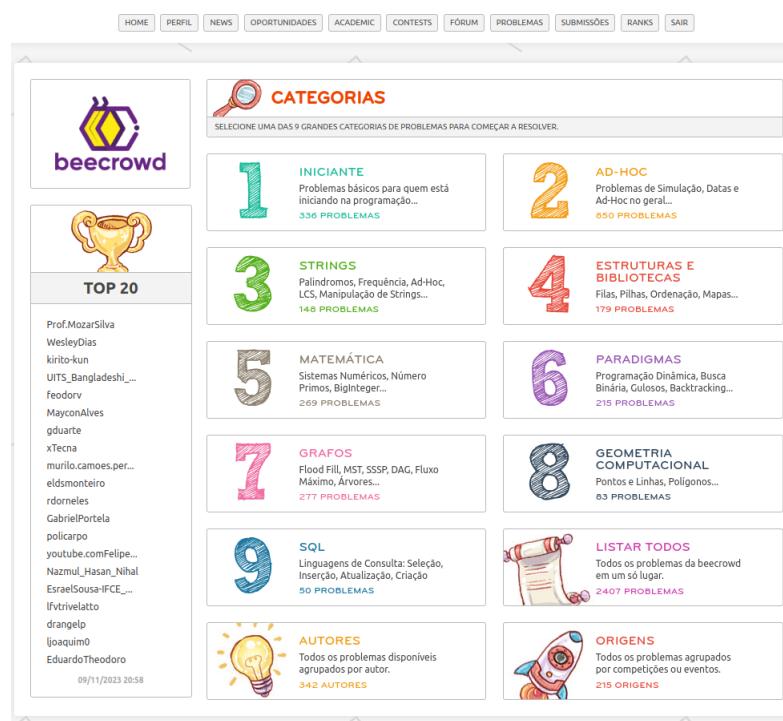
Fonte: ([BEECROWD, 2021](#))

Além disso, o Beecrowd contém problemas no estilo do ICPC - International Collegiate Programming Contest da ACM, e disponibiliza um juiz online que avalia em tempo real as submissões dos algoritmos dos usuários ([BERSSANETTE; FRANCISCO, 2018](#),

p. 350). Os problemas são categorizados por assunto e por nível de dificuldade, e há flexibilidade na escolha da linguagem de programação com a qual se deseja resolver os exercícios (BEZ; TONIN, 2012, p. 1).

A figura 2 mostra as diferentes categorizações dos exercícios, que são: Iniciante: problemas básicos para quem está iniciando na programação; Ad-Hoc: problemas de simulação, datas e Ad-Hoc no geral; Strings: palíndromos, frequência, ad-hoc, LCS, manipulação de strings; Estruturas e Bibliotecas: filas, pilhas, ordenação, mapas; Matemática: sistemas numéricos, números primos, BigInteger; Paradigmas: programação dinâmica, busca binária, algoritmos gulosos, backtracking; Grafos: flood fill, MST, SSSP, DAG, fluxo máximo, árvores; Geometria Computacional: pontos e linhas, polígonos; e SQL: linguagens de consulta - seleção, inserção, atualização, criação.

Figura 2 – Portal Beecrowd - Categorias



Fonte: (BEECROWD, 2021)

Já na figura 3 estão listados diversos exercícios dentro da categoria Paradigmas, e, para cada problema, há um nível classificado (destacado na imagem com um retângulo azul). O nível representa a dificuldade da questão, e pode variar de 1 até 10, sendo 1 o nível mais fácil, e 10 o nível mais difícil (BEECROWD, 2021). Dessa maneira, impede-se que estudantes iniciantes enfrentem frustrações ao tentar resolver problemas que demandam conhecimento mais avançado, prática e técnicas (BEZ; TONIN, 2014, p. 239).

Ademais, o portal Beecrowd não apenas fornece uma plataforma desafiadora para competições de programação, mas também se destaca como um recurso educacional abrangente. Entre os seus diferenciais, evidenciam-se materiais de estudo,

Figura 3 – Portal Beecrowd - Exercícios dentro da categoria Paradigmas

The screenshot shows the Beecrowd interface. At the top, there's a navigation bar with links like HOME, PERFIL, NEWS, OPORTUNIDADES, ACADEMIC, CONTESTS, FÓRUM, PROBLEMAS, SUBMISSIONS, RANKS, and SAIR. Below the navigation is a logo for 'beecrowd' and a trophy icon labeled 'TOP 20'. To the right of the trophy is a section titled 'PARADIGMAS' with the sub-instruction 'SELECIONE UM DOS SEGUINTE PROBLEMAS PARA RESOLVER.' Below this is a table listing 14 programming problems. The table has columns for '#', 'NOME', 'FAVORITO', 'RESOLVIDO', and 'NÍVEL'. The problems listed include 'Onda Crítica', 'Elhoarci, Quantas Chamadas?', 'Quantas Chamadas Recurativas?', 'Festival de Estátuas de Gelo', 'Sapo Dinâmico', 'Soma Permutada Elegante', 'Zonas de Ataque Independentes', 'Apagando e Ganhando', 'Dragster', 'Torre de Hanói, Novamente!', 'Pontas de São Petersburgo', 'Produção Ótima de Ótima Vodka', 'Cartões', 'Elevador Espacial', 'Combate ao Câncer', 'Palavras', 'DJ da Computação', 'ICPC Ataca Novamente', 'Organizando Pacotes', 'Caixas e Pedras', 'Motoboy', 'Canhão de Destruição', 'Jogo de Mosaicos', 'Lucro', and 'Empilhamento de Bolas'. At the bottom of the table, it says '1 OF 9' and has buttons for 'PRIMEIRO', 'ANTERIOR', 'PRÓXIMO', and 'ÚLTIMO'.

#	NOME	FAVORITO	RESOLVIDO	NÍVEL
1027	Onda Crítica	♥	886	9
1029	Elhoarci, Quantas Chamadas?	♥	16.523	4
1033	Quantas Chamadas Recurativas?	♥	1.284	6
1034	Festival de Estátuas de Gelo	♥	3.693	8
1054	Sapo Dinâmico	♥	887	7
1055	Soma Permutada Elegante	♥	900	8
1058	Zonas de Ataque Independentes	♥	189	8
1084	Apagando e Ganhando	♥	3.120	8
1106	Dragster	♥	231	7
1166	Torre de Hanói, Novamente!	♥	1.661	2
1203	Pontas de São Petersburgo	♥	1.350	7
1210	Produção Ótima de Ótima Vodka	♥	704	5
1224	Cartões	♥	1.097	8
1226	Elevador Espacial	♥	425	8
1229	Combate ao Câncer	♥	441	7
1231	Palavras	♥	200	7
1265	DJ da Computação	♥	279	6
1269	ICPC Ataca Novamente	♥	131	7
1282	Organizando Pacotes	♥	262	7
1283	Caixas e Pedras	♥	100	5
1286	Motoboy	♥	4.095	5
1288	Canhão de Destruição	♥	2.798	5
1299	Jogo de Mosaicos	♥	111	7
1310	Lucro	♥	4.533	4
1312	Empilhamento de Bolas	♥	784	6

Fonte: adaptado de ([BEECROWD, 2021](#))

tutoriais sobre algoritmos e programação, e um fórum que facilita a colaboração entre os usuários. Além desses aspectos, o portal foi projetado com características específicas que o tornam uma ferramenta eficaz de apoio às aulas de Algoritmos e Estruturas de Dados. Por meio de problemas que abordam conceitos fundamentais dessas disciplinas, o Beecrowd oferece uma abordagem prática que contribui para uma compreensão mais sólida por parte dos estudantes ([BEZ; TONIN, 2012](#), p. 2).

A figura 4 mostra o fórum do exercício 1063 do Beecrowd, que pode ser acessado facilmente em Fórum -> Estruturas e Bibliotecas -> 1063, ou pelo próprio exercício 1063, no seu botão “Fórum”. Nesse fórum, diversas perguntas foram criadas para os alunos conversarem entre si sobre erros e problemas que tiveram.

Além de ser um recurso fundamental para o aprimoramento das habilidades de programação dos alunos, o Beecrowd disponibiliza recursos diferenciados que se mostram valiosos para professores na área de Tecnologia da Informação. Essas características não apenas enriquecem o ambiente de aprendizado, mas também facilitam o processo de ensino e avaliação em cursos específicos dessa área. Os professores podem, por exemplo, criar disciplinas com alunos e listas de exercícios, separadas por assunto e delimitadas por prazos, caso desejado, e acompanhar a evolução dos alunos ([BEZ; TONIN, 2014](#), p. 239).

Na figura 5 está a página Academic, do portal Beecrowd, onde alocam-se as disciplinas do usuário. Dentro de cada disciplina se encontram as listas de exercícios disponibilizadas aos alunos pelos professores, bem como algumas orientações aos usuários.

Considerando que a construção do Beecrowd foi completamente centrada nas necessidades dos professores e, principalmente, nas necessidades dos alunos ([BEZ;](#)

Figura 4 – Portal Beecrowd - Fórum do Problema 1063

The screenshot displays the Beecrowd forum interface for Problem 1063. At the top, there's a navigation bar with links for HOME, PERFIL, NEWS, OPORTUNIDADES, ACADEMIC, CONTESTS, FÓRUM, PROBLEMAS, SUBMISSÕES, RANKS, and SAIR. Below the navigation is the Beecrowd logo and the problem identifier '1063'. To the right of the problem identifier, there's a sidebar with links for Descrição, Tela Cheia, Ranking, Fórum, and uDebug, along with a note '+ 4.2 PONTOS' and a link to ESTRUTURAS E BIBLIOTECAS. The main content area shows several forum posts. One prominent post is 'JavaScript - Atenção com espaços em branco' by RafaelOgelas, dated 01/12/2021 at 23:07. Another post is 'WA 100%, matches all test cases' by ViniciusSouzaMartins, dated 25/03/2021 at 15:03. Other posts include 'Recebendo WA (5%) mesmo forçando a resposta errada', 'Dica para visualizar melhor o problema.', 'AJUDA LINGUAGEM C, URI 1063 (Wrong answer (5%))', 'Wrong answer (5%)', and 'PROBLEM 1063 - URI Fórum 1.0'. Each post includes a small profile icon, the author's name, the date, and a brief description.

Fonte: (BEECROWD, 2021)

Figura 5 – Portal Beecrowd - Academic

The screenshot shows the Beecrowd Academic section. At the top, there's a navigation bar with links for HOME, PERFIL, NEWS, OPORTUNIDADES, ACADEMIC, CONTESTS, FÓRUM, PROBLEMAS, SUBMISSÕES, RANKS, and SAIR. On the left, there's a sidebar with the Beecrowd logo and a 'TOP 20' section listing student names: Prof.MozzSilva, WesleyDas, kirito-kun, UITs_Bangladesh, feedor, MayconAlves, gduarte, xTecno, murilo.camoes.per..., elismontteiro, rdornelles, GabriePortela, policarpo, poli...confelipe..., Nazmi_Jasen_Nhal, ErnaiSouzaIFCE..., lheirevelatto, drangeip, lisaquim0, EduardoTheodoro. Below this is a timestamp '09/11/2022 21:17'. To the right, there's a 'DISCIPLINAS' section showing a course titled 'PARADIGMAS DE PROGRAMAÇÃO 2022/01' with a '2022 23 STUDENTES 2 HOMEWORK' badge. There are also four large empty boxes for other courses, with 'ABRIR' buttons below them. At the bottom, there are navigation buttons for '1 OF 1', 'ANTERIOR', and 'PRÓXIMO'.

Fonte: (BEECROWD, 2021)

TONIN, 2014, p. 239), ele integra recursos dos melhores portais de programação do mundo e, adicionalmente, oferece funcionalidades exclusivas, como um nível de problemas destinado a iniciantes, um sistema de recompensas por meio de *badges*, e um módulo acadêmico completo para o acompanhamento das listas de exercícios pelos alunos, incluindo o *ranking* dos estudantes por universidade, entre outras inovações (BEZ; TONIN, 2014, p. 239).

A Figura 5 exemplifica um exercício do Beecrowd, o 1001: Extremamente Básico. Cada exercício é acompanhado por sua descrição, requisitos de entrada para o programa e a saída esperada do algoritmo a ser desenvolvido. Na seção à direita, o usuário pode selecionar a linguagem de programação desejada e utilizar a IDE integrada para desenvolver seu algoritmo. Ao concluir, o usuário deve clicar no botão "Enviar" para que a plataforma avalie sua solução. Após o envio, o aluno é redirecionado para uma tela onde o resultado é exibido na seção "Resposta"(Figura 6).

Figura 6 – Portal Beecrowd - Exercício 1001

Fonte: ([BEECROWD, 2021](#))

As respostas possíveis do Juiz Online do Beecrowd são: Accepted, Compilation Error, Runtime Error, Time Limited Exceeded, Presentation Error, Wrong Answer, Closed. O significado de cada resposta pode ser encontrado na plataforma, oferecendo ao usuário uma indicação do que pode estar incorreto em seu algoritmo ([BEECROWD, 2021](#)).

O Beecrowd também possui o ambiente Academic, e é esse ambiente que fornece ao professor diversos recursos ([ACADEMIC, 2021](#)), como:

- Criar disciplinas, podendo convidar alunos para participar dela;
- Criar listas de exercícios e acompanhar facilmente o progresso de seus estudantes (Veja figura 8), podendo ter uma melhor visão sobre como eles estão aplicando os tópicos apresentados em aula e onde eles estão encontrando dificuldades;
- Selecionar exercícios de programação do banco de dados do Beecrowd para essas listas, o qual possui mais de 2.000 problemas distintos;

Figura 7 – Portal Beecrowd - Exercício Submetido

Fonte: ([BEECROWD, 2021](#))

- Visualizar todas soluções enviadas pelos alunos para suas listas de exercícios através de uma organizada linha do tempo;
- Definir data de início, prazo e duração das listas, considerando somente as submissões recebidas durante o tempo determinado;
- Visualizar *feedback* sobre soluções copiadas entre estudantes e de repositórios online.

Figura 8 – Portal Beecrowd Academic - Progresso dos Alunos

STUDENT	1192	1193	1199	1026	1198	1093	1197	1216	1217	1218	1170	1161	1029	1234	1235	1238	1239	1240	1241	1243	PROGRESS	
10 Andre Tonin	✗	✗	✗	-	-	-	✓	✗	✗	✗	-	✓	-	-	✓	-	-	-	-	25.00%		
56 Michele Selivon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100.00%	
68 Ricardo Fávero Júnior	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	✓	✓	-	-	✓	✓	-	60.00%	
69 Matheus William Auler	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100.00%
82 Gustavo Lanzana	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	80.00%
83 Mateus Lazarotto	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	85.00%

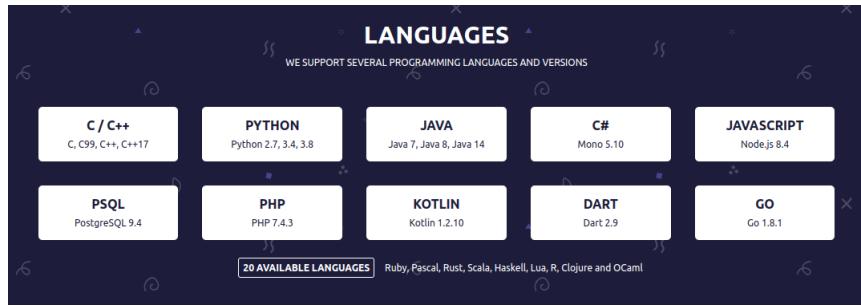
Fonte: ([BEZ; TONIN, 2012](#), p. 244)

Por fim, o Beecrowd oferece suporte a várias linguagens de programação e versões, dentre as quais o usuário pode escolher para desenvolver suas soluções. Todas essas linguagens estão listadas abaixo, na figura 9.

2.3 BOCA ONLINE CONTEST ADMINISTRATOR

O BOCA Online Contest Administrator é um ambiente utilizado para gerenciar competições que envolvem programação de computadores ([GALASSO; MOREIRA,](#)

Figura 9 – Portal Beecrowd Academic – Linguagens e Versões



Fonte: ([BEECROWD, 2021](#))

[2014](#), p. 22), e foi desenvolvido para ser usado na Maratona de Programação da Sociedade Brasileira de Computação ([CAMPOS; FERREIRA, 2004](#), p. 2).

Por ser um juiz online ([BEZ; TONIN, 2012](#), p. 1), realiza a avaliação automática dos algoritmos submetidos, classificando-os como corretos ou incorretos ([GALASSO; MOREIRA, 2014](#), p. 22). Ou seja, na competição, a correção dos programas enviados pelos times ocorre de forma online, e o resultado deve ser transmitido ao time o mais rápido possível. Após os competidores concluírem uma questão, enviam-na para o BOCA, que julga a solução fornecida e indica se está correta ou incorreta. Se estiver incorreta, o aluno tem a oportunidade de corrigi-la e submetê-la novamente, ainda durante o tempo da prova, mas sem ter conhecimento específico sobre o erro na questão ([CAMPOS; FERREIRA, 2004](#), p. 4).

A utilização de interfaces web possibilita que o sistema BOCA seja aplicado na organização de competições distribuídas, permitindo que equipes estejam localizadas em qualquer lugar com acesso à internet. Essa abordagem é adequada para simulações e outras modalidades de competições ([CAMPOS; FERREIRA, 2004](#), p. 20).

Além disso, o BOCA proporciona a flexibilidade de ser utilizado em competições com várias sedes (ou sites), todos operando simultaneamente, com a correção centralizada de submissões e dúvidas. Isso viabiliza competições em larga escala, alcançando regiões distantes de um país, como o Brasil, como evidenciado na Maratona de Programação. A centralização no processamento de submissões e dúvidas é essencial para assegurar igualdade de condições a todos os participantes, independentemente de sua localização, uma vez que tudo é processado pelo mesmo grupo de juízes ([CAMPOS; FERREIRA, 2004](#), p. 10-11).

Um benefício adicional do sistema é sua aplicabilidade no contexto de aprendizado, oferecendo suporte a disciplinas que envolvem a submissão e correção de trabalhos de programação ([CAMPOS; FERREIRA, 2004](#), p. 2). No entanto, ele foi utilizado por alunos de Ciência da Computação na URI por algum tempo, em sala de aula, para resolver problemas relacionados aos conteúdos abordados. Contudo, a cada semestre, era necessário realizar uma limpeza no site e registrar novos problemas, incluindo todos os seus arquivos de entrada e saída, enunciado, e tempos

limites, dependendo da matéria ensinada ([BEZ; TONIN, 2012](#), p. 1). Os autores [Bez e Tonin \(2012\)](#), p. 1. destacaram que lidar com essa quantidade de problemas não é viável, pois o sistema não foi criado para esse propósito. Cada vez que o sistema era limpo, tanto a classificação dos problemas quanto a classificação dos usuários eram perdidas.

Assim, apesar de o BOCA não precisar de cadastro em nenhum portal, e ainda ter acesso a todo código fonte dos exercícios, ele não é a opção ideal para ser o sistema de juiz online utilizado em sala de aula. Ainda assim, o sistema revela-se uma escolha excelente para competições de programação. O crescimento extraordinário dessas competições tem despertado um interesse crescente entre os alunos de cursos de computação no Brasil. Esse interesse é extremamente positivo, pois motiva os alunos a estudarem conceitos fundamentais da computação, tais como estruturas de dados, análise de algoritmos, técnicas e linguagens de programação ([CAMPOS; FERREIRA, 2004](#), p. 11).

2.4 VPL

O Virtual Programming Lab (VPL) é um módulo de atividades para o Moodle que gerencia tarefas de programação ([VPL, 2021](#)). Suas características distintas incluem a capacidade de editar o código-fonte diretamente no navegador, permitindo aos alunos executar programas de forma interativa sem sair do ambiente Moodle. Tanto alunos quanto professores podem realizar testes para revisar os programas, enquanto a funcionalidade de pesquisa de similaridade entre arquivos auxilia na detecção de plágio. Além disso, o VPL oferece a possibilidade de definir restrições de edição e prevenir a colagem de texto externo, contribuindo para um ambiente acadêmico mais controlado e transparente ([VPL, 2021](#)).

Segundo [Rodríguez-Del-Pino, Royo e Figueroa \(2012](#), p. 1), o VPL é um ambiente de desenvolvimento simples para os alunos, oferecendo recursos de avaliação automática. Para os instrutores, funciona como um sistema de gerenciamento do trabalho dos alunos, facilitando a preparação de tarefas, gerenciando envios, verificando plágio e conduzindo avaliações eficientes e flexíveis com base em testes de programas. Além disso, é independente da linguagem de programação utilizada nos exercícios e prioriza questões críticas de segurança.

Para criar uma atividade VPL, é necessário preencher um formulário de configuração básica, seguindo o padrão comum a outros módulos do Moodle. Esse formulário abrange elementos essenciais, como nome da atividade, breve descrição, período de disponibilidade (com datas de início, término e visibilidade), opções de avaliação, agrupamento, entre outros. Adicionalmente, a atividade VPL permite a inclusão de dados específicos, como o número máximo de arquivos a serem enviados, o tamanho máximo desses arquivos, bem como restrições relacionadas à edição, rede, senha, entre

outros ([VPL, 2021](#)).

Ao concluir o preenchimento do formulário de configuração básica, o criador da tarefa tem a possibilidade de ajustar cinco grupos adicionais de recursos: descrição completa, casos de teste, opções, arquivos solicitados e recursos avançados. Assim, caso o professor queira que os códigos submetidos pelos alunos sejam testados, ele mesmo precisa inserir todos os casos de teste manualmente na tarefa ([VPL, 2021](#)).

Assim, sempre que o professor desejar apresentar novas tarefas de programação para suas turmas a cada semestre, será necessário criar uma nova atividade utilizando o VPL, preenchendo integralmente o formulário de configuração básica e incluindo os casos de teste necessários.

2.5 CODERUNNER

O CodeRunner é um plugin de código aberto gratuito para o Moodle, projetado para avaliar respostas de programação em várias linguagens. Utilizado principalmente em cursos de programação de computadores, permite que os alunos enviem código em resposta a perguntas específicas e visualizem os resultados dos testes imediatamente. Os professores podem executar programas para avaliar as respostas dos alunos, utilizando uma abordagem adaptativa ([CODERUNNER, 2023](#)). Assim, os alunos desenvolvem e testam seu código usando um ambiente de desenvolvimento normal e enviam o código ao CodeRunner por meio de um navegador da Web somente quando acreditam que ele está correto ([LOBB; HARLOW, 2016](#), p. 47).

Funcionando como um tipo de pergunta padrão no Moodle, o CodeRunner foi desenvolvido para integrar-se facilmente com outros tipos de perguntas computadorizadas, como múltipla escolha, numérica, resposta curta, correspondência e até mesmo questões de redação avaliadas por humanos ([LOBB; HARLOW, 2016](#), p. 48).

Ao contrário do VPL, o CodeRunner opera no modo de teste adaptativo, onde os alunos podem verificar imediatamente se o código passa nos testes definidos na pergunta. Eles podem corrigir e reenviar o código, geralmente com uma pequena penalidade. A avaliação é baseada nos casos de teste aprovados pelo aluno ([MOODLE, 2023](#)).

Adicionalmente, o CodeRunner proporciona um sistema flexível de penalidades, permitindo que os criadores ajustem a avaliação conforme o contexto. Por exemplo, é possível configurar algumas perguntas sem penalidades para reenvios, outras que concedem um ou dois envios gratuitos antes de aplicar uma penalidade incremental de 10% para cada tentativa subsequente errada, e ainda outras que impõem uma penalidade de 100% mesmo para um único reenvio equivocado. Vale ressaltar a alta escalabilidade do CodeRunner: ele acomoda desde questões simples de codificação até tarefas significativamente complexas ([LOBB; HARLOW, 2016](#), p. 48).

Atualmente, o CodeRunner é compatível com Python2 (considerado obsoleto),

Python3, C, C++, Java, PHP, JavaScript (NodeJS), Octave e Matlab. A arquitetura permite a fácil expansão para outras linguagens ([MOODLE, 2023](#)).

Para garantir a segurança, o CodeRunner necessita do software sandbox ("Jobe") instalado em uma máquina separada com medidas adequadas de segurança. Recomenda-se o uso de um servidor Moodle separado para questionários baseados no CodeRunner, especialmente em testes e exames finais, tanto por motivos de carga quanto para que vários recursos de comunicação do Moodle, como bate-papo e mensagens, possam ser desativados sem afetar outras turmas ([CODERUNNER, 2023](#)).

Assim como no VPL, é necessário que o criador da pergunta configure seus detalhes e seus casos de teste ([CODERUNNER, 2023](#)).

A Universidade de Canterbury utiliza o CodeRunner para ministrar cursos de programação em Python, C, Octave e Matlab. Esse plugin é especialmente eficaz em cursos introdutórios de programação, proporcionando prática intensiva com pequenos problemas de programação que ensinam diversas construções e técnicas de linguagem. No entanto, o CodeRunner também é aplicável em níveis mais avançados, sendo utilizado em cursos teóricos de ciência da computação, inteligência artificial (com programação em Clojure) e programação na web para avaliação de sites desenvolvidos pelos alunos ([LOBB; HARLOW, 2016](#), p. 48).

2.6 MOODLE

“Moodle é uma plataforma de aprendizagem projetada para fornecer a educadores, administradores e alunos um único sistema robusto, seguro e integrado para criar ambientes de aprendizagem personalizados” ([MOODLE, 2023](#)). Ou seja, o Moodle é um Sistema de Gerenciamento de Aprendizagem (LMS) poderoso e altamente extensível ([MOODLE, 2023](#)).

Desenvolvido pelo projeto Moodle, a plataforma possui uma interface web amplamente acessível, permitindo que estudantes, tutores e administradores realizem tarefas diárias de qualquer lugar do mundo. Com suporte para várias opções técnicas, garante bom desempenho, sendo utilizado por instituições de destaque, como o Open Polytechnic da Nova Zelândia, que atende mais de 45.000 estudantes e oferece 6.500 cursos registrados ([MOODLE, 2023](#)).

Projetado e auditado para segurança, o Moodle conta com funções específicas para administradores, professores, alunos e visitantes, com um conjunto claro de privilégios. Comprometido com a segurança e privacidade, mantém controles atualizados contra acesso não autorizado, perda de dados e uso indevido. Pode ser implantado em nuvem privada segura ou servidor para controle total ([MOODLE, 2023](#)).

Reconhecido globalmente em ambientes de aprendizagem, está disponível em mais de 60 idiomas e é utilizado por instituições renomadas, incluindo London School of Economics, Shell e Microsoft, contando com mais de 213 milhões de usuários em

nível acadêmico e empresarial ([MOODLE, 2023](#)).

O Moodle oferece um conjunto robusto de ferramentas centradas no aluno e ambientes colaborativos para facilitar o ensino e aprendizagem. Com uma interface simples, é fornecido gratuitamente como software de código aberto sob a GNU General Public License, permitindo adaptação para projetos comerciais e não comerciais sem custos de licenciamento. Sua constante revisão atende às necessidades em evolução dos usuários e permite escalabilidade para turmas pequenas e grandes organizações ([MOODLE, 2023](#)).

Como software de código aberto, possibilita personalização e adaptação, com estrutura modular e design interoperável para o desenvolvimento de plugins e integração de aplicativos externos. Proporciona flexibilidade excepcional para apoiar a aprendizagem combinada e cursos totalmente online, configurável conforme necessário e integrando facilmente diversas ferramentas colaborativas ([MOODLE, 2023](#)).

Contando com forte apoio de uma comunidade global ativa, equipe de desenvolvedores dedicados e parceiros certificados, o Moodle recebe atualizações rápidas, correções de bugs e melhorias em lançamentos significativos a cada seis meses ([MOODLE, 2023](#)).

Atualmente, diversas instituições educacionais buscam modernizar seus sistemas de aprendizagem para facilitar o acesso ao conhecimento. Nesse contexto, o ambiente Moodle tem se destacado como uma escolha frequente. O Moodle adota o conceito construcionista social de educação, conforme destacado por ([GALASSO; MOREIRA, 2014](#), p. 22). Esse enfoque pedagógico enfatiza as demandas contemporâneas por métodos de ensino mais dinâmicos e participativos.

2.6.1 Características Gerais do Moodle

O Moodle, flexível e de fácil instalação em plataformas que suportam PHP, exige apenas uma base de dados, de qualquer marca. Destaca-se por sua ênfase na segurança, implementando verificações rigorosas em formulários, validação de dados, e codificação de cookies. Também permite que um site Moodle suporte milhares de cursos, estes podendo ser categorizados e pesquisados.

2.6.1.1 Administração do site

- O site é administrado por um usuário administrador, permitindo ajustes de aparência por meio de extensões (plug-ins) de temas;
- Oferece suporte a extensões de módulos de atividade e pacotes de idiomas, possibilitando compatibilidade com mais de 60 idiomas;
- O código PHP, licenciado sob GPL, é claro e facilmente modificado para atender às necessidades individuais.

2.6.1.2 Administração dos usuários

- Objetiva reduzir a intervenção do administrador, suportando diversos mecanismos de autenticação, como email, LDAP, IMAP, POP3, NNTP, e base de dados externa;
- Permite que cada pessoa possua uma única conta para todo o servidor, com diferentes acessos configuráveis;
- Uma conta de administrador controla a criação de cursos e cria professores através da inscrição de usuários aos cursos;
- Somente é permitida a uma conta de criador de cursos criar e dar aula nos cursos;
- Os professores podem acrescentar uma “chave de inscrição” a seus cursos para manter fora os não inscritos. Eles podem fornecer essa chave diretamente ou através do email particular de cada um;
- Os professores podem incluir e excluir alunos manualmente;
- Cada usuário pode especificar faixas de horário para que cada compromisso seja ajustado a esses horários, e escolher o idioma de interface.

2.6.1.3 Administração do Curso

- Professores têm controle total sobre os parâmetros do curso, podendo restringir outros professores;
- Oferece flexibilidade na composição de atividades, como fóruns, jornais, questionários, recursos, pesquisas de opinião, tarefas e chats.
- Todas as notas para os fóruns, jornais, questionários e tarefas podem ser vistas em uma página;
- Rastreamento detalhado dos usuários, relatórios de atividade com gráficos, e história detalhada do envolvimento de cada aluno, como postagens.

2.6.2 LTI External tools

Ferramentas Externas LTI do Moodle são aplicativos adicionais que você pode integrar curso, como conteúdo interativo, atividades ou avaliações. Os professores podem vincular essas atividades diretamente na página do curso no Moodle, e os alunos podem acessá-las sem sair do curso no Moodle ou precisar fazer login em um sistema diferente. Onde disponível, e dependendo de cada ferramenta, os professores também podem fazer com que as notas sejam enviadas de volta ao Moodle ([MOODLE](#),

2023).

A conexão entre o Moodle e essas ferramentas é feita através do padrão Learning Tools Interoperability (LTI), que integra plataformas de aprendizado com ferramentas de aprendizado para criar uma experiência de aprendizado mais rica e contínua. Esencialmente, o padrão LTI permite uma troca segura e bidirecional de informações entre o Moodle e ferramentas de aprendizado externas ([MOODLE, 2023](#)).

O administrador tem permissão para gerenciar Ferramentas Externas LTI, podendo adicionar novas ferramentas aos seus cursos usando o botão 'Adicionar ferramenta' na página de Ferramentas Externas LTI ([MOODLE, 2023](#)).

As ferramentas adicionadas no nível do curso aparecerão na sua lista de atividades por padrão, sendo possível removê-las da lista de atividades acessando a página do curso > Mais > Ferramentas Externas LTI e desmarcando-as na coluna 'Mostrar na lista de atividades' ([MOODLE, 2023](#)).

Para configurar uma ferramenta externa LTI, é necessário ir até "Configurações da ferramenta", fornecer um nome para a ferramenta (e, opcionalmente, uma descrição). Para preencher os outros campos nesta seção, incluindo a versão LTI, deve-se seguir as instruções do fornecedor da sua ferramenta. Se a ferramenta não fornecer informações sobre algumas dessas configurações, você pode deixá-las em branco ([MOODLE, 2023](#)).

De acordo com [Moodle \(2023\)](#), os campos a serem preenchidos para configuração de uma ferramenta externa LTI são:

- URL da ferramenta – Esta é a URL para conectar ao site. Se o site Moodle usar SSL (estiver em HTTPS), só é possível usar uma ferramenta que também use SSL. É necessário certificar-se de que a URL da ferramenta tenha HTTPS antes de tentar usá-la;
- Versão LTI – A versão LTI da ferramenta que está sendo adicionada;
- Chave do consumidor – Isso informa ao site LTI compatível conectado que o Moodle está autorizado a se conectar. O "fornecedor da ferramenta", ou seja, o gerente do site LTI compatível conectado, fornecerá essa chave. Se o administrador estiver apenas vinculando a uma ferramenta sem acesso seguro ou compartilhamento de notas, não será necessário uma chave do consumidor. Se estiver vinculando a um curso ou atividade de outro site Moodle, o administrador pode adicionar qualquer chave do consumidor;
- Segredo compartilhado – Este é o "senha" para conectar à ferramenta – o site LTI compatível;
- Parâmetros personalizados – Na maioria das vezes, esse campo pode ficar em branco. O fornecedor da ferramenta pode usar isso para permitir que o adminis-

trador exiba um recurso específico;

- Contêiner de lançamento padrão – Esta é a forma como a ferramenta externa será exibida;
- Incorporar – O conteúdo aparecerá dentro da página de atividade onde a ferramenta é usada, sem afastar os usuários da página, mas ocultando quaisquer blocos que o curso possa ter;
- Incorporar sem blocos – O conteúdo será exibido na guia ou janela existente. Os usuários terão que navegar de volta para o curso usando o botão ‘Voltar’ assim que terminarem;
- Janela existente – O conteúdo será exibido na guia ou janela existente. Os usuários terão que navegar de volta para o curso usando o botão ‘Voltar’ assim que terminarem;
- Nova janela – A ferramenta externa será aberta em uma nova janela. (Uma nova janela ou guia será aberta com a ferramenta externa e a janela do navegador antiga contendo a página do curso não será alterada);
- URL de seleção de conteúdo – A URL de Seleção de Conteúdo será usada para abrir a página de seleção de conteúdo do fornecedor da ferramenta. Se estiver vazia, a URL da ferramenta será usada;
- URL do ícone – É possível exibir um ícone diferente do ícone padrão da Ferramenta Externa inserindo sua URL nesse campo;
- URL do ícone seguro – É possível inserir URL de um ícone diferente aqui se os alunos estiverem acessando o Moodle com segurança via SSL

2.7 LTI

A especificação Learning Tools Interoperability (LTI) 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; BOOTH, 2014](#), p.4).

Assim, o padrão LTI permite uma troca segura de dados entre o 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](#)).

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, Bratko e Rajkovic \(1990\)](#) definem sistemas especialistas como sistemas inteligentes de informação que imitam, em certo grau, o comportamento de um especialista humano no domínio em questão.

Também chamados de sistemas baseados em conhecimento, esses programas de computador armazenam o conhecimento específico de um ou mais especialistas humanos ([SINGLA, 2013](#)). Os principais componentes de um sistema especialista são a base de conhecimento e a máquina de inferência. A base de conhecimento armazena informações sobre um domínio específico de problemas, enquanto a máquina de inferência utiliza esse conhecimento para resolver problemas indicados pelo usuário, gerando explicações orientadas a ele sobre as soluções propostas ([BOHANEC; BRATKO; RAJKOVIC, 1990](#)).

Assim, a base de conhecimento contém o conhecimento do domínio necessário para resolver problemas, geralmente na forma de regras, um dos paradigmas mais populares para a representação de conhecimento; e o motor de inferência é o núcleo do sistema, responsável por gerar conclusões a partir da base de conhecimento através de inferência ou raciocínio ([SINGLA, 2013](#)). Portanto, um sistema especialista para tomada de decisão precisa estabelecer uma base de conhecimento apropriada e usá-la para o cálculo de utilidade ([BOHANEC; BRATKO; RAJKOVIC, 1990](#)).

O autor [Singla \(2013\)](#) ressaltou que as principais características de um sistema especialista incluem interface com o usuário, representação de dados, inferência, capacidade de explicação e habilidade de lidar com incertezas. Entre as vantagens, esse tipo de sistema oferece resposta rápida, maior confiabilidade, redução de custos e de erros, uso de múltiplas expertises, banco de dados inteligente e mitigação de riscos. Contudo, o sistema apresenta algumas limitações, como a ausência de senso comum, a incapacidade de responder a casos excepcionais e a dificuldade de adaptação a ambientes em constante mudança ([SINGLA, 2013](#)).

2.8.0.1 Prolog

De acordo com [Clark, McCabe e McCabe \(1980\)](#), o Prolog, uma linguagem de programação baseada na lógica de predicados proposta por Kowalski, foi desenvolvida pelo grupo de Alain Colmerauer em Marselha. Sua implementação mais conhecida é o compilador DEC-10 de Edimburgo, que apresenta eficiência comparável à de LISP, mas com maior suporte a inferências, essencial em sistemas especialistas. Em Marselha, é aplicada no processamento de linguagem natural, enquanto em Edimburgo é usada para planejamento, resolução de problemas e desenvolvimento de compiladores ([CLARK; MCCABE; MCCABE, 1980](#)).

Clark, McCabe e McCabe (1980) também explicam que, na Hungria, a linguagem de programação Prolog é amplamente utilizada em sistemas especialistas, especialmente na farmacologia, incluindo previsão de atividades biológicas e interações medicamentosas. Experiências como um sistema de diagnóstico de falhas no Imperial College demonstram o potencial de Prolog para sistemas especialistas, incentivando novas aplicações (CLARK; MCCABE; MCCABE, 1980).

Já Russell e Norvig (2013) mostram que o Prolog é, de longe, a linguagem de programação lógica mais amplamente utilizada, sendo popular para prototipação rápida e manipulação simbólica, como no desenvolvimento de compiladores e análise de linguagem natural. Além disso, diversos sistemas especialistas foram implementados em Prolog para áreas como direito, medicina (SINGLA, 2013), finanças e outros domínios especializados (RUSSELL; NORVIG, 2013).

Além disso, o Prolog é ideal para desenvolver serviços Web robustos devido à sua semântica segura e gerenciamento automático de memória (WIELEMAKER; HUANG; MEIJ, 2008).

Programas em Prolog consistem em conjuntos de cláusulas escritas com a seguinte notação: as variáveis são indicadas por letras maiúsculas, enquanto as constantes são escritas em letras minúsculas; e a estrutura das cláusulas, ao invés de A B C, utiliza-se C :- A, B, com vírgulas separando os elementos da conjunção (RUSSELL; NORVIG, 2013).

2.8.0.2 SWI-Prolog

O SWI-Prolog é uma implementação de Prolog baseada em um subconjunto da WAM (Warren Abstract Machine), que possui um ambiente robusto de desenvolvimento, priorizando compatibilidade, portabilidade, escalabilidade e estabilidade (WIELEMAKER, 2012).

O SWI-Prolog, projeto de código aberto, foi moldado para desenvolver protótipos de pesquisa, especialmente em sistemas interativos e que exigem intensivo uso de conhecimento (WIELEMAKER; SCHRIJVERS *et al.*, 2012). Além do sistema básico, foram integrados ao SWI-Prolog várias interfaces e bibliotecas de restrições. Assim, o SWI-Prolog atua como uma ferramenta integradora e versátil, permitindo a interação com diversos recursos externos e apoioando inovações da comunidade Prolog (WIELEMAKER; SCHRIJVERS *et al.*, 2012).

Como o SWI-Prolog inclui uma extensa gama de biblioteca, que permite integração com gráficos (XPCE), bancos de dados, redes, outras linguagens de programação e diferentes formatos de documentos, ele é uma escolha poderosa para o desenvolvimento de aplicações que requerem integração e lógica sofisticada, mantendo uma estrutura de código organizada e eficiente (WIELEMAKER, 2014, p. 1).

Outra vantagem do SWI-Prolog é a disponibilidade de bibliotecas para servi-

dor HTTP, o que facilita o desenvolvimento de controles de servidor (WIELEMAKER; LAGER; RIGUZZI, 2015). Essas bibliotecas foram projetadas para que o Prolog se comunique com outros componentes de aplicações Web por meio do protocolo HTTP padrão. Com isso, a arquitetura é capaz de gerenciar protocolos de transferência e realizar parsing, representação e geração dos principais formatos de documentos Web, como HTML, XML e RDF, de forma integrada e padronizada (WIELEMAKER; HUANG; MEIJ, 2008).

2.9 WEB API

Os Web Services (Serviços da Web) são servidores da Web dedicados a propósitos específicos, atendendo às demandas de um site ou de outros aplicativos. Para interagir com esses serviços, os programas clientes utilizam Interfaces de Programação de Aplicações (APIs). Em essência, uma API expõe conjuntos de dados e funções, facilitando a comunicação entre programas de computador e viabilizando a troca de informações. Uma API da Web representa a interface de um serviço da Web, sendo responsável por receber e responder diretamente às solicitações dos clientes (MASSÉ, 2011, p. 5).

A interação com uma Web API envolve a realização de solicitações (requests) por meio dos protocolos de comunicação HTTP ou HTTPS. Essas solicitações são essenciais para a comunicação entre programas clientes e serviços da Web, permitindo a troca de informações de maneira estruturada (RICHARDSON; AMUNDSEN; RUBY, 2011, p. 18-20).

Quando um programa cliente envia uma solicitação para uma Web API, ele está requisitando a execução de uma operação específica ou a obtenção de dados do serviço da Web. Essa solicitação é processada pelo serviço da Web, que fornece uma resposta (response) ao cliente. O formato comum para a estruturação dessas respostas é o JSON (JavaScript Object Notation), um padrão amplamente adotado para representar estruturas de dados (RICHARDSON; AMUNDSEN; RUBY, 2011, p. 19-21).

Essa dinâmica de solicitação e resposta é a base do funcionamento de uma Web API. As solicitações podem conter parâmetros que informam ao serviço da Web sobre a operação desejada, e a resposta contém os dados solicitados ou uma confirmação da operação realizada. Esse modelo facilita a integração de sistemas e a comunicação entre diferentes componentes de software, sendo fundamental em ambientes de desenvolvimento web e integração de aplicativos (RICHARDSON; AMUNDSEN; RUBY, 2011, p. 20-23).

3 TRABALHOS RELACIONADOS

Neste capítulo, inicialmente é apresentado um artigo que descreve um modelo de plano de ensino baseado na plataforma Beecrowd, destacando suas justificativas e aspectos positivos. Em seguida, são analisados quatro artigos sobre a integração de diferentes ambientes de desenvolvimento com o Moodle: BOCA, CodeRunner, VPL, e um estudo sobre a integração de juízes online em geral com o Moodle. Por fim, são discutidos dois estudos focados na criação de sistemas especialistas web, explorando suas características e abordagens.

3.1 UTILIZAÇÃO DA PLATAFORMA BEECROWD DE MARATONA DE PROGRAMAÇÃO COMO ESTRATÉGIA PARA O ENSINO DE ALGORITMOS

Neste estudo conduzido por ([CRUZ et al., 2022](#)), é apresentada uma proposta de plano de ensino que se baseia no aproveitamento da plataforma Beecrowd para maratonas de programação, alinhada a abordagens pedagógicas ativas. Dessa maneira, são integrados conceitos como sala de aula invertida e gamificação, proporcionando uma abordagem dinâmica e envolvente no processo de aprendizagem e explorando as potencialidades da plataforma Beecrowd para oferecer uma experiência educacional inovadora.

Os autores têm como objetivo aplicar essa metodologia de maneira a diminuir a curva de aprendizagem dos alunos, apoiar os professores na implementação de abordagens pedagógicas ativas em sala de aula e ampliar as oportunidades de interação entre os alunos, contribuindo assim para o desenvolvimento do aprendizado.

Os autores destacam a gamificação e a sala de aula invertida como estratégias que integram métodos de ensino com diversos recursos de comunicação, incluindo materiais escritos, orais e audiovisuais, junto a atividades, desafios e informações contextualizadas. A gamificação é apresentada como a aplicação de elementos característicos de jogos, uma abordagem eficaz para transformar a forma como os estímulos são apresentados aos alunos, visando engajá-los e incentivá-los a resolver problemas da maneira mais eficiente possível.

De maneira semelhante, a sala de aula invertida é descrita como uma proposta para dinamizar a transmissão de informações durante as aulas, utilizando ferramentas online estruturadas e bem planejadas. Essa abordagem visa incentivar a iniciativa dos alunos ao estudarem previamente determinados conteúdos antes das aulas ministradas pelos professores.

Os autores ressaltaram a importância da disciplina de algoritmos como a base fundamental para o ensino de programação em diversos cursos na área da Informática. Essa disciplina aborda princípios essenciais de lógica de programação, visando desenvolver a capacidade dos estudantes em analisar e resolver problemas por meio

da criação de algoritmos. Contudo, enfrenta desafios significativos, evidenciados pelo elevado índice de evasão e reprovação, tornando-se um ponto crítico nos cursos de graduação e impactando a continuidade dos alunos.

A complexidade reside, em grande parte, na diversidade de alunos e em seus distintos estilos e ritmos de aprendizado, conforme indicado pelos autores. A adaptação eficiente a essa heterogeneidade representa um desafio no ensino de algoritmos e programação para novos estudantes. Nesse contexto, os autores propõem que a utilização de ferramentas online amplamente disponíveis para a aprendizagem ativa pode ser uma estratégia eficaz para permitir que cada aluno progride em seu próprio ritmo e velocidade ([CRUZ et al., 2022](#), p. 5).

No que diz respeito à integração de questões de maratonas de programação em sala de aula, os autores destacam sua relevância ao promover a busca ativa pela resolução de problemas. Ao buscar soluções, os alunos não apenas adquirem conhecimento sobre tópicos ainda não abordados em aula, mas também consolidam conceitos previamente ensinados.

O Beecrowd se destaca como uma ferramenta online que não apenas viabiliza a resolução de problemas em competições de programação, mas também se apresenta como uma valiosa aliada para o trabalho do professor. Os autores do estudo evidenciam que os desafios oferecidos pela plataforma Beecrowd estãometiculosamente organizados em categorias e níveis de dificuldade. As categorias abrangem temas como Iniciante, AD-HOC, Strings, Estruturas e Bibliotecas, Matemática, Paradigmas, Grafos, Geometria Computacional e SQL. A escala de dificuldade varia de 1 a 10, sendo 1 as questões mais acessíveis e 10 as mais desafiadoras.

No âmbito do artigo analisado, foram selecionadas questões da categoria Iniciante, e a partir delas, elaborou-se um plano de ensino voltado para a implementação da metodologia ativa da maratona de programação. Este plano foi aplicado em seis semestres na disciplina de algoritmos, envolvendo alunos que, em sua maioria, não possuíam conhecimento prévio em programação. A avaliação da proposta foi conduzida ao longo de três semestres que adotaram as metodologias ativas e outros três semestres que não as utilizaram, de forma intercalada.

Através desse experimento e da análise de seus resultados, os autores constataram que a aplicação das metodologias de gamificação e sala de aula invertida contribuiu significativamente para o desempenho da turma, observando-se um crescimento notável. Além disso, as notas mais elevadas foram mais consistentes ao longo do tempo, demonstrando a eficácia dessas abordagens no contexto educacional.

Dessa forma, constatou-se que integrar o Beecrowd como um suporte para a implementação da gamificação como metodologia ativa na disciplina de algoritmos foi eficaz. A adoção do formato de maratona de programação possibilita ao professor envolver os alunos de forma mais imersiva, enquanto a plataforma oferece suporte adicional para o desenvolvimento da disciplina. Isso não apenas aprimora o processo

de ensino e aprendizagem, mas também contribui para a redução da evasão em cursos superiores de informática.

3.2 INTEGRAÇÃO DO AMBIENTE BOCA COM O AMBIENTE MOODLE PARA AVALIAÇÃO AUTOMÁTICA DE ALGORITMOS

O estudo de [Galasso e Moreira \(2014\)](#) explora a integração entre o sistema BOCA Online Contest Administrator e a plataforma Moodle, direcionada para o ensino de algoritmos. Essa integração visa proporcionar uma experiência simplificada para alunos e professores, facilitando o processo de avaliação automática de algoritmos no Moodle, já que o BOCA, além de gerenciar competições de programação, permitindo, assim, eventos como maratonas de programação, possui também um juiz automático para correção de algoritmos. Essa funcionalidade possibilita que os alunos recebam respostas imediatas sobre a correção de seus algoritmos, contribuindo para um aprendizado mais ágil e eficiente.

Na integração proposta, os professores podem criar questões de programação no Moodle, onde os alunos submetem códigos-fonte que são automaticamente avaliados pelo BOCA. Os algoritmos submetidos são compilados e executados com os dados de entrada predefinidos no Moodle, e suas saídas são comparadas com uma saída padrão para determinar se estão corretos. A resposta ao aluno é binária (correta ou incorreta), sem avaliação parcial, mas Moodle permite múltiplas tentativas, oferecendo a oportunidade de corrigir erros como problemas de compilação, formatação ou limite de tempo. Adicionalmente, o professor pode revisar manualmente os resultados, caso deseje realizar uma avaliação mais detalhada e parcial das respostas.

A figura 10 representa uma edição da tela de cadastro da questão, proporcionando uma visão ilustrativa desta fase do processo, já a figura 11 oferece uma representação visual da avaliação de uma questão como correta.

Para manter o desempenho do Moodle, a integração foi concebida de modo a garantir que os ambientes envolvidos fossem implementados em servidores independentes. Essa separação minimiza a possibilidade de sobrecarga no Moodle durante a avaliação dos algoritmos no BOCA. A integração foi desenvolvida com o PostgreSQL, e os bancos de dados do Moodle e do BOCA se comunicam via Dblink, um recurso que permite o acesso remoto a tabelas específicas.

As tabelas principais do BOCA envolvidas na avaliação automática incluem a “problemable”, para cadastro de questões, “runtable”, para submissões dos alunos, “answertable”, para dados de correção, e “langtable”, com as linguagens de programação suportadas (C, C++ e Java). Os autores também criaram duas tabelas auxiliares, “tempaluno” e “tempprofessor”, para facilitar a transferência de dados entre o Moodle e o BOCA.

Os gatilhos (triggers) configurados no banco de dados são fundamentais para o

Figura 10 – Integração BOCA-Moodle - Edição de tela de cadastro de questão

Adding a Juiz Online question

Geral

Categoria: Padrão para M.Programação (25)

Nome da pergunta*: Tobogan de bolinhas

Texto da questão*:

```
A primeira linha de um caso de teste contém um inteiro N indicando o número de aletas do brinquedo. A segunda linha contém dois inteiros L e H, indicando mais alta (a segunda a ser descrita) tem a extremidade ligada à haste direita, assim alternadamente.
```

'Input'.*: 9857 9482 9853
9794 4871 9790
9787 9871 9780
9779 7830 9777
9784 9177 9789

'Output'.*: 9.24
20.24
7.22
0.47
9.47

Para a resposta do aluno: Você deve definir uma linguagem de programação, para que o mesmo possa submeter seus algoritmos na linguagem de programação: * C++

Fonte: (GALASSO; MOREIRA, 2014, p. 26)

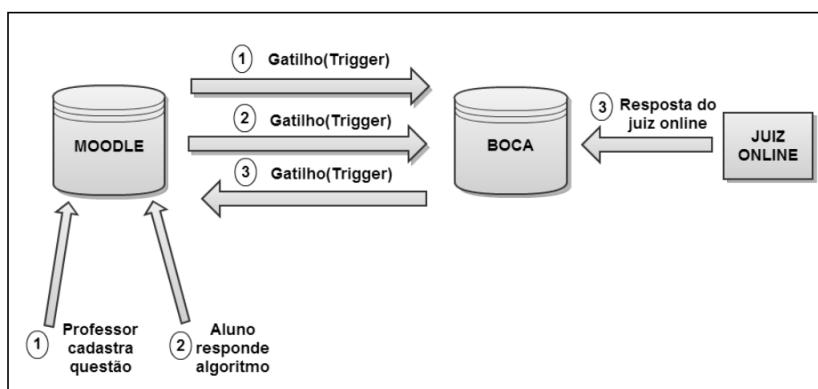
Figura 11 – Integração BOCA-Moodle - Visualização do retorno da avaliação da questão

Tentativa	Estado	Notas / 1,00	Nota / 100,00	Revisão	Comentários
Visualização prévia	Finalizadas Enviada(s) quarta, 17 julho 2013, 17:15	1,00	100,00	Revisão	Correto Algoritmo

Fonte: (GALASSO; MOREIRA, 2014, p. 27)

fluxo de dados entre Moodle e BOCA, eliminando a necessidade de alterações no código. O fluxo começa quando o professor cobra uma questão no Moodle, acionando gatilhos que transferem os dados para o BOCA, onde a questão é registrada. Da mesma forma, quando um aluno responde, gatilhos adicionais transferem as submissões para a tabela de execução no BOCA.

Figura 12 – Integração BOCA-Moodle - Fluxo da interação das bases de dados do Moodle e do BOCA



Fonte: (GALASSO; MOREIRA, 2014, p. 29)

Durante o julgamento das respostas, o BOCA executa scripts em intervalos de

um minuto, verificando novas submissões e atualizando o status no Moodle. Os alunos visualizam uma área verde para respostas corretas e vermelha para incorretas. Figuras incluídas no estudo ilustram etapas do cadastro das questões e o *feedback* visual ao aluno.

Figura 13 – Integração BOCA-Moodle - retorno da avaliação da questão como correta

Questão 1
Completo
Atingiu 1,00 de 1,00
Marcar questão
Editar questão

Uma fábrica quer produzir um tobogã de brinquedo como o da figura abaixo, composto de duas hastas de madeira sustentando aletas que se alternam nas duas hastas. Uma bolinha de aço é solta na aleta mais alta do tobogã, sob efeito da gravidade, a bolinha desliza pelas aletas, terminando por sair do brinquedo.

O projeto do brinquedo, contendo as especificações do tamanho, posição e inclinação das hastas e de cada aleta, foi feito pelo dono da fábrica, e milhares de unidades já estão sendo confeccionadas na China. O gerente da fábrica foi incumbido de comprar as bolinhas de aço, mas antes de fazer o pedido das milhares de bolinhas quer saber o diâmetro máximo da bolinha, para que esta não pare no meio do brinquedo.

Insira o seu Código fonte:

```
#include <stdio.h>
#include <math.h>

#include <algorithm>
#include <utility>

using namespace std;

const double EPS = 1e-10;
#define MAX 1000

int cmp(double x, double y = 0,
double tol = EPS) {
    return (x <= y + tol) ?
(x + tol < y) ? -1 : 0 : 1;
}
```

Linguagem de Programação definida pelo professor:

C++

Fonte: (GALASSO; MOREIRA, 2014, p. 26)

Houve desafios técnicos, como a tentativa inicial de enviar arquivos “zip” diretamente para o banco de dados, o que levou a problemas de codificação. Para contornar isso, os arquivos foram transferidos via FTP, necessitando a instalação de um servidor FTP no ambiente do BOCA. Ao final, os autores ressaltam a importância de uma instalação simplificada e recomendam uma versão do BOCA já configurada para integração com o Moodle. Essa abordagem visa facilitar a adoção da solução e sua eficácia no ensino de algoritmos.

3.3 CODERUNNER: A TOOL FOR ASSESSING COMPUTER PROGRAMMING SKILLS

O estudo de Lobb e Harlow (2016) examina as dificuldades encontradas na avaliação de alunos em disciplinas de programação utilizando métodos tradicionais, como provas de papel e caneta. O artigo destaca que a complexidade dos códigos e a natureza da programação moderna tornam a avaliação precisa um desafio, pois esses métodos não oferecem a possibilidade de realizar testes ou debug. O artigo

propõe o uso de ferramentas de avaliação automatizadas, como o CodeRunner, que proporciona correção e *feedback* imediato para os alunos.

Pelo CodeRunner, plugin do Moodle, os alunos recebem um *feedback* binário, onde as respostas corretas são indicadas por marcas verdes e as incorretas por cruzes vermelhas. A ferramenta adota o método de avaliação "tudo ou nada", ou seja, uma resposta só é considerada correta se atender completamente aos requisitos do teste, sendo a nota zero atribuída para respostas incorretas. Contudo, o sistema permite que os alunos façam múltiplas tentativas, oferecendo a chance de corrigir falhas como erros de compilação, formatação inadequada ou problemas de tempo de execução, com a possibilidade de aplicar penalidades em caso de envio tardio ou repetido.

Figura 14 – CodeRunner - A simples pergunta sobre Python, respondida erroneamente

Question 2
Incorrect
Mark 0.00 out of 1.00
[Flag question](#) [Edit question](#)

Write a Python function `squares(n)` that returns a list of the squares of all integers from 1 to n inclusive. Returns the empty list if n < 1.

For example:

Test	Result
<code>print(squares(5))</code>	[1, 4, 9, 16, 25]

Answer:

```
1- def squares(n):
2     return [i * i for i in range(n)]
```

Check

	Test	Expected	Got
✗	<code>print(squares(5))</code>	[1, 4, 9, 16, 25]	[0, 1, 4, 9, 16]
✗	<code>print(squares(1))</code>	[1]	[0]
✗	<code>print(squares(7))</code>	[1, 4, 9, 16, 25, 36, 49]	[0, 1, 4, 9, 16, 25, 36]
✓	<code>print(squares(0))</code>	[]	[]
✓	<code>print(squares(-10))</code>	[]	[]
✗	<code>print(squares(2))</code>	[1, 4]	[0, 1]

Some hidden test cases failed, too.
Your code must pass all tests to earn any marks. Try again.

Incorrect
Marks for this submission: 0.00/1.00. This submission attracted a penalty of 0.10.

Fonte: (LOBB; HARLOW, 2016, p. 48)

O CodeRunner também apresenta uma abordagem flexível para a criação de perguntas, permitindo que os desenvolvedores de questões criem protótipos personalizados. Esses protótipos podem ser usados para perguntas que exigem funcionalidades adicionais ou para impor restrições, como o uso de determinado tipo de estrutura de controle no código. Além disso, a execução do código do aluno ocorre em um ambiente seguro, isolado do servidor principal, o que evita riscos de segurança.

Apesar de o CodeRunner ter se revelado altamente eficaz em diversas atividades

Figura 15 – CodeRunner - A simples pergunta sobre Python, respondida corretamente

Test	Expected	Got
print(squares(5))	[1, 4, 9, 16, 25]	[1, 4, 9, 16, 25]
print(squares(1))	[1]	[1]
print(squares(7))	[1, 4, 9, 16, 25, 36, 49]	[1, 4, 9, 16, 25, 36, 49]
print(squares(0))	[]	[]
print(squares(-10))	[]	[]
print(squares(2))	[1, 4]	[1, 4]

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives 0.90/1.00.

Fonte: ([LOBB; HARLOW, 2016](#), p. 48)

de avaliação, os autores destacaram algumas limitações fundamentais:

- O CodeRunner destaca-se em tarefas relativamente simples com especificações claras. Embora, teoricamente, qualquer pergunta com resposta mensurável por um programa de computador possa ser formulada como uma pergunta do CodeRunner, o esforço demandado para criar avaliadores para tarefas complexas muitas vezes inviabiliza essa abordagem, especialmente em turmas de maior tamanho;
- Ferramentas de qualidade de código, como o pylint, provaram ser muito valiosas para aumentar a conscientização sobre o estilo e melhorar a qualidade do código, mas ainda ocorrem abusos das regras de estilo, e a qualidade dos comentários e identificadores não pode ser avaliada por computador. Por isso, a avaliação humana ainda é reservada para a qualidade do código;
- A resposta a uma pergunta do CodeRunner deve ser apresentada como um único bloco de texto, predominantemente composto por código. Embora os autores das perguntas possam gerar diversos formatos de *feedback*, inclusive gráficos, a apresentação padrão dos resultados é tabular, com uma linha para cada caso de teste e correspondência direta entre a saída esperada e a saída real. Caso

os testes exijam blocos extensos de código ou resultem em saídas volumosas, a visualização dos resultados pode dificultar a compreensão pelos alunos do motivo pelo qual uma resposta foi marcada como incorreta;

- (d) A avaliação de perguntas com resultados gráficos representa um desafio. Foi criado uma simulação do kit de ferramentas GUI tkinter em Python para avaliar interfaces gráficas na disciplina, incluindo perguntas que verificam a precisão de gráficos gerados por chamadas à biblioteca de gráficos do Matlab. Contudo, a avaliação da correção de imagens ou mesmo da saída de programas que envolvem gráficos complexos, como os da biblioteca de tartarugas, foi considerada uma tarefa difícil, senão impossível;
- (e) Por último, é importante ressaltar que a elaboração de perguntas de qualidade e testes eficazes pode demandar considerável tempo, mesmo em cursos de programação de nível básico. O compartilhamento de bancos de dados de perguntas entre professores seria altamente benéfico nesse contexto.

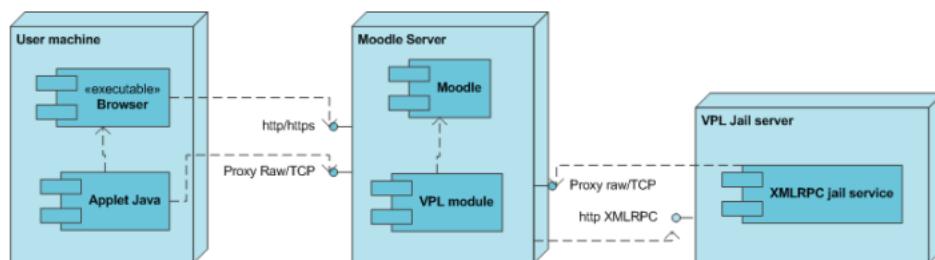
3.4 A VIRTUAL PROGRAMMING LAB FOR MOODLE WITH AUTOMATIC ASSESSMENT AND ANTI-PLAGIARISM FEATURES

[Rodríguez-Del-Pino, Royo e Figueroa \(2012\)](#) descreveram neste artigo o módulo Virtual Programming Lab (VPL) para o Moodle, desenvolvido na Universidade de Las Palmas de Gran Canaria (ULPGC), e lançado para uso gratuito sob a licença GNU/GPL.

Para empregar uma linguagem de programação específica no VPL, basta garantir que o compilador correspondente esteja devidamente instalado no sistema de execução. Atualmente, estão disponíveis sistemas de execução com instalações para diversas linguagens, incluindo Ada, C, C, C#, FORTRAN, Haskell, Java, Octave, Pascal, Perl, PHP, Prolog, Python, Ruby, Scheme, SQL e VHDL.

O VPL é composto por três elementos: um módulo do Moodle, um editor de código baseado em navegador e um componente Jail, mostrados na figura 16.

Figura 16 – VPL - Componentes



Fonte: ([RODRÍGUEZ-DEL-PINO; ROYO; FIGUEROA, 2012](#), p. 2)

O editor de código, construído em Java, possibilita a edição, execução, depuração e avaliação de programas em um ambiente simples de desenvolvimento. Já o módulo do Moodle oferece características típicas, como backup, integração com o livro de notas e controle de eventos, além de recursos específicos, como gerenciamento de submissões, suporte à avaliação e prevenção contra plágio. E o componente Jail é um servidor que compila e executa códigos de alunos em um ambiente seguro, usando o comando linux chroot para restrições de leitura.

O VPL permite configurar, gerenciar e avaliar atividades de aprendizagem, classificadas por tipo (exemplos, exercícios de cloze, exercícios de desenvolvimento de código) e escopo (tarefas fora da sala de aula ou exames em sala).

A criação de uma atividade VPL inicia-se com o preenchimento do formulário de configuração básica no Moodle, incluindo nome, descrição, período de disponibilidade, opções de avaliação e outras configurações. Além disso, é possível especificar detalhes como número máximo de arquivos, tamanho máximo, restrições de edição, rede e senha. Veja na imagem 18.

Figura 17 – VPL - Configuração básica de restrições

Submission restrictions	
Maximum number of files	2
Submission by means of restricted Code Editor*	No
This activity acts as example*	No
Maximum upload file size*	Select
Password*	<input type="password"/>
<input type="checkbox"/> Unmask Allowed submission from net*	

Fonte: ([RODRÍGUEZ-DEL-PINO; ROYO; FIGUEROA, 2012](#), p. 3)

Após preencher o formulário básico, o instrutor pode ajustar cinco grupos adicionais de recursos na ferramenta VPL. A guia "Descrição completa" permite fornecer detalhes sobre o problema a ser resolvido. Em "Casos de teste", é possível configurar testes com descrição, entrada, saída esperada e penalizações em caso de falha. A guia "Options" configura aspectos gerais, como a base da atividade, permissões dos alunos e se os resultados automáticos contam para a nota final. "Arquivos solicitados" determina os nomes obrigatórios dos arquivos a serem enviados.

Essas configurações são suficientes, mas a guia "Advanced" oferece recursos adicionais para testes mais elaborados, como testes de funcionalidade avançados e outros tipos de teste, possibilitando uma avaliação abrangente da qualidade do código.

A figura 19 mostra um exemplo de configuração de casos de teste, e a figura 20 mostra a tela de configuração de casos de teste mais avançados.

O VPL oferece avaliação automática e assistida por computador, permitindo a personalização através das opções na guia correspondente. É essencial configurar os testes para respaldar a avaliação. O VPL realiza automaticamente os testes configurados, fornecendo um relatório com testes falhos, comentários explicativos e uma

Figura 18 – VPL - Exemplo de configuração de casos de teste

```
vpl_evaluate.cases
1 case = Divisible por 4 pero no por 100
2 input = 2012
3 output = 2012 es bisiesto
4 grade reduction = 25%
5 case = Divisible por 4 y por 100, pero no por .
6 input = 1900
```

Fonte: ([RODRÍGUEZ-DEL-PINO; ROYO; FIGUEROA, 2012](#), p. 3)

Figura 19 – VPL - Arquivos de execução para testes avançados

Execution

This script prepares the submitted program execution

This script prepares the submitted program debug

This script evaluates the submitted program

Write here cases of execution to evaluate the submitted program

File	Name	Edit	Rename	Delete
File 1	test_case_unit.adb	<input type="button" value="Edit"/>	<input type="button" value="Rename"/>	<input type="button" value="Delete"/>
File 2	test_case_unit.ads	<input type="button" value="Edit"/>	<input type="button" value="Rename"/>	<input type="button" value="Delete"/>
File 3	test_main.adb	<input type="button" value="Edit"/>	<input type="button" value="Rename"/>	<input type="button" value="Delete"/>
File 4	pruebas.txt	<input type="button" value="Edit"/>	<input type="button" value="Rename"/>	<input type="button" value="Delete"/>
File 5	evaluate.sh	<input type="button" value="Edit"/>	<input type="button" value="Rename"/>	<input type="button" value="Delete"/>

Fonte: ([RODRÍGUEZ-DEL-PINO; ROYO; FIGUEROA, 2012](#), p. 4)

proposta de nota. O avaliador humano tem a flexibilidade de ajustar o relatório, excluindo ou adicionando comentários, reutilizando itens da lista e recalculando a nota conforme necessário.

O plágio é abordado pelo VPL através de uma ferramenta de verificação de plágio no código-fonte. Essa ferramenta procura identificar plágio entre os envios de uma tarefa em um curso, considerando também fontes como envios anteriores ou tarefas similares de outros cursos. O processo de identificação de semelhanças entre os arquivos de origem envolve três etapas: tokenização, comparação e clusterização.

A tokenização visa obter uma assinatura normalizada para facilitar a comparação eficiente, envolvendo análise lexical, filtragem e normalização. Esse processo cria uma assinatura do programa, que é uma representação normalizada dos arquivos de código-fonte de um usuário, otimizando o processo de comparação, permitindo a detecção eficaz de plágio.

Em resumo, o Virtual Programming Lab (VPL) apresenta-se como uma ferramenta abrangente para o gerenciamento de atividades de programação. Com recursos

que vão desde a configuração detalhada de atividades no Moodle até a execução de testes avançados e a detecção de plágio, o VPL oferece uma variedade de funcionalidades para apoiar a aprendizagem prática e avaliação de habilidades de programação. Destaca-se, além disso, a necessidade de configurar todos os exercícios e seus casos de teste, para disponibilização destes aos alunos.

3.5 UMA FERRAMENTA BASEADA EM JUÍZES ONLINE PARA O APOIO ÀS ATIVIDADES DE PROGRAMAÇÃO DE COMPUTADORES NO MOODLE

No projeto conduzido por [Chaves et al. \(2013\)](#), foi desenvolvido o Módulo de Integração com Juízes Online (MOJO), ferramenta que integra os Juízes Online, como o Timus Online Judge e o Beecrowd (conhecido como URI Online Judge na época) ao Moodle, para evitar que os professores de disciplinas de programação ficassem sobre-carregados ao ter que elaborar, submeter, abalizar e fornecer o *feedback* necessário das questões aos alunos. O MOJO automatizou o processo de Elaboração, Submissão e Avaliação (ESA) das atividades de programação, fornecendo um ambiente unificado que facilita o acompanhamento dos alunos e amplia o acesso a um maior número de questões.

O MOJO, portanto, integra o Moodle com os Juízes Online para atividades de programação, sem interação direta entre as plataformas. Ele automatiza o processo ESA (Elaboração, Submissão e Avaliação), permitindo que professores submetam questões no Moodle, estudantes enviem suas respostas, e os Juízes Online avaliem automaticamente o código, com os resultados retornando ao Moodle para análise de professores e alunos. O professor pode monitorar o progresso dos estudantes e acessar os códigos submetidos, enquanto o Repositório de Integração armazena as questões para uso futuro.

Na arquitetura geral do MOJO, o Moodle gerencia a interface e o controle das atividades de programação, enquanto o MOJO se encarrega da comunicação e interação com os Juízes Online. A arquitetura conta com o Módulo Principal (MOP), responsável pelas funcionalidades essenciais e pelo controle do Módulo de Carga e Atualização (MOCA), que, por sua vez, realiza o carregamento inicial e atualizações constantes das questões, armazenando-as no Repositório de Integração. Esse repositório mantém as questões e resultados para uso e acompanhamento contínuo no Moodle.

Como os juízes online não oferecem APIs ou serviços Web, foram necessárias implementações específicas para cada juiz, utilizando PHP (a mesma linguagem do Moodle) e JavaScript para a interface. O banco de dados PostgreSQL foi escolhido por ser open-source e por permitir a criação de tabelas personalizadas para armazenar dados dos alunos, viabilizando um *feedback* direcionado e adequado.

O objetivo do MOJO é reduzir a carga de trabalho dos professores no processo ESA, permitindo um acompanhamento mais próximo dos alunos. Em uma pesquisa

inicial com sete professores, os resultados apontaram que 71,4% acreditam que a ferramenta pode economizar tempo; 28,6% gostariam de poder criar e avaliar suas próprias atividades automaticamente; 42,8% mencionaram questões de legibilidade do código; 28,6% observaram a clareza dos resultados; 85,7% valorizam o feedback mais rápido aos alunos; e 100% acreditam que a redução de tarefas possibilitará um acompanhamento mais eficaz, especialmente para alunos com dificuldades.

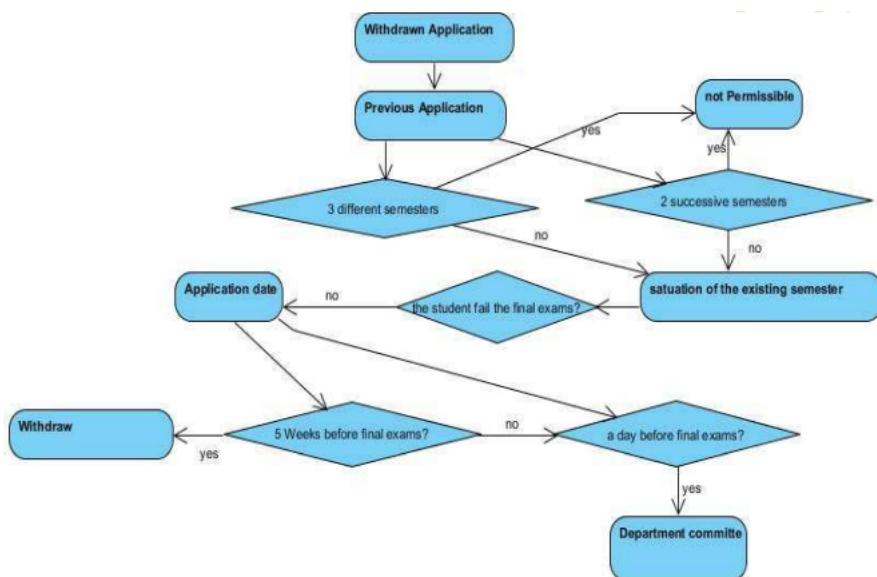
3.6 BUILDING OF A RULE-BASED EXPERT SYSTEM FOR ACADEMIC ADVISING VIA WEB EXPERT SYSTEM TOOLS

O projeto conduzido por [NASR, TALAB e AL-GAHTANI \(2018\)](#) aborda o desenvolvimento de um sistema especialista voltado para otimizar o aconselhamento acadêmico na Universidade King Khalid. A proposta central do estudo é criar uma ferramenta que simule as funções do conselheiro acadêmico humano, utilizando técnicas de inteligência empresarial para auxiliar na tomada de decisões informadas por estudantes e funcionários, baseadas nas regulamentações acadêmicas da instituição.

O processo de desenvolvimento do sistema é dividido em várias fases, cada uma com objetivos específicos:

Fase de Análise: Esta etapa inicial é dedicada à coleta de informações sobre as questões mais recorrentes entre os estudantes, além de procedimentos acadêmicos frequentes, como matrícula em cursos, transferências de departamento e pedidos de suspensão. A análise dessas necessidades foi realizada por meio de entrevistas com estudantes e conselheiros acadêmicos, utilizando uma abordagem de árvore de decisão (UML), mostrada na Figura 20, para entender melhor os requisitos do sistema.

Figura 20 – Sistema especialista baseado em regras - Árvore de decisão



Fonte: ([DUNSTAN, 2013](#), p. 305)

Fase de Design: Com base na análise anterior, o design do sistema foi estruturado utilizando uma metodologia baseada em regras (rule-based). Esse método foi escolhido por sua adequação à modelagem dos processos de aconselhamento acadêmico, que envolvem uma lógica condicional do tipo "se-então". Durante essa fase, também foi projetada uma interface amigável, alinhada com o estilo visual e os padrões de interação do portal da universidade, garantindo que os usuários pudessem facilmente se adaptar ao sistema.

Fase de Implementação: A construção do sistema foi realizada utilizando o ES Builder, uma ferramenta de código aberto que facilita a criação de sistemas especialistas baseados em regras. O sistema foi projetado para ser acessível via plataforma web, o que possibilita seu uso por uma grande variedade de usuários, incluindo estudantes e conselheiros acadêmicos.

Fase de Testes e Avaliação: O sistema foi testado com diferentes cenários, utilizando tanto estudantes quanto conselheiros acadêmicos, a fim de validar sua eficácia. Os resultados indicaram que o sistema é capaz de fornecer respostas adequadas e de auxiliar no processo de aconselhamento de forma precisa e eficiente.

O estudo conclui que a implementação de sistemas especialistas no âmbito acadêmico tem um grande potencial para melhorar a eficiência do aconselhamento acadêmico, promovendo uma interação mais eficaz entre estudantes e conselheiros. Além disso, o sistema proposto pode se tornar uma ferramenta valiosa para a tomada de decisões informadas, não apenas para os estudantes, mas também para os docentes, que poderão fornecer orientações mais precisas e baseadas em dados. O sistema mostrou-se alinhado às expectativas dos usuários, que manifestaram interesse em expandir sua funcionalidade para cobrir mais áreas do aconselhamento acadêmico.

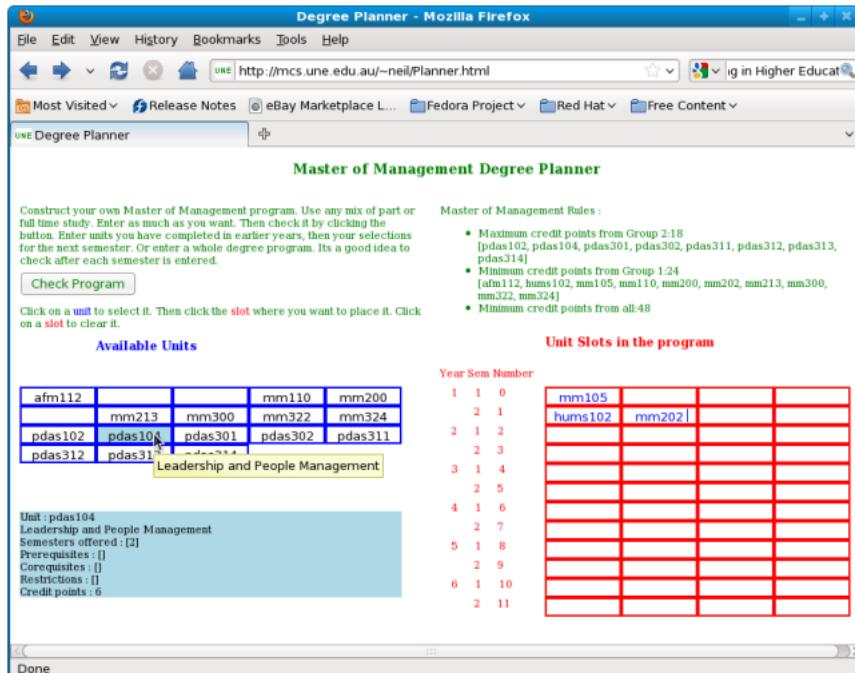
Este trabalho evidencia o papel transformador que a tecnologia pode desempenhar no setor educacional, especialmente no contexto do aconselhamento acadêmico, apontando para a necessidade de continuar a pesquisa e o desenvolvimento de sistemas dessa natureza para atender de forma mais abrangente às necessidades dos estudantes e professores.

3.7 AN INTERACTIVE WEB-BASED EXPERT SYSTEM DEGREE PLANNER

O artigo desenvolvido por [Dunstan \(2013\)](#) descreve o desenvolvimento de um planejador de grau acadêmico baseado na web (seu layout é mostrado na Figura 21), utilizando XML (Extensible Markup Language) e sistemas especialistas. O sistema possui um servidor que hospeda arquivos XML para cada grau acadêmico, descrevendo unidades e regras associadas. Essas regras são importadas para uma interface web genérica, permitindo a personalização do visual com uma paleta de unidades e horários disponíveis para os semestres. As regras do grau acadêmico em formato XML são convertidas em código Prolog, que permite ao sistema responder a consultas

complexas relacionadas ao planejamento de cursos.

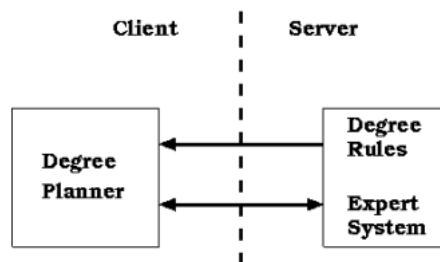
Figura 21 – Planejador de grau acadêmico - Layout da página



Fonte: (DUNSTAN, 2013, p. 305)

O design do sistema é baseado na separação clara entre o tratamento da interface do usuário no cliente e o processamento de regras no servidor (Figura 22). O servidor utiliza um sistema especialista em Prolog para resolver consultas mais complexas, enquanto o cliente lida com informações mais simples por meio de JavaScript. O sistema é acessível via navegador e não requer plugins específicos, sendo adequado para dispositivos móveis, como smartphones.

Figura 22 – Planejador de grau acadêmico - Interação cliente-servidor



Fonte: (DUNSTAN, 2013, p. 304)

A interface do usuário é composta por várias seções, incluindo uma paleta de unidades, um cronograma de programa semestral e um painel de exibição de detalhes das unidades selecionadas. Os estudantes podem explorar opções de inscrição e verificar se suas escolhas atendem aos requisitos do curso. O sistema permite que os

alunos ajustem seu programa acadêmico, respondendo a perguntas sobre requisitos de unidades e pré-requisitos, de forma interativa e personalizada.

Conclui-se que o uso de sistemas especialistas em planejadores de grau acadêmico baseados na web pode melhorar a interatividade e fornecer orientações mais precisas aos alunos, ajudando-os a construir um programa de graduação adequado aos seus interesses e cronograma, sem a necessidade de plugins ou instalações específicas.

3.8 COMPARAÇÃO ENTRE OS TRABALHOS

O estudo de ([CRUZ et al., 2022](#)) ressalta a eficácia da integração da plataforma Beecrowd em um plano de ensino inovador para disciplinas de programação. Ao incorporar estratégias como sala de aula invertida e gamificação, a pesquisa conclui que o Beecrowd, utilizado como suporte para a gamificação, não só melhora o processo de ensino e aprendizagem, oferecendo suporte aos educadores e motivando os alunos, mas também contribui para a redução da evasão em cursos superiores de informática.

A aplicação das metodologias propostas no estudo dos autores demonstrou um impacto significativo no desempenho dos alunos ao longo do tempo na universidade dos autores, resultando em um crescimento notável e consistente nas notas. A ferramenta Beecrowd, destacada no estudo, é elogiada pela sua organização meticolosa de desafios em categorias e níveis de dificuldade, além de sua abordagem de maratona de programação, revelando-se eficaz em envolver os alunos de maneira imersiva.

A Tabela 1 apresenta uma comparação detalhada das vantagens do Beecrowd em relação a outras ferramentas integradas ao Moodle, como BOCA, VPL e CodeRunner, conforme discutido nos trabalhos relacionados. O objetivo é analisar os benefícios específicos do uso do Beecrowd, oferecendo uma base sólida para este estudo. Vale ressaltar que este trabalho propõe o incentivo ao uso da integração entre Moodle e Beecrowd por meio de LTI. Em contraste, o BOCA e o CodeRunner funcionam como tipos de questões no Moodle, enquanto o VPL é um módulo próprio da plataforma.

A análise da tabela destaca o Beecrowd como notavelmente diferenciado em relação aos outros sistemas, sobretudo devido à presença de questões pré-existentes para os professores. Essa característica elimina a necessidade de cadastramento manual das questões de programação e seus respectivos testes, uma exigência presente nas plataformas BOCA, VPL e CodeRunner. Como confirmação, uma das limitações do CodeRunner apontadas por [Lobb e Harlow \(2016\)](#) é o fato de a criação de perguntas e testes de qualidade demandar tempo, além de ser mais adequado para tarefas simples e bem definidas, já que a criação de testes para atividades complexas exige muito esforço, especialmente em turmas grandes.

Um aspecto distintivo e relevante nos sistemas BOCA, VPL e CodeRunner é a abordagem de desenvolvimento que incorpora a utilização de servidores separados

Tabela 1 – Comparação entre quatro integrações de sistemas, que auxiliam no ensino da programação, com o Moodle: Beecrowd, BOCA, CodeRunner e VPL

Aspecto	Beecrowd	BOCA	VPL	CodeRunner
Possui Juiz Online	X	X	X	X
Fornece <i>feedback</i> Imediato	X	X	X	X
Criação Manual de Questões		X	X	X
Questões Prontas para Professores	X			
Questões Prontas classificadas por categoria	X			
Inserção Manual de Casos de Teste		X	X	X
Suporta Diversas Linguagens	X	X	X	X
Pré-processamento de Submissões				X
Comentários de Testes Falhados			X	
Prevenção de Plágio	X		X	
Incentiva participar de maratonas de programação	X			

Fonte: Produzido pela autora.

para o Moodle e os sistemas juízes online. Essa estratégia visa garantir a execução independente e eficiente de cada componente, proporcionando benefícios tanto em termos de desempenho quanto de segurança.

A adoção de servidores separados para o Moodle e os sistemas juízes online tem como objetivo otimizar a distribuição de carga, prevenindo sobrecargas e garantindo uma operação mais estável e responsiva. Essa separação também fortalece a segurança, pois ao isolar cada plataforma, diminui-se os riscos de conflitos e vulnerabilidades. Da mesma forma, ela possibilita uma escalabilidade mais eficiente, permitindo que cada sistema seja dimensionado conforme suas necessidades específicas, sem afetar o desempenho do outro. Isso é particularmente importante em ambientes acadêmicos, onde a demanda por plataformas e sistemas pode variar significativamente.

O trabalho de [Rodríguez-Del-Pino, Royo e Figueroa \(2012\)](#) mostra que o módulo Virtual Programming Lab (VPL) possui um editor de código em Java para editar, executar, depurar e avaliar programas em um ambiente simples de desenvolvimento. Dessa forma, para utilizar todas as funcionalidades é necessário um navegador com suporte a JavaScript e applets Java. Já os trabalhos de [Lobb e Harlow \(2016\)](#) e [Galasso e Moreira \(2014\)](#) mostram que tanto o CodeRunner quanto o BOCA disponibilizam caixas de texto nas quais os usuários podem escrever ou colar o código, permitindo sua subsequente avaliação.

Neste trabalho, a integração LTI fornecida pelo Beecrowd será utilizada, de modo que a criação de listas de exercícios, a resolução de exercícios pelos alunos e a correção das soluções pelo juiz online ocorrerão diretamente nos servidores do Beecrowd.

Além disso, a prevenção contra plágio é uma característica crucial e distintiva tanto no Beecrowd quanto no VPL, destacando-se como um diferencial significativo dessas plataformas. Essa funcionalidade visa salvaguardar a integridade acadêmica, garantindo que os trabalhos submetidos pelos alunos sejam autênticos e originais.

No contexto do Beecrowd, de acordo com (CRUZ *et al.*, 2022), a prevenção contra plágio pode envolver mecanismos avançados de análise de código-fonte, identificando padrões suspeitos que indicam possível cópia indevida. Além disso, o sistema pode empregar algoritmos e técnicas que comparam soluções de diferentes alunos, buscando similaridades que ultrapassem limites aceitáveis de coincidência.

No caso do VPL, de acordo com Rodríguez-Del-Pino, Royo e Figueroa (2012), a prevenção contra plágio pode ser implementada através de verificações rigorosas durante o processo de submissão. Isso pode incluir a comparação automática de códigos submetidos em busca de trechos idênticos ou substancialmente semelhantes. Além disso, o VPL pode adotar métodos de análise mais avançados, como a detecção de técnicas de programação específicas que são características de trabalhos plagiados.

O projeto de Chaves *et al.* (2013) descreve o desenvolvimento do Módulo de Integração com Juízes Online (MOJO), uma ferramenta criada para integrar juízes online, como o Beecrowd (anteriormente URI Online Judge), ao Moodle. O MOJO visa delegar ao Moodle o gerenciamento da interface e do controle das atividades de programação, enquanto centraliza a comunicação e interação com os juízes online. Devido à ausência de APIs ou serviços web nos juízes à época, foram necessárias implementações específicas para cada plataforma, utilizando PHP (a mesma linguagem do Moodle) e JavaScript para a interface. O banco de dados PostgreSQL foi adotado por ser open-source e pela flexibilidade de criação de tabelas personalizadas para armazenamento de dados dos alunos, o que possibilita um *feedback* direcionado e preciso.

Este trabalho, contudo, utilizará a integração LTI fornecida pelo Beecrowd, a qual facilita a interação de docentes e alunos com a plataforma. Além disso, será disponibilizado um manual para orientar o uso da LTI, juntamente com um sistema especialista que auxiliará os alunos na resolução das questões e os docentes no esclarecimento de dúvidas dos estudantes.

É relevante notar que 100% dos professores entrevistados para avaliar a eficácia do MOJO acreditam que a redução de tarefas permitirá um acompanhamento mais eficaz dos estudantes, especialmente daqueles que apresentam maiores dificuldades.

Por fim, os projetos de NASR, TALAB e AL-GAHTANI (2018) e Dunstan (2013) abordam a construção de sistemas especialistas para o aconselhamento e planejamento acadêmico, porém apresentam enfoques e abordagens técnicas distintas.

No primeiro trabalho, NASR, TALAB e AL-GAHTANI (2018) desenvolve um sistema especialista baseado em regras para o aconselhamento acadêmico na Universidade King Khalid. Este sistema é focado em replicar o processo de tomada de decisão dos conselheiros acadêmicos humanos, utilizando uma metodologia de árvore de de-

cisão, implementada através do ES Builder, uma ferramenta de código aberto para sistemas especialistas baseados em regras, com uma interface web simples e uma lógica de "se-então". Essa escolha reflete a necessidade de simular de forma prática e direta o trabalho dos conselheiros humanos sem uma complexidade computacional elevada.

No segundo trabalho, [Dunstan \(2013\)](#) foca em um planejador de grau acadêmico, utilizando uma abordagem híbrida com XML e Prolog para gerenciar e responder a consultas complexas. As regras para cada programa acadêmico são estruturadas em XML e convertidas em Prolog, possibilitando que o sistema lide com consultas interativas e personalizadas. A arquitetura desse sistema é distribuída, com o cliente manipulando a interface e o servidor processando as regras em Prolog. Essa separação facilita o uso em dispositivos móveis e oferece flexibilidade ao permitir que os alunos construam e verifiquem seu programa acadêmico de maneira independente e intuitiva, sem plugins específicos.

Já neste trabalho, será utilizado o Prolog como servidor da aplicação, responsável pelo gerenciamento de requisições HTTP, armazenamento de dados de sessão e comunicação contínua com o front-end. O servidor proporcionará um ambiente dinâmico para o carregamento modular de arquivos específicos de cada questão, permitindo a adaptação das perguntas e a personalização das respostas conforme as interações do usuário. Além disso, configurará permissões de CORS para integrar facilmente com a interface web, que será desenvolvida utilizando o React, uma biblioteca JavaScript para construção de interfaces de usuário interativas e reativas.

4 DESENVOLVIMENTO

Nesta seção, será abordada a LTI do Beecrowd, que simplifica a integração entre o Moodle e a plataforma Beecrowd. Serão apresentados também os detalhes da implementação da aplicação web do sistema especialista, desenvolvida para auxiliar professores e alunos no esclarecimento de dúvidas recorrentes sobre as questões do Beecrowd, facilitando a adaptação de ambos ao uso da plataforma.

4.1 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 (Apêndice A).

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 (Apêndice A). 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 (Apêndice B), 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 (Apêndice B), duas atividades ficam disponíveis na página do curso, apresentadas como "botões" visuais. A primeira atividade, que é ocultada para os estudantes (Imagem 23), permite ao professor acessar o Beecrowd Academic (Imagem 24) para configurar disciplinas, criar listas de exercícios para os alunos e enviar as notas para o Moodle.

Figura 23 – Atividade para o professor acessar o Beecrowd Academic

▼ General



Fonte: Produzido pela autora.

Figura 24 – Beecrowd Academic

ID	NAME	OPTIONS
011533	Moodle Demo II	
011532	Moodle Demo I	
011477	Mackenzie	
010038	Disciplina do Administrador	
010034	Disciplina do coach	
009872	Principal	
009322	SQL	

Fonte: Produzido pela autora.

A segunda atividade, acessível aos estudantes (Imagem 25), redireciona-os diretamente para a página no Beecrowd da lista de exercícios criada pelo professor (Imagem 26), 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.

Figura 25 – Atividade para o aluno acessar o Beecrowd

▼ Topic 1 ↗

Fonte: Produzido pela autora.

Figura 26 – Beecrowd do aluno

The screenshot shows a Beecrowd student profile. At the top, there are navigation buttons: HOME, PROFILE, NEWS (highlighted in red), OPPORTUNITIES, ACADEMIC, CONTESTS, FORUM, PROBLEMS, SUBMISSIONS, RANKS, and SIGN OUT. Below the navigation is a challenge summary:

DISCIPLINE:	Moodle Demo I
PROFESSOR:	New Task
HOMEWORK:	1 problems
EXERCISES:	1 problems
STARTS:	9/11/23, 9:00 PM
ACCEPT:	PYTHON 3.9

Below the challenge summary is a section titled 'INSTRUCTIONS' containing the text 'testing moodle access'. Underneath is a 'PROGRESS' bar which is mostly filled with a green gradient.

A 'TOP ZBIZZ' section lists the student's top users:

- EGJanke0
- aluno.beecrowd
- aluno.ava4253
- celso.barboza+can...
- alunost

Finally, a table shows the student's solved problems:

#	PROBLEM	SUBMISSION	ACCEPTED	LEVEL	WEIGHT
1	1000 ✓ Hello World!	02	35493426	5	100

Fonte: Produzido pela autora.

4.2 SISTEMA ESPECIALISTA WEB

O objetivo do sistema especialista é oferecer suporte na resolução de dúvidas recorrentes dos estudantes em relação a questões do Beecrowd. A proposta é desenvolver 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, o sistema especialista será estruturado com um conjunto de dados contendo perguntas direcionadas aos alunos e respostas pre-definidas 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.

4.2.1 Coleta de Dados

Nesta fase, foram realizadas duas coletas de dados: uma com dicas de resolução fornecidas pelo professor para uma lista de exercícios, e outra com as dúvidas dos alunos sobre o uso de listas em Python, com foco em questões específicas.

Na primeira coleta, professor da disciplina de Programação Orientada a Objetos I, da Universidade Federal de Santa Catarina (UFSC), forneceu dicas de resoluções para cada questão de uma lista de exercícios que ainda não tinha sido repassada aos alunos, com o objetivo de avaliar a utilidade da aplicação após ela ser desenvolvida. Assim, quando os alunos fossem resolver a lista de exercícios, poderiam testar a aplicação para ser sua eficácia. As questões abordadas, que receberam as dicas de solução, incluem: 1261, 1281, 1430, 1449, 1483, 1763, 1991, 1953, 2091, 2482, 2492, 2654, 2949 e 2987.

Na segunda coleta, foram registradas as dúvidas dos alunos da disciplina de Programação Orientada a Objetos I, da UFSC, coletadas no dia 30 de outubro de 2024. Nessa data, os alunos estavam trabalhando em uma lista de exercícios sobre o uso de listas em Python. As questões abordadas incluíram os problemas de números 1187, 1435, 1715, 2436, 1383, 1184, 1181 e 1185. Um aspecto notável foi que, durante essa mesma aula, quatro alunos apresentaram a mesma dúvida em relação à questão 1435, especificamente perguntando: "Como resolvo essa questão?"

Durante as 2h30 de aula, a questão 1435 foi a que suscitou o maior número de dúvidas entre os estudantes. Em uma turma de 26 alunos – embora nem todos tenham alcançado essa questão no tempo da aula –, quatro solicitaram uma explicação geral sobre a abordagem para resolvê-la, enquanto dois buscaram orientação sobre a conversão dessa abordagem para o código. Adicionalmente, três alunos enfrentaram o erro "Presentation Error" e um aluno recebeu o erro "Run-time Error".

As dúvidas em relação à questão 1181 refletiram dificuldades comuns no uso de laços de repetição e na manipulação de matrizes. Muitos estudantes não obtinham a resposta correta, mesmo com uma solução aparentemente desenvolvida, devido à confusão na seleção dos elementos da matriz, com erros ao iterar pelas linhas em vez das colunas. Os alunos também enfrentaram dificuldades na estruturação do laço de repetição, especialmente ao calcular a soma dos elementos de uma linha específica e ao determinar o divisor correto para o cálculo da média. Problemas de formatação, que resultaram no erro "Presentation Error", também foram recorrentes, sendo necessário ajustar a saída para uma casa decimal com '.

Na questão 1184, as dificuldades estavam relacionadas à estruturação do laço de repetição para percorrer corretamente os elementos abaixo da diagonal principal da matriz. Os alunos foram orientados a usar um laço duplo, com a sequência correta de `for i in range(0, 12) and for j in range(0, i)`. Além disso, houve dúvidas sobre a exclusão dos elementos da diagonal principal, que não deveriam ser somados. Como na questão

1181, os alunos enfrentaram dificuldades de formatação, com orientações para ajustar a saída utilizando ' '.

As dúvidas relativas à questão 1185 envolveram a manipulação de elementos acima da diagonal secundária da matriz, com os alunos sendo orientados a percorrer as linhas até o último índice da coluna, subtraindo o número da linha. A configuração correta dos laços de repetição foi outra dúvida comum, com muitos alunos invertendo os laços, o que afetou os resultados. A correção do divisor para o cálculo da média e os problemas de formatação também foram pontos críticos, com instruções para usar ' '.

A questão 1187 gerou dúvidas relacionadas aos laços de repetição necessários para somar os elementos da matriz, com os alunos questionando sobre o ajuste dos índices para evitar a soma dos elementos das diagonais. Também surgiram dúvidas sobre como os laços percorriam as linhas e colunas para formar a região triangular desejada, além de questionamentos sobre a lógica para calcular a média e evitar erros de divisão.

A questão 1383, sobre as regras do Sudoku, suscitou dúvidas sobre como validar se uma solução estava correta, além de dificuldades na transformação da lógica para código. A orientação dada foi garantir a não repetição de números em cada linha, coluna, e bloco 3x3, com orientações sobre o uso de depuração com o udebug e Thonny.

A questão 1715 gerou dúvidas sobre a multiplicação dos elementos das linhas para identificar jogadores que marcaram gols em todas as partidas. Os alunos tiveram dificuldades em entender como aplicar esse método corretamente e como garantir que o contador fosse atualizado. A depuração no udebug e Thonny foi útil para a maioria dos alunos, que ajustaram seus códigos após o *feedback*.

Por fim, a questão 2465 também gerou dúvidas em relação ao uso do comando while True para verificar os vizinhos da matriz. A principal dificuldade foi evitar que o robô retornasse à posição anterior após avançar para um vizinho com valor '1', sendo sugerido que a posição fosse marcada como zero antes de seguir para o próximo vizinho. As orientações permitiram que os alunos avançassem na resolução do problema, superando as dificuldades iniciais.

4.2.2 Requisitos da Aplicação

Esta seção descreve os requisitos essenciais para o desenvolvimento da aplicação, divididos em Regras de Negócio, Requisitos Funcionais e Não Funcionais. As Regras de Negócio definem os processos do sistema, enquanto os Requisitos Funcionais especificam as funcionalidades necessárias. Já os Requisitos Não Funcionais tratam de aspectos de qualidade, como desempenho, segurança e arquitetura do sistema.

4.2.2.1 Regras de Negócio

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.

4.2.2.2 Requisitos Funcionais

RF1 – **Comunicação por Chat:** 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 [2.2](#).

4.2.2.3 Requisitos Não Funcionais

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.

4.2.3 Tecnologias Usadas

4.2.3.1 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](#)).

4.2.3.2 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](#)).

4.2.3.3 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.

4.2.3.4 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* ([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.

4.2.3.5 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 questionNumber 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.

4.2.3.6 React e Typescript (Frontend)

O *frontend* foi feito em React¹ e Typescript², 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, tanto para a web quanto para aplicativos nativos. Ela permite criar interfaces a partir de unidades individuais chamadas componentes, que podem ser reutilizados e combinados para formar interfaces complexas e 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 JavaScript³, como TypeScript.

¹ <https://react.dev/>

² <https://www.typescriptlang.org/>

³ <https://www.javascript.com/>

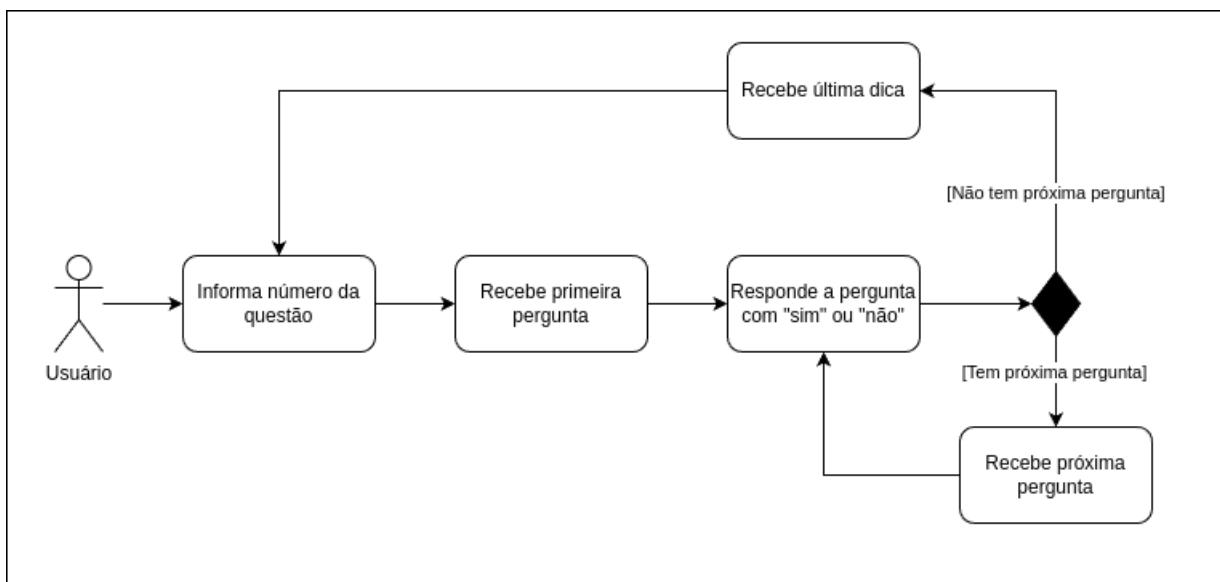
TypeScript é uma linguagem de programação que é um superconjunto do JavaScript, adicionando tipagem estática. Isso permite que os desenvolvedores definam tipos para variáveis e funções, ajudando a identificar erros antes da execução do código. TypeScript melhora a manutenção e segurança do código, especialmente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

4.2.4 Arquitetura de 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 27 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.

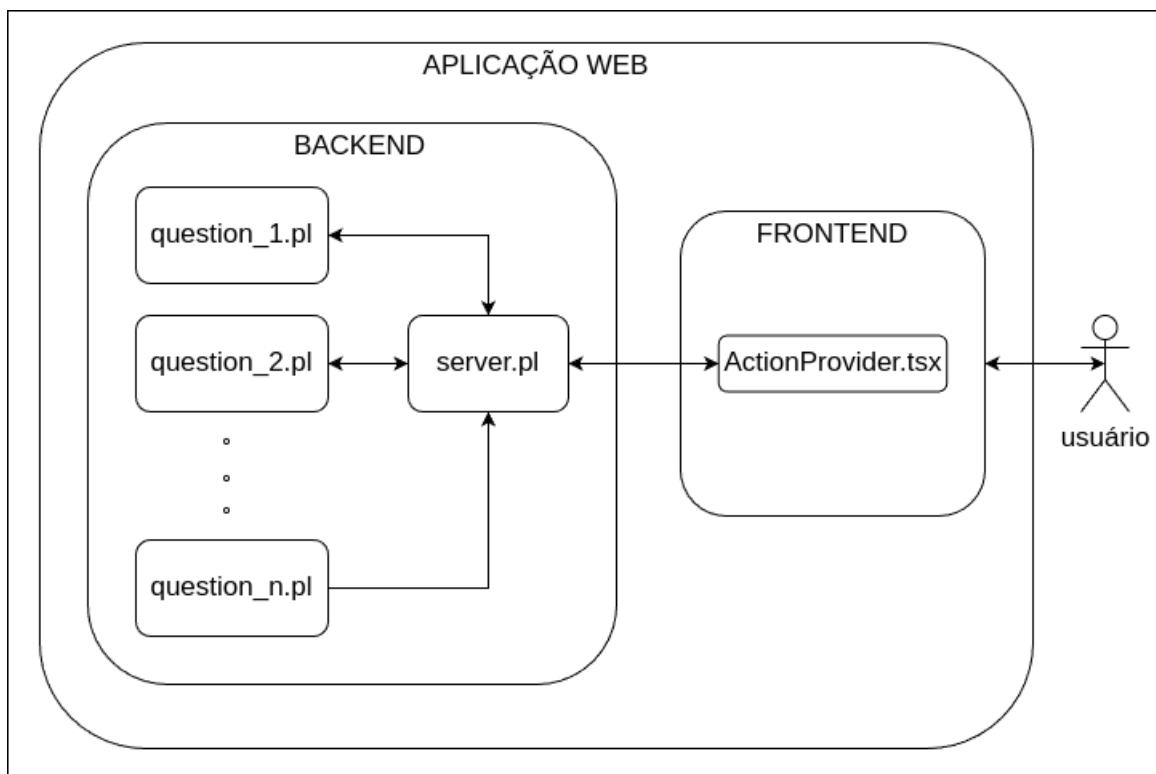
Figura 27 – Diagrama de Atividades do Usuário da Aplicação



Fonte: Produzido pela autora.

A Figura 28 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 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.

Figura 28 – Arquitetura da Aplicação do Sistema Especialista



Fonte: Produzido pela autora.

4.2.5 Elementos do Frontend

O *frontend* da aplicação foi desenvolvido utilizando as tecnologias React e TypeScript, conforme especificado no requisito não funcional **RNF6**, descrito na seção 4.2.2. 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 FredrikOseberg⁴ como estrutura principal para a criação do chatbot, combinado com o repositório *Chatbot-using-React-Chatbot-Kit* de Subid Das⁵, que forneceu recursos adicionais para personalização da interface e 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.

Pelo Código 1 é possível observar que, para a construção do componente Chat, foi usado o componente Chatbot da biblioteca *react-chatbot-kit*. O Chatbot recebe o MessageProvider, responsável por processar as respostas do usuário e acionar as funções do ActionProvider, que gerencia o envio das requisições ao backend.

Código 1 – Componente Chat

```
import { useState } from "react";
import Chatbot from "react-chatbot-kit";
import "react-chatbot-kit/build/main.css";
import { styled } from "styled-components";
import config from "../../bot/config";
import ActionProvider from "../../bot/ActionProvider";
import MessageParser from "../../bot/MessageParser";

const Chat = () => {
  const [ isOpened, setIsOpened ] = useState<boolean>(false)
  return (
    <StyledChat>
      <div className="intro">
        <h2>Aqui você pode tirar dúvidas sobre algumas questões do Beecrowd!</h2>
        <button onClick={() => setIsOpened((prev) => !prev)}>
          Comece agora!
        </button>
    </StyledChat>
  )
}
```

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

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

```

        </div>
        {isOpened && (
          <Chatbot
            config={config}
            messageParser={MessageParser}
            actionProvider={ActionProvider}
          />
        )}
      </StyledChat>
    );
  };

export default Chat;

```

Para atender ao Requisito Funcional **RF5**, descrito na seção 4.2.2, foi criado um componente `Details` para a rota `details` para mostrar as descrições das mensagens de resposta das questões do Beecrowd (Código 2).

Código 2 – Rotas do frontend

```

<Routes location={location} key={location.pathname}>
  <Route path="/" element={<Chat />} />
  <Route path="/details" element={<Details />} />
  <Route path="*" element={<NotFound />} />
</Routes>

```

4.2.6 Comunicação Frontend-Backend

O Código 3 representa como o arquivo `ActionProvider.tsx`, do *frontend*, informa para o *backend* o número da questão. Ele define duas funções assíncronas: `fetchData` e `handleFirstMessage`, usadas para interagir com um *backend* via requisições HTTP.

A função `fetchData` recebe uma URL e um corpo de dados (`URLSearchParams`) para fazer uma requisição POST. Ela define cabeçalhos para enviar os dados como `application/x-www-form-urlencoded`, garantindo que o *backend* receba os dados em um formato compatível e de fácil manipulação, especialmente em linguagens que interpretam parâmetros de URL de forma nativa. Além disso, a função inclui cookies nas requisições com `credentials`: 'include' para permitir que a sessão seja mantida entre as requisições, o que é crucial para a continuidade da comunicação com o *backend* e possibilita cumprir o Requisito Não Funcional **RNF4** descrito na seção 4.2.2. Caso a resposta não tenha o status 200, a função trata o erro, retornando uma mensagem de erro em vez do JSON da resposta.

A função `handleFirstMessage` constrói um corpo de dados contendo question-

Number e chama fetchData para enviar esses dados ao *backend* no *endpoint* "endereço/server". Em seguida, cria uma estrutura de mensagens adicionais vazia e invoca handleMessageResponse para manipular a resposta, preenchendo essa estrutura conforme a resposta recebida.

Código 3 – Funções assíncronas para requisição HTTP

```
const fetchData = async (url: string, body: URLSearchParams) => {
    try {
        const response = await fetch(url, {
            method: "POST",
            headers: { 'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8' },
            credentials: 'include',
            body,
        });

        if (response.status !== 200) throw new Error("Erro ao consultar o backend.");

        return await response.json();
    } catch (error) {
        return { error: "Erro ao consultar o backend." };
    }
};

const handleFirstMessage = async (questionNumber?: string) => {
    const body = new URLSearchParams({ questionNumber: questionNumber?.toString() || '' });
    const data = await fetchData("{endereço}/server", body);
    let additionalMessages: {
        messages: {
            message: string;
            type: string;
            id: number;
            loading?: boolean;
            widget?: string | undefined;
            delay?: number | undefined;
            payload?: any;
        }[];
    }[] = [];

    await handleMessageResponse(data, additionalMessages);
}
```

};

4.2.7 Elementos do Backend

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 4.2.2.

Além disso, conforme ilustrado na Figura 28, 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.

4.2.7.1 Arquivos no formato *questao_{numeroDaQuestao}.pl*

O Código 4 apresenta o arquivo *questao_1181.pl* como um exemplo de estrutura para arquivos do tipo *questao_{numeroDaQuestao}.pl*.

Código 4 – Arquivo *questao_1181.pl*

```

:- module(questao_1181, [questao/3, diagnostico/3]).


% Predicado para fornecer perguntas com base na sequência de respostas
questao(1181, [], "Você já desenvolveu sua solução, mas não está saíndo a
resposta correta?").

questao(1181, ["sim"], "Tome cuidado que a questão quer todos os elementos da
linha, e não da coluna! Verifique \ccomo você está iterando sobre a matriz: se
está pegando todos os elementos de uma linha \c ou todos os elementos de uma
coluna. Mais uma dica sobre isso?").

questao(1181, ["sim", "sim"], "Se sua matriz é feita de forma que é iterada
assim: matriz[linha][coluna], verifique \cse o 'for' está correto. Quer saber
como o for fica?").

questao(1181, ["sim", "sim", "sim"], "considerando que l é a entrada do
usuário que indica a linha que será a \cconsiderada para operação: \nfor i in
range(0, 12): soma += matriz[l][i]. \n\cPróxima dica?").

questao(1181, ["sim", "sim", "sim", "sim"], "Verifique se o número pelo qual
você está tentando dividir para \cpara conseguir a média está correto. Próxima
pergunta?").

questao(1181, Respostas, "Deu Presentation Error?") :-
    Respostas = ["não"];
    Respostas = ["sim", "não"];

```

```
Respostas = ["sim", "sim", "não"];
Respostas = ["sim", "sim", "sim", "não"];
Respostas = ["sim", "sim", "sim", "sim", "sim"].
```

questao(1181, Respostas, "Perceba que a formatação que a questão pede é:
Imprima o resultado solicitado \c(a soma ou média), com 1 casa após o ponto decimal. Ou seja, '%.1f'. Deu certo?") :-

```
Respostas = ["não", "sim"];
Respostas = ["sim", "não", "sim"];
Respostas = ["sim", "sim", "não", "sim"];
Respostas = ["sim", "sim", "sim", "não", "sim"];
Respostas = ["sim", "sim", "sim", "sim", "sim"].
```

% Diagnóstico com base nas respostas completas

diagnostico(1181, Respostas, "Legal! Parabéns!") :-

```
Respostas = ["não", "sim", "sim"];
Respostas = ["sim", "não", "sim", "sim"];
Respostas = ["sim", "sim", "não", "sim", "sim"];
Respostas = ["sim", "sim", "sim", "não", "sim", "sim"];
Respostas = ["sim", "sim", "sim", "sim", "não", "sim"].
```

diagnostico(1181, _, "Atente-se bem às dicas já enviadas ou pergunte novamente! Além disso, você pode \cusrar o udebug para verificar a diferença entre as saídas do seu código, e as \csaidas esperadas pela questão! Também é uma boa ideia debugar seu código no Thonny, \crevisando linha por linha, ou usando prints para entender o que ele está executando!").

Esse arquivo Prolog define um módulo especializado para auxiliar usuários na resolução da questão 1181 por meio de um sistema especialista baseado em regras. O módulo *questao_1181* implementa dois predicados principais: *questao/3* e *diagnostico/3*, que são responsáveis por operacionalizar a lógica de suporte do sistema.

- **Predicado *questao/3*:** Este predicado orienta o usuário de maneira sequencial, adaptando as instruções com base nas respostas fornecidas. Ele recebe três parâmetros: o identificador da questão, uma lista de respostas (que mantém o histórico de interações), e uma mensagem de orientação. A matriz de decisões, implementada neste predicado, direciona o fluxo de perguntas e respostas com base no histórico de interações do usuário, promovendo uma progressão lógica que facilita a resolução da questão. Essa matriz está estruturada de forma que cada combinação de respostas leva a uma nova instrução, que oferece dicas cada vez mais específicas até que o usuário obtenha a solução correta.

A tabela 2 apresenta a matriz de decisões para o predicado *questao/3*, em for-

mato de tabela, representando as combinações de respostas possíveis e as respectivas instruções emitidas:

Tabela 2 – Matriz de Decisões para o Predicado *questao/3* na Questão 1181

Questão	Respostas	Instrução
1181	[]	Você já desenvolveu sua solução, mas não está saindo a resposta correta?
1181	[sim]	Tome cuidado que a questão quer todos os elementos da linha, e não da coluna! Verifique como você está iterando sobre a matriz: se está pegando todos os elementos de uma linha ou todos os elementos de uma coluna. Mais uma dica sobre isso?
1181	[sim, sim]	Se sua matriz é feita de forma que é iterada assim: matriz[linha][coluna], verifique se o 'for' está correto. Quer saber como o for fica?
1181	[sim, sim, sim]	Considerando que l é a entrada do usuário que indica a linha que será considerada para operação: <code>for i in range(0, 12): soma += matriz[l][i]</code> Próxima dica?
1181	[sim, sim, sim, sim]	Verifique se o número pelo qual você está tentando dividir para calcular a média está correto. Próxima pergunta?
1181	[não], [sim, não], [sim, sim, não], [sim, sim, sim, não], [sim, sim, sim, sim, sim]	Deu Presentation Error?
1181	[não, sim], [sim, não, sim], [sim, sim, não, sim], [sim, sim, sim, não, sim], [sim, sim, sim, sim, sim, sim]	Perceba que a formatação que a questão pede é: Imprima o resultado solicitado (a soma ou média), com 1 casa após o ponto decimal. Ou seja, %.1f. Deu certo?

- **Predicado *diagnostic0/3*:** Este predicado fornece um diagnóstico final com base nas respostas acumuladas ao longo da interação. Atuando como *feedback* conclusivo, ele emite uma mensagem de sucesso caso as respostas indiquem que o usuário compreendeu e solucionou a questão. Caso contrário, sugere ferramentas como o *udebug* e o depurador *Thonny*, de modo que o usuário possa inspecionar o comportamento do seu código em detalhes.

Dessa forma, arquivos do formato *questao_{numeroDaQuestao}.pl* devem seguir essa estrutura, onde cada questão contém um predicado *questao/3* para fornecer orientações sequenciais e um predicado *diagnostico/3* para avaliação final. A matriz de decisões implementada no predicado *questao/3* permite um processo dinâmico e adaptativo de interação, guiando o usuário de forma direcionada e progressiva na resolução da questão específica.

Assim, para a inclusão de novas questões, basta criar um arquivo no formato especificado, seguindo a mesma estrutura. Essa abordagem facilita a expansão do sistema, conforme estabelecido no requisito não funcional **RNF5**, descrito na seção [4.2.2](#).

4.2.7.2 Arquivo server.pl

4.2.7.2.1 Configurando servidor HTTP

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.

O Código 5 exemplifica a configuração de um servidor HTTP em Prolog utilizando o módulo *http_dispatch*, no qual as rotas são definidas por meio de *http_handler* para manipular requisições *GET*, *POST* e *OPTIONS*. Especificamente, a requisição *POST* é tratada pela rota *server*, sendo manipulada pela função *handle_post_request*.

Código 5 – Arquivo *server.pl* - Handlers e Manipuladores de requisições

```
% Define o handler para receber as requisições HTTP POST
:- http_handler(root(server), handle_post_request, [method(post)]).
:- http_handler(root(.), handle_options, [method(options)]).

% Manipulador de requisições POST
handle_post_request(Request) :-
    member(method(post), Request), !,

    % Garante que a sessão está ativa para o cliente
    http_session_id(_SessionID),

    % Lê os dados da requisição
    http_read_data(Request, Data, [encoding(utf8)]),

    % Definindo os headers para permitir CORS
    format('Access-Control-Allow-Origin: {endereço}~n'),
```

```

format('Access-Control-Allow-Credentials: true~n'),
format('Content-Type: application/json; charset=UTF-8'),
cors_enable(Request, [methods([post])]),

( % Caso seja a primeira requisição da questão
  member(questionNumber=QN, Data)
-> % Inicializa a sessão e a questão
  http_session_retractall(questionNumber(_)),
  http_session_retractall(answers("Answers", _)),
  http_session_assert(answers("Answers", [])),

  http_session_assert(questionNumber(QN)),
  start_question(QN, FirstQuestion),

  reply_json_dict(FirstQuestion)

; % Caso receba respostas subsequentes
  member(answer=Answer, Data),

  % Garante que a sessão tenha a variável questionNumber
  ( http_session_data(questionNumber(QN)) ->
    continue_question(QN, Answer, NextQuestionOrResult),
    reply_json_dict(NextQuestionOrResult)
  ; % Erro caso a sessão não tenha um questionNumber
    reply_json_dict({_error: "Número da questão não encontrado na sessão"})
  )
).

```

O **manipulador de requisições POST** (`handle_post_request`) tem como função processar requisições *POST* enviadas ao servidor, gerenciando o fluxo de perguntas e respostas entre o cliente e o servidor. Inicialmente, ele habilita o CORS, seguindo o Requisito Não Funcional **RNF6**, descrito na seção 4.2.2. Isso permite requisições provenientes do *frontend* no endereço endereço, e configura os cabeçalhos necessários para o envio de credenciais, como cookies de sessão. A função verifica se a sessão está ativa, garantindo a associação com um ID de sessão único, cumprindo os Requisitos Não Funcionais **RNF3** e **RNF4**.

Dependendo do tipo de requisição, a função processa a primeira pergunta ou continua a sequência de perguntas baseadas nas respostas anteriores. Se for a primeira requisição, o número da questão é extraído, e a sessão é inicializada com os dados necessários. Além disso, o arquivo correspondente à questão é carregado dinamicamente.

Caso seja uma resposta subsequente, a função valida a presença do número da questão na sessão, atualiza as respostas e retorna a próxima pergunta ou o resultado final. Se os dados da sessão estiverem faltando, um erro é enviado indicando que o número da questão não foi encontrado.

4.2.7.2.2 Carregando os arquivos das questões

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.

No Código 6, é mostrado o trecho do *server.pl* que realiza esse carregamento dinâmico. O predicado *start_question/2* tem como objetivo inicializar uma nova questão dentro de um sistema interativo, baseado no número da questão fornecido. O primeiro passo do código é construir o caminho para o arquivo que contém a definição da questão, concatenando o prefixo *questao_* com o número da questão (Question-Number) e adicionando a extensão *.pl* para formar o nome completo do arquivo. Isso é realizado através do predicado *atom_concat3*, que concatena átomos (strings) para formar o caminho completo do arquivo a ser carregado.

Código 6 – Arquivo *server.pl* - Handlers e Manipuladores de requisições

```
start_question(QuestionNumber, FirstQuestion) :-
    atom_concat('questao_', QuestionNumber, FileName),
    atom_concat(FileName, '.pl', FilePath),

    % Verifica se o arquivo existe antes de tentar carregá-lo
    ( exists_file(FilePath)
    -> % Carrega dinamicamente o arquivo da questão
        abolish(questao/3),
        abolish(diagnostico/3),

        ensure_loaded(FilePath)
    ).
```

Depois, o código verifica se o arquivo gerado existe no sistema de arquivos utilizando o predicado *exists_file1*. Caso o arquivo exista, o sistema prossegue para carregá-lo dinamicamente com o predicado *ensure_loaded1*. Antes de carregar o arquivo, o predicado *abolish1* é utilizado para garantir que qualquer definição anterior dos predicados *questao3* e *diagnostico3* seja removida da memória, evitando conflitos ou sobrecarga de definições antigas.

4.2.7.2.3 Recebe respostas

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. O Código 7 apresenta a função **continue_question**, responsável por processar respostas subsequentes enviadas pelo usuário para uma questão em andamento, armazenando a resposta na sessão e determinando o próximo passo com base nas respostas acumuladas. Inicialmente, a função adiciona a nova resposta à lista de respostas da sessão, atualizando a variável de sessão *answers* ("Answers").

Código 7 – Arquivo *server.pl* - Handlers e Manipuladores de requisições

```

continue_question(QuestionNumber, Answer, Response) :-
    string(Answer) -> AW = Answer
    ; atom(Answer) -> atom_string(Answer, AW)
    ; term_string(Answer, AW)
    ),
    atom_number(QuestionNumber, QN),

    % Salva a resposta na sessão
    http_session_data(answers("Answers", AnswerList)),
    append(AnswerList, [AW], UpdatedAnswers),
    http_session_retractall(answers("Answers", _)),
    http_session_assert(answers("Answers", UpdatedAnswers)),

    % Chama a questão correspondente com a lista atualizada de respostas
    ( questao(QN, UpdatedAnswers, NextQuestion)
    -> Response = {_question: NextQuestion}
    ; ( diagnostico(QN, UpdatedAnswers, Result)
        -> true
        ; Result = "Sem conclusão final."
        ),
        Response = {_result: Result}
    ).
```

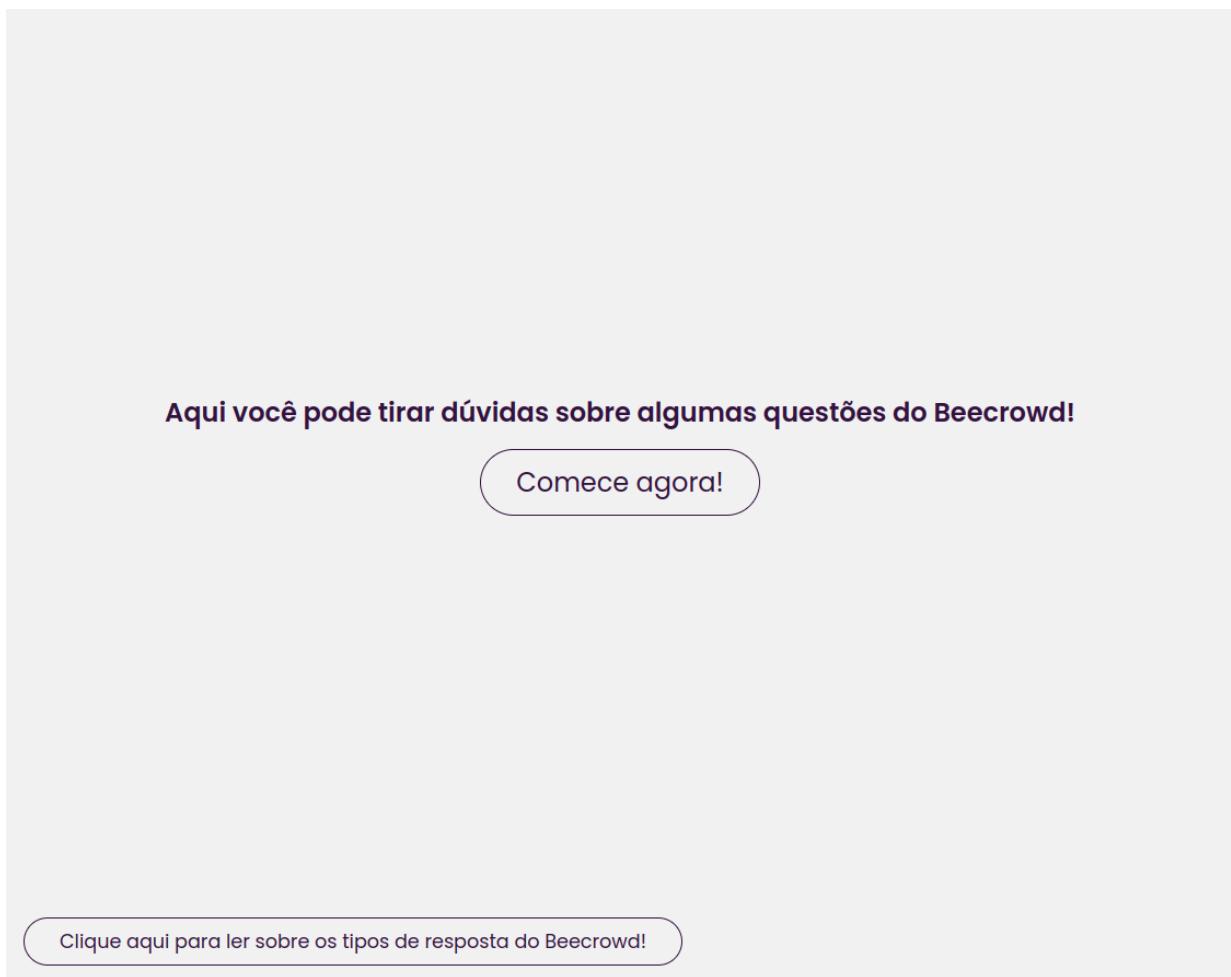
Em seguida, tenta gerar a próxima pergunta por meio da chamada ao predicado *questao/3*. Se *questao/3* retornar uma nova pergunta (*NextQuestion*), esta é enviada ao cliente para dar continuidade à interação. Caso contrário, a função verifica se uma resposta final pode ser gerada utilizando o predicado *diagnostico/3*. O resultado final, seja uma nova pergunta ou uma conclusão, é então enviado ao cliente no formato JSON, permitindo que a interação prossiga ou seja concluída de acordo com o progresso da sessão, como consta no Requisito Não Funcional **RNF6**, descrito na seção [4.2.2](#).

4.2.8 Expert Bee: Sistema Especialista de Apoio à Resolução de Dúvidas nos Exercícios do Beecrowd

A aplicação foi desenvolvida para ser acessada via navegador e tem como objetivo ajudar os usuários com dúvidas na resolução dos exercícios do Beecrowd. Por meio de um chat interativo, o usuário pode informar qual questão precisa de orientação. A partir disso, o sistema especialista realiza perguntas específicas para compreender melhor as necessidades do usuário e, com base nas respostas, fornece dicas e orientações personalizadas. Ou seja, à medida que o usuário responde "sim" ou "não" a cada pergunta, o sistema ajusta as questões subsequentes e as sugestões, guiando o usuário de forma progressiva até fornecer uma solução ajustada ao contexto das respostas acumuladas.

Nesta seção, serão apresentadas imagens que ilustram o resultado final da aplicação.

Figura 29 – Expert Bee: Tela Inicial



Fonte: Produzido pela autora.

A Figura 29 exibe a tela inicial da aplicação Web, que contém dois botões. Ao clicar no botão "Clique aqui para ler sobre os tipos de resposta do Beecrowd!", o usuário

é redirecionado para a tela mostrada na Figura 30. Nessa tela, são apresentados detalhes sobre cada tipo de resposta do Beecrowd, atendendo ao Requisito Funcional **RF5**, descrito na seção 4.2.2. Além disso, caso o usuário clique no botão "Voltar para a página inicial", ele retorna à tela inicial.

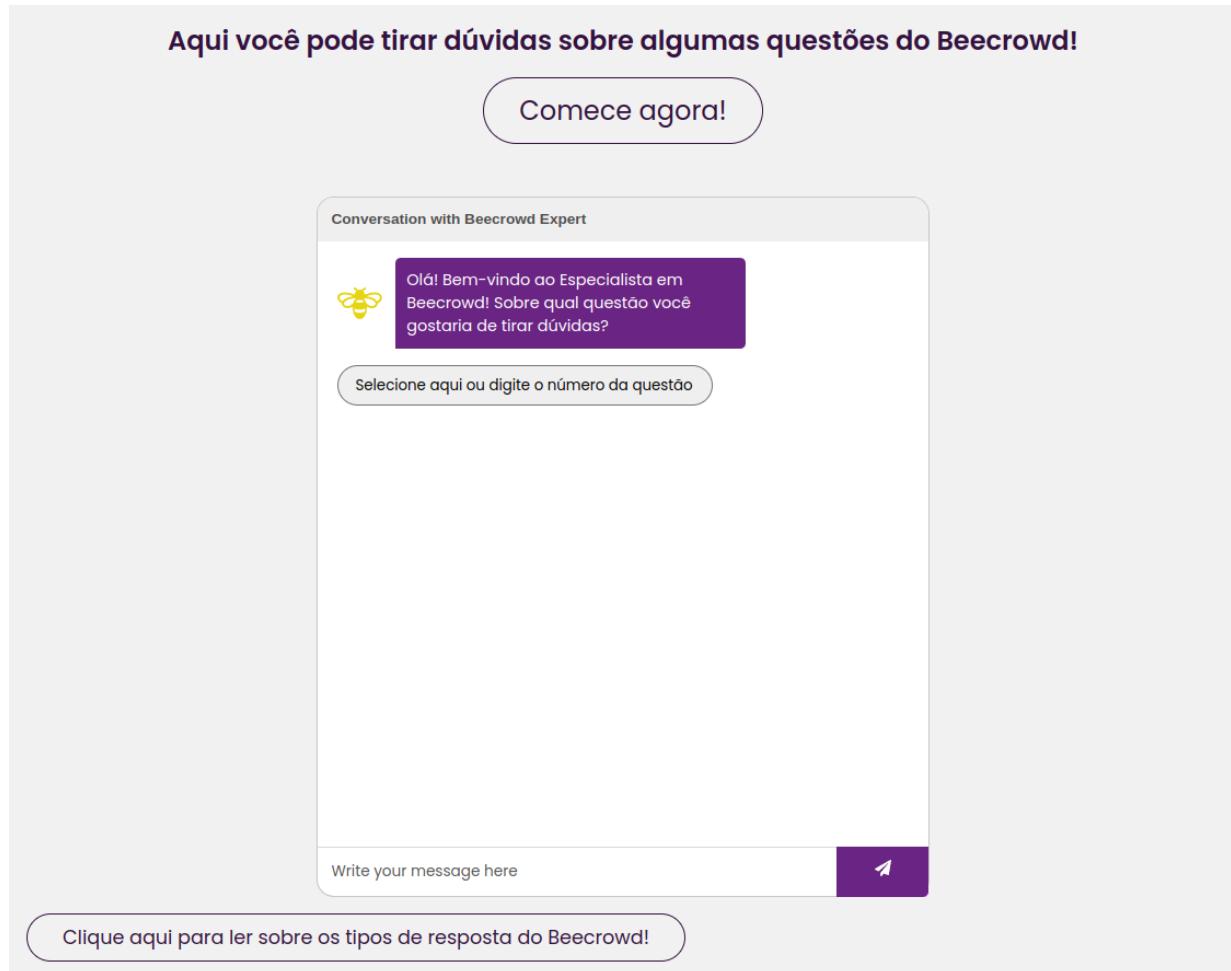
Figura 30 – Expert Bee: Detalhes das Respostas do Juiz Online do Beecrowd

The screenshot shows a user interface for handling errors in a programming competition. At the top, a large text area is titled 'Resposta errada/Wrong answer'. It contains a detailed explanation in Portuguese about what to do if the program fails a test case, mentioning the use of 'uDebug' and Thonny for debugging. Below this are four expandable sections: 'Tempo limite excedido/Time limit exceeded', 'Erro de compilação/Compile error', 'Erro em tempo de execução/Run-time error', and 'Erro de apresentação/Presentation error'. At the bottom left is a button labeled 'Voltar para a página inicial' (Return to the initial page).

Fonte: Produzido pela autora.

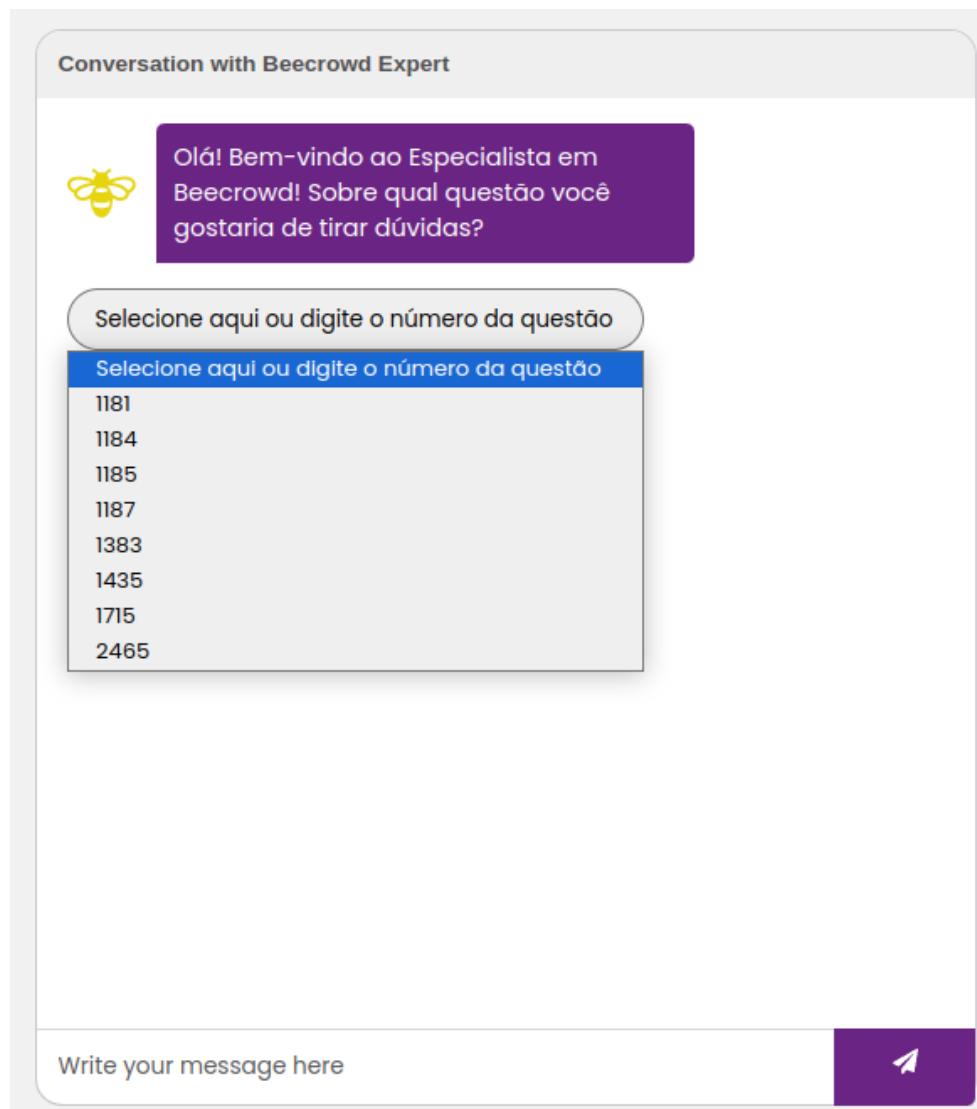
Na página inicial, ao clicar no botão "Comece agora!", é aberto o chat, conforme o Requisito Funcional **RF1**, ilustrado na Figura 31. Através deste chat, o usuário deve inicialmente informar o número da questão sobre a qual deseja tirar dúvidas, atendendo ao Requisito Funcional **RF2**. O usuário pode selecionar o número da questão a partir da lista apresentada (Figura 32) ou optar por digitá-lo manualmente (Figura 33). Caso o número informado não corresponda a uma questão existente no sistema, uma mensagem informando que a questão não foi encontrada será exibida (Figura 34).

Figura 31 – Expert Bee: Chat



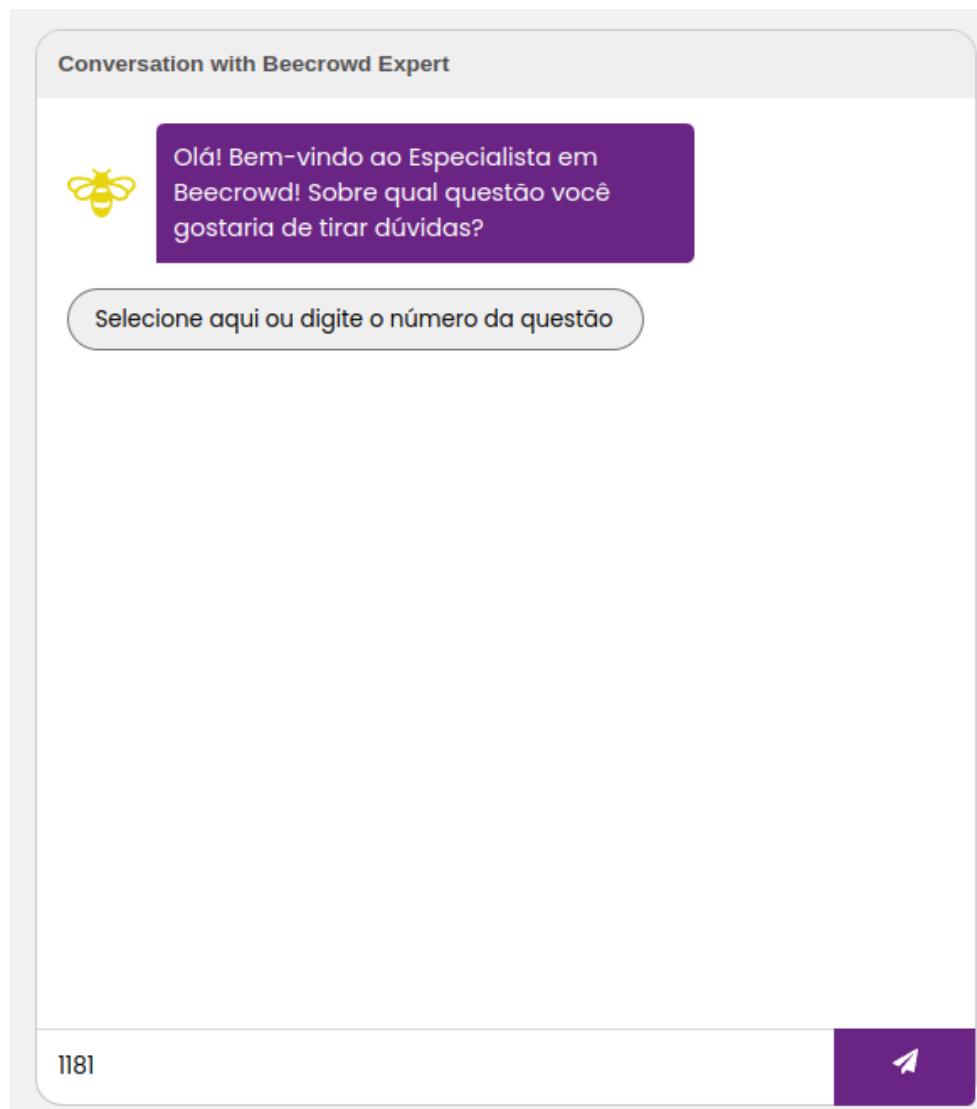
Fonte: Produzido pela autora.

Figura 32 – Expert Bee: Escolhendo a Questão



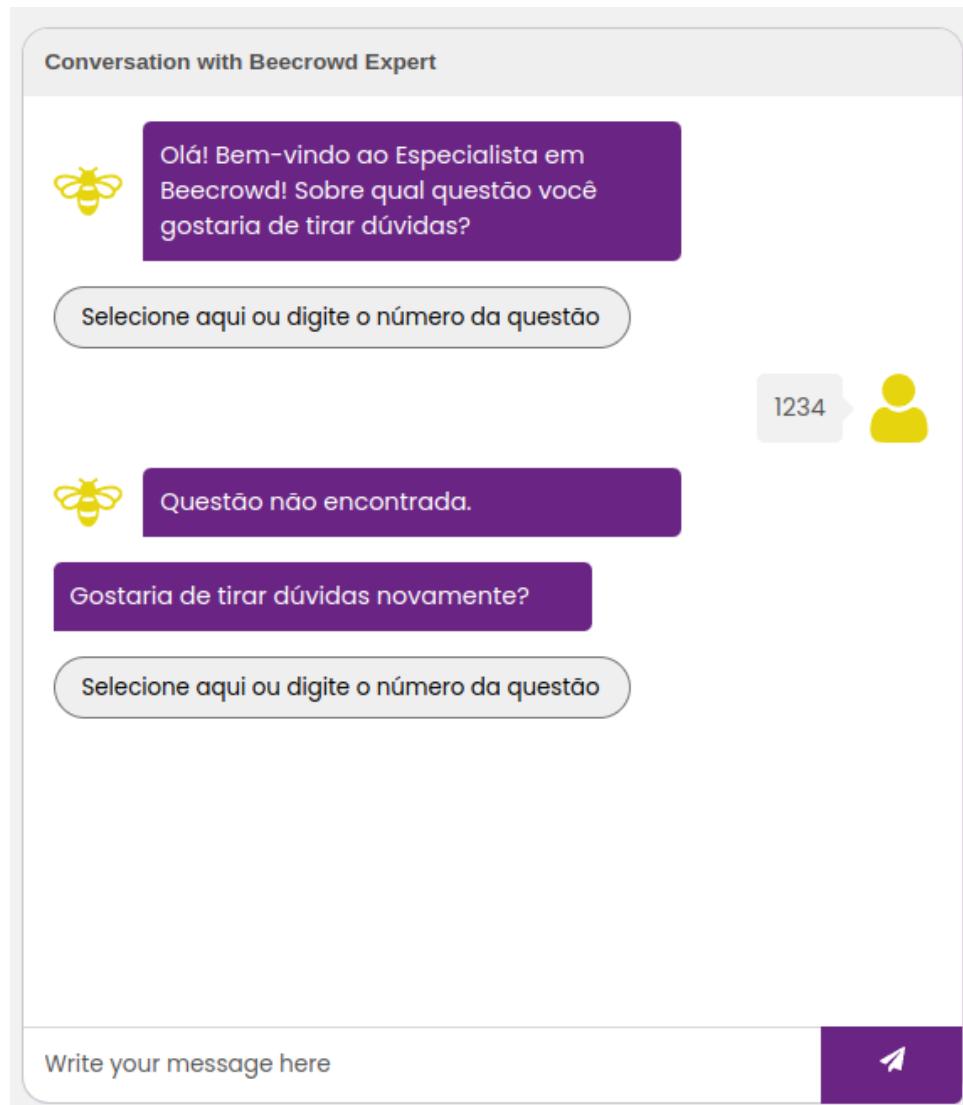
Fonte: Produzido pela autora.

Figura 33 – Expert Bee: Digitando o Número da Questão



Fonte: Produzido pela autora.

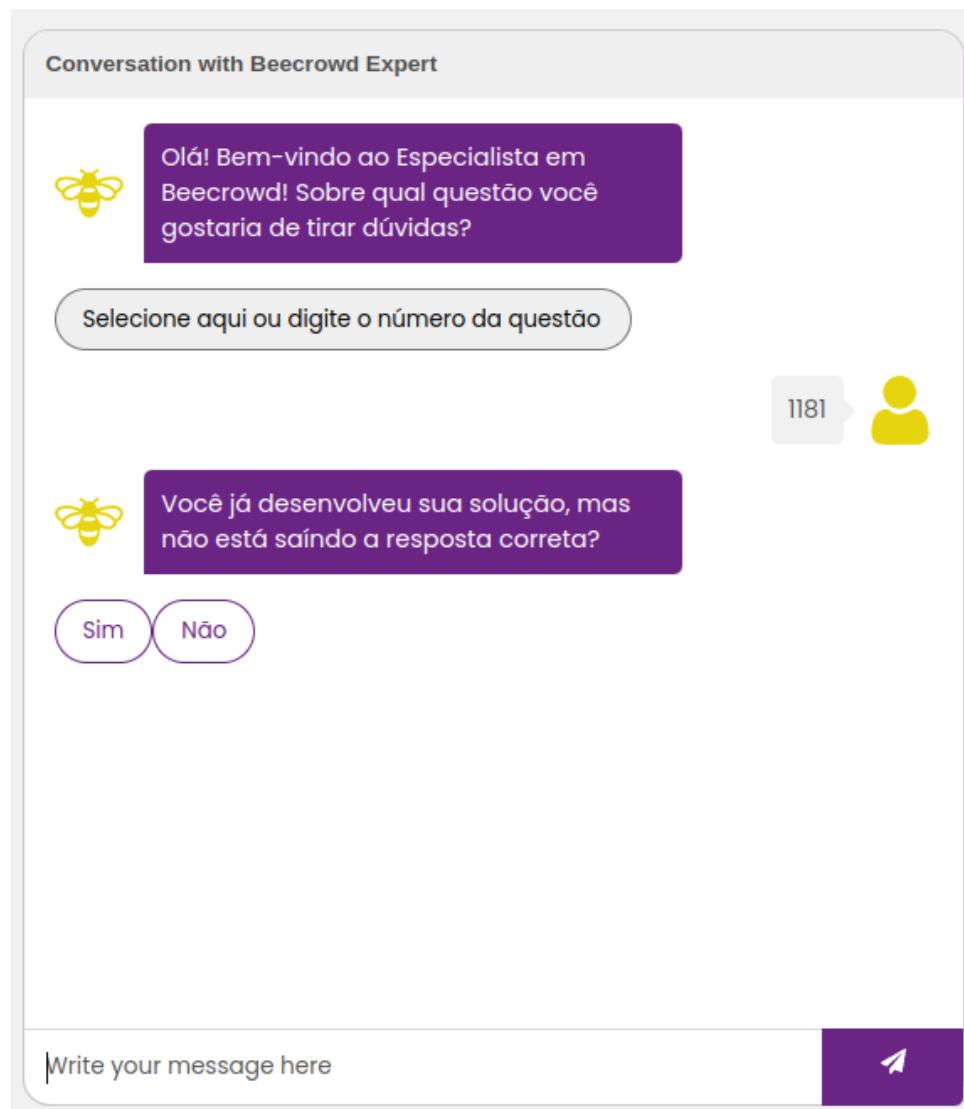
Figura 34 – Expert Bee: Questão não encontrada



Fonte: Produzido pela autora.

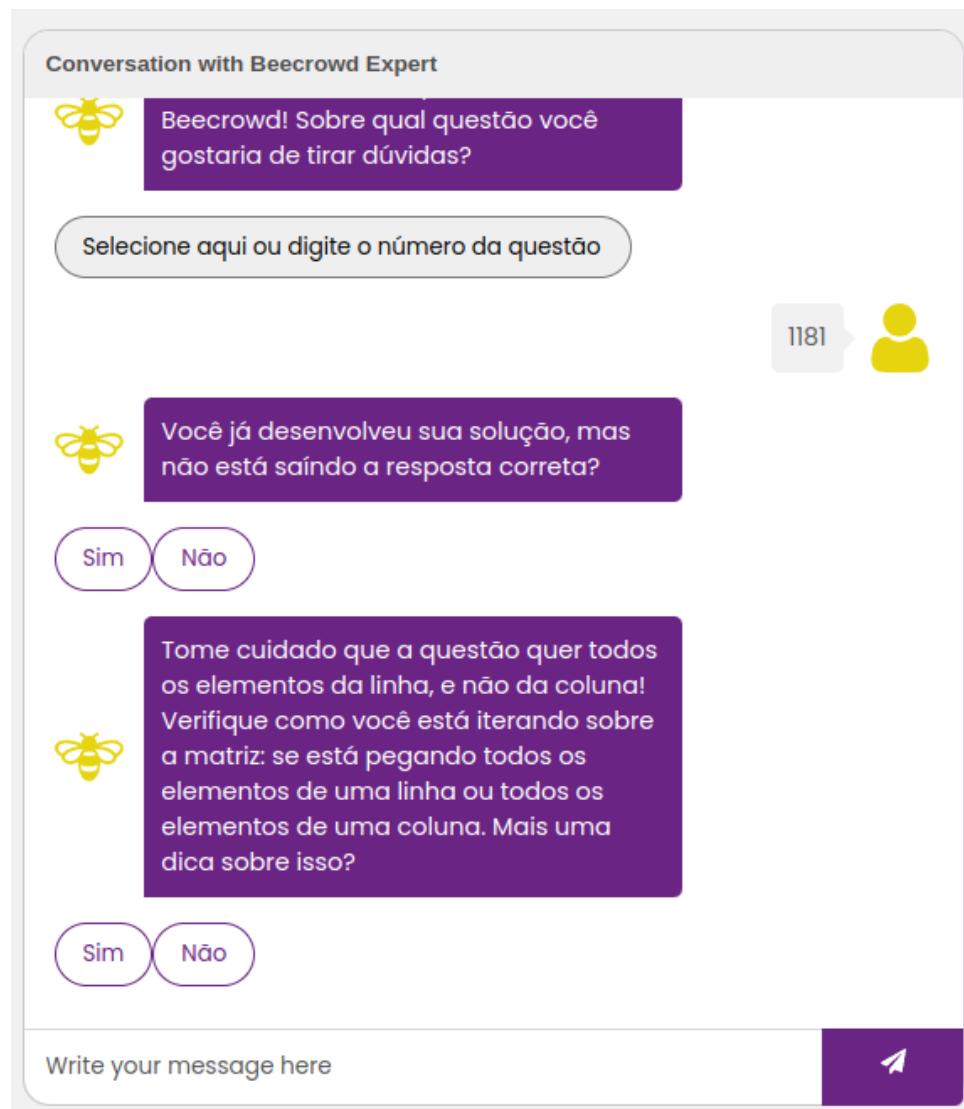
Se a questão for localizada, o chat começa a enviar as perguntas do sistema especialista (Figura 35), com opções de resposta "sim" ou "não", conforme o Requisito Funcional **RF3**. O usuário deve responder clicando no botão "sim" ou "não". Se a resposta for "sim", a resposta adequada para aquela pergunta é exibida junto com a próxima pergunta (Figura 36). Se a resposta for "não", uma nova pergunta será apresentada (Figura 37). Assim, o Requisito Funcional **RF4** é atendido, pois o sistema adapta as perguntas conforme as respostas fornecidas pelo usuário.

Figura 35 – Expert Bee: Primeira Pergunta



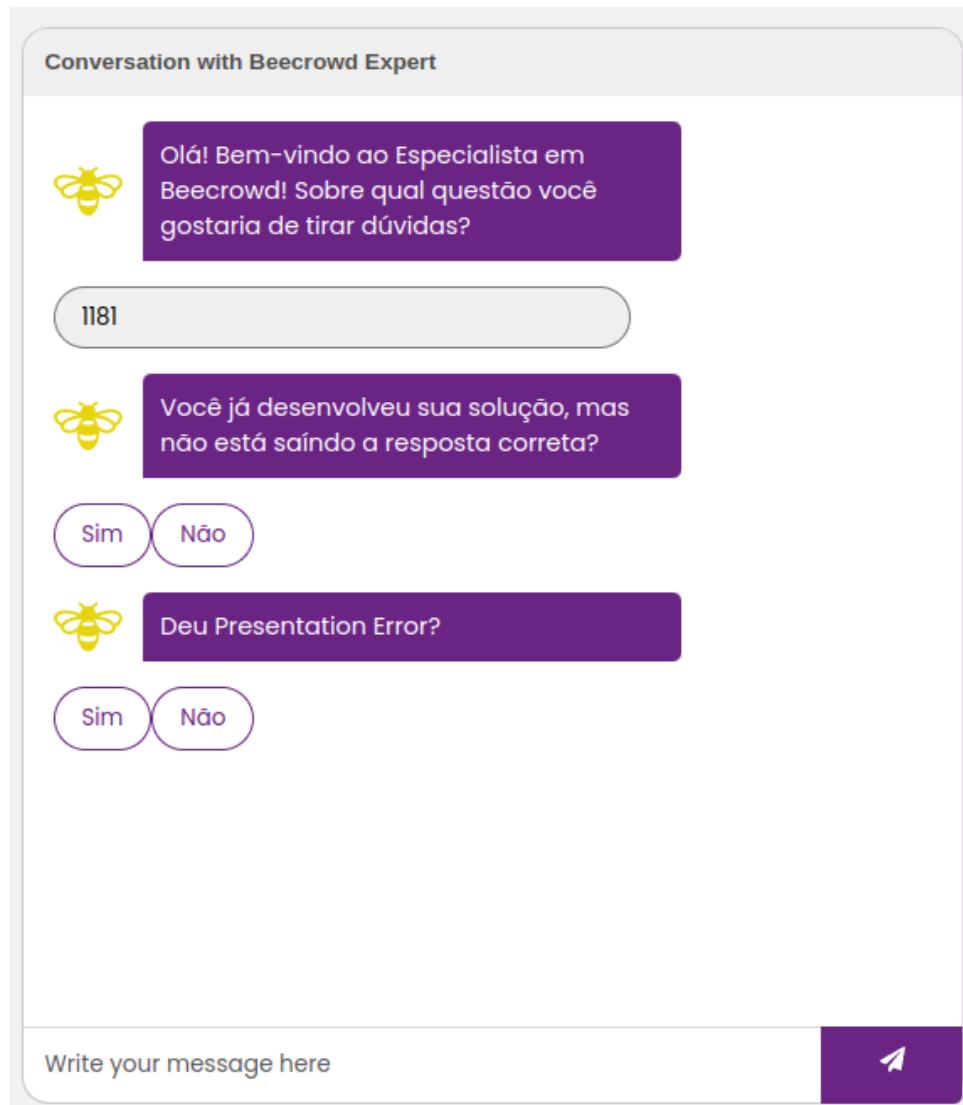
Fonte: Produzido pela autora.

Figura 36 – Expert Bee: Sim para a Primeira Pergunta



Fonte: Produzido pela autora.

Figura 37 – Expert Bee: Não para a Primeira Pergunta



Fonte: Produzido pela autora.

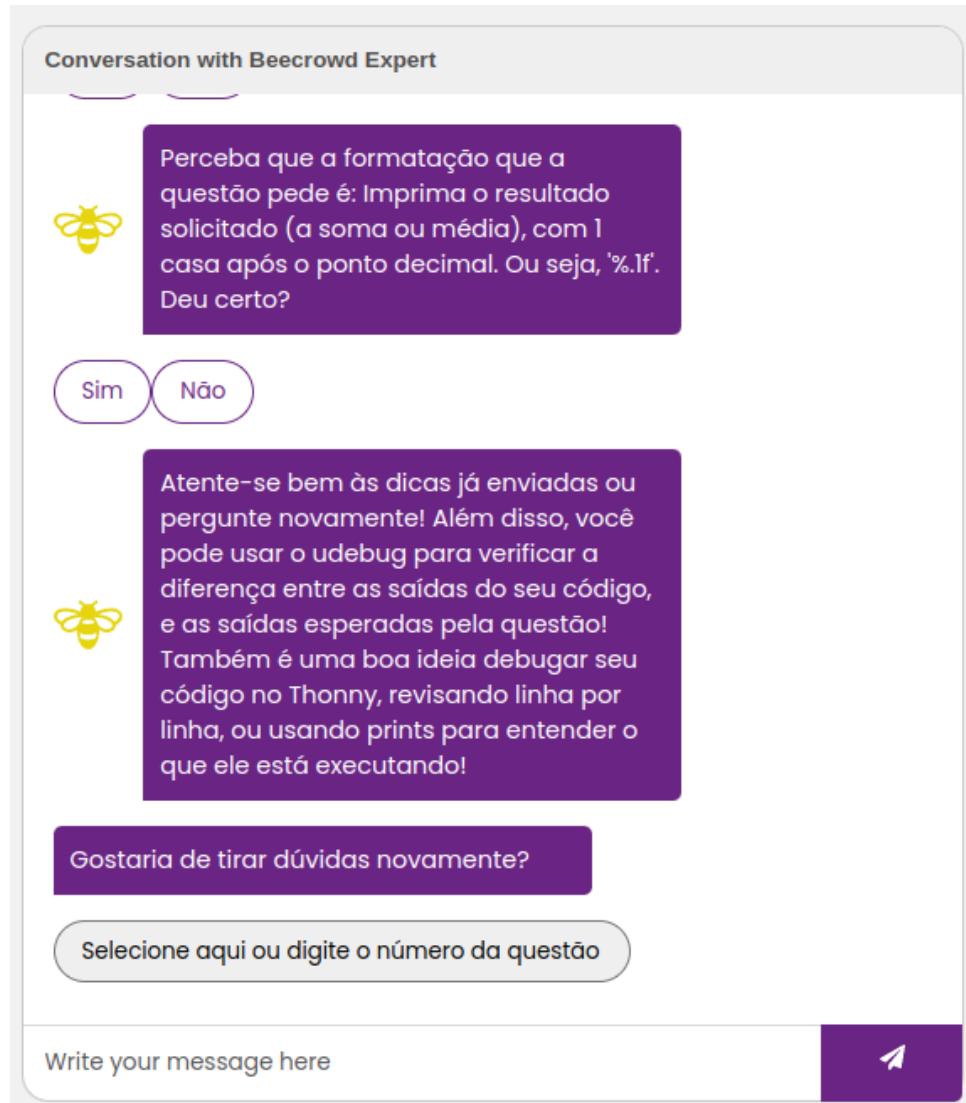
A Figura 38 mostra a resposta final, caso o usuário indique que conseguiu resolver a questão. Já a Figura 39 exibe a resposta com uma dica final, caso o usuário informe que não conseguiu resolver o problema. Em ambos os casos, o usuário tem a opção de inserir um novo número de questão, para iniciar uma nova sessão de dúvidas.

Figura 38 – Expert Bee: Usuário conseguiu



Fonte: Produzido pela autora.

Figura 39 – Expert Bee: Usuário não conseguiu



Fonte: Produzido pela autora.

4.2.9 Adicionar novas questões no Expert Bee

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 `diagnostic/3`.

O predicado `questao/3` associa uma sequência de respostas do usuário a uma nova pergunta, enquanto o predicado `diagnostic/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 `diagnostic/3`.

Com essa estrutura, o sistema será capaz de processar novas questões de forma automática, bastando apenas adicionar o arquivo correspondente ao diretório. Essa abordagem atende e completa o Requisito Não Funcional **RNF5**, conforme descrito na seção [4.2.2](#).

Além disso, foi criado um manual (Apêndice [C](#)) 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.

5 VALIDAÇÃO

Nesta seção, será realizada uma análise crítica da proposta, abordando as vantagens e desvantagens da integração entre o Moodle e o Beecrowd via LTI (Learning Tools Interoperability), bem como da aplicação da aplicação com sistema especialista desenvolvida. Além disso, será apresentada uma avaliação prática sobre o uso da aplicação pelos alunos da disciplina *Programação Orientada a Objetos I* da Universidade Federal de Santa Catarina (UFSC) no segundo semestre de 2024, com o objetivo de verificar se o sistema especialista conseguiu resolver efetivamente as dúvidas dos estudantes.

5.1 INTEGRAÇÃO LTI ENTRE O MOODLE E O BEECROWD

5.1.1 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.

5.1.2 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.

5.2 SISTEMA ESPECIALISTA PARA RESOLUÇÃO DE DÚVIDAS NO BEECROWD

5.2.1 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.

5.2.2 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.

5.3 AVALIAÇÃO PRÁTICA

A avaliação prática foi conduzida por meio da utilização do sistema especialista pelos alunos da disciplina *Programação Orientada a Objetos I* da UFSC, durante o segundo semestre de 2024. Para isso, foram realizados os seguintes passos:

1. **Aplicação do sistema:** Os alunos foram incentivados a usar o sistema especialista para resolver dúvidas relacionadas às questões do Beecrowd. Durante esse período, o uso do sistema foi monitorado para avaliar sua eficácia em termos de resolução de dúvidas.
2. **Análise de resultados:** Foram avaliados o número de dúvidas resolvidas, as dúvidas não esclarecidas e os feedbacks fornecidos pelos alunos.
3. **Melhorias das questões:** Com base na análise dos resultados, dúvidas não resolvidas pelo Bee Expert foram incluídas no sistema 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 4.2.1, para que fosse possível testar a eficácia da ferramenta na hora em que as dúvidas dos alunos surgiram.

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

5.3.1 Resultados

Na Avaliação Prática realizada em 18 de novembro de 2024, foi disponibilizada aos alunos uma lista com as seguintes questões do Beecrowd: 1261, 1281, 1430, 1449, 1483, 1763, 1911, 1953, 2091, 2482, 2492, 2654, 2949 e 2987, para que começassem a resolver durante a aula. 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.

As dúvidas relatadas pelos alunos foram as seguintes:

- **Questão 1911:** 4 alunos apresentaram dúvidas, e todos conseguiram resolvê-las apenas com o auxílio do Bee Expert.
- **Questão 2654:** 2 alunos tiveram dúvidas, todas solucionadas pelo Bee Expert.
- **Questão 1430:** 1 aluno apresentou dúvida e conseguiu resolvê-la com o Bee Expert.

- **Questão 2949:** 1 aluno apresentou dúvida, solucionada pelo Bee Expert.
- **Questão 1483:** 1 aluno teve dúvida, também solucionada pelo Bee Expert.
- **Questão 1261:** 2 alunos apresentaram dúvidas específicas que não puderam ser resolvidas, pois o Bee Expert, à época, fornecia apenas uma ideia geral de resolução.

No total, de 11 dúvidas relatadas, o Bee Expert conseguiu resolver 9, resultando em uma taxa de sucesso de aproximadamente 81%.

5.3.2 Pontos de Melhoria

- **Questão 1261:** As dúvidas específicas dos alunos foram incorporadas ao sistema, enriquecendo o conteúdo da questão para turmas futuras.
- **Questões 1911 e 2654:** Apesar do sucesso na resolução, foi identificado um ponto de melhoria: o Bee Expert apresentava diretamente a solução completa após a primeira interação. No entanto, os alunos demonstraram interesse em orientações iniciais sobre como resolver as questões utilizando dicionários em Python. Esse ajuste foi implementado, permitindo ao usuário solicitar essas informações antes da solução.
- **Questão 1430:** A dúvida do aluno surgiu por falta de entendimento do método `split()` do Python. Como melhoria, o arquivo dessa questão no Bee Expert agora inclui uma explicação detalhada sobre o uso do método.

6 CONSIDERAÇÕES FINAIS

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.

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. Foi realizada uma análise detalhada do Moodle, destacando suas funcionalidades e como ele pode ser integrado de forma eficiente ao Beecrowd para potencializar o processo de aprendizagem. A proposta de integração dessas duas plataformas, por meio de um plugin LTI (Learning Tools Interoperability) desenvolvido pelo Beecrowd, visa simplificar o ensino e tornar as atividades de programação mais acessíveis e eficazes para os alunos.

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.

6.1 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 (ChatGPT):** 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.

REFERÊNCIAS

ACADEMIC, Beecrowd. **Beecrowd Academic.** [S.l.: s.n.], 2021. Disponível em: <<https://www.beecrowd.com.br/academic/pt>>. Acesso em: 9 nov. 2023. Citado na p. 29.

BARTH, A. **HTTP State Management Mechanism.** [S.l.: s.n.], 2011. Disponível em: <<https://datatracker.ietf.org/doc/html/rfc6265>>. Acesso em: 13 nov. 2024. Citado na p. 68.

BEECROWD. **Beecrowd.** [S.l.: s.n.], 2021. Disponível em: <<https://www.beecrowd.com.br/judge/pt>>. Acesso em: 9 nov. 2023. Citado nas pp. 25–31.

BERSSANETTE, João Henrique; FRANCISCO, Antonio Carlos de. 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. Araranguá: [s.n.], 2018. P. 348–354. Disponível em: <<https://www.academia.edu/download/83230343/428-25-1253-1-10-20180622.pdf>>. Acesso em: 20 jul. 2023. Citado nas pp. 18, 25.

BEZ, Jean Luca; TONIN, Neilor A. URI Online Judge e a Internacionalização da Universidade. **Revista Eletrônica de Extensão da Uri**, Erechim, v. 10, n. 18, p. 237–249, 2014. Disponível em: <<https://docplayer.com.br/11099240-Uri-online-judge-e-a-internacionalizacao-da-universidade-uri-online-judge-and-the-university-internationalization.html>>. Acesso em: 20 jul. 2023. Citado nas pp. 18, 19, 25–28.

BEZ, Jean Luca; TONIN, Neilor A. 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). Athens: [s.n.], 2012. P. 1–5. Disponível em: <<https://search.proquest.com/openview/33325d7946128d0f1096abbe8b0b3664/1?pq-origsite=gscholar&cbl=1976352>>. Acesso em: 20 jul. 2023. Citado nas pp. 18, 19, 25–27, 30–32.

BOHANEC, Marko; BRATKO, I.; RAJKOVIC, V. Expert system for decision making. **Sistemica**, v. 1, n. 1, p. 145–157, 1990. Disponível em: <<http://www-ai.ijs.si/MarkoBohanec/pub/IFIP1983.pdf>>. Acesso em: 9 nov. 2024. Citado na p. 39.

BRAY, T. **The JavaScript Object Notation (JSON) Data Interchange Format.**

[S.I.: s.n.], 2017. Disponível em: <<https://www.rfc-editor.org/rfc/rfc8259>>.

Acesso em: 13 nov. 2024. Citado na p. 67.

CAMPOS, Cassio P. de; FERREIRA, Carlos E. BOCA: um sistema de apoio a competições de programação. In: SOCIEDADE Brasileira de Computação. São Paulo: [s.n.], 2004. Disponível em: <<https://research.tue.nl/en/publications/boca-um-sistema-de-apoio-a-competi%C3%A7%C3%B5es-de-programa%C3%A7%C3%A3o>>. Acesso em: 8 nov. 2023. Citado nas pp. 23, 24, 31, 32.

CHAVES, José Osvaldo *et al.* Uma Ferramenta Baseada em Juízes Online para Apoio às Atividades de Programação de Computadores no Moodle. **Revista Novas**

Tecnologias na Educação, v. 11, n. 3, 2013. Disponível em:

<https://www.researchgate.net/profile/Angelica-Castro-4/publication/332584946_Uma_Ferramenta_Baseada_em_Juizes_Online_para_Apoyo_as_Atividades_de_Programacao_de_Computadores_no_Moodle/links/61bb3d081d88475981f2cc8b/Uma-Ferramenta-Baseada-em-Juizes-Online-para-Apoyo-as-Atividades-de-Programacao-de-Computadores-no-Moodle.pdf>. Acesso em: 10 nov. 2024. Citado nas pp. 52, 58.

CLARK, Keith L.; MCCABE, Frank; MCCABE, F. G. **PROLOG: a language for implementing expert systems.** 1980. Diss. (Mestrado) – Imperial College of Science e Technology. Department of Computing. Disponível em:

<https://d1wqxts1xzle7.cloudfront.net/101473629/classics_210D58B0-libre.pdf?1682420871=&response-content-disposition=inline%3B+filename%3DProlog_a_language_for_implementing_exper.pdf&Expires=1731202575&Signature=fuXhBL~FdUeD8lPTRewUWnxVLkcskz28DYkkhpqSjflp4oDoPpBriAYMvFMZ-LXnkV69Ff3DdM13ofVGfLxqVdEGwBXiD3wYZzMp-hx633ZZ7tJobEHWbXQRkZeY0Y2COAZW~maaTEBmlU4NoVfp9-iVNTNSaDrtV~jm-LrqNhgLKXHYQVBP0zRMqtM0RgfaXVacxeLieUzCeVNli3wP78USukC1scfa7T-RoZSMR9YJstlyeoXgka53UQXG4EZLpT0VCd0LFiuBbF7Jj5BZPdfSOwm7WKAeKX-JyT-x3ZxxLQPAOKnqL0d9gEvo9PjiUjQ6hHJwnB-bumwc7melNw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>. Acesso em: 6 nov. 2023. Citado nas pp. 39, 40.

CODERUNNER. **CodeRunner.** [S.I.: s.n.], 2023. Disponível em:

<<https://coderunner.org.nz/>>. Acesso em: 13 nov. 2023. Citado nas pp. 33, 34.

CRUZ, Allan Kássio Beckman Soares da *et al.* 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). [S.I.]: Sociedade Brasileira de Computação, 2022. P. 754–764. Disponível em: <https://doi.org/10.5753/sbgames_estendido.2022.225898>. Acesso em: 20 jul. 2023. Citado nas pp. 17, 18, 23, 42, 43, 56, 58.

DUNSTAN, Neil. 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**, p. 302–308, 2013. Disponível em: <https://www.researchgate.net/profile/Angelica-Castro-4/publication/332584946_Uma_Ferramenta_Baseada_em_Juizes_Online_para_Apoio_as_Atividades_de_Programacao_de_Computadores_no_Moodle/links/61bb3d081d88475981f2cc8b/Uma-Ferramenta-Baseada-em-Juizes-Online-para-Apoio-as-Atividades-de-Programacao-de-Computadores-no-Moodle.pdf>. Acesso em: 11 nov. 2024. Citado nas pp. 53–55, 58, 59.

FERREIRA, Rafael Makaha Gomes. **BROMS: Brazilian Online Marathon Scoreboard**. 2022. Diss. (Mestrado) – Faculdade Unb Gama, Universidade de Brasília, Brasília. Disponível em: <<https://bdm.unb.br/handle/10483/30740>>. Acesso em: 6 nov. 2023. Citado nas pp. 18, 19.

FIELDING, R. **Hypertext Transfer Protocol – HTTP/1.1**. [S.I.: s.n.], 1999. Disponível em: <<https://datatracker.ietf.org/doc/html/rfc2616>>. Acesso em: 13 nov. 2024. Citado na p. 67.

FRANÇA, Allyson Bonetti; SOARES, José Marques. Sistema de apoio a atividades de laboratório de programação via Moodle com suporte ao balanceamento de carga. In: ANAIS do XXII SBIE - XVII WIE. [S.I.]: SBIE, 2011. P. 710–719. Disponível em: <https://www.researchgate.net/profile/Jose-Soares-14/publication/268337343_Sistema_de_apoio_a_atividades_de_laboratorio_de_programacao_via_Moodle_com_suporte_ao_balanceamento_de_carga/links/564309dd08aeacf8938a73a/Sistema-de-apoio-a-atividades-de-laboratorio-de-programacao-via-Moodle-com-suporte-ao-balanceamento-de-carga.pdf>. Acesso em: 19 jul. 2023. Citado na p. 17.

FRANCISCO, Rodrigo; JÚNIOR, Cleon Pereira; AMBRÓSIO, Ana Paula. Juiz Online no ensino de Programação Introdutória - Uma Revisão Sistemática da Literatura. In: ANAIS do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016). [S.I.: s.n.], 2016. P. 11–20. Disponível em: <https://www.researchgate.net/profile/Cleon-Pereira-Junior/publication/309778045_Juiz_Online_no_Esino_de_Programacao_Introdutoria_-Uma_Revisao_Sistematica_da_Literatura/links/582318d608aeb45b58893e6d/Juiz-Online_no_Ensino_de_Programacao_Introdutoria_-Uma_Revisao_Sistematica_da_Literatura.pdf>. Acesso em: 19 jul. 2023. Citado na p. 17.

Online-no-Ensino-de-Programacao-Introductoria-Uma-Revisao-Sistematica-da-Literatura.pdf>. Acesso em: 8 nov. 2023. Citado nas pp. 17, 23, 24.

FREITAS, Larissa Mage de. **Análise de Usabilidade do Módulo Laboratório Virtual de Programação do Moodle**. 2016. Diss. (Mestrado) – Universidade Federal de Santa Catarina, Araranguá. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/165178>>. Acesso em: 19 jul. 2023. Citado na p. 17.

GALASSO, Rafael Hernandez; MOREIRA, Benjamin Grando. Integração do ambiente BOCA com o ambiente Moodle para avaliação automática de algoritmos. In: COMPUTER ON THE BEACH. Ponta Grossa: [s.n.], 2014. P. 22–31. Disponível em: <<https://periodicos.univali.br/index.php/acotb/article/view/5292>>. Acesso em: 9 nov. 2023. Citado nas pp. 30, 31, 35, 44–46, 57.

GALVÃO, Leandro; FERNANDES, David; GADELHA, Bruno. Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. In: ANAIS do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016). [S.I.: s.n.], 2016. P. 140–149. Disponível em: <https://www.researchgate.net/profile/Leandro-Carvalho-3/publication/309892315_Juiz_online_como_ferramenta_de_apoio_a_uma_metodologia_de_ensino_hibrido_em_programacao/links/5da882a792851caa1babdcc5/Juiz-online-como-ferramenta-de-apoio-a-uma-metodologia-de-ensino-hibrido-em-programacao.pdf>. Acesso em: 8 nov. 2023. Citado na p. 24.

KESTEREN, A. van. **Cross-Origin Resource Sharing, W3C Working Draft WD-cors-20100727**. [S.I.: s.n.], 2010. Disponível em: <<http://www.w3.org/TR/2010/WD-cors-20100727/>>. Acesso em: 13 nov. 2024. Citado na p. 67.

LIMA, Gustavo Marques. **Gamma Online Judge: projeto de criação de uma plataforma para o armazenamento das questões das maratonas UnB de programação**. 2022. Diss. (Mestrado) – Faculdade Unb Gama, Universidade de Brasília, Brasília. Disponível em: <<https://bdm.unb.br/handle/10483/34516>>. Acesso em: 6 nov. 2023. Citado nas pp. 19, 24.

LIMA, José Maria Maciel. Plataforma Moodle: a educação por mediação tecnológica. **Revista Científica Multidisciplinar Núcleo Do Conhecimento**, v. 7, n. 6, p. 17–37, 2021. Disponível em:

<<https://www.nucleodoconhecimento.com.br/educacao/plataforma-moodle>>. Acesso em: 19 jul. 2023. Citado na p. 17.

LOBB, Richard; HARLOW, Jenny. Coderunner: a tool for assessing computer programming skills. In: 1. ACM Inroads. Ponta Grossa: [s.n.], 2016. P. 47–51. Disponível em:

<https://dl.acm.org/doi/abs/10.1145/2810041?casa_token=cV5554XMVswAAAAA:8SVdEhKBneuW43kadkaSv8ELNdN41F376mTCSo7a98yPV9CJFAp-8vADXTAbD9nURn0Lmzne-W_f>. Acesso em: 16 nov. 2023. Citado nas pp. 18, 33, 34, 46–48, 56, 57.

MASSE, Mark. **REST API design rulebook**. [S.l.: s.n.], 2011. Disponível em: <[https://www.swi-prolog.org/pldoc/doc_for?object=section\(%27packages/http.html%27\)](https://www.swi-prolog.org/pldoc/doc_for?object=section(%27packages/http.html%27))>. Acesso em: 13 nov. 2024. Citado na p. 67.

MASSÉ, Mark. **REST API Design Rulebook**. [S.l.]: Sebastopol: O'reilly Media, Inc., 2011. Disponível em: <https://books.google.com.br/books?hl=pt-PT&lpg=&id=eABpzyTcJNIC&oi=fnd&pg=PR3&dq=REST+API+Design+Rulebook&ots=vB0vZ_jdMD&sig=tvXanZ39ViurHJ7fWrRIOvBwlglgI>. Acesso em: 13 nov. 2023. Citado na p. 41.

MOODLE. **Moodle Developer Resource centre**. [S.l.: s.n.], 2023. Disponível em: <<https://moodledev.io/>>. Acesso em: 10 nov. 2023. Citado nas pp. 33–37.

NASR, Osman A.; TALAB, Ahmed; AL-GAHTANI, Said S. **Building of a Rule-Based Expert System for Academic Advising via Web Expert System Tools**. 2018. Diss. (Mestrado) – FKING Khalid University - College of Business - Dept. of MIS. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/57393851/17-44-2-11-libre.pdf?1537170728=&response-content-disposition=inline%3B+filename%3DBuilding_of_a_Rule_Based_Expert_System_f.pdf&Expires=1731285754&Signature=C37Ci002J7H1n9uza0UAvGlGagiZrKvuAZEfy~W62aqwIJ16r7EE2p~dfpFUYSdCeB4nuDA7T5-Q2cnEZQ3kcuHanae21k2emCT0KW6IsfNmCmr15uEMMnL3LYIc~54H3R4cC7dehTcCUSV-LqCJXBxNawDWj~VZHGWsqUu5pqU1NDIx059ZZvEWLr512Le8rKIF~XTIKW7x1V8vCe8KYjtacSSt4f8hZID~a5AC61al2hGuv2DWqbt2I1hy7gZLGqu5iIBuDh11I~Zvw6gco~rGB42xA31G1LCDRzqJ93ja7KExsDFuVXglwPSLb5qIED4~cj~TMnIUNq2~aZg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>. Acesso em: 11 nov. 2024. Citado nas pp. 53, 58.

PIEKARSKI, Ana Elisa Tozetto *et al.* PROGRAMAÇÃO COMPETITIVA EM UM PROJETO DE EXTENSÃO PARA O ENSINO TÉCNICO EM INFORMÁTICA. **Revista Conexão UEPG**, Ponta Grossa, v. 19, n. 1, p. 1–15, 2023. Disponível em:

<<https://revistas.uepg.br/index.php/conexao/article/view/21239>>. Acesso em: 9 nov. 2023. Citado na p. 25.

RIBEIRO, Ralph Breno *et al.* 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). [S.I.: s.n.], 2018. P. 805–814. Disponível em:

<<http://milanesa.ime.usp.br/rbie/index.php/sbie/article/view/8040>>. Acesso em: 8 nov. 2023. Citado na p. 23.

RICHARDSON, Leonard; AMUNDSEN, Mike; RUBY, Sam. **RESTful Web APIs**. [S.I.]: Sebastopol: O'reilly Media, Inc., 2011. Disponível em:

<<https://books.google.com.br/books?hl=pt-PT&lpg=&id=wWnGAAAQBAJ&oi=fnd&pg=PR2&dq=restful+web+apis&ots=Fi9itG-75e&sig=q64KUvXJzi4zFGQhnJnIzxjL2LM>>. Acesso em: 13 nov. 2023. Citado na p. 41.

RODRÍGUEZ-DEL-PINO, Juan Carlos. **Moodle plugins directory: Virtual Programming Lab**. [S.I.: s.n.], 2023. Disponível em:

<https://moodle.org/plugins/mod_vpl>. Acesso em: 19 jul. 2023. Citado na p. 17.

RODRÍGUEZ-DEL-PINO, Juan Carlos; ROYO, Enrique Rubio; FIGUEROA, Zenón Hernández. 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: [s.n.], 2012. Disponível em: <<http://hdl.handle.net/10553/9773>>. Acesso em: 16 nov. 2023. Citado nas pp. 32, 49–51, 57, 58.

RUSSELL, Stuart; NORVIG, Peter. Inteligência Artificial: tradução da terceira edição. In: Disponível em: <[https://www.kufunda.net/publicdocs/Intelig%C3%AAncia%20Artificial%20\(Peter%20Norvig,%20Stuart%20Russell\).pdf](https://www.kufunda.net/publicdocs/Intelig%C3%AAncia%20Artificial%20(Peter%20Norvig,%20Stuart%20Russell).pdf)>. Acesso em: 10 nov. 2024. Citado na p. 40.

SALES, André Barros de; JUNIOR, Edson Costa; SALES, Márcia Barros de. Utilização de Problemas da Maratona de Competição de Programação e Juízes Eletrônicos como Estratégia de Ensino em um Curso de Graduação em Engenharia de Software. In: ANAIS do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016). [S.I.: s.n.], 2016. P. 210–219. Disponível em: <<http://milanesa.ime.usp.br/rbie/index.php/sbie/article/view/6701>>. Acesso em: 8 nov. 2023. Citado na p. 24.

SANTOS, Joanna C. S.; RIBEIRO, Admilson R. L. JOnline: proposta preliminar de um juiz online didático para o ensino de programação. In: ANAIS do Simpósio Brasileiro de Informática na Educação (SBIE 2011). [S.I.: s.n.], 2011. P. 964–967. Disponível em: <<http://milanesa.ime.usp.br/rbie/index.php/sbie/article/view/1863>>. Acesso em: 8 nov. 2023. Citado nas pp. 23, 24.

SINGLA, Jimmy. The diagnosis of some lung diseases in a prolog expert system. **International Journal of Computer Applications**, v. 78, n. 15, p. 37–40, 2013. Disponível em: <https://d1wqxts1xzle7.cloudfront.net/47387099/pxc3891435-libre.pdf?1469050904=&response-content-disposition=inline%3B+filename%3DThe_Diagnosis_of_Some_Lung_Diseases_in_a.pdf&Expires=1731202594&Signature=K2HWohOs96t-T0see9D56TwDTQ8bnsejjvZ~htc5V5ip0dTMBYdFB2976WEyKqDFfbNMOLVpMd9BW-3mEROPqGwLFr361QYC5WyCXXNtM00Y8s6kJUzH05P1ULctIg9EVVorrTMyyRCyB65Azm-pA3eWtwKtfFzwx7x5RSnSiuhYqFYbpF0BqOGmUZ8DhkGZt8~y07J14W9xsTcMgtdYYaiajRbPcxgixepiWlOBbYZRwmDgN0W03qBrhXKLqy69tFcNzPNWs1PU~Nzh5uDgb71zBTQvi9UYU3R61g6zwokJF1m7Am6GfL6vk~MMje6dtXQL0Jq3HzDN3iUH~jQUmA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>. Acesso em: 9 nov. 2024. Citado nas pp. 39, 40.

VERDAGUER, Júlia. **O que é LTI e como ele pode melhorar seu ecossistema de aprendizagem**. [S.I.: s.n.], 2021. Disponível em: <<https://moodle.com/pt-br/news/o-que-e-e-como-pode-melhorar-seu-ecossistema-de-aprendizagem/>>. Acesso em: 14 nov. 2023. Citado nas pp. 20, 38.

VICKERS, Stephen P; BOOTH, Simon. Creating Environments For Learning Through Instigating A Community Of Developers: A JISC-funded Project. **Learning Tools Interoperability (LTI): A Best Practice Guide**, 2014. Disponível em: <https://ltiapps.net/guide/LTI_Best_Practice.pdf>. Acesso em: 14 nov. 2023. Citado na p. 38.

VPL. Virtual Programming Lab for Moodle (VPL). [S.I.: s.n.], 2021. Disponível em: <<https://vpl.dis.ulpgc.es/>>. Acesso em: 13 nov. 2023. Citado nas pp. 32, 33.

WASIK, Szymon *et al.* A Survey on Online Judge Systems and Their Applications. **Acm Computing Surveys**, v. 51, n. 1, p. 1–34, 2018. Disponível em: <<https://dl.acm.org/doi/abs/10.1145/3143560>>. Acesso em: 8 nov. 2023. Citado na p. 23.

WIELEMAKER, J.; HUANG, Z.; MEIJ, L. VAN DER. SWI-Prolog and the web. In: **THEORY and Practice of Logic Programming**. [S.I.: s.n.], 2008. P. 363–392. Disponível

em: <<https://doi.org/10.1017/S1471068407003237>>. Acesso em: 10 nov. 2024. Citado nas pp. 40, 41.

WIELEMAKER, J.; SCHRIJVERS, T. *et al.* SWI-Prolog. In: THEORY and Practice of Logic Programming. [S.I.: s.n.], 2012. P. 67–96. Disponível em: <<https://doi.org/10.1017/S1471068411000494>>. Acesso em: 10 nov. 2024. Citado na p. 40.

WIELEMAKER, Jan. In: SWI-PROLOG 6.1: Reference Manual. [S.I.: s.n.], 2012. Disponível em: <<https://rabbit.eng.miami.edu/class/een594/SWI-Prolog-6.1.2.pdf>>. Acesso em: 10 nov. 2024. Citado na p. 40.

WIELEMAKER, Jan. **SWI-Prolog HTTP support**. [S.I.: s.n.], 2024. Disponível em: <[https://www.swi-prolog.org/pldoc/doc_for?object=section\(%27packages/http.html%27\)](https://www.swi-prolog.org/pldoc/doc_for?object=section(%27packages/http.html%27))>. Acesso em: 13 nov. 2024. Citado na p. 66.

WIELEMAKER, Jan. SWI-Prolog version 7 extensions. In: WORKSHOP on Implementation of Constraint and Logic Programming Systems and Logic-based Methods in Programming Environments. [S.I.: s.n.], 2014. Disponível em: <<https://www.swi-prolog.org/download/publications/swi7.pdf>>. Acesso em: 10 nov. 2024. Citado na p. 40.

WIELEMAKER, Jan; LAGER, Torbjorn; RIGUZZI, Fabrizio. SWISH: SWI-Prolog for Sharing. In: Disponível em: <<https://arxiv.org/pdf/1511.00915.pdf>>. Acesso em: 10 nov. 2024. Citado na p. 41.

Apêndices

APÊNDICE A – MANUAL DE CONFIGURAÇÃO DA LTI BEECROWD NO MOODLE

O administrador do Moodle deve configurar a LTI do Beecrowd.

1. Acessar o moodle com uma conta de administrador
2. Selecionar a opção: **Plugins**

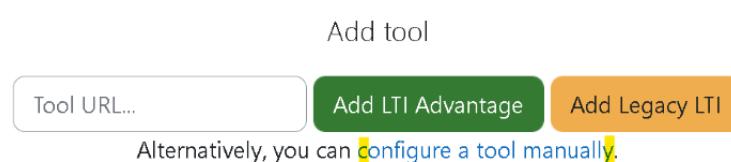
Site administration



3. Em Activity Modules, selecionar **External tool -> Manage tools**

A screenshot of the Moodle Activity modules page. The left sidebar lists various activity types: Manage activities, Common activity settings, Assignment, Assignment settings, Submission plugins, Manage assignment submission plugins, File submissions, Online text submissions, Feedback plugins, Manage assignment feedback plugins, Feedback comments, Annotate PDF, File feedback, Offline grading worksheet, Book, Chat, Database, External tool, and Manage tools. The "Manage tools" link is highlighted with a yellow box.

4. Selecionar a opção **configure a tool manually**



5. Na página a seguir, preencher os campos conforme abaixo:

Tool name: *beecrowd*

Tool URL: <https://api.beecrowd.com/>

Tool description: *beecrowd LTI1.3 moodle integration*

LTI version: *LTI 1.3*

Public key type: *Keyset URL*

Public keyset: <https://api.beecrowd.com/lti/jwks>

Initiate login URL: <https://api.beecrowd.com/lti/login>

Redirection URI(s): <https://api.beecrowd.com/lti/launch>

Custom parameters: “Manter vazio”

Tool configuration usage: *Show in activity chooser and as a preconfigured tool*

Default launch container: *New window*

Supports Deep Linking (Content-Item Message): “*Assinalar o check box*”

Content Selection URL: <https://api.beecrowd.com>

Icon URL: <https://moodle.beecrowd.io/pluginfile.php/15/course/summary/lgo.png>

Secure icon URL: <https://moodle.beecrowd.io/pluginfile.php/15/course/summary/lgo.png>

Services

IMS LTI Assignment and Grade Services: *Use this service for grade sync and column management*

IMS LTI Names and Role Provisioning: *Use this service to retrieve members' information as per privacy settings*

Tool Settings: *Use this service*

Privacy

Share launcher's name with tool: *Always*

Share launcher's email with tool: *Always*

Accept grades from the tool: *Always*

Force SSL: “*Não assinalar*”

Miscellaneous

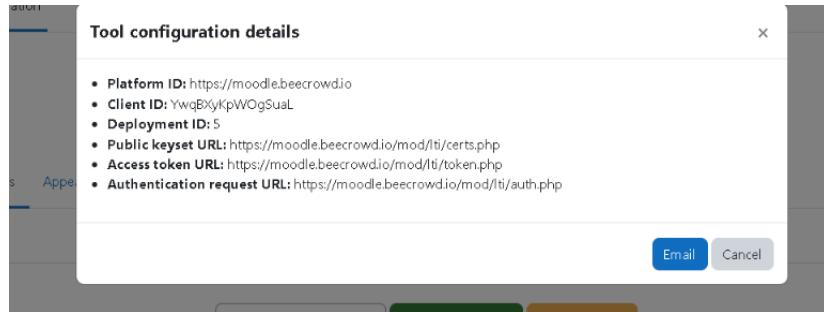
Default organisation ID: *Site hostname*

Organisation ID: “*Manter Vazio*”

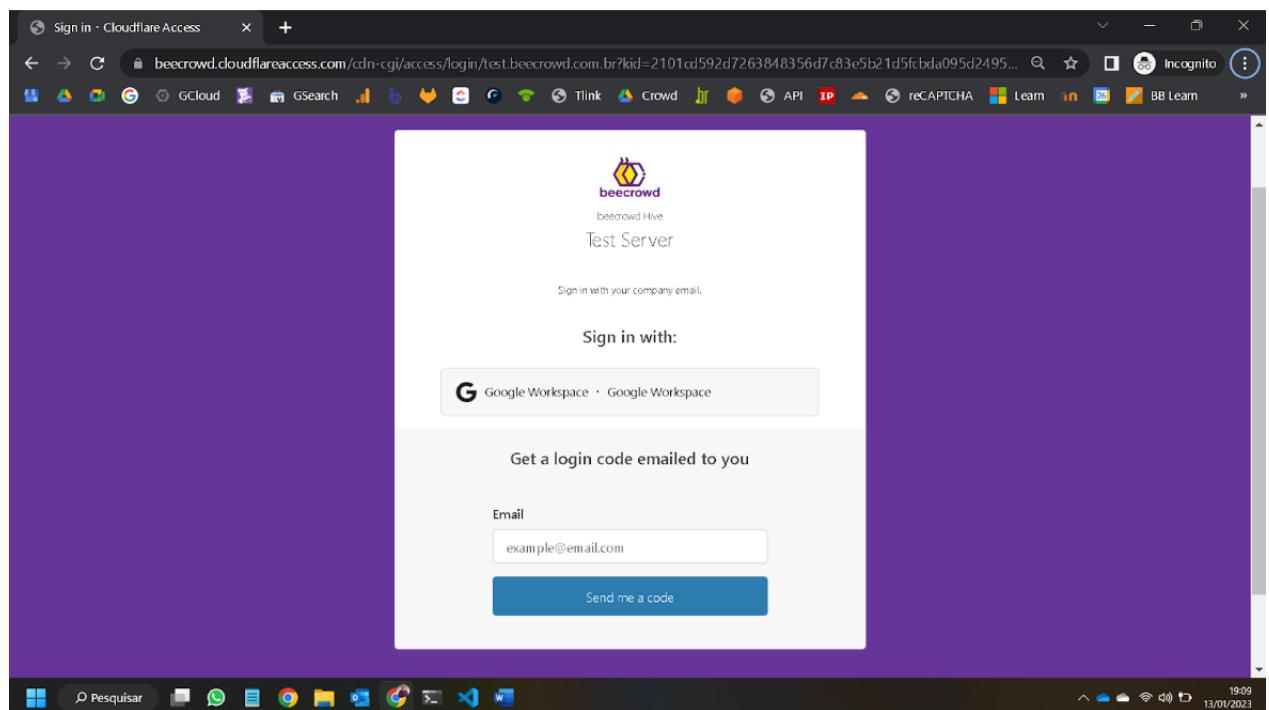
Organisation URL: <https://beecrowd.com/>

6. Após configurar as informações acima, você deve enviar para o Beecrowd a

captura de tela da seguinte tela (exemplo na foto abaixo) através do botão de e-mail com os dados que eles devem inserir na parte deles para estabelecer a comunicação:



7. Será exibida uma página de autorização. Basta preencher o email e aguardar a aprovação. Uma vez feito, é necessário recarregar a sessão e iniciar os testes novamente.



APÊNDICE B – MANUAL DE USO LTI BEECROWD NO MOODLE

B.1 UTILIZAR LTI DO BEECROWD JÁ CONFIGURADA NO MOODLE

Este Manual mostra como o professor cria uma atividade Beecrowd no Moodle, como um aluno acessa essa atividade, e como o professor pode transferir as notas do Beecrowd para o Moodle.

B.1.1 Criando a atividade

1. Entre em modo edição
2. Vá em adicionar uma atividade ou recurso:

The screenshot shows the Moodle course settings interface. At the top, there are tabs: Course, Settings, Participants, Grades, Reports, and More. The 'Settings' tab is active. Below the tabs, there's a section titled 'General' with a pencil icon. A blue callout box highlights the '+ Add an activity or resource' button. The 'Add topic' link is visible below it. On the right side of the 'General' section, there are 'Collapse all' and a three-dot menu icons.

3. Selecione a atividade “beecrowd”

The screenshot shows the 'Add an activity or resource' modal. At the top, there's a search bar and tabs for Starred, All, Activities, and Resources. The Activities tab is selected. In the main area, there are two rows of icons. The first row includes Assignment, beecrowd (which has a cursor icon over it), Book, Chat, Choice, and Database. The second row includes External tool, Feedback, File, Folder, Forum, and Glossary. Each icon has a star and info button below it. The 'beecrowd' icon is highlighted with a mouse cursor.

4. Na página que abrir:

- a) Em **General > Activity Name**, escreva o nome da atividade - como essa primeira vai ser para uso do professor, para entrar no Beecrowd Academic, você pode escolher um nome tipo “Acesso ao Beecrowd Acadêmico”;
- b) Em **Grade > Type**, selecione *None*;
- c) Em **Common module settings > Availability**, selecione “*Hidden on Course Page*”;
- d) Clique em **Save and return to course**.

✖ Adding a new External tool

The screenshot shows the configuration interface for adding a new external tool in Moodle. It includes sections for General, Grade, Common module settings, and various other options like Restrict access, Tags, and Competencies. The 'Type' dropdown in the Grade section is highlighted with a blue border.

Activity name: Academic Access

Type: None

Availability: Hide on course page

Save and return to course | Save and display | Cancel

5. Agora, a atividade que você criou vai aparecer na página do curso. Clique nela.

▼ General ↗



- Você vai ser redirecionado para a página do Beecrowd Academic. Caso você já esteja logado, vai aparecer a lista de suas disciplinas.

The screenshot shows the 'LIST DISCIPLINES' page on beecrowd.com.br/academic/en/home. The page title is 'beecrowd academic'. It displays a table with columns 'ID' and 'NAME'. The listed disciplines are:

ID	NAME
011533	Moodle Demo II
011532	Moodle Demo I
011477	Mackenzie
010038	Disciplina do Administrador
010034	Disciplina do coach
009872	Principal
009322	SQL

Você pode criar uma disciplina, ou selecionar uma disciplina já existente. Para cada disciplina, você pode criar tarefas, e, para cada tarefa, você pode selecionar diferentes questões, e arrumar outras configurações, como data de início, de fim, etc (Veja as imagens da próxima página).

Criar ou editar tarefa:

The screenshot shows the 'HOMEWORK' page for the discipline 'Moodle Demo I' on beecrowd.com.br/academic/en/disciplines/view/11532. The page title is 'beecrowd academic'. It shows a table with one homework entry:

#	ID	HOMEWORK	DEADLINE
1	038593	New Task	September 30, 2023 at 12:00 PM

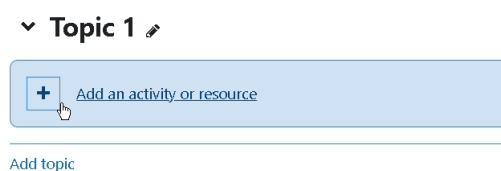
Editando a tarefa:

The screenshot shows the 'Edit Homework' page in Moodle. At the top, it says 'YOU ARE HERE: DISCIPLINES / MOODLE DEMO I / EDIT HOMEWORK'. Below that, under 'DISCIPLINE INFORMATION', the 'DISCIPLINE' field contains 'Moodle Demo I'. Under 'HOMEWORK INFORMATION', the 'HOMEWORK TITLE' field contains 'New Task'.

Selecionando questões:

The screenshot shows the 'Edit Homework' page in Moodle. It includes fields for 'START' (09/11/2023 18:00) and 'DEADLINE' (09/30/2023 12:00). Under 'PROGRAMMING LANGUAGES', 'Python 3.9 (Python 3.9.4)' is selected. A notice box states: 'NOTICE: YOU MUST EXPLICITLY SELECT ALL THE LANGUAGES AND VERSIONS YOU WANT TO ACCEPT: BY CHOOSING C++, FOR INSTANCE, THE ACADEMIC WILL NOT ACCEPT SUBMISSIONS IN C++17. SIMILAR BEHAVIOR CAN BE EXPECTED FROM PYTHON AND OTHER LANGUAGES.' Under 'EXERCISES', '1000 - Hello World!' is listed. A 'SUGGESTION' button is visible. There is also a checkbox for 'ENABLE ANTI-PLAGIARISM MECHANISMS'.

7. Agora que você tem a disciplina com a tarefa desejada, volte ao Moodle e clique para adicionar uma atividade ou recurso:



8. Selecione a atividade Beecrowd:

Add an activity or resource

The screenshot shows a Beecrowd activity card. It has a search bar at the top. Below it, tabs for 'Starred', 'All', 'Activities', and 'Resources' are visible. The 'Activities' tab is selected. A card for 'beecrowd' is shown, featuring its logo, the text 'beecrowd', and two icons: a star and an info symbol.

9. Clique em “Selecionar conteúdo”:

beecrowd Integration Demo

Course Settings Participants Grades Reports More ▾

Adding a new External tool to Topic 1

General

Activity name



Show more...

Select content



Privacy

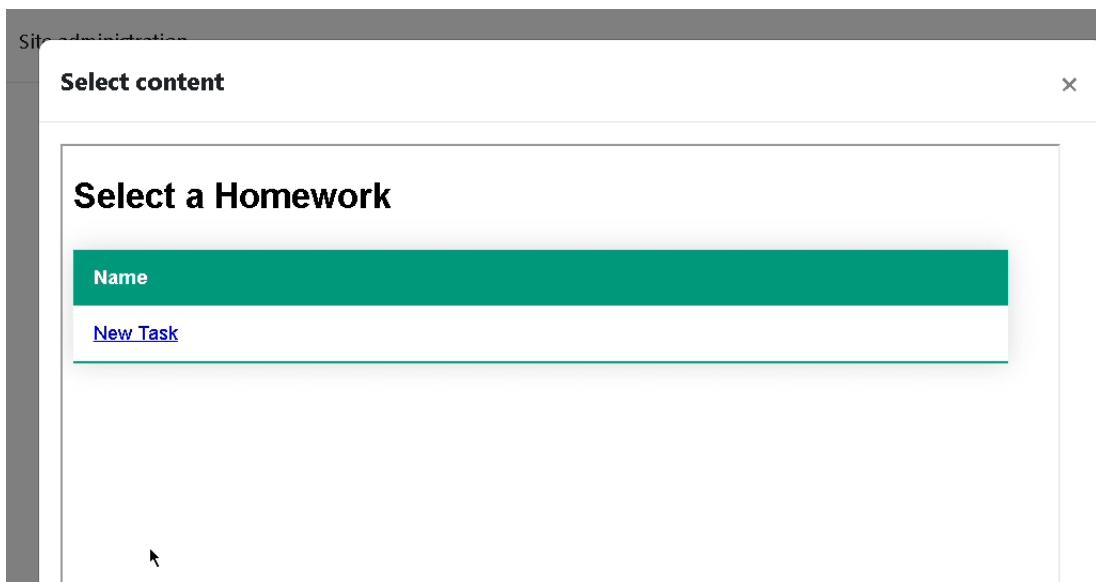
Grade

10. Selecione a Disciplina criada no Beecrowd Academic:

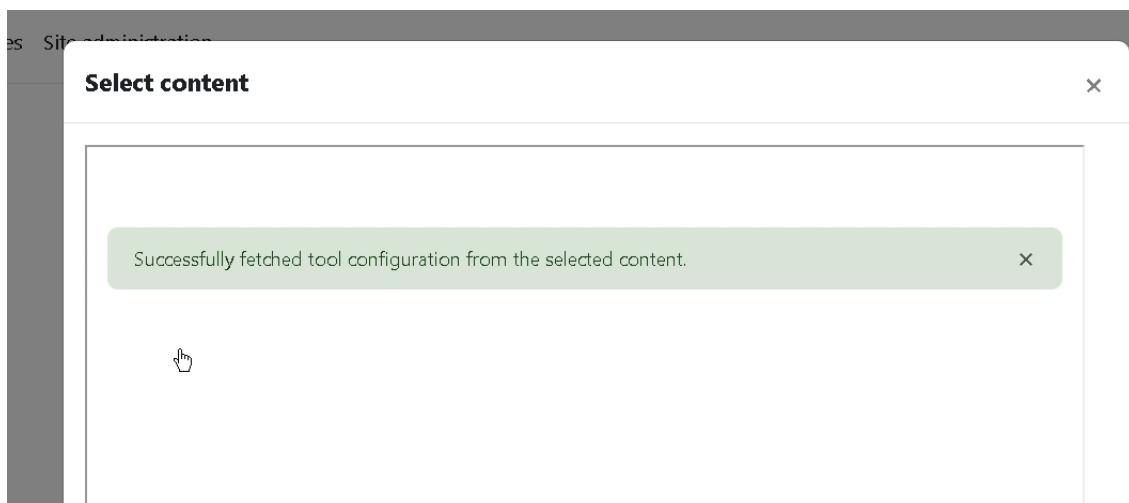
The screenshot shows a modal dialog titled "Select content" with a sub-titile "Select a Discipline". The list contains the following items:

Name
SQL
Principal
Disciplina do coach
Disciplina do Administrador
Mackenzie
<u>Moodle Demo I</u>
<u>Moodle Demo II</u>

11. Selecione a tarefa criada no Beecrowd Academic:



Vai aparecer uma mensagem de confirmação que a tarefa foi selecionada com sucesso:



12. Você vai ser redirecionado de volta para a tela anterior, e agora é só clicar em “Save and return do course”:

Site administration

Activity name ! Tema de Casa: New Task

Show more... Select content

> Privacy

> Grade

> Common module settings

> Restrict access

> Tags

> Competencies

Send content change notification ?

Save and return to course Save and display Cancel

Agora a tarefa aparecerá dessa forma na página do curso:

Site administration

Hidden from students

+ Add an activity or resource

Add topic

▼ Topic 1 ✎

 EXTERNAL TOOL
Tema de Casa: New Task ✎ ✖

+ Add an activity or resource

Add topic

B.1.2 Entrando como aluno

- O aluno clica na tarefa do Beecrowd, criada pelo professor:

The screenshot shows a Moodle course page titled "beecrowd Integration Demo". The navigation bar includes "Course", "Participants", "Grades", and "Competencies". Below the navigation, there are two expandable sections: "General" and "Topic 1". Under "Topic 1", there is a card for a task named "Tema de Casa: New Task". The card features the Beecrowd logo and a blue button labeled "SOLVED". A cursor arrow points to the "SOLVED" button.

General

Topic 1

Tema de Casa: New Task

Topic 2

- O aluno é direcionado à tarefa do Beecrowd, sem precisar criar um usuário, e já pode começar a resolver os exercícios da tarefa:

The screenshot shows the Beecrowd task page for "Hello World!". The top navigation bar includes links for HOME, PROFILE, NEWS (with 527 notifications), OPPORTUNITIES, ACADEMIC, CONTESTS, FORUM, PROBLEMS, SUBMISSIONS, RANKS, and SIGN OUT. The task details section shows the following information:

- DEADLINE:** Moodle Demo I
- PROFESSOR:** New Task
- HOMEWORK:** 1 problems
- STARTS:** 9/11/23, 9:00 PM
- ACCEPT:** PYTHON 3.9

The task instructions say "testing moodle access". The progress bar is nearly full. The submission table shows one submission from user "1000" with status "ACCEPTED".

#	PROBLEM	SUBMISSION	ACCEPTED	LEVEL	WEIGHT
1	1000 ✓ Hello World!	02	35493426	5	100

Resolvendo um exercício:

The screenshot shows a Beecrowd challenge titled "Hello World!". The challenge details are as follows:

- Author:** beecrowd | 1000
- Title:** Hello World!
- Description:** Jean Bez, beecrowd Brasil
- Time limit:** 1
- Source code:** (shown below)
- Language:** Python 3.9

The source code provided is:

```
1 # -*- coding: utf-8 -*-
2 ...
3 ...
4 Escreva a sua solução aqui
5 Code your solution here
6 Escriba su solución aquí
7 ...
```

A large red speech bubble icon with the text "hello!" is displayed on the left side of the challenge page.

Digita o código e clica em Submit:

The screenshot shows the Beecrowd submission interface. The "SOURCE CODE" field contains the following code:

```
1 print('Hello World!')
```

Below the code editor, there is a message: "s print this". At the bottom of the interface are two buttons: "CODE YOUR SOLUTION AND SUBMIT!" and a green "SUBMIT" button, which has a cursor pointing to it.

O código é avaliado pelo juiz online da plataforma:

The screenshot shows a Beecrowd interface. At the top, there is a navigation bar with links: PROFILE, NEWS (with a red badge showing 527), OPPORTUNITIES, ACADEMIC, CONTESTS, FORUM, PROBLEMS, SUBMISSIONS, RANKS, and SIGN OUT. Below the navigation bar, a green banner displays the message "Code received! Your submission was placed in the judging queue...". The main area features a "SOURCE CODE" section with a blue icon of a document. To the right of the icon is a "EDIT & SUBMIT" button. Below this, a sub-section titled "SUBMISSION # 35495777" provides detailed information about the submission:

PROBLEM:	1000 - Hello World!
ANSWER:	Accepted
LANGUAGE:	Python 3.9 (Python 3.9.4) [+1s]
RUNTIME:	0.063s
FILE SIZE:	21 Bytes
MEMORY:	-
SUBMISSION:	9/12/23, 1:04:29 AM

Below this, another "SOURCE CODE" section shows the submitted Python code:

```
1 print('Hello World!')
```

B.1.3 Notas

- Após seguir os passos anteriores descritos nesse documento, o professor pode clicar na tarefa do Beecrowd que ele criou para os alunos:

The screenshot shows the Moodle course settings interface. At the top, there are tabs: Course (which is selected), Settings, Participants, Grades, Reports, and More. Below the tabs, there are two expandable sections: 'General' and 'Topic 1'. In the 'General' section, there is a card for 'Academic Access' with a blue icon, the text 'Academic Access', and a button labeled 'Hidden from students'. In the 'Topic 1' section, there is a card for 'Tema de Casa: New Task' with a blue icon, the text 'Tema de Casa: New Task', and a small hand cursor icon pointing at the text.

- O professor pode visualizar quem respondeu as questões, e, ao clicar em “Send Grades”, as notas são enviadas para o Moodle, e o professor consegue visualizar as notas lá:

Página da tarefa no Beecrowd Academic:

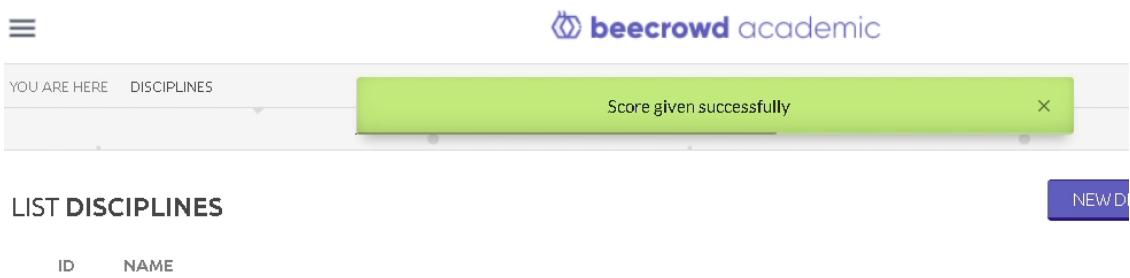
The screenshot shows the Beecrowd Academic task progress page for a task titled 'HOMEWORK NEW TASK'. The page includes details like the discipline ('Moodle Demo I'), creation date ('September 11, 2023 at 8:56 PM'), update date ('September 11, 2023 at 11:20 PM'), and languages ('PYTHON 3.9'). It also shows the start date ('09/11/2023') and deadline ('09/30/2023'). On the right, there's a summary box showing '1 PROBLEMS' and '18 DAYS TO SOLVE'. Below this, the 'PROGRESS New Task' section displays student statistics. A student named 'Estudante Moodle beecrowd' has completed 100% of the problems. The 'SEND GRADES' button is highlighted with a mouse cursor. Other buttons visible include 'EXPORT CSV', 'HORUS PLAGIARISM', 'DUPLICATE HOMEWORK', and 'STUDENT LIST'.

Clica em “Send Grades”:

DUPLICATE HOMEWORK | SEND GRADES

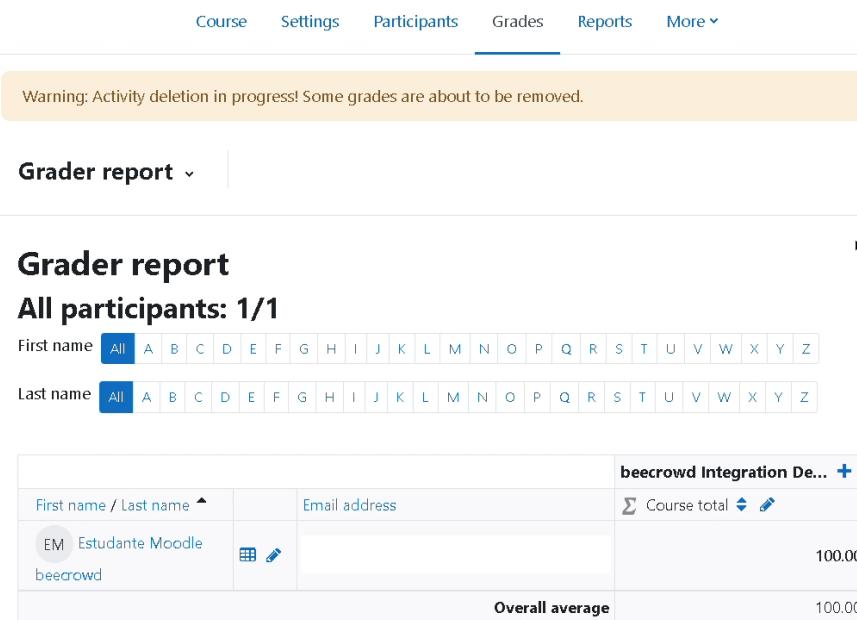

SCORE * TOTAL

Mensagem de sucesso que as notas foram enviadas:



The screenshot shows the Beecrowd Academic interface. At the top, there's a navigation bar with 'YOU ARE HERE' and 'DISCIPLINES'. A green notification bar in the center says 'Score given successfully'. Below it, there's a section titled 'LIST DISCIPLINES' with columns for 'ID' and 'NAME'. A blue button labeled 'NEWD...' is visible on the right.

Notas agora estão moodle:



The screenshot shows the Moodle Grades page. At the top, there are tabs: Course, Settings, Participants, Grades (which is selected), Reports, and More. A yellow warning box says 'Warning: Activity deletion in progress! Some grades are about to be removed.' Below that, there's a 'Grader report' section with a dropdown menu. The main part shows a table for 'Grader report' with 'All participants: 1/1'. The table has columns for 'First name' (with a dropdown menu from 'All' to 'Z'), 'Last name' (with a dropdown menu from 'All' to 'Z'), 'Email address' (containing 'EM Estudante Moodle beecrowd'), and 'beecrowd Integration De... +'. The table shows one participant with a score of 100.00. The overall average is also listed as 100.00.

APÊNDICE C – COMO ADICIONAR RESPOSTAS PARA NOVAS QUESTÕES NO EXPERT BEE

Basta adicionar um arquivo dentro do diretório backend, com o nome no formato questao_numeroDaQuestao.pl. Cada arquivo deve seguir o seguinte padrão:

C.1 ESTRUTURA DO ARQUIVO

C.1.1 Definição do módulo

O arquivo deve começar com a declaração do módulo. O nome do módulo deve ser igual ao nome do arquivo (sem a extensão .pl). Além disso, o módulo deve exportar dois predicados principais:

- questao/3
- diagnostico/3

Exemplo:

```
:– module(questao_2465, [questao/3, diagnostico/3]).
```

C.1.2 Predicado questao/3

O predicado questao/3 deve ser usado para mapear uma sequência de respostas a uma nova pergunta. Ele segue o formato:

```
questao(NúmeroDaQuestao, RespostasAteAgora, Pergunta).
```

- NúmeroDaQuestao: Número da questão a que o arquivo se refere.
- RespostasAteAgora: Lista de respostas fornecidas pelo usuário até o momento.
- Pergunta: Pergunta a ser exibida com base nas respostas recebidas.

Exemplo:

```
questao(2465, [], "Você quer uma dica de qual solução adotar pra esse problema?").  
questao(2465, ["sim"], "Você pode usar um while True para verificar, a partir da posição
```

C.1.3 Predicado diagnostico/3

O predicado diagnostico/3 deve ser usado para fornecer um diagnóstico final ou mensagem de feedback com base nas respostas completas do usuário. Ele segue o formato:

```
diagnostico(NúmeroDaQuestao, RespostasCompletas, Diagnóstico).
```

- NumeroDaQuestao: Número da questão.
- RespostasCompletas: Lista de respostas fornecidas.
- Diagnostico: Diagnóstico ou feedback final.

Exemplo:

```
diagnostico(2465, ["sim", "sim", "sim"], "Legal! Parabéns!").  
diagnostico(2465, _, "Atente-se bem às dicas já enviadas ou pergunte novamente!").
```

C.2 CONSIDERAÇÕES IMPORTANTES

- Certifique-se de que todas as perguntas e diagnósticos estejam relacionados ao número da questão especificado.
- O sistema utilizará a lista de respostas fornecida até o momento para determinar qual pergunta exibir em seguida. Portanto, as sequências de respostas devem ser consistentes.
- Para mensagens genéricas ou casos em que as respostas não correspondam a nenhuma sequência específica, use _ como curinga no campo RespostasCompletas no predicado diagnostico/3.

Com essa estrutura, o sistema poderá processar novas questões automaticamente, bastando que o arquivo correspondente seja adicionado ao diretório.