

UNIVERSIDADE FEDERAL DO PARANÁ

RAFAEL STAVITSKI COSTA DE OLIVEIRA

REBAEV: UMA ABORDAGEM PARA RECOMENDAÇÃO DE QUESTÕES UTILIZANDO
O CAMINHO DO BOM APRENDIZ E EVIDÊNCIAS DE APRENDIZADO

CURITIBA PR

2022

RAFAEL STAVITSKI COSTA DE OLIVEIRA

REBAEV: UMA ABORDAGEM PARA RECOMENDAÇÃO DE QUESTÕES UTILIZANDO
O CAMINHO DO BOM APRENDIZ E EVIDÊNCIAS DE APRENDIZADO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Andrey Ricardo Pimentel.

CURITIBA PR

2022

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA CIÊNCIA E TECNOLOGIA

Oliveira, Rafael Stavitski Costa de

REBAEV : uma abordagem para recomendação de questões
utilizando o caminho do bom aprendiz e evidências de aprendizado /
Rafael Stavitski Costa de Oliveira. – Curitiba, 2022.

1 recurso on-line : PDF.

Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de
Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: Prof. Dr. Andrey Ricardo Pimentel.

1. Informática – Estudo e ensino. 2. Computadores. 3. Programação
(Computadores). 4. Ensino gerenciando por computador. I. Pimentel,
Andrey Ricardo. II. Universidade Federal do Paraná. Programa de Pós-
Graduação em Informática. IV. Título.

Bibliotecário: Nilson Carlos Vieira Junior CRB-9/1797

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **RAFAEL STAVITSKI COSTA DE OLIVEIRA** intitulada: **ReBaEv: Uma Abordagem para Recomendação de questões utilizando o caminho do bom aprendiz e evidências de aprendizado.**, sob orientação do Prof. Dr. ANDREY RICARDO PIMENTEL, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 09 de Março de 2022.

Assinatura Eletrônica

11/03/2022 13:48:04.0

ANDREY RICARDO PIMENTEL

Presidente da Banca Examinadora

Assinatura Eletrônica

10/03/2022 09:38:44.0

EDUARDO TODT

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica

10/03/2022 11:05:44.0

ALEXANDER ROBERT KUTZKE

Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

*Esta dissertação é dedicada aos que
sempre acreditaram e apoiaram pe-
los desafios da vida.*

RESUMO

Esta dissertação tem como objetivo descrever a abordagem ReBaEv, que consiste da recomendação da próxima questão, contextualizada no ensino da programação de computadores em cursos de computação. Esta abordagem adota duas estratégias de recomendação, o caminho do bom aprendiz e as evidências de aprendizado. O estado da arte foi identificado por meio de um mapeamento sistemático da literatura e revelou as técnicas empregadas em recomendação de questões. Considera-se o modelo do estudante, representado por um grafo genético, que é utilizado para investigar e monitorar a evolução dos aprendizes, por meio das suas habilidades. Objetiva-se utilizar o caminho do bom aprendiz, que é gerado por alunos que obtiveram êxito na realização dos exercícios, como forma de replicar um modelo competente a outros aprendizes, ao considerar o campo do aprendizado. Faz-se uso de um método de identificação automática de evidências de aprendizado, através de códigos fonte de programação, de modo a extrair a quantidade de evidências utilizadas nos exercícios. As evidências de aprendizado são empregadas na recomendação ao calcular um peso que é atribuído para os exercícios. Experimentos realizados com o objetivo de validar a abordagem mostram que as estratégias empregadas, tomando-se como base o caminho do bom aprendiz e as evidências de aprendizado, foram promissoras e favoráveis para os três cenários considerados no trabalho, quanto aos quesitos de adequação e engajamento. Por meio da avaliação com especialistas percebe-se que os exercícios recomendados, referentes aos cenários propostos, foram adequados para os diferentes tipos de alunos, e que a recomendação em uma ordem mais interessante, contribuiu com o aumento do engajamento dos alunos. Finalmente, a abordagem ReBaEv recomenda exercícios recebendo como parâmetro o estado do aluno, sendo assim, relevante com o objetivo de aprimorar o processo de ensino-aprendizagem, ao melhor atender o aluno de acordo com as suas necessidades.

Palavras-chave: Recomendação da Próxima Questão, Caminho do Bom Aprendiz, Avaliação Automática de Evidências de Aprendizado, Ensino da Programação de Computadores

ABSTRACT

This dissertation aims to describe the ReBaEv approach, which consists of the recommendation of the next question, contextualized in the teaching of computer programming in computing courses. This approach adopts two recommendation strategies, the good learner path and the learning evidence. The state of the art was identified through a systematic mapping of the literature and revealed the techniques used in question recommendation. The student's model is considered, represented by a genetic graph, which is used to investigate and monitor the learners' evolution, through their abilities. The objective is to use the path of the good learner, which is generated by students who were successful in performing the exercises, as a way to replicate a competent model to other learners, when considering the field of learning. A method of automatic identification of learning evidence is used, through programming source codes, in order to extract the amount of evidence used in the exercises. The learning evidence is used in the recommendation when calculating a weight that is assigned to the exercises. Experiments carried out with the aim of validating the approach show that the strategies employed, taking as a basis the path of the good learner and the evidence of learning, were promising and favorable for the 3 scenarios considered in the work, regarding the items of adequacy and engagement. Through the evaluation with experts, it can be seen that the recommended exercises, referring to the proposed scenarios, were suitable for different types of students, and that the recommendation, in a more interesting order, contributed to increased student engagement. Finally, the ReBaEv approach recommends exercises taking the student's state as a parameter, thus being relevant in order to improve the teaching-learning process, by better serving the student according to their needs.

Keywords: Recommendation of the Next Question, Path of the Good Learner, Automatic Assessment of Learning Evidence, Teaching Computer Programming

LISTA DE FIGURAS

2.1	Arquitetura de um Sistema Tutor Inteligente. Fonte: Nkambou et al. (2010) . . .	19
2.2	Habilidades no domínio da programação. Fonte: Porfirio (2020)	24
2.3	Habilidades no domínio da programação. Fonte: Porfirio (2020)	25
2.4	Exemplo de grafo genético no domínio da programação. Fonte: Oliveira (2016) .	26
2.5	Interface do BOCA. Fonte: de Campos e Ferreira (2004)	30
2.6	Interface do The Huxley. Fonte: de Barros Paes et al. (2013)	30
2.7	Esquema do FARMA-ALG. Fonte: Kutzke (2015).	31
2.8	Execução do parser resultado em uma árvore sintática abstrata. Fonte: Porfirio (2020)	35
2.9	Exemplo de código fonte na linguagem C. Fonte: Porfirio (2020)	35
3.1	Ilustração do processo de busca e filtros utilizados na seleção dos trabalhos. . . .	41
3.2	Distribuição ao longo dos anos.. . . .	42
3.3	Tipos de algoritmos.	42
3.4	Soluções utilizadas no modelo do estudante.. . . .	43
3.5	Domínio dos trabalhos.	44
3.6	Trabalhos que realizaram experimentos para validação.	45
4.1	Esquema da abordagem.	48
4.2	Exemplo de progresso das questões corretas dos alunos.. . . .	49
4.3	Exemplo de aluno regular.	49
4.4	Arquitetura de recomendação para aluno regular.	50
4.5	Exemplo de aluno travado.	50
4.6	Arquitetura para aluno travado e ótimo.	52
4.7	Exemplo de aluno ótimo.	52
4.8	Exemplo de conjunto de bons aprendizes.	53
4.9	Distribuição das notas dos bons aprendizes.	54
4.10	Caminho de exercícios.	54
4.11	Demonstração da ordem dos exercícios.	55
4.12	Caminho de exercícios de um aprendiz.	56
4.13	Exercícios selecionados - bom aprendiz 1.	56
4.14	Exercícios selecionados - bom aprendiz 2.	57
4.15	Exemplo de identificação automática de evidências de um código fonte.	60
4.16	Ilustração da atualização do modelo do aprendiz.	64
4.17	Ilustração da atualização do modelo do aprendiz.	64

4.18	Ilustração da atualização do modelo do aprendiz.	65
4.19	Sugestão de exercícios.	65
4.20	Caminho de exercícios do aprendiz.	66
4.21	Exercícios recomendados para o aluno regular.. . . .	66
4.22	Caminho de exercícios do aprendiz.	67
4.23	Exercícios recomendados para o aluno travado.	67
4.24	Caminho de exercícios do aprendiz.	68
4.25	Exercícios recomendados para o aluno ótimo.	68
5.1	Quantidade de anos de ensino de programação de computadores dos participantes.	71
5.2	Concordância.. . . .	72
5.3	Resultados sobre os exercícios recomendados para aluno regular são adequados..	72
5.4	Concordância.. . . .	72
5.5	Resultados sobre o engajamento para aluno regular.	73
5.6	Concordância.. . . .	73
5.7	Resultados sobre os exercícios recomendados para aluno travado são adequados..	73
5.8	Concordância.. . . .	73
5.9	Resultados sobre se os exercícios similares são úteis para ajudar alunos travados.	74
5.10	Concordância.. . . .	74
5.11	Resultados sobre os exercícios recomendados para aluno ótimo são adequados. .	74
5.12	Concordância.. . . .	75
5.13	Resultados sobre o engajamento para aluno ótimo	75
5.14	Concordância.. . . .	75
5.15	Resultados sobre a contribuição da recomendação para continuar utilizando a ferramenta e resolver os exercícios.. . . .	76
5.16	Concordância.. . . .	76
5.17	Resultados sobre exercícios recomendados em ordem mais adequada contribuem para o engajamento.. . . .	76
C.1	Slide 1.	95
C.2	Slide 2.	95
C.3	Slide 3.	96
C.4	Slide 4.	96
C.5	Slide 5.	97
C.6	Slide 6.	97
C.7	Slide 7.	98
C.8	Slide 8.	98
C.9	Slide 9.	99
C.10	Slide 10.. . . .	99

D.1	Questionário - página 1..	100
D.2	Questionário - página 2..	101
D.3	Questionário - página 3..	102
D.4	Questionário - página 4..	103
D.5	Questionário - página 5..	104

LISTA DE TABELAS

2.1	Comportamentos de Sistemas Tutores Inteligentes. Fonte: Vanlehn (2006)	19
2.2	Habilidades computadas pela detecção automática. Fonte: Porfirio (2020)	36
3.1	Strings de busca.	40
3.2	Critérios de inclusão e exclusão.	40
3.3	Questões de pesquisa.	41
3.4	Resultados QP2.. . . .	43
3.5	Resultados QP4.. . . .	44
3.6	Resultados QP5.. . . .	45
4.1	Exercícios e distâncias relativas.	56
4.2	Exercícios e distâncias relativas - iteração 1.	57
4.3	Exercícios e distâncias relativas - iteração 2.	58
4.4	Exercícios e quantidades médias de evidências.	62
4.5	Porcentagem cumprida e somatório das evidências.	63
4.6	Exercícios e distâncias relativas.	67

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CARACTERIZAÇÃO DO PROBLEMA	14
1.2	OBJETIVOS	14
1.2.1	Geral	15
1.2.2	Específicos	15
1.3	JUSTIFICATIVA	15
1.4	ORGANIZAÇÃO DO DOCUMENTO	16
2	CONCEITOS FUNDAMENTAIS	17
2.1	ENSINO DA PROGRAMAÇÃO	17
2.2	SISTEMAS TUTORES INTELIGENTES	18
2.2.1	Modelo Domínio	20
2.2.2	Modelo do Estudante	20
2.2.3	Modelo Tutorial	22
2.3	MODELAGEM DO ESTUDANTE NO CONTEXTO DO ENSINO DE PRO- GRAMAÇÃO DE COMPUTADORES.	23
2.3.1	Grafo Genético	23
2.4	SISTEMAS DE RECOMENDAÇÃO.	26
2.5	JUÍZES ONLINE	28
2.6	FARMA-ALG.	29
2.6.1	Armazenamento.	32
2.6.2	Recuperação.	32
2.6.3	Manipulação	33
2.6.4	Visualização.	33
2.6.5	Mediação do Erro	33
2.7	AVALIAÇÃO AUTOMÁTICA DE EVIDÊNCIAS DE APRENDIZADO EM CÓDIGOS FONTE	34
3	TRABALHOS RELACIONADOS	38
3.1	MAPEAMENTO SISTEMÁTICO DA LITERATURA	38
3.1.1	Método de Pesquisa	39
3.1.2	Resultados e Discussão	40
3.1.3	Distribuição do Tempo (QP1).	41
3.1.4	Tipo de Algoritmo (QP2)	41
3.1.5	Características Utilizadas na Escolha da Próxima Questão (QP3)	42
3.1.6	Modelo do Estudante (QP4)	43

3.1.7	Domínio (QP5)	43
3.1.8	Experimentos (QP6)	44
3.1.9	Conclusão	45
4	ABORDAGEM REBAEV	48
4.1	ARQUITETURA REBAEV	48
4.1.1	Recomendação Utilizando o Caminho do Bom Aprendiz	49
4.1.2	Recomendação Utilizando Evidências de Aprendizado Aluno Travado	50
4.1.3	Recomendação Utilizando Evidências de Aprendizado Aluno Ótimo	51
4.2	GERAÇÃO DOS CAMINHOS DOS BONS APRENDIZES	53
4.3	CÁLCULO DA DISTÂNCIA RELATIVA	54
4.3.1	Cálculo Aplicado	55
4.4	SELEÇÃO DOS CÓDIGOS FONTE DOS EXERCÍCIOS	57
4.5	IDENTIFICAÇÃO AUTOMÁTICA DE EVIDÊNCIAS DE PROGRAMAÇÃO	58
4.6	CÁLCULO DA MÉDIA DE EVIDÊNCIAS DE PROGRAMAÇÃO POR EXERCÍCIO	62
4.7	GERAÇÃO DA TABELA COM A QUANTIDADE MÉDIA DE EVIDÊNCIAS	62
4.8	HABILIDADES.	62
4.8.1	Construção do Modelo do Aprendiz	63
4.8.2	Atualização do Modelo do Aprendiz	63
4.9	RECOMENDAÇÃO DAS QUESTÕES	64
4.9.1	Recomendação para Aluno Regular	65
4.9.2	Recomendação para Aluno Travado	66
4.9.3	Recomendação para Aluno Ótimo	68
5	VALIDAÇÃO.	69
5.1	METODOLOGIA.	69
5.1.1	Participantes.	69
5.1.2	Método de Pesquisa	69
5.1.3	Cenários.	70
5.1.4	Instrumento de Pesquisa	70
5.2	RESULTADOS ESPERADOS	70
5.3	DESCRIÇÃO DOS EXPERIMENTOS	71
5.4	RESULTADOS	71
5.5	DISCUSSÃO	76
5.5.1	Aluno Regular.	77
5.5.2	Aluno Travado	77
5.5.3	Aluno Ótimo	78
5.5.4	Geral	79

5.6	AMEAÇAS À VALIDADE DOS EXPERIMENTOS	79
5.7	REBAEV E TRABALHOS CORRELATOS	80
5.8	CONSIDERAÇÕES	81
6	CONCLUSÃO	82
6.1	TRABALHOS FUTUROS	83
	REFERÊNCIAS	84
	APÊNDICE A – CAMINHOS DOS BONS APRENDIZES	89
	APÊNDICE B – CONJUNTO DE EXERCÍCIOS DE OPERAÇÕES ARIT- MÉTICAS	91
	APÊNDICE C – SLIDES DA VALIDAÇÃO	95
	APÊNDICE D – QUESTIONÁRIO DA VALIDAÇÃO	100

1 INTRODUÇÃO

A construção do conhecimento é um processo natural que ocorre ao aprender algum assunto novo. O processo de ensino-aprendizagem caracteriza esta construção. Pode-se considerar um assunto como dominado quando se atinge certo nível de maestria a respeito deste. O caminho para chegar até este nível de conhecimento pode ser diferente entre os diferentes aprendizes. Isto é evidenciado pelo fato que alguns podem assimilar um conteúdo sem dificuldade enquanto que outros podem sentir dificuldade e necessitar de alguma ajuda a mais que influencie no seu processo de aquisição do conhecimento (Bauman e Tuzhilin, 2014).

O modelo tradicional de ensino da programação, em uma sala de aula com alunos e um professor, apresenta desafios que precisam ser superados (Zanetti e Oliveira, 2015). Para muitos iniciantes, o aprendizado da programação é bastante difícil conforme evidenciado pela grande taxa de evasão e desistência em disciplinas e cursos que necessitam da programação (Miliszewska e Tan, 2007). Em muitos momentos os alunos necessitam de uma atenção individual e o professor não consegue atender a todos. Essa falta de atenção sugere que existe a necessidade de buscar por alternativas que possam suprir este problema.

Uma alternativa que surge em um mundo digital em constante movimento é o e-learning, que é o aprendizado facilitado e suportado por tecnologias da informação e comunicação. Este se torna uma opção viável para a educação e o ensino em diversas possíveis aplicações computacionais. O modelo tradicional de ensino é limitado pelo tempo e espaço. Chang e Chen (2009) e Chang et al. (2009) mostram que o e-learning pode fornecer um aprendizado de alto nível e conveniente devido a sua grande flexibilidade. O e-learning pode fornecer um ambiente de aprendizado universal, contudo a falta de um instrutor humano ainda é o gargalo ao fornecer um aprendizado efetivo.

Sistemas Tutores Inteligentes (STI) agem em contrapartida ao modelo tradicional de ensino ao considerar as características específicas dos aprendizes de forma a aprimorar o processo de ensino-aprendizagem e torná-lo mais eficaz (Strayer, 2007). Enquanto o e-learning tradicional contribui com a melhora da flexibilização do aprendizado e custos reduzidos, é um STI que tenta cobrir as especificidades dos aprendizes. Um dos principais problemas do e-learning é o fato de fornecer a mesma experiência de aprendizado desconsiderando os conhecimentos prévios, experiências e as preferências entre os diferentes aprendizes evidenciado em (Alhathli et al., 2020). Uma possível solução para este problema é fornecer um caminho de aprendizado personalizado para o aluno. Com o aprendizado personalizado, é possível aumentar a satisfação de um aluno com o processo de aprendizagem junto com a melhora na inovação, coesão, orientação, cooperação, individualização e equidade (Strayer, 2007).

A seleção adaptativa de atividades pode ser considerada como uma maximização do potencial individual de aprendizado dos alunos. Um dos pontos chave é a caracterização individual das fraquezas e qualidades de um aprendiz de modo que um bom caminho de aprendizado considere estes pontos ao ser traçado por uma seleção adaptativa. Em (Huo et al., 2018) destacam-se a remediação da aprendizagem e o planejamento do caminho de aprendizado personalizado como pontos focais para aprimorar a performance durante o aprendizado.

Elaborando a remediação, é necessária a escolha de material exclusivo que seja restringido ao tratamento de uma fraqueza particular em comparação à um material padronizado que causa sobrecarga de informação para o aluno. O domínio de uma determinada habilidade é categorizada por uma componente de conhecimento e isto deve ser considerado ao escolher o material para o aluno. Analisando este eixo, entram os sistemas de recomendação (SR), que

investigam as características de um usuário. Aplicando sistemas de recomendação no ensino personalizado, chega-se perto ao conceito de Sistemas Tutores Inteligentes que será descrito no próximo capítulo.

1.1 CARACTERIZAÇÃO DO PROBLEMA

O ato de ensinar é desafiador considerando que cada estudante possui uma maneira diferente de aprender. Um método de ensino pode ser eficaz para um grupo de alunos, entretanto pode ser falho para outro. Como tornar o ensino abrangente de forma a cobrir os diferentes grupos de aprendizes de modo a dar atenção às particularidades de cada um? Em uma sala de aula, atribui-se a um instrutor a tarefa de ensino. Este geralmente utiliza uma ordem de exercícios de ensino destinada à massa de alunos em que se deixa de lado estas características específicas de cada um.

O processo de ensino-aprendizagem acaba por ser prejudicado ao não considerar as especificidades de cada aprendiz. Ao aplicar uma ordem de exercícios fixa ao todo, há um desequilíbrio entre os diferentes nichos. Isto acaba causando problemas que são perceptíveis no desempenho destes a exemplo de aprendizes com alto desempenho e aqueles que possuem alguma dificuldade. Tendo em vista este cenário, levanta-se a questão: Com um direcionamento personalizado, será que um aprendiz com desempenho aquém ao de um aprendiz bem sucedido não poderia melhorar o seu desempenho no aprendizado? A partir desta perspectiva, deriva-se o problema de como tratar os diferentes grupos de alunos em um contexto educacional.

O problema situado neste trabalho pode ser observado em alunos no contexto do aprendizado da programação. Ao enfrentar exercícios de programação para exercitar os diversos conceitos é comum o aluno se deparar com a situação de errar exercícios em sequência devido a problemas como o não entendimento dos conceitos. Esta situação pode ser muito desmotivadora para o aluno que pode desistir de fazer os exercícios o que acarreta no fracasso da assimilação dos conteúdos por não conseguir prosseguir com os exercícios. Ao não assimilar os conceitos, o aluno tem um desempenho ruim, notas ruins e somado da desmotivação ocorre a evasão ou abandono das disciplinas.

A desmotivação também pode vir de uma repetição sucessiva de exercícios que não apresentem uma sequência interessante e engajadora ao aluno. Dessa forma, o desinteresse do aluno pode agir de maneira disruptiva no processo de aprendizado.

1.2 OBJETIVOS

Delimitando o escopo desta pesquisa, esta dissertação foca em exercícios de programação de computadores. O problema abordado é contextualizado por meio da abordagem de recomendação de questões e a descrição de um modelo do estudante que serve para monitorar a evolução das habilidades dos estudantes. Seguindo, os objetivos gerais que o trabalho espera atingir são descritos em conjunto de alguns objetivos específicos traçados para atingir o objetivo principal.

Esse trabalho é baseado no trabalho de Porfirio (2020), que descreve e caracteriza a avaliação automática de evidências de aprendizado em códigos fonte. Também utiliza-se o modelo do bom aprendiz, que é descrito em Meliana e Nurjanah (2018), na abordagem de recomendação de questões. Finalmente, foca-se nas perícias de programação seguindo o modelo de habilidades descrito no trabalho de Krynski (2013). Esses trabalhos serão contextualizados no segundo capítulo, onde descrevem-se os conceitos fundamentais para este trabalho, e no terceiro capítulo, capítulo referente ao mapeamento sistemático da literatura.

1.2.1 Geral

Este trabalho tem como objetivo criar uma abordagem de escolha da próxima questão baseada no caminho do bom aprendiz e em evidências que permita auxiliar os aprendizes no desenvolvimento das habilidades de programação de computadores. Considerando o caminho de exercícios seguido pelos aprendizes, busca-se recomendar exercícios levando em consideração o que o aluno já fez.

1.2.2 Específicos

Somando-se a abordagem de recomendação de questões junto com o modelo do estudante, espera-se construir um modelo de recomendação da próxima questão e uma forma que permita observar a evolução do conhecimento dos aprendizes. Para alcançar esses objetivos, estabelecem-se os seguintes objetivos específicos:

- Identificar o estado da arte e elaborar um mapeamento sistemático da literatura;
- Investigar abordagens e estratégias de recomendação de exercícios;
- Personalizar a escolha da próxima questão de acordo com o caminho de exercícios dos aprendizes;
- Desenvolver mecanismo para detectar evidências de habilidades em soluções modelo de exercícios;
- Implementar modelo do estudante que permita investigar o estado e evolução do aprendiz;
- Validar a solução proposta de recomendação de questões.

1.3 JUSTIFICATIVA

A seleção personalizada da próxima questão com base no progresso do estudante contribui para guiar os alunos durante a realização dos exercícios por fornecer uma ordem de questões mais apropriada com as suas necessidades. Conforme descrito na Seção 4.1.2, é comum que um aluno apresente dificuldade e fique "travado" em alguma questão. Considerando este fato, o trabalho objetiva uma intervenção de recomendação por meio de questões de igual dificuldade ou até mais fáceis para que haja a promoção da assimilação e entendimento no momento em que a dificuldade foi percebida. Destaca-se, também, o aumento do engajamento e a motivação dos estudantes com a presença da orientação do sistema. Por isso, seguir uma seleção personalizada com base no estado do estudante aprimora o processo de ensino-aprendizagem e é considerada relevante de acordo com (Dang e Ghergulescu, 2018). Para acompanhar a evolução dos alunos, evidencia-se o modelo do estudante, um subconjunto do domínio, que serve para representar o conhecimento que é adquirido pelos aprendizes durante o tempo e refletir o conhecimento em dado momento.

Este trabalho busca apresentar uma alternativa à maneira tradicional de sugestão de exercícios que é focada em um grupo homogêneo de alunos para cobrir as necessidades específicas dos diferentes aprendizes. Espera-se, que com a personalização da recomendação de exercícios, possa-se aumentar o engajamento e a motivação dos alunos durante o processo de ensino-aprendizagem e aumentar a eficácia ao adquirir os conhecimentos em programação bem como a autonomia. Abordagens automáticas de recomendação também auxiliam professores na sugestão

de exercícios o que ajuda a reduzir a carga de trabalho que geralmente consiste em atender um número grande de alunos.

Para aprimorar este processo de ensino, este trabalho sugere uma forma de seleção da próxima questão para os alunos em que seja selecionado um caminho personalizado de exercícios ao aluno de acordo com o seu estado. O estado do aprendiz varia de aprendiz para aprendiz e é refletido por meio dos exercícios que o aluno já resolveu. O processo é beneficiado ao utilizar um sistema de recomendação de questões que possa auxiliar alunos a encontrar um caminho de exercícios. Para isso, é utilizado a recomendação seguindo o caminho do bom aprendiz bem como a recomendação baseada em evidências de aprendizado para um aluno que esteja travado ou esteja indo bem, descritos na seção 4.1 da abordagem. Para que o aluno faça os exercícios de acordo com a sua situação, recomendam-se questões de dificuldade mais fácil ou mais difícil para que o aluno possa continuar engajado enquanto faz exercícios.

Um possível problema envolvendo a recomendação é mesmo que com uma recomendação de algum exercício mais fácil, um aluno que não conseguiu assimilar a teoria, muito provavelmente não conseguirá realizar mesmo exercícios mais fáceis. Nesta situação, deve-se ponderar sobre diferentes estratégias para guiar o aluno até que haja o entendimento do assunto abordado.

1.4 ORGANIZAÇÃO DO DOCUMENTO

A organização deste trabalho segue da seguinte forma: O Capítulo 2 fornece uma definição sobre os conceitos utilizados no trabalho. O Capítulo 3 apresenta um mapeamento sistemático da literatura sobre a escolha da próxima questão, soluções utilizadas no laço externo de sistemas tutores inteligentes, tipos de algoritmos, características utilizadas na escolha da próxima questão, soluções utilizadas na modelagem do estudante. O Capítulo 4 demonstra a abordagem deste trabalho. O capítulo 5 mostra como é feita a validação do trabalho. Por fim, o Capítulo 6 traz as considerações finais e a conclusão.

2 CONCEITOS FUNDAMENTAIS

Descreve-se, a seguir, um panorama compreensivo de conceitos necessários para situar pesquisadores sobre os temas vigentes abordados no trabalho. Este capítulo apresenta alguns temas que são relevantes para a categorização da abordagem de recomendação de questões no contexto de sistemas tutores inteligentes. As seções que seguem buscam apresentar os conceitos e discorrer de acordo com o encontrado na literatura.

2.1 ENSINO DA PROGRAMAÇÃO

O ensino da programação é implementado em disciplinas introdutórias, em cursos relacionados a computação. Visto que a ciência da computação é fundamentada pela programação, é de importância que se disponha uma assimilação de seus conceitos iniciais para que haja uma compreensão sobre tópicos mais avançados. Considerando isto, é indispensável um processo de ensino-aprendizagem eficiente, que possa facilitar este ensino e deixar mais atrativo aos alunos.

Programação em computadores é uma tarefa de complexidade elevada que exige bastante da capacidade cognitiva (Robins et al., 2003). Um algoritmo é uma sequência de comandos que é descrita através de linguagens de programação que possuem como objetivo resolver um problema. O computador interpreta a linguagem e executa os passos programados. As linguagens de programação possuem sintaxe e semântica próprias.

De acordo com Robins et al. (2003), pode-se destacar alguns pilares para a compreensão das disciplinas de programação:

- Algoritmos - são comandos que objetivam resolver um problema por meio do conjunto de passos executáveis em uma sequência finita.;
- Lógica de programação - habilidade necessária para desenvolver algoritmos considerando regras com o objetivo de construir o algoritmo em uma implementação através de uma linguagem de programação;
- Linguagem de programação - conjunto de regras e comandos com sintaxe e semântica específicos utilizado para traduzir algoritmos criados utilizando a lógica de programação em programas executáveis.

Para Aureliano e Tedesco (2012), os alunos iniciantes de programação parecem entender os conceitos e as estruturas que compõem uma linguagem de programação, porém o insucesso acontece na aplicação e durante a construção dos seus próprios programas.

Rocha et al. (2010) evidencia a atribuição de um conceito final de acordo com o quão longe um aluno chegou durante a realização das atividades na disciplina. As atividades são separadas em níveis e com base nisso é feita a atribuição da nota final. Isso é relevante, pois revela em que ponto ou nível houve deficiência na aprendizagem em contrapartida a somente uma nota final que indica que o aluno somente não alcançou o mínimo para a aprovação. O fracasso repetitivo gera frustração ao aluno, e ao repetir a disciplina no caso de reprovação é possível utilizar alguma intervenção específica de acordo com o nível em que o aluno chegou na última vez que ele cursou a disciplina ao invés de aplicar a repetição integral da disciplina. Com isso, é possível dar melhor atenção individual de acordo com o ritmo individual do aluno e dar continuidade a partir do ponto em que foi apresentado dificuldade.

Na abordagem de Garlet et al. (2016), é feita menção ao ensino da lógica de programação desde o ensino básico. Com isso, é possível preparar os alunos para o ensino superior e torná-los mais aptos a realizar a atividade de programação. O ensino de programação para crianças pode desenvolver o pensamento computacional bem como os passos para a resolução de problemas. O desenvolvimento da habilidade de programação estimula a pensar com maior criatividade e acaba desenvolvendo outras habilidades. Estas são habilidades bastante importantes a serem desenvolvidas mesmo que não seja escolhido um curso de informática posteriormente. O pensamento algorítmico é uma habilidade que pode ser aplicada não só na programação, mas em diversas outras áreas (Futschek, 2006).

2.2 SISTEMAS TUTORES INTELIGENTES

Os sistemas tutores inteligentes são sistemas de computadores cujo objetivo é fornecer um ensino personalizado aos aprendizes. O aprendizado objetivado efetivo pelos sistemas é geralmente implementado utilizando alguma solução computacional que traga características que promovam uma diferença significativa no processo de ensino. Esses sistemas têm como característica, atender aos diferentes tipos de estudantes e fornecer um ensino de alta qualidade adaptado às especificidades de seus usuários. O sistema utilizado como referência neste trabalho é um arcabouço para o ensino da programação de computadores e age como sistema tutor inteligente na mediação do erro.

O principal objetivo é aprimorar o processo de ensino-aprendizagem para que os diferentes grupos de alunos sejam atendidos com as soluções mais convenientes e fornecer um caminho de aprendizado efetivo. Essa é uma das principais características de sistemas tutores inteligentes em contrapartida com um sistema tradicional. Os STI são sistemas que utilizam estratégias de forma ativa para promover o ensino efetivo.

Esse tipo de sistema é utilizado, pois muitas vezes os professores são sobrecarregados nas salas de aula e não conseguem fornecer atenção individual aos alunos. Com uma recomendação específica a um aluno sobre qual caminho percorrer em um momento de dúvida, por exemplo, na hora de escolher algum exercício para resolver. Fornecer atenção individual especializada é custoso e contrário ao modelo tradicional que muitas vezes emprega o ensino em massa. É neste momento em que os sistemas tutores inteligentes podem entrar como protagonistas ao auxiliar no ensino e mostrar a sua funcionalidade.

A implementação de sistemas tutores inteligentes requer planejamento, pois deve-se pensar nos diferentes grupos de alunos e como atender a esses de maneira efetiva e que se enquadre ao estilo específico necessário para fornecer o melhor tratamento individualizado. Considerando isto, um sistema tutor inteligente consegue interferir no processo de ensino-aprendizagem de forma efetiva e consegue trazer benefícios positivos aos alunos (Crow et al., 2018).

Segundo Vanlehn (2006), sistemas tutores são caracterizados por ter dois laços, o laço externo e o interno:

- O laço externo é responsável por selecionar a próxima tarefa apropriada ao estudante e itera uma vez a cada tarefa. Uma tarefa é um problema com muitos passos que pode ser pulada ou trocada por outra tarefa;
- O laço interno é executado uma vez a cada passo realizado na solução de uma tarefa e pode fornecer comentários ou dicas.

Outra característica é avaliar as competências do estudante e atualizar o modelo do estudante que age em conjunto do laço externo. Quando um aluno executa uma iteração no laço

Termo	Definição
Domínio de tarefas	As informações e habilidades ensinadas pelo tutor
Tarefa	Uma tarefa que pode ser "pulada" ou trocada para outras tarefas.
Passo	Completar uma tarefa consiste de vários passos. Cada passo é um evento de interface.
Componente de conhecimento	Um conceito do domínio de tarefas. Qualquer fragmento que deve ser utilizado para completar tarefas.
Evento de aprendizado	A construção ou aplicação de uma componente de conhecimento, geralmente ao atingir uma tarefa.
Laço externo	Laço externo (iterações) sobre tarefas.
Laço interno	Laço interno sobre passos.
Incorreto	Inconsistência com o objetivo instrucional de um tutor.

Tabela 2.1: Comportamentos de Sistemas Tutores Inteligentes.
Fonte: Vanlehn (2006)

externo, ou conclui uma tarefa, é possível atualizar e refletir uma mudança no seu modelo do aprendiz.

Para Vanlehn (2006) um sistema tutor pode ser definido de acordo com os termos apresentados na Tabela 2.1. É possível caracterizar o domínio de tarefas, as tarefas, os passos, as componentes de conhecimento, os eventos de aprendizado, o laço externo, o laço interno e o que é o incorreto referente ao objetivo instrucional dos sistemas tutores inteligentes.

Para categorizar sistemas tutores inteligentes, consideram-se 3 componentes estruturais como pilares da arquitetura desse sistema: os modelos do domínio, estudante e tutorial. Na Figura 2.1, evidencia-se uma representação dos modelos que compõem um sistema tutor inteligente. Os modelos do domínio, estudante e tutorial se comunicam dando o sentido e caracterizando o sistema tutor inteligente. Por sua vez, o STI apresenta uma interface de aprendizado que é a ponte entre as interações do aprendiz que utiliza o sistema e o tutor. A interface pode apresentar o conhecimento do domínio por meio de diferentes ambientes de aprendizagem como simulações, hipermídias ou micromundos. A seguir detalham-se os modelos que compõem a arquitetura dos sistemas tutores inteligentes.

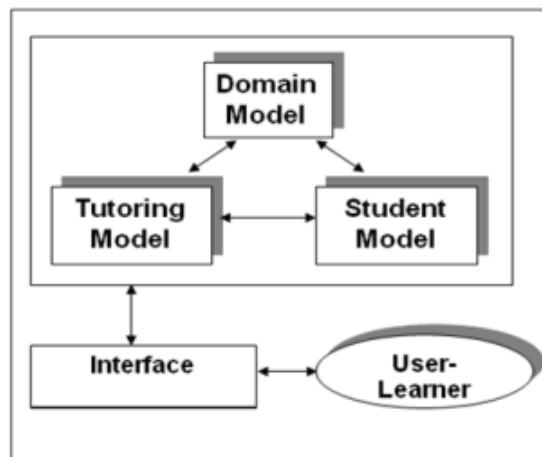


Figura 2.1: Arquitetura de um Sistema Tutor Inteligente.
Fonte: Nkambou et al. (2010)

2.2.1 Modelo Domínio

O modelo domínio contém conceitos, regras e estratégias de resolução de problemas do domínio que devem ser aprendidos. Também é definido pelos conhecimentos do domínio que devem ser aprendidos pelos aprendizes. O domínio pode ser organizado por um programa de estudos que possui os conhecimentos de acordo com uma sequência pedagógica desejável. O modelo pode servir como a representação de um expert que é o conhecimento desejável que um aprendiz deveria obter.

Analisando o modelo domínio, Nkambou (2010) aponta para 3 modelos que são identificados e conceituados em conexão com sistemas tutores inteligentes:

- Modelo Caixa Preta
- Modelo Caixa de Vidro
- Modelo Cognitivo

O modelo de caixa preta é uma representação que gera como resultado apenas o resultado ou saída final. Um exemplo clássico de sistema que usa o modelo caixa preta é o SOPHIE I (Brown e Burton, 1974), que é um sistema tutor para avaliar um circuito por meio de uma consulta no modelo domínio. O tutor resolve os problemas e pode recomendar ações para o contexto, entretanto o estudante é o responsável por decidir a descrição do contexto da resolução do problema que melhor se encaixa com a ação tomada pelo tutor.

O modelo caixa de vidro é um modelo que constrói uma ponte intermediária no conhecimento de um expert humano. Nesse modelo, a maior diferença está no fato de que cada passo pode ser inspecionado. Um exemplo de sistema caixa de vidro é o GUIDON (Mark e Clancey, 1985), que é um sistema para diagnóstico médico. O GUIDON foi construído com base no MYCIN, que é um sistema expert para o tratamento de infecções bacterianas. Dessa forma, durante o aprendizado utilizando o GUIDON, as ações do estudante são analisadas e confrontadas com base no que o sistema MYCIN faria.

O modelo cognitivo busca ligar representações de formalismo e inferência com a cognição do ser humano. O campo do módulo de conhecimento do domínio é um campo bastante relevante na tutoria inteligente. Abordagens cognitivas buscam desenvolver o módulo de cognição contido no conhecimento domínio. Para isso, busca-se representar o conhecimento da mente humana para replicar na resolução de problemas em um sistema tutor inteligente. A discrepância entre as outras abordagens é o fato de que o modelo cognitivo possui como objetivo suportar o raciocínio que seja cognitivamente plausível. Isto é, aplicar o mesmo estilo de representação para codificar o conhecimento ao aprendiz. O sistema Sherlock (Lesgold et al., 1988) é um exemplo que emprega esse modelo.

2.2.2 Modelo do Estudante

O modelo do estudante é um dos componentes mais importantes de um sistema tutor inteligente. O modelo representa a evolução de um estudante e suas características específicas. De acordo com Nkambou et al. (2010), alguns dos papéis do modelo do estudante são:

- corretivo - ajudar a corrigir falhas no conhecimento;
- elaborativo - ajudar a elaborar conhecimentos insuficientes;
- estratégico - indicar estratégias na estratégia tutorial;

- diagnóstico - ajudar a diagnosticar falhas de conhecimento;
- preditivo - prever a resposta para ações tutoriais;
- avaliativo - avaliar o estudante ou sistema tutor inteligente.

O modelo do estudante é geralmente visto como um modelo dinâmico que implementa funções. Para Wenger (1987), o modelo do estudante é composto por 3 funções que devem:

- coletar dados explícitos e implícitos por meio de inferências do aprendiz;
- usar os dados coletados para criar uma representação do conhecimento do estudante e o seu processo de aprendizado;
- verificar as informações que dizem respeito ao estudante por meio de um diagnóstico sobre o estado e a melhor forma de selecionar uma estratégia pedagógica para melhor apresentar as informações do domínio ao estudante.

Em sistemas tutores inteligentes, a modelagem do estudante representa as competências e os aprendizados de um estudante. O modelo representa inferências sobre usuários que incluem aspectos relevantes de acordo com as interações feitas com o sistema. O modelo do estudante em um sistema tutor inteligente investiga as ações de um estudante e cria uma representação qualitativa do conhecimento cognitivo e afetivo. Dessa forma, o modelo serve para auxiliar na escolha de um feedback que seja mais apropriado para o aluno.

Woolf (2010b) aponta que a experiência adquirida por professores se dá através dos anos de experiência. Muitos professores utilizam as características dos alunos para aprimorar a estratégia de compreender o estado do aprendizado dos estudantes. Assim sendo, aprendem-se as melhores estratégias no campo pedagógico que apresentam melhor aproveitamento para os diferentes nichos específicos de aprendizes. A interação aluno-professor traz como ponto importante o reconhecimento de uma proposta que melhor trata as características específicas e individuais de um estudante. De forma parecida, um sistema tutor inteligente busca construir um modelo de inferência que permite melhor ajustar as estratégias de ensino e customizar a experiência para as necessidades dos alunos.

Espera-se construir um modelo que consiga identificar as características e responder de forma efetiva às necessidades dos estudantes. Levando esses pontos em consideração, cita-se a representação e a atualização do conhecimento do estudante como pilares relevantes para a modelagem do estudante que serão descritos a seguir.

A representação do conhecimento pode ser construída por diversos formatos. É possível ranquear um conhecimento em uma escala numérica ou até mesmo construir redes entre as diferentes características que podem gerar representações entre conhecimentos. Outro ponto possível é evidenciar conceitos equívocos dos aprendizes que incluem erros comuns até a insuficiência de conhecimento.

Outro campo de representação são as características afetivas que incluem as emoções ou atitudes dos aprendizes. Como exemplo é possível citar representações de confusão, frustração, tédio, motivação, confiança e fadiga. A computação afetiva considera esses fatores que podem ser detectados por meio de sensores. Para abordar esses aspectos, utiliza-se alguma intervenção que possa mitigar um efeito negativo.

Para Woolf (2010b) a atualização do modelo do estudante é bastante relevante para inferir o estado no momento determinado do conhecimento do estudante e comparar com uma sequência de ações apropriada. Espera-se que um estudante consiga conquistar todo o conhecimento do domínio de forma que aconteça uma transição de um estudante iniciante, que seria o estado

inicial, para um especialista, que é categorizado por compreender os conhecimentos do domínio. O modelo do estudante pode indicar a falta de algum conhecimento que está no domínio ou apresentar conhecimentos equívocos. Considerando esses aspectos, é possível investigar planos de ações para indicar caminhos ou estratégias para que o aluno siga e dessa forma evidenciar estratégias que funcionam melhor para diferentes tipos de características dos alunos.

2.2.3 Modelo Tutorial

A modelagem tutorial é qualquer forma em que pesquisadores tentam conceituar e operacionalizar funções e variáveis tutoriais (Baker, 2000). O modelo tutorial recebe como entrada o modelo de domínio e o modelo do estudante e gera como saída decisões, estratégias e ações tutoriais. De acordo com os princípios definidos, o modelo tutorial toma ações como, quando ou não fazer alguma intervenção e de que forma isso é feito.

A pesquisa em sistemas tutores inteligentes sobre o design de funções tutoriais consistem do planejamento, o diálogo e o design das interações tutoriais. Recentemente, esta área está investigando a modelagem tutorial com base em técnicas de aprendizado de máquinas que permitem processar os dados dos estudantes com base em buscas e na identificação de padrões. Bourdeau e Grandbastien (2010) apontam para alguns desafios que seguem desta área:

- integrar com o objetivo de obter adaptabilidade e efetividade na tutoria;
- usar elementos do contexto em tempo real;
- melhorar a acessibilidade;
- avaliar metodologias e ferramentas para promover as dimensões de tutoria com os sistemas tutores inteligentes.

O uso de emoções em sistemas tutores inteligentes é estudado de forma que permita aos sistemas compreender os estados emotivos e cognitivos para que a melhor adaptação seja feita. O objetivo do modelo tutorial é conquistar uma tutoria inteligente que avalie o aprendizado e forneça suporte e serviço que seja comparável ao de um ótimo tutor humano. O objetivo de um sistema tutor inteligente é tentar ser tão efetivo quanto um bom tutor humano. Para isso, surge a questão sobre como replicar tutores humanos que possuem um conhecimento que permita compreender o currículo, as questões cognitivas, o aprendizado e estratégias tutoriais. Em que contexto é possível abrir o modelo tutorial? A abertura pode envolver a preparação de uma sequência de aprendizado e o comportamento do sistema tutor durante a execução desta sequência. É possível ponderar estratégias de acordo com preferências pessoais e extrair os resultados em pós-sessões de feedback. Sistemas tutores inteligentes apresentam um número expressivo de variáveis tutoriais: Quem, Quando, Como e Onde a tutoria pode acontecer são indagações presentes que possuem suas especificidades e desafios. As interdependências entre as variáveis é uma questão de bastante complexidade (Bourdeau e Grandbastien, 2010).

De acordo com Bourdeau e Grandbastien (2010), deve-se responder, primeiramente, quem é o aprendiz, quais são as suas características: idade, conhecimentos prévios, habilidades, motivação, dificuldades, entre outros pontos. É imperativo conhecer o aprendiz para modelar uma solução tutorial adaptativa. Outro ponto relevante é o conteúdo que está sendo tutorado. Características intrínsecas do domínio são bastante relevantes para a modelagem. Outra questão é sobre as ações tutoriais: estratégias, planejamento, remediação, dicas, intervenções, estímulos, suporte ao erro, visualizações, seleção de estratégias entre outras. Também deve-se questionar a efetividade da tutoria. VanLehn et al. (2007) acrescenta a seguinte ponderação: "Quando

diálogos tutoriais são mais efetivos que ler?" Por fim, o contexto em que a tutoria ocorre deve ser considerado, o físico, cultural, cognitivo ou afetivo. O conhecimento limitado acerca dos contextos é uma desvantagem que acontece no desenvolvimento de sistemas tutores inteligentes.

Considerando a avaliação de modelos tutoriais e as suas funcionalidades, o objetivo é testar os parâmetros de acurácia das estratégias, a efetividade das ferramentas por meio do diagnóstico e a adaptabilidade às diferentes situações individuais encontradas. Sawyer (2006) faz menção ao fato de que dados coletados por meio de estudos grandes são bastante significativos para justificar o uso de ações tutoriais no campo de sistemas tutores inteligentes. Metodologias e ferramentas para avaliação de funções tutoriais permitem o progresso na pesquisa de sistemas tutores inteligentes. Woolf (2010a) acrescenta que os problemas encontrados na avaliação de funções tutoriais reside no fato de que essas funções estão mal definidas, enviesadas, apresentam variáveis confusas e demonstram fraqueza no design de avaliação ou na interpretação de resultados.

2.3 MODELAGEM DO ESTUDANTE NO CONTEXTO DO ENSINO DE PROGRAMAÇÃO DE COMPUTADORES

O modelo do estudante é a representação das habilidades desenvolvidas ou não por aprendizes necessárias para a realização da programação de computadores. Para medir as habilidades na programação de computadores, estabelece-se um conjunto de capacidades que são utilizadas como métricas de conhecimento. Seguindo as 41 habilidades definidas por Krynski (2013) e o modelo utilizando por Porfirio (2020) estabelece-se um modelo do estudante através de uma rede bayesiana que representa um conjunto com as habilidades julgadas necessárias no domínio da programação de computadores e uma valoração que indica o quanto uma habilidade foi masterizada, representado pelas Figura 2.2 e 2.3 que ilustram o modelo e domínio em questão.

O modelo é importante em um sistema tutor inteligente para que se possa medir e ilustrar o estado atual de conhecimento dos aprendizes. Analisando o estado atual, é possível diagnosticar o nível de domínio do conhecimento do aprendiz em relação às habilidades. Com essa capacidade, realiza-se a seleção de questões mais apropriadas para que os aprendizes possam resolver no contexto da programação de computadores. Um sistema tutor inteligente consegue fornecer um caminho personalizado de exercícios ao aprendiz de acordo com as características indicadas pelo modelo do estudante. Dessa forma, avalia-se qual habilidade desse domínio necessita mais atenção e utilizam-se as estratégias de recomendação necessárias para que o aprendiz tenha um caminho de aprendizado mais efetivo.

Deve-se considerar que algumas habilidades são precedentes a outras, por isso deve-se selecionar o caminho em que o aprendiz deve realizar os exercícios de forma coerente que siga o caminho evidenciados pelas habilidades que são representadas pelo modelo do estudante.

2.3.1 Grafo Genético

Um tipo de modelagem do estudante pode ser caracterizado por meio do grafo genético, ilustrado pela Figura 2.4, que é uma estrutura utilizada para representar componentes de conhecimento. Oliveira (2016) faz uso desta estrutura para representar as habilidades ou perícias no domínio da programação. O grafo genético representa a evolução do conhecimento através dos vértices e arestas. Os vértices do grafo genético demonstram as habilidades dentro do domínio em evidência. Pode-se considerar o processo de aprendizagem na completude destas habilidades que serão desenvolvidas.

As relações estabelecidas dentro do grafo genético entre os vértices por meio de arestas são:

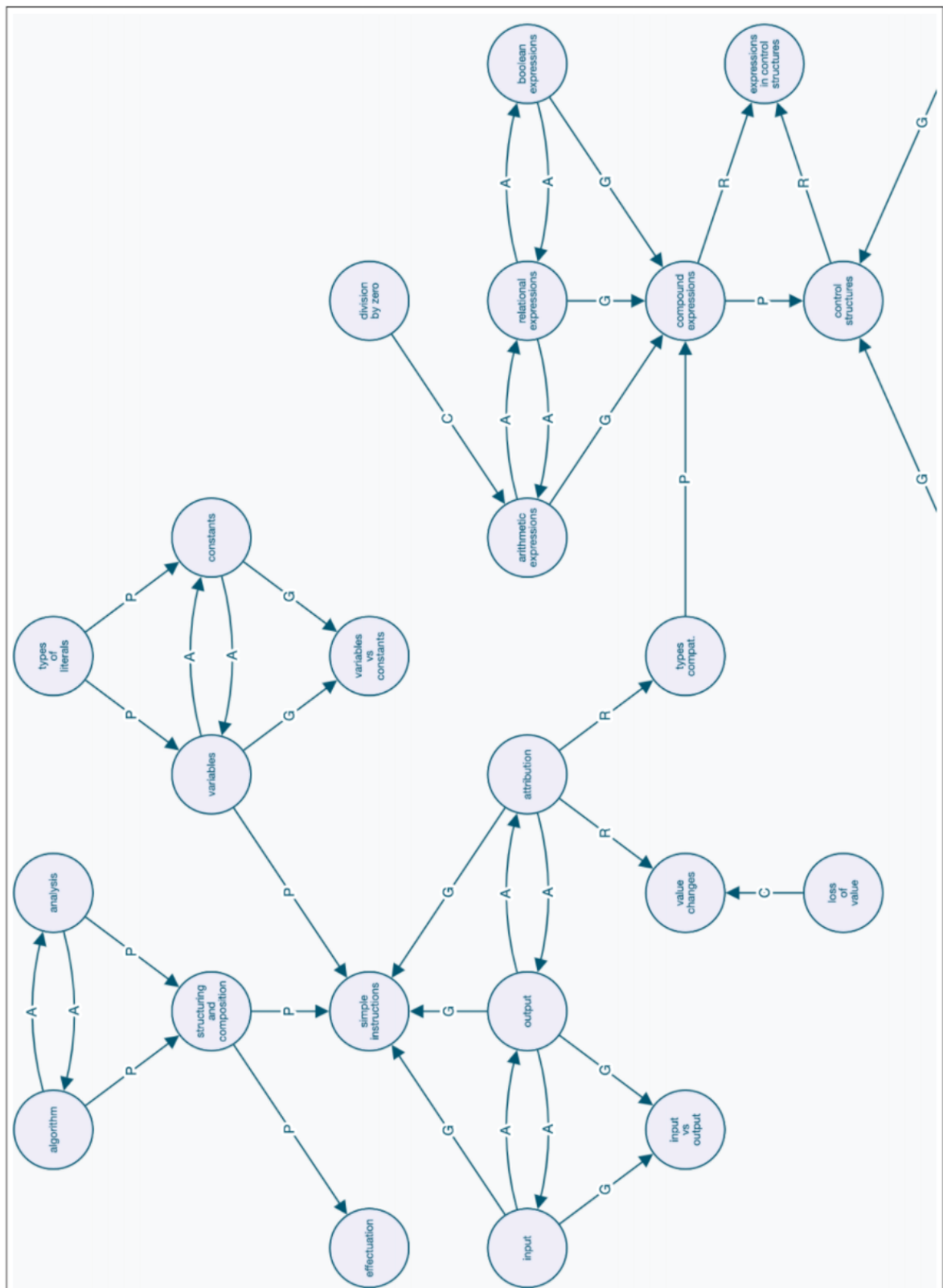


Figura 2.2: Habilidades no domínio da programação.

Fonte: Porfirio (2020)

- Generalização/Especialização;
- Analogia;

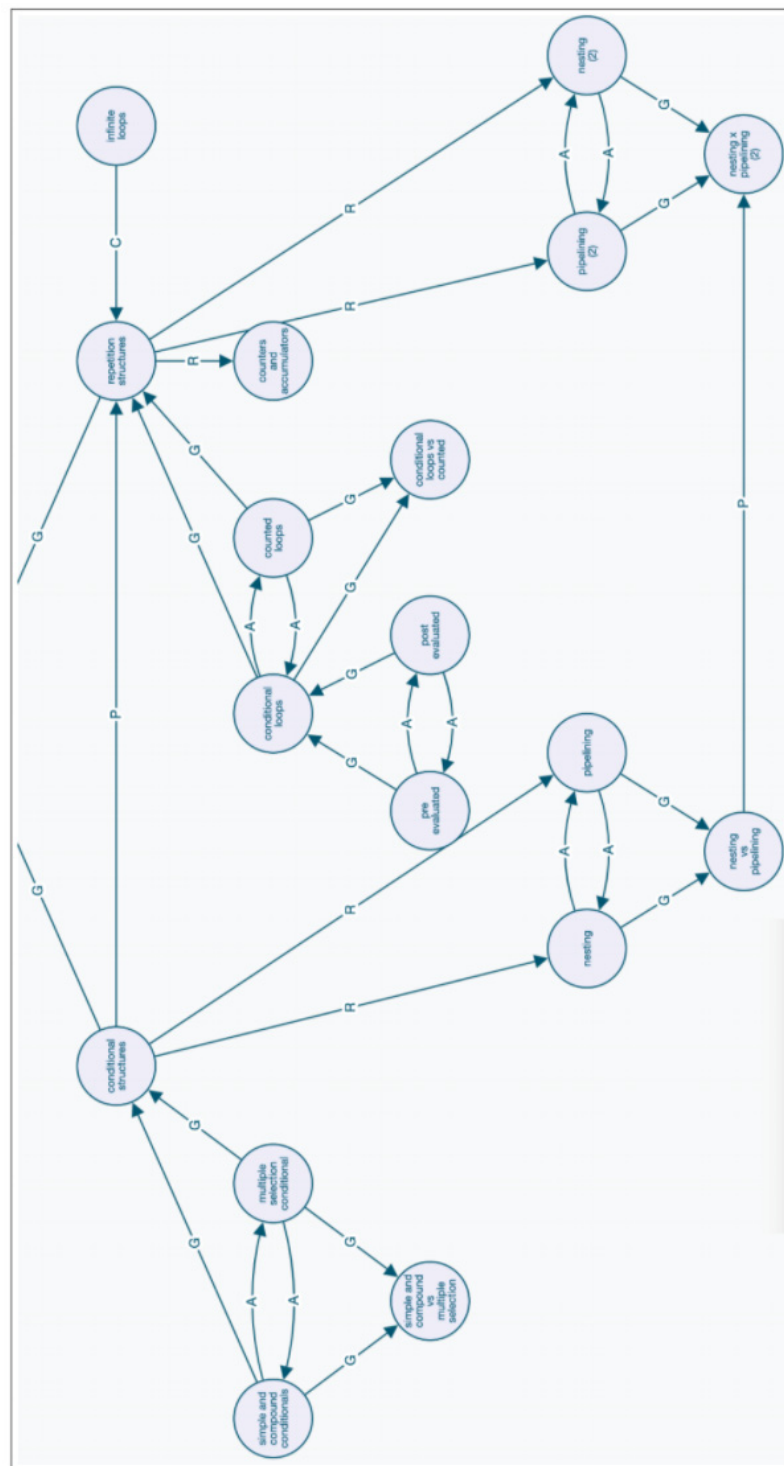


Figura 2.3: Habilidades no domínio da programação.

Fonte: Porfirio (2020)

- Simplificação/Refinamento;
- Desvio/Correção.

Outras ferramentas de representação dentro do grafo genético são o agrupamento de habilidades dentro de ilhas de conhecimento, conhecimento declarativo versus conhecimento

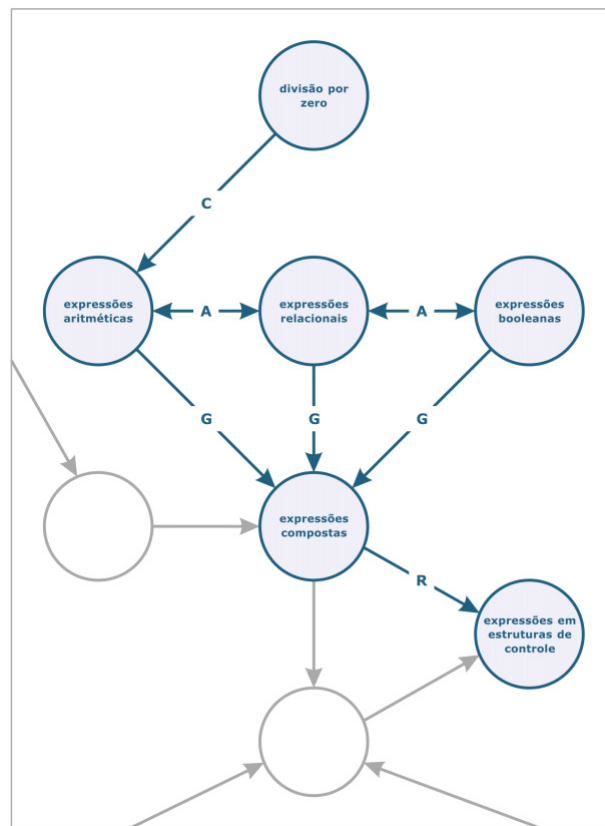


Figura 2.4: Exemplo de grafo genético no domínio da programação.
Fonte: Oliveira (2016)

procedimental e a ordem que compreende pré e pós-requisitos entre as habilidades (Oliveira, 2016).

A sobreposição no grafo genético representa a parte da representação do estudante através do conhecimento do domínio. O conhecimento de um aprendiz é um subgrafo do grafo do domínio que contém todas as habilidades. Desta forma, através das relações dos vértices e arestas, modela-se o aprendizado do estudante. A modelagem do estudante é compreendida como um passeio em cima do grafo. O passeio ilustra as habilidades adquiridas pelo estudante dentro do domínio.

De acordo com Oliveira (2016), o grafo genético fornece uma fronteira de habilidades com base no estado atual do aprendiz sobre quais habilidades podem ser exploradas em sequência pelo aprendiz. Essa fronteira é obtida entre o subgrafo do modelo do aprendiz e o grafo do domínio.

2.4 SISTEMAS DE RECOMENDAÇÃO

Sistemas de recomendação são utilizados em diversas aplicações onde o perfil do usuário e suas características específicas são relevantes na hora de tomar alguma decisão. Alguns exemplos de áreas que utilizam esses sistemas são: plataformas de streaming, plataformas de vídeos, plataformas de músicas, plataformas de compras. Nesses casos, o sistema de recomendação atua para achar os interesses do usuário e fazer as melhores recomendações de acordo com as especificidades do usuário Lü et al. (2012).

A recomendação é bastante relevante para esses serviços citados. A fundamentação destes se apoia em explorar estas características do usuário para poder oferecer efetivamente os

seus serviços. De acordo com (Lü et al., 2012), Sistemas de recomendação têm como objetivo fazer uma seleção personalizada com base nos gostos ou características do usuário e fornecer aquelas opções que sejam mais coerentes ou atrativas.

A representação de sistemas de recomendação segundo Pazzani e Billsus (2007), pode ser feita através de representações dos itens, perfis dos usuários, o aprendizado do modelo de um usuário e algoritmos de aprendizado para aprender os diferentes usuários e a escolha desses algoritmos com base no contexto.

A representação dos itens segue a organização dos itens por diferentes atributos. Um simples exemplo de atributo bem estruturado seria a recomendação de restaurantes. Para o atributo tipo de cozinha dentre as opções: francesa, italiana e mexicana. Nesse caso, muitos algoritmos de aprendizado de máquinas aprendem o perfil do usuário e fazem recomendações de acordo. Um usuário que avaliou 4 de 5 restaurantes de cozinha francesa como positivos apresenta 80% de chance se interessar por um restaurante desta cozinha.

Outra vertente são atributos não estruturados. Um exemplo é um texto contendo a descrição de um restaurante. Por exemplo "A melhor carne preparada pela cozinha francesa já produzida", "Para degustar o melhor da cozinha francesa, prove a nossa carne". Para estes exemplos, há a atribuição de um certo peso para termos chave nesses textos. Um usuário que avaliou positivamente restaurantes com estas descrições possivelmente está interessado em cozinha francesa associada de carne. Logo, os algoritmos de recomendação podem computar itens com descrições parecidas com estas para recomendar ao usuário. O domínio em que a recomendação está inserida ajuda a estabelecer os termos de maior relevância a serem utilizados pelos algoritmos de recomendação.

O perfil do usuário possui diversos tipos de informação, entretanto foca-se nos dois seguintes itens:

- Um modelo da preferência dos usuários, por exemplo a descrição de tipos de itens que interessam ao usuário. Uma representação é uma função que para qualquer item calcule a probabilidade do usuário estar interessado nesse item. Uma aplicação seria pesquisar n itens que possuem maior probabilidade de interesse para o usuário;
- O histórico das interações do usuário com o sistema de recomendação. Este pode incluir itens visualizados junto de outras informações sobre a interação do usuário. Por exemplo salvar termos utilizados pelo usuário ao realizar uma busca por restaurantes: "Restaurante italiano no código postal 952150".

O aprendizado do modelo de um usuário é o aprendizado de classificação do usuário. Por exemplo as categorias binárias em que, por exemplo, um usuário gosta de um item ou não gosta. Isso é avaliado através de uma interação direta do usuário com um sistema de avaliação ou implicitamente investigando as interações do usuário com os itens. Um exemplo de interação é se um usuário compra um item e avalia que gostou do item ou se o item foi retornado implica que o usuário não gostou. Implicitamente há incerteza se a previsão do gosto do usuário realmente demonstra a sua preferência. Em contraste, quando o usuário avalia diretamente um item, há menor incerteza.

Muitos dos algoritmos utilizados na recomendação criam uma função que estima a probabilidade de um usuário demonstrar preferência por algum item. Essas probabilidades podem ser utilizadas para agrupar os itens que seriam possivelmente interessantes ao usuário. Algumas técnicas de recomendação utilizadas são: árvores de decisão, indução de regra, k-vizinhos mais próximos, feedback de relevância, algoritmo de Rocchio, classificadores lineares, métodos probabilísticos, classificador Naive Bayes, tendências, dentre outros.

Um sistema de recomendação dentro do contexto de sistemas tutores inteligentes pode ser encontrado em Pimentel e Direne (1998b) através do RUI (Representations for Understanding Images), que apresenta uma abordagem genérica de medidas cognitivas para a descrição e ordenação da base de imagens radiológicas de um ITS destinado ao ensino de conceitos visuais. No trabalho em questão, o processo de sequenciamento é aprimorado ao utilizar as medidas cognitivas. O objetivo é replicar uma boa escolha de imagem a ser trabalhada com o aprendiz, de modo a simular a experiência com um professor, minimizando o tempo de aprendizado. O sistema RUI, é dividido em três partes: o nível conceitual, o nível de produção e o nível de ensino. O nível conceitual é o que permite a descrição de classes de anomalia e todo o conhecimento relativo a elas. Cada imagem é considerada como uma instância de uma classe de anomalia e é descrita no nível de produção. O nível de ensino é o responsável pela interface do sistema com o aprendiz. O trabalho traz como contribuição modificar o modelo tutorial para permitir uma melhor ordem de apresentação de imagens. As medidas cognitivas permitem quantificar o potencial das imagens em exercitar determinada habilidade para o aprendiz. Dessa forma é possível aplicar uma estratégia pedagógica que melhor se enquadre aos aprendizes.

Outro sistema é descrito em Pimentel e Direne (1998a), que traz as medidas cognitivas destinadas ao ensino da programação de computadores. O objetivo consiste em possibilitar a automação de uma boa escolha do próximo programa a ser desenvolvido pelo aprendiz de modo a simular um professor, reduzindo o tempo de aprendizado do tema trabalhado. As medidas cognitivas são uma forma de conseguir meta conhecimentos sobre os exemplos de uma base de conhecimento de um STI. As características descritas de um perito em programação de computadores mais significativas segundo o trabalho são:

- precisão sintática/semântica;
- análise do problema;
- reutilização de soluções;
- simulação mental dos estados do computadores durante a execução.

Para obter uma medida que quantifique cognitivamente um exemplo, deve-se analisar o quanto um exemplo contribui para a evolução do aprendizado. O quanto cada programa que é apresentado ao aluno contribui para a aquisição do estereótipo padrão de um programador perito. Dessa forma, um sistema de recomendação pode empregar essas medidas cognitivas ao indicar em seu modelo tutorial programas exemplo para que um aprendiz se torne um perito. Estratégias pedagógicas também são relevantes para manter o aprendiz em um estado motivacional alto. Um modelo tutorial necessita das informações que o modelo do estudante recebe sobre métricas de desempenho até o estado motivacional do aluno. Desse modo, a comunicação entre os módulos de um sistema tutor inteligente são relevantes para que um STI consiga atender aos seus objetivos de maneira efetiva.

2.5 JUÍZES ONLINE

Sistemas juízes online são sistemas designados para a avaliação confiável de códigos fonte de algoritmos submetidos por usuários que são compilados e testados em um ambiente homogêneo (Wasik et al., 2018). A utilização desses sistemas tem se tornado cada vez mais popular. A classificação para estes sistemas pode ser através de seus objetivos principais tais como sistemas suportando a organização de concursos de programação competitiva, aprimoramento da educação e processos de recrutamento, facilitar a resolução de desafios de mineração de dados,

compiladores online e o desenvolvimento de plataformas integradas como componentes de outros sistemas.

O procedimento de avaliação utilizado por um sistema juiz online consiste de 3 etapas:

- Submissão;
- Avaliação;
- Pontuação.

Durante a etapa de submissão, o código fonte submetido é compilado e verificado se a execução pode ser realizada com sucesso no ambiente de avaliação. Com uma verificação bem sucedida, a submissão é avaliada se atende às especificidades dos casos de teste. Para a execução de cada caso de teste para uma submissão particular, verifica-se os seguintes passos são atendidos:

- O processo de execução ocorreu sem erros;
- Algum problema de limitação de recursos foi excedido;
- A saída recebida está de acordo com a definição do problema.

Sistemas Juízes Online são caracterizados como um sistema centrado na verificação das saídas que os programas submetidos geram conforme as entradas fornecidas. Dessa forma, as entradas pertencentes a um conjunto válido cujas saídas geradas pelo programa são debatidas com saídas válidas. Por fim, computa-se uma pontuação de acordo com os resultados dos casos de teste para a submissão que é específica ao juiz online utilizado.

Um exemplo de sistema juiz online é o descrito em de Campos e Ferreira (2004). O sistema BOCA, ilustrado pela Figura 2.5, é utilizado no gerenciamento de competições de programação. O sistema possui as características de um juiz online: envio de submissões, os times podem fazer perguntas ao sistema e o juiz pode responder essas perguntas, autenticação de usuários, perfil dos times que participam na competição e administradores que acompanham a competição e possuem acesso às informações da competição. Ao utilizar uma interface web é possível a criação de competições distribuídas em que equipes podem estar espalhadas em diferentes locais desde que haja conexão com a internet.

Alguns sistemas Juízes Online pedem por um processo mais complexo na elaboração e publicação de problemas. Isso ocorre no sentido de necessitar permissão pela equipe responsável pela administração do sistema. Um sistema que se destaca neste sentido, por sua simplicidade ao publicar novos problemas com formato semelhante a exercícios de fixação sem muita burocracia é a The Huxley, Figura 2.6, descrito por de Barros Paes et al. (2013).

2.6 FARMA-ALG

A ferramenta FARMA-ALG (Kutzke, 2015), derivada do FARMA (Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem), atua no processo de ensino de programação de computadores e possui como principais características: a autoria de objetos de aprendizagem aos professores e o armazenamento automático de respostas e contextos.

O FARMA-ALG é uma aplicação WEB que pode ser utilizada tanto em computadores pessoais quanto dispositivos móveis e busca contribuir para uma maior aproximação entre alunos e professores no ensino da programação de computadores.

Runs	Score	Clarifications	Tasks	Options	Logout
Run #	Time	Problem	Language	Answer	
2	37	Problema 1	C	No - Compilation error	
3	37	Problema 3	Java	Yes	
4	83	Problema 1	C++	Not answered yet	

To submit a program, just fill in the following fields:

Problem:

Language:

Source code:

Figura 2.5: Interface do BOCA.
Fonte: de Campos e Ferreira (2004)

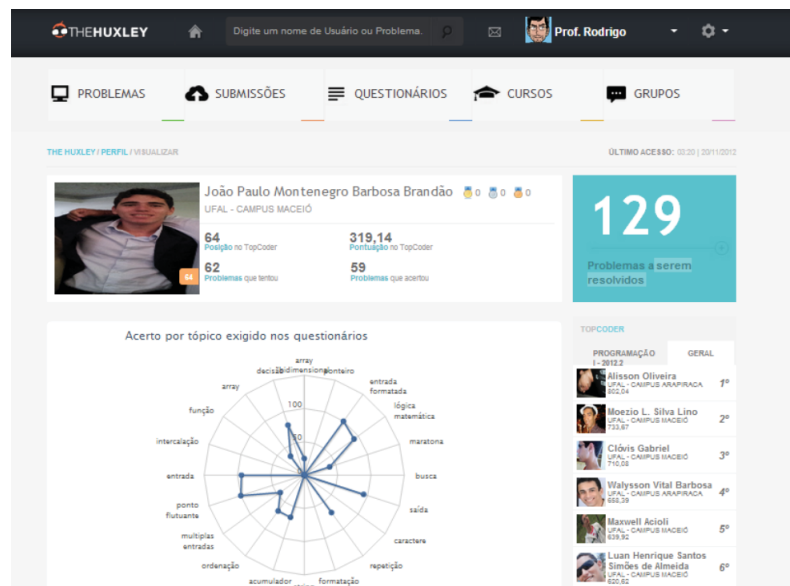


Figura 2.6: Interface do The Huxley.
Fonte: de Barros Paes et al. (2013)

Os objetos de aprendizagem utilizados no FARMA-ALG são exercícios de programação. As respostas submetidas pelos alunos são códigos de programação em uma linguagem (C, Pascal ou Ruby) definida pelo professor durante a criação de exercícios.

Para avaliar a corretude do exercício, o FARMA-ALG compila a solução submetida pelo aluno, executa e compara com casos de teste cadastrados pelo professor da mesma forma que Juízes Online. Se o resultado for incorreto, o aluno recebe um feedback sobre qual caso de teste em que houve problemas.

Outro ponto de relevância do sistema é o fato que para cada submissão do aluno, o sistema guarda as respostas e as informações relacionadas à submissão. Com esses dados, é possível buscar padrões que se repetem em grupos de alunos e criar uma respectiva intervenção que mitigue o acontecimento de algum erro recorrente.

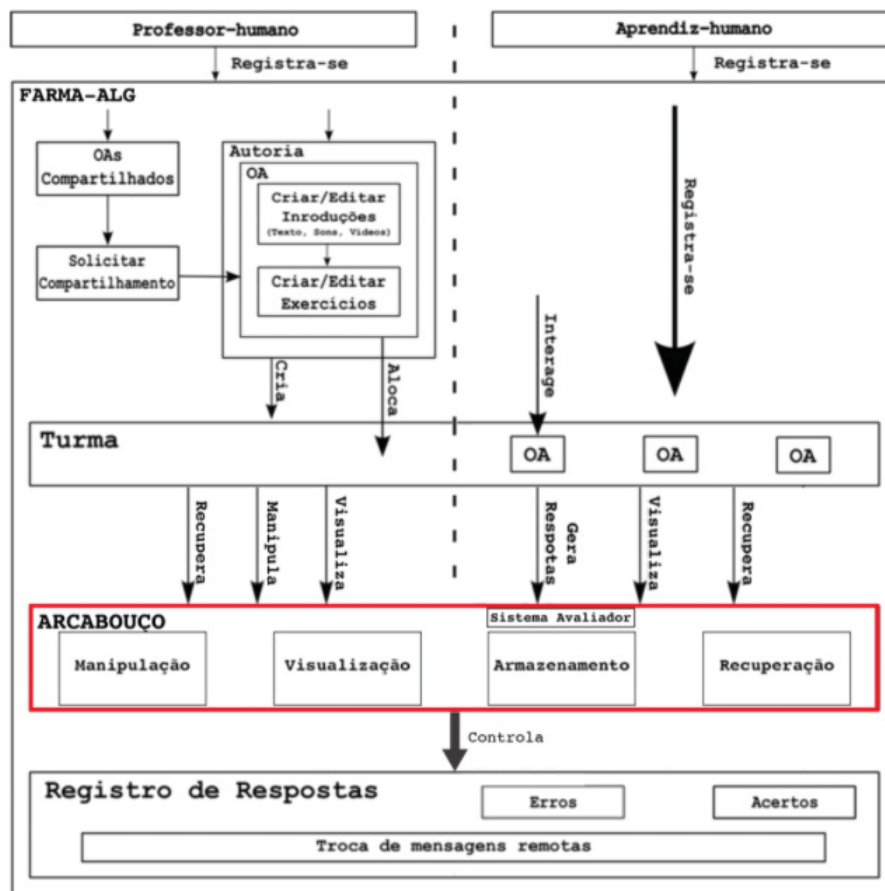


Figura 2.7: Esquema do FARMA-ALG.
Fonte: Kutzke (2015)

Existem dois tipos de usuário dentro da ferramenta: professor e aluno. O professor pode criar turmas e exercícios e tem uma visão geral sobre as formas de manipulação. Os alunos podem se matricular em turmas criadas por professores, responder exercícios disponíveis e visualizar suas submissões.

O FARMA-ALG possui as mesmas características de um sistema juiz online, visto que recebe submissões de códigos fonte de programação pelos alunos, avalia-os através de casos de teste e pontua-os como corretos ou incorretos. Considera-se o FARMA-ALG como um sistema pedagógico educacional, visto que não só faz a avaliação das soluções, mas guarda registros das soluções tanto as soluções corretas quanto as incorretas para realizar a mediação do erro. Outra característica pedagógica é a questão da criação de um ambiente educacional que consiste nas turmas que são administradas por professores e englobam alunos. Neste ambiente, há uma aproximação maior entre o professor e o aluno. O sistema vai além de ser um simples juiz online que avalia soluções e promove esta aproximação pedagógica.

Algumas das principais características do FARMA-ALG, demonstradas pela Figura 2.4, que serão descritas a seguir:

- Armazenamento;
- Recuperação;
- Manipulação;
- Visualização.

2.6.1 Armazenamento

O armazenamento é a função de guardar as respostas submetidas. Tendo o armazenamento em evidência, implementam-se duas componentes no FARMA-ALG.

- Uma das componentes implementadas é a máquina de similaridade que é a capacidade de definir a semelhança entre duas respostas. O grau de semelhança é calculado seguindo parâmetros no contexto de exercícios de programação;
- A outra componente é o grafo de similaridade que é uma estrutura de dados que é formado conforme as respostas são submetidas. Buscam-se semelhanças nas respostas dos códigos fontes.

2.6.2 Recuperação

Recuperação é a capacidade de recuperar dados relevantes que são utilizados para as funcionalidades busca e recomendação.

- A funcionalidade de busca é a capacidade, disponibilizada ao professor e ao aluno, de recuperar dados que sejam relevantes para a mediação do erro;
- A outra funcionalidade é a recomendação que é a função de encontrar automaticamente dados de respostas e questões que possam ser de relevância pedagógica para um grupo de alunos. Essa função é apresentada ao professor para tentar encontrar registros potencialmente relevantes.

2.6.3 Manipulação

A manipulação das respostas traçadas pelo arcabouço fornecem algumas ações: classificação de resposta e controle das relações de similaridade.

- A classificação é a capacidade de atribuir palavras-chave aos registros de respostas. Essa função é feita automaticamente pela ferramenta e pode ser feita de forma manual pelo professor;
- O controle da relação de similaridade compreende as operações: adição, remoção, aceitação e rejeição de relações. Estas são disponibilizadas aos professores e servem para manipular as relações de similaridade.

2.6.4 Visualização

Visualização é a maneira de ilustrar os registros de respostas armazenados. A seguir descreve-se os tipos de visualizações implementados no FARMA-ALG.

- A funcionalidade de visualização gráfica através de um grafo de similaridade que oferece a manipulação e abstração das respostas. Através do grafo, é possível analisar e interagir com os registros;
- Outro tipo de visualização é a linha do tempo que é a capacidade de acessar as respostas sobre uma linha do tempo. Nesta representação, mostra-se a sequência de respostas cronologicamente;
- Por fim, existe também a visualização de respostas de forma contrária a visualização rápida. Nesta, é feita uma exibição de maneira mais elaborada que oferece maior interação entre aluno e professor. É possível a troca de comentários entre aluno e professor. O professor pode também fazer a correção de um registro de resposta de forma manual, caso o corretor valore uma resposta correta como incorreta.

2.6.5 Mediação do Erro

Um dos pilares principais na tese de Kutzke (2015) é a mediação do erro. É discorrido sobre a perspectiva histórica do erro através da evolução dos tópicos sobre a perspectiva tradicional, perspectiva da escola nova, perspectiva tecnicista. Aborda-se sobre as diferenças individuais e de grupo. A problemática do erro é categorizada como um objeto de estudo para alunos e professores. Não se deve atribuir a culpa para um ou outro. Deve-se compreender o caminho do desenvolvimento individual do aluno e através de uma ação educacional promover o aprendizado ao aluno em um nível que ocorra o seu desenvolvimento.

O erro é um ponto enraizado negativamente durante o processo de aprendizagem (Serconek, 2006). Considera-se o erro como o fracasso associado ao aluno que foi incapaz de suceder. A negatização do erro precede o fato de que o erro pode ser fitado sob uma perspectiva de que para que a mediação ocorra, deve-se compreender o erro e os fatores que levaram este a acontecer. Sem o registro do erro, é impossível compreender os motivos que fizeram que o erro acontecesse e cancela a perspectiva de aprendizado da recorrência do erro para que seja possível aprender a evitar ou minimizar o acontecimento deste pelos aprendizes.

Para que haja a mediação do erro, é importante fornecer os mecanismos que permitam acessar e investigar o erro. O aluno não deve somente ficar preso ao sentimento de errar, mas sim ver e conhecer o seu erro. É importante compreender e consolidar como este erro se tornou a

ser através da mediação do professor. Para tanto, é importante salvar os registros de respostas submetidos pelos alunos para que, dessa forma, entenda-se a trajetória do aluno até que haja o entendimento do erro.

A importância do erro na formação de conceitos científicos pode ser ilustrada ao detectar erros de um aluno durante a realização de exercícios de programação. É possível que o aluno não tenha assimilado os conceitos de forma correta necessários para a resolução. Caso essa situação ocorra, o aluno terá insucesso em resolver exercícios e isso o desmotiva a continuar praticando. Neste contexto, um sistema tutor inteligente pode agir de maneira ativa ao detectar tal situação em que o erro foi cometido e aplicar alguma intervenção corretiva para ajudar o aluno a entender o conceito com dificuldade.

Um trabalho que foi realizado em cima da ferramenta FARMA-ALG é o de Ogawa (2017), que descreve uma metodologia orientada ao aumento da interação entre professores e alunos que utilizam o FARMA-ALG. Foi constatado que a intervenção do professor no FARMA-ALG constitui grande contribuição para que os alunos possam sanar as dúvidas e progredir na realização dos problemas. Observou-se uma quantidade reduzida de interações, que foram causadas pela falta de um método para guiar os professores durante a utilização do FARMA-ALG. Nesse contexto, propôs-se uma metodologia direcionada na utilização da ferramenta focando na interação entre os professores e alunos. Utilizando conceitos da abordagem sociointeracionista professor aluno e da Zona de Desenvolvimento Proximal que serviu para a construção das atividades que o professor deve desempenhar durante a utilização da ferramenta de auxílio para o ensino de algoritmos.

2.7 AVALIAÇÃO AUTOMÁTICA DE EVIDÊNCIAS DE APRENDIZADO EM CÓDIGOS FONTE

A avaliação automática de evidências de aprendizado em códigos fonte, descritas por Porfirio (2020), é caracterizada por meio da análise sintática, lê de uma entrada de dados que segue regras específicas e monta uma estrutura intermediária para organizar as informações extraídas. Utilizando essa técnica, recebe-se como entrada o código fonte de uma solução de um exercício de programação e como retorno, gera-se um conjunto de informações sobre o código avaliado. As informações retornadas são estruturadas por meio de uma representação em uma árvore sintática abstrata, Figura 2.8, que é uma estrutura de dados capaz de representar as evidências extraídas do código. Cui et al. (2010) define os seguintes passos para gerar a árvore sintática abstrata:

- Pré-processamento da fonte;
- Análise léxica;
- Parsing;
- Geração da árvore.

Esta representação gerada é utilizada para fazer uma análise sobre o código fonte e extrair informações relevantes a respeito das evidências de programação utilizadas. Uma representação da avaliação automática, utilizando a técnica citada, sendo aplicada para um exercício na linguagem de programação C é ilustrado pela Figura 2.9. Na representação em árvore, é possível identificar as evidências e contabilizar a quantidade de vezes que uma solução utiliza uma evidência que está ligada a alguma habilidade de programação.

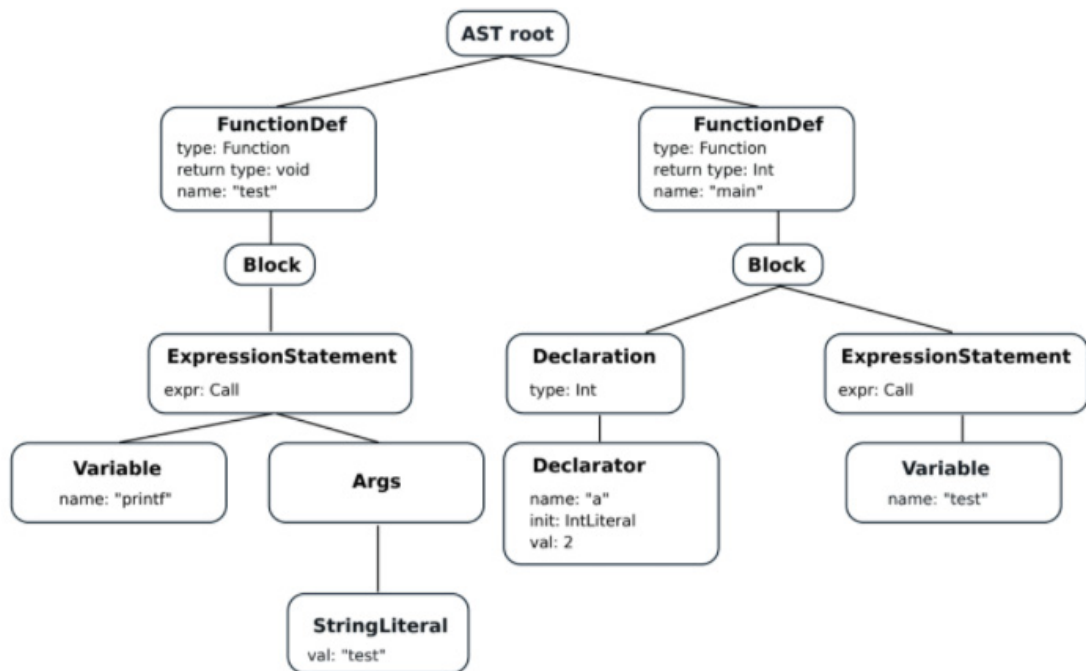


Figura 2.8: Execução do parser resultado em uma árvore sintática abstrata.

Fonte: Porfirio (2020)

```

1 void test(){
2     printf("test!");
3 }
4 int main(){
5     int a = 2;
6     test();
7 }
  
```

Figura 2.9: Exemplo de código fonte na linguagem C.

Fonte: Porfirio (2020)

De acordo com Porfirio (2020), as evidências possíveis de computar por meio da estratégia de detecção automática para o conjunto de habilidades definidas por (Krynski, 2013), são as seguintes de acordo com a Tabela 2.2.

O trabalho revelou que a avaliação automática de aspectos de programação é uma tarefa que pode muitas vezes revelar situações inesperadas que causam falhas em estratégias tutoriais. A criatividade dos estudantes é imprevisível. Aprendizizes podem seguir as direções do professor e escrever códigos de acordo ou tentar por conta própria até atingir uma solução válida. Ao seguir por conta própria o estudante muitas vezes mistura metodologias o que pode causar que estratégias tutoriais não consigam avaliar as soluções dos aprendizes de maneira devida. Com isso, comportamentos estranhos são encontrados nos dados que causam impactos negativos nas estratégias de avaliação pelo motivo de avaliar evidências incorretamente.

A proposta, implementação e demonstração do uso de estratégias automatizadas como um meio de alto nível e a avaliação baseada em habilidades pode ser vista como impactos positivos sobre os métodos de avaliação existentes, especialmente ao empregar as informações resultantes como forma de monitorar o progresso das habilidades do aluno e detectar possíveis falhas de

Habilidades	Descrição
output	saídas
input	entradas
attribution	atribuições
types of literals	tipos de literais
variables	variáveis
constants	constantes
arithmetic expressions	expressões aritméticas
relational expressions	expressões relacionais
division by zero	divisões por zero
boolean expressions	expressões booleanas
multiple selection conditional	condicionais de seleção múltipla
simple and compound conditionals	condicionais simples e compostos
nesting	aninhamentos
infinite loops	loops infinitos
counted loops	loops contados
pre-evaluated	pré avaliações
post evaluated	pós avaliações
nesting (2)	aninhamentos
arrays	vetores
matrices	matrizes
functions	funções

Tabela 2.2: Habilidades computadas pela detecção automática.
 Fonte: Porfirio (2020)

conceito. Em relação ao nicho de pesquisa investigado, o estudo, contribuiu na atualização e expansão do estado da arte por meio de revisão sistemática da literatura feita no trabalho.

Comparando vários estados do modelo do estudante, é possível detectar quais conjuntos de habilidades se manifestam em diferentes conjuntos de códigos fonte analisados. Outro ponto, a comparação permite a visualização se os alunos progrediram ou não ao longo do curso e permite identificar tópicos de programação de computadores que requerem atenção especial ao ser ensinado. Aplicar evidências de aprendizagem identificadas automaticamente como fonte para alimentar o modelo do estudante é considerado um ponto forte do trabalho evidenciado. Avaliando vários aspectos de um único o código fonte é possível fornecer informações sobre as habilidades de programação empregadas pelo aluno durante a escrita e realização de um exercício. A visualização dessas informações no modelo do estudante fornece uma ideia sobre a visão geral das habilidades exercitadas em um código fonte.

3 TRABALHOS RELACIONADOS

Neste capítulo, descrevem-se os passos utilizados para a condução de um mapeamento sistemático da literatura que é utilizado para identificar trabalhos conceituados pelo estado da arte relacionados com o tema explorado no trabalho.

3.1 MAPEAMENTO SISTEMÁTICO DA LITERATURA

Para compreender o estado da arte, é necessário situar-se sobre as contribuições já existentes, as limitações e os desafios presentes no campo de pesquisa. Para solucionar isso, evidencia-se o mapeamento sistemático da literatura para ilustrar os trabalhos da área.

O mapeamento sistemático da literatura surge da necessidade de investigar e analisar o estado da arte quanto ao tema “Escolha da próxima questão”. Este tema compreende soluções utilizadas no laço externo de sistemas tutores inteligentes bem como a modelagem do estudante. Com isso, é possível encontrar lacunas de conhecimento nesta área para posterior elaboração de pesquisa.

Mapeamentos sistemáticos da literatura são úteis para nortear o caminho de pesquisadores, visto que a área é "mapeada", isto é, destaca-se os principais achados e busca-se agrupar resultados parecidos nos trabalhos de modo a sumarizar resultados.

Para a realização do mapeamento, utiliza-se um protocolo rigoroso de busca que tem como característica a reprodutividade, isto é: se outro pesquisador realizar o mapeamento, chega-se aos mesmos resultados. Esta característica garante a confiabilidade deste protocolo e torna o método bastante completo.

Os pontos chave do mapeamento são: questões de pesquisa, processo de busca, critérios de inclusão e exclusão e extração de dados.

Este tema compreende trabalhos que propõem soluções utilizadas na ordenação/recomendação/seleção/sequenciamento de questões de acordo com o modelo do estudante. As soluções de escolha da próxima questão estavam caracterizadas dentro dos seguintes termos:

- Content-based: são soluções que utilizam características dos aprendizes podendo fazer uso de ações anteriores do aprendiz para selecionar exercícios;
- IA: são soluções que utilizam alguma técnica de inteligência artificial, incluindo técnicas de otimização como algoritmos genéticos, para a realização de escolha da próxima questão.

O modelo do estudante mapeia o conhecimento atual do estudante (rastreamento de conhecimento). Diversas soluções são utilizadas na modelagem do estudante. Algumas delas são:

- Grafos genéticos são um conjunto de vértices que representam capacidades interconectados por arestas que são as relações estabelecidas entre os vértices;
- BKT (Bayesian Knowledge Tracing) é um algoritmo utilizado na área de sistemas tutores inteligentes para modelar o nível de conhecimento de um aprendiz a partir de um conhecimento sendo tutorado;

- DKT (Deep Knowledge Tracing) é a modelagem do aprendizado do estudante utilizando redes neurais recorrentes.

Este capítulo busca evidenciar os algoritmos e soluções utilizadas para a escolha de questões personalizada. Resultados apontam para um panorama sobre o estado da arte nos últimos anos.

O capítulo organiza-se da seguinte forma: A seção 3.1.1 apresenta o método de pesquisa. A seção 3.1.2 apresenta os resultados. A seção 3.1.3 conclui o capítulo.

3.1.1 Método de Pesquisa

Para analisar e revisar o estado da arte quanto ao tema "Escolha da próxima questão", foi conduzido um mapeamento sistemático da literatura de forma a selecionar artigos relevantes e extrair os dados necessários. Seguindo as diretrizes de Kitchenham e Charters (2007), o mapeamento cobre desde a definição de questões de pesquisa até os critérios de inclusão e exclusão bem como a extração de dados e análise.

Questões de pesquisa: Foram listadas 6 questões de pesquisa principais:

QP1: Como as publicações sobre a escolha da próxima questão estão distribuídas ao longo do tempo? Resultados esperados: verificar como ocorre a evolução dos trabalhos ao longo dos anos;

QP2: Como é o algoritmo de escolha da próxima questão? Resultados esperados: descrever como é o algoritmo de escolha da próxima questão;

QP3: Com base em quais características é feita a escolha da próxima questão? Resultados esperados: descrever quais características são levadas em consideração na escolha da próxima questão;

QP4: Qual é a solução utilizada na modelagem do estudante? Resultados esperados: identificar como é feita a modelagem do estudante;

QP5: Qual é o domínio do estudo? Resultados esperados: identificar qual área o estudo está inserido;

QP6: O estudo evidenciado apresenta experimento empírico? Resultados esperados: verificar se foi conduzido experimentação utilizando a solução proposta.

Processo de Busca: Para conduzir a pesquisa, foi escolhida a biblioteca digital do Google Scholar. Foi utilizado o intervalo de tempo entre 2015-2020, de modo a selecionar os trabalhos mais recentes na literatura. Esta biblioteca foi escolhida devido à simplicidade e praticidade em realizar buscas e por retornar artigos relevantes dentro da área de informática. Algumas restrições consistem na sumarização dos dados. Por se tratar de um tema em que várias soluções diferentes são propostas, torna-se difícil classificar e agrupar os trabalhos. Não existe uma taxonomia própria para sintetizar os dados dos trabalhos. Com isso, faz-se necessário analisar os trabalhos e descrevê-los individualmente.

A string é composta pelos termos *intelligent tutoring systems, example, exercise, question, ordering, recommendation, selection e sequencing*. Devido a um limite de caracteres, a string de busca foi dividida em 4 sub strings de modo a tornar possível a pesquisa com todos os termos

String de Busca
"intelligent tutoring systems"AND ("example ordering"OR "exercise ordering"OR "question ordering")
"intelligent tutoring systems"AND ("example recommendation"OR "exercise recommendation"OR "question recommendation")
"intelligent tutoring systems"AND ("example selection"OR "exercise selection"OR "question selection")
"intelligent tutoring systems"AND ("example sequencing"OR "exercise sequencing"OR "question sequencing")

Tabela 3.1: Strings de busca.

Critério	Descrição
CI1	abordar soluções de seleção de questão em um contexto educacional
CI2	descrever sobre o rastreamento de conhecimento.
CE1	não atender aos critérios de inclusão
CE2	não estar escrito em inglês ou português
CE3	ser duplicata
CE4	ser literatura cinza
CE5	artigo não encontrado na base de dados ou acesso restrito

Tabela 3.2: Critérios de inclusão e exclusão.

definidos respeitando as limitações de tamanho na hora de preencher o campo de pesquisa da biblioteca de busca. As strings de busca derivam-se, para a biblioteca digital Google Scholar, Tabela 3.1.

Critérios de Inclusão e Exclusão: Os artigos retornados pela busca foram selecionados através de filtros, de forma a selecionar os artigos relevantes e excluir os que não possuem relação aos objetivos de pesquisa. Para conduzir este mapeamento sistemático, foi considerado que os estudos deveriam estar de acordo com os seguintes critérios de inclusão/exclusão, Tabela 3.2.

O procedimento adotados para a seleção de artigos é o seguinte, ilustrado pela Figura 3.1:

- 1º Filtro: leitura do título, resumo e palavras-chave
- 2º Filtro: leitura do artigo

Extração de Dados: Após a busca na biblioteca digital, utilizando a string de busca, aplica-se o primeiro filtro que consiste na leitura do título, resumo e palavras-chave. Retornados os resultados do primeiro filtro, aplica-se o segundo filtro que consiste na leitura do artigo. Após a aplicação de todas etapas de seleção de artigos, faz-se a extração de dados relevantes para o mapeamento. No caso do tema “Escolha da próxima questão”, definiu-se os seguintes dados a serem extraídos de acordo com as questões de pesquisa, Tabela 3.3.

3.1.2 Resultados e Discussão

Esta seção mostra os detalhes sobre o processo e os dados obtidos nos processos de busca e filtros dos artigos. É demonstrado a extração de dados para as questões de pesquisa.

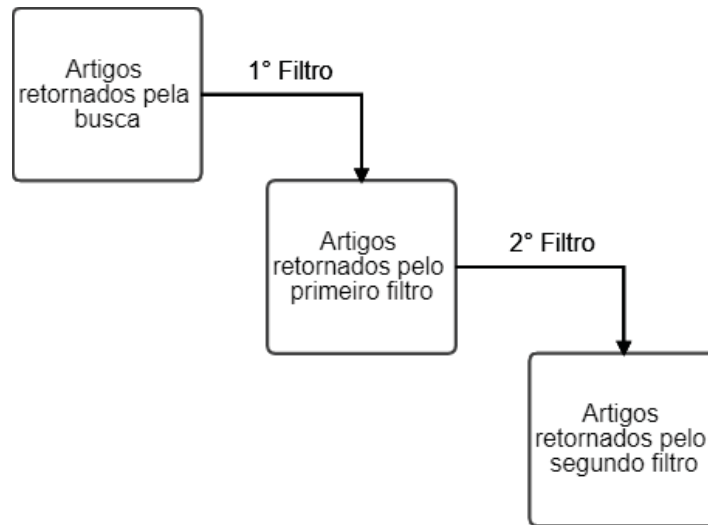


Figura 3.1: Ilustração do processo de busca e filtros utilizados na seleção dos trabalhos.

QP	Atributo
QP1	Ano da publicação
QP2	Algoritmo de escolha da próxima questão
QP2	Como é o tipo do algoritmo (IA/Content-Based/Algoritmo Genético/Outro)
QP3	Características consideradas na escolha da próxima questão
QP4	Modelo do estudante
QP5	Domínio
QP6	Possui experimento empírico

Tabela 3.3: Questões de pesquisa.

3.1.3 Distribuição do Tempo (QP1)

A distribuição dos artigos ao longo do tempo é equilibrada, evidenciado pela Figura 3.2. Em 2018 houve uma maior quantidade de artigos referentes ao tema “Escolha da próxima questão”. Percebe-se que nos últimos anos, há uma tendência pelo uso da inteligência artificial nos trabalhos, entretanto algoritmos content-based ainda seguem como a maioria encontrada nos trabalhos.

3.1.4 Tipo de Algoritmo (QP2)

A grande maioria (67.7%) dos artigos utiliza uma abordagem de algoritmo content-based, Figura 3.3. Esses algoritmos de recomendação utilizam características relacionadas ao estudante como parâmetros para a recomendação de questões. Com base nestas características, é utilizada alguma técnica computacional e faz-se uma recomendação com base nas necessidades do usuário. 12.9% dos artigos utiliza técnicas de IA para fazer uma recomendação. Algumas técnicas evidenciadas são: reinforcement learning, neural networks, neural collaborative filtering. 6.5% utilizam algoritmos genéticos. Os algoritmos genéticos também podem ser considerados técnicas de IA, mas foram relacionados nesse mapeamento separadamente devido a categoria de artigos que os utilizam. Estes exploram a compatibilidade entre o conteúdo de aprendizagem com os aprendizes e a geração de testes baseados na performance de grupos de estudantes. Outras técnicas (12.9%) também se mostram presentes: concept-based similarity, deeper knowledge



Figura 3.2: Distribuição ao longo dos anos.

tracing by modeling skill application, syntactic parse-key tree based approach, students' memory for application adaptive educational systems, Tabela 3.4.



Figura 3.3: Tipos de algoritmos.

3.1.5 Características Utilizadas na Escolha da Próxima Questão (QP3)

Investigando as características utilizadas na escolha da próxima questão, variáveis utilizadas na personalização da seleção podem ser descritas tais como: habilidade menos desenvolvida, histórico de exercícios, performance, esforço mental, nível de conhecimento, nível de conhecimento prévio, estilo de aprendizagem, nível de taxonomia de bloom, traços de personalidade (autoestima, conscienciosidade, estabilidade emocional, extroversão, abertura para experiências), dentre outras. Esta parte é bastante específica entre os trabalhos analisados. Cada trabalho foca em uma característica ou um conjunto delas para realizar a seleção personalizada.

Tipo de Algoritmo	Trabalhos
Content-Based	(Dang e Ghergulescu, 2018) (Teusner et al., 2018) (Oliveira et al., 2015) (Oliveira et al., 2016) (Okpo et al., 2016) (Okpo et al., 2018) (Jurenoka e Jurenoks, 2018) (Okpo et al., 2017) (Hou e Fidopiastis, 2017) (Alhathli et al., 2018) (Zhang e VanLehn, 2017) (Alhathli et al., 2016) (Alhathli et al., 2020) (Maestro-Prieto e Simon-Hurtado, 2018) (Saleh e Salama, 2018) (González-González et al., 2019) (Pelánek et al., 2017) (Huo et al., 2018) (Chrysafiadi et al., 2018) (Meliana e Nurjanah, 2018) (Huo et al., 2020)
IA	(Huo et al., 2020) (Ai et al., 2019) (Chaplot et al., 2016) (Sidi e Klein, 2020)
Algoritmo Genético	(Christudas et al., 2018) (Wu et al., 2020)
Outro	(Fang et al., 2018) (Hosseini e Brusilovsky, 2017) (Huang, 2016) (Pelánek, 2015)

Tabela 3.4: Resultados QP2.

3.1.6 Modelo do Estudante (QP4)

Considerando as soluções utilizadas na modelagem do estudante, a grande maioria dos trabalhos (33.3%) utiliza soluções baseadas em BKT (bayesian knowledge tracing). Percebe-se que as soluções utilizadas são bastante específicas entre os trabalhos. Os trabalhos diversificam a forma que a modelagem do estudante acontece. 20% utilizam alguma forma de modelagem que trabalha com traços de personalidade do aprendiz. Outras abordagens encontradas utilizavam estilos de aprendizagem (3.3%), DKT (deep knowledge tracing) (10%), Q-Matrix (6.7%), entre outras soluções (26.7%) que usavam diferentes abordagens, representado pela Figura 3.4 e Tabela 3.5.

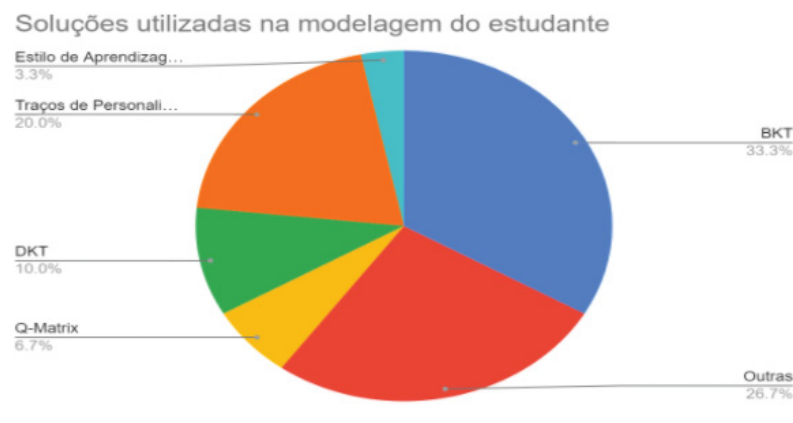


Figura 3.4: Soluções utilizadas no modelo do estudante.

3.1.7 Domínio (QP5)

Investigando o domínio dos artigos, percebe-se que há uma abrangência grande quanto às áreas estudadas. Como destaque, cita-se o ensino da programação (22.6%) é a área em que se encontram a maior parte dos estudos. Em seguida, evidenciam-se a matemática e o ensino de

Modelagem do Estudante	Trabalhos
BKT	(Dang e Ghergulescu, 2018) (Zhang e VanLehn, 2017) (Huang, 2016) (Hou e Fidopiastis, 2017) (Saleh e Salama, 2018) (Oliveira et al., 2015) (Oliveira et al., 2016) (Meliana e Nurjanah, 2018) (Jurenoka e Jurenoks, 2018) (González-González et al., 2019)
Traços de Personalidade	(Alhathli et al., 2020) (Alhathli et al., 2016) (Alhathli et al., 2018) (Okpo et al., 2016) (Okpo et al., 2017) (Okpo et al., 2018)
Estilo de Aprendizagem	(Christudas et al., 2018)
DKT	(Wu et al., 2020) (Ai et al., 2019) (Chaplot et al., 2016)
Q-Matrix	(Huo et al., 2020) (Huo et al., 2018)
Outras	(Maestro-Prieto e Simon-Hurtado, 2018) (Hosseini e Brusilovsky, 2017) (Chrysafiadi et al., 2018) (Pelánek et al., 2017) (Pelánek, 2015) (Fang et al., 2018) (Sidi e Klein, 2020) (Teusner et al., 2018)

Tabela 3.5: Resultados QP4.

línguas estrangeiras como assuntos bastante abordados (12.9%). 9.7% dos artigos caracterizam abordagens genéricas, isto é, que podem ser aplicadas em diferentes áreas de conhecimento. Outras áreas de destaque foram álgebra (6.5%), aritmética (6.5%), inglês (6.5%) e geografia (6.5%). Diversas outras áreas (16.1%) também foram utilizadas nos estudos, ilustrado pela Figura 3.5 e Tabela 3.6.

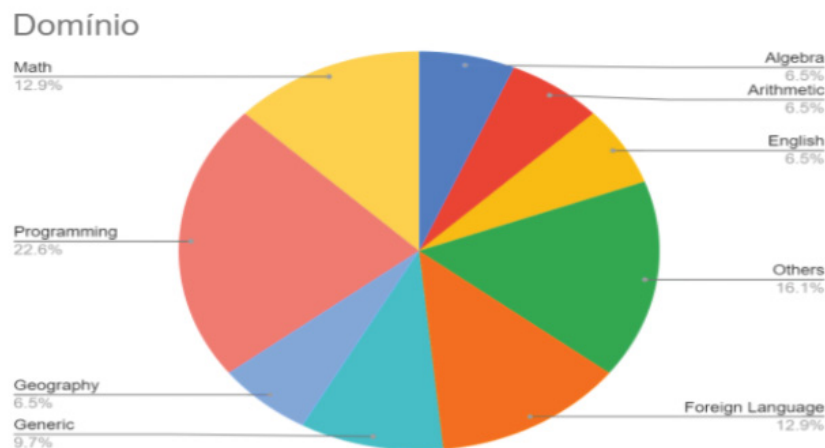


Figura 3.5: Domínio dos trabalhos.

3.1.8 Experimentos (QP6)

Quanto aos experimentos conduzidos nos trabalhos, a grande maioria (86.7%) realizou validações e obteve resultados empíricos que são evidenciados nos artigos, Figura 3.6. Aqueles que não realizaram experimentos (13.3%) apresentavam soluções e descreviam qualitativamente, entretanto deixavam como trabalhos futuros as experimentações.

Domínio	Trabalhos
Programação	(Chrysafiadi et al., 2018) (Hosseini e Brusilovsky, 2017) (Oliveira et al., 2015) (Oliveira et al., 2016) (Meliana e Nurjanah, 2018) (Christudas et al., 2018) (Teusner et al., 2018)
Matemática	(Wu et al., 2020) (Dang e Ghergulescu, 2018) (Ai et al., 2019) (Okpo et al., 2016)
Língua Estrangeira	(Chrysafiadi et al., 2018) (Huo et al., 2020) (Alhathli et al., 2020) (Alhathli et al., 2016)
Genérico	(Huang, 2016) (Alhathli et al., 2018) (Jurenoka e Jurenoks, 2018)
Álgebra	(Chaplot et al., 2016) (Sidi e Klein, 2020)
Aritmética	(Okpo et al., 2017) (Okpo et al., 2018)
Inglês	(Fang et al., 2018) (Huo et al., 2018)
Geografia	(Pelánek et al., 2017) (Pelánek, 2015)
Outros	(Zhang e VanLehn, 2017) (Maestro-Prieto e Simon-Hurtado, 2018) (Hou e Fidopiastis, 2017) (González-González et al., 2019) (Saleh e Salama, 2018)

Tabela 3.6: Resultados QP5.

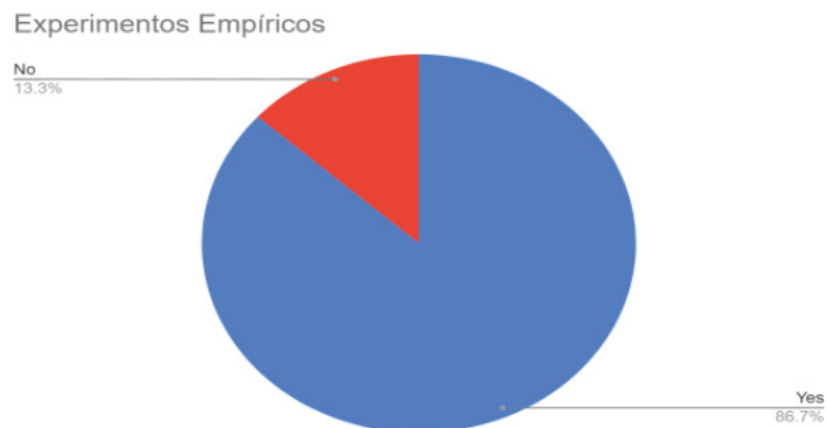


Figura 3.6: Trabalhos que realizaram experimentos para validação.

3.1.9 Conclusão

Esta pesquisa apresenta um mapeamento sistemático da literatura sobre o tema "Escolha da próxima questão" que compreende soluções utilizadas na seleção de exercícios bem como a modelagem do estudante. O protocolo de pesquisa apresentado, resultou na seleção, utilizando o Google Scholar, de 30 trabalhos oriundos de diversos veículos de publicação de artigos científicos. Os resultados revelaram soluções algorítmicas na seleção de exercícios sendo: content-based, utilizando técnicas de inteligência artificial, algoritmos genéticos bem como outras técnicas. Percebe-se que nos anos mais recentes, há uma crescente onda por técnicas de IA.

A primeira questão de pesquisa (QP1) demonstrou a evolução de trabalhos seguindo o tema da "Escolha da próxima questão" e a quantidade de artigos por ano. Dentro do escopo de

artigos selecionados, nota-se que os seguintes trabalhos foram realizados por um mesmo autor em comum, comprovado por mais de um trabalho na área:

- Okpo et al. (2016), Okpo et al. (2017), Okpo et al. (2018);
- Oliveira et al. (2015), Oliveira et al. (2016);
- Pelánek (2015), Pelánek et al. (2017);
- Alhathli et al. (2016), Alhathli et al. (2018), Alhathli et al. (2020).

A segunda questão de pesquisa (QP2) teve como objetivo localizar as soluções mais utilizadas na escolha da próxima questão. Três categorias foram utilizadas para classificar as soluções. Citando elas, foram: content-based, IA e algoritmos genéticos. Isto serviu para classificar as abordagens utilizadas nos trabalhos. Subsequentemente, a (QP3) tratou das características utilizadas nas soluções de escolha da próxima questão. Os dados revelaram que diversas são as características utilizadas pelos trabalhos. Notou-se que algumas características foram repetidas em diversos trabalhos, entretanto há bastante divergência na utilização dessas informações. É difícil classificar características por trabalhos. Este é um ponto individual e específico dos trabalhos. Tornou-se difícil sumarizar e agrupar as características utilizadas para a escolha da próxima questão nos trabalhos. Adicionalmente, a (QP4) buscou identificar soluções utilizadas para modelar os estudantes. Algumas técnicas foram mais evidenciadas como estilos de aprendizagem, BKT e DKT. Outras soluções diferentes também foram evidenciadas entre os trabalhos e alguns não descreviam como o estudante era modelado. A questão (QP5) buscou ilustrar os diferentes domínios em que os trabalhos estavam inseridos. A grande maioria estava inserida na programação de computadores. Por fim, (QP6) investigou a condução de experimentos empíricos. A grande maioria dos trabalhos fez experimentos e analisou as soluções propostas. Uma pequena parcela de trabalhos permaneceu apenas na teoria e outra somente descreveu as soluções qualitativamente sem apresentar dados quantitativos.

A literatura demonstrou uma grande variação nesta área. Foram encontrados diferentes trabalhos que empregaram diferentes soluções de escolha da próxima questão para diferentes domínios. Entre os trabalhos selecionados, foi possível evidenciar técnicas que eram aplicadas a determinados domínios do conhecimento. Uma questão levantada é: será uma solução de escolha da próxima questão melhor adequada para um determinado domínio? Estudos que evidenciam as melhores técnicas para tratar aprendizes de domínios específicos é uma questão interessante de ser aprofundada.

Um dos trabalhos retornados pelo mapeamento sistemático da literatura que contribuiu para ideia do bom aprendiz foi o trabalho de Meliana e Nurjanah (2018).

De acordo com Meliana e Nurjanah (2018), a realização de exercícios baseada no bom aprendiz, constata-se que o aprendizado pode ser adquirido ao imitar modelos competentes de bons aprendizes. Com isso, constrói-se uma sequência de questões para os alunos seguirem em contrapartida a uma ordem fixa determinada por professores. Espera-se que um sistema tutor inteligente seja capaz de prever quais exercícios são mais apropriado para os alunos, com o mínimo de ajuda de especialistas. A pesquisa de (Meliana e Nurjanah, 2018) é uma tentativa de encorajar a implementação de sistemas tutores inteligentes em universidades com um número abaixo da média de especialistas. Dessa forma, a limitação dessas universidades, pode ser superada por meio da injeção de uma experiência dinâmica de seleção de questões envolvendo o bom aprendiz no sistema. Pode-se concluir que injetar a experiência de bons alunos no sistema deve melhorar a capacidade do sistema em recomendar melhores exercícios para ajudar os alunos

a se tornarem bons alunos. Os resultados do trabalho revelaram que, em média, a nova abordagem produziu 19% a mais de bons alunos.

Adotando a ideia da utilização do modelo do bom aprendiz no processo de ensino-aprendizagem, busca-se, nesse trabalho, expandir a ideia do bom aprendiz ao agregar este modelo na recomendação de questões no domínio da programação de computadores.

4 ABORDAGEM REBAEV

Nesta seção detalha-se a arquitetura da abordagem de recomendação, que consiste na definição dos componentes utilizados e como se relacionam. Um esquema da abordagem é representado pela Figura 4.1. O FARMA-ALG possui seu banco de dados que fornece as questões para a abordagem. A abordagem de recomendação utilizando as questões fornecidas pelo FARMA-ALG, recebe o estado de um aprendiz do FARMA-ALG e com base nas informações recebidas gera a recomendação que é recebida pelo FARMA-ALG.

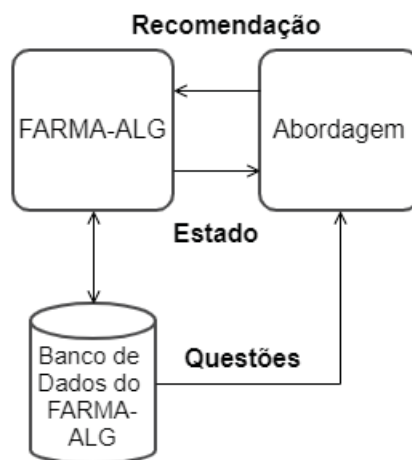


Figura 4.1: Esquema da abordagem.

Neste trabalho, por meio dos conceitos de Vanlehn (2006), descritos no Capítulo 2, foca-se no laço externo através da recomendação de questões de programação dentro do contexto de aprendizes resolvendo exercícios. Desta forma, as iterações, contextualizadas dentro do trabalho, seriam a recomendação de exercícios com base nos passos anteriores do aluno que são os exercícios em que o aluno já fez. Um evento de aprendizado seria quando o aprendiz resolve os exercícios com êxito que é refletido através de uma componente de conhecimento referente à habilidade desenvolvida dentro do modelo do estudante.

Considerando a abordagem de recomendação, caracterizam-se cenários no contexto do FARMA-ALG em que a recomendação de questões é objetivada.

4.1 ARQUITETURA REBAEV

No FARMA-ALG, existem diversas turmas que contém vários alunos. Dentro dessas turmas, é possível visualizar o progresso dos alunos observando-se a quantidade de resolução dos exercícios das listas, ilustrado pela Figura 4.2. O primeiro aluno resolveu 114 questões que corresponde à 100% da lista. O segundo aluno resolveu 91 exercícios. O terceiro 24 exercícios. O quarto aluno 2 exercícios e o quinto 89 exercícios.

Em relação ao progresso dos alunos, é possível investigar diferentes situações durante a resolução dos exercícios por estes aprendizes, mas que são recorrentes dentro do sistema tutor inteligente. Inspeccionando o caminho de diferentes alunos, é possível evidenciar diferentes cenários que são caracterizados a seguir.



Figura 4.2: Exemplo de progresso das questões corretas dos alunos.

4.1.1 Recomendação Utilizando o Caminho do Bom Aprendiz

A recomendação para aluno regular consiste na resolução de exercícios alternativos do sistema. A recomendação utiliza como parâmetros o caminho do aluno bem como o caminho gerado pelos bons aprendizes. Calcula-se a distância relativa dos exercícios a partir do último exercício em que o aluno resolveu.

Aluno regular - este é o cenário evidenciado pela Figura 4.3, que representa um aluno progredindo pela lista de exercícios em um ritmo normal sem que ele trave em algum exercício;

• Exercício 105	Correto 1 tentativas
• Exercício 106	Correto 6 tentativas
• Exercício 107	Correto 8 tentativas
• Exercício 108	Correto 3 tentativas

Figura 4.3: Exemplo de aluno regular.

A arquitetura da recomendação de questões para um aluno regular é ilustrada pela Figura 4.4. A partir dos dados do FARMA-ALG, foi gerada a sequência de exercícios percorrida pelos bons aprendizes, e utilizando-se do cálculo das distâncias relativas aos caminhos percorridos, sugeriu-se a resolução dos exercícios mais próximos em relação ao último exercício que o aluno fez. Gerou-se uma tabela com as distâncias relativas dos exercícios proximais com distância d , definida no algoritmo. Aqueles exercícios que apresentam a menor distância são selecionados como opções de recomendação para o aluno.

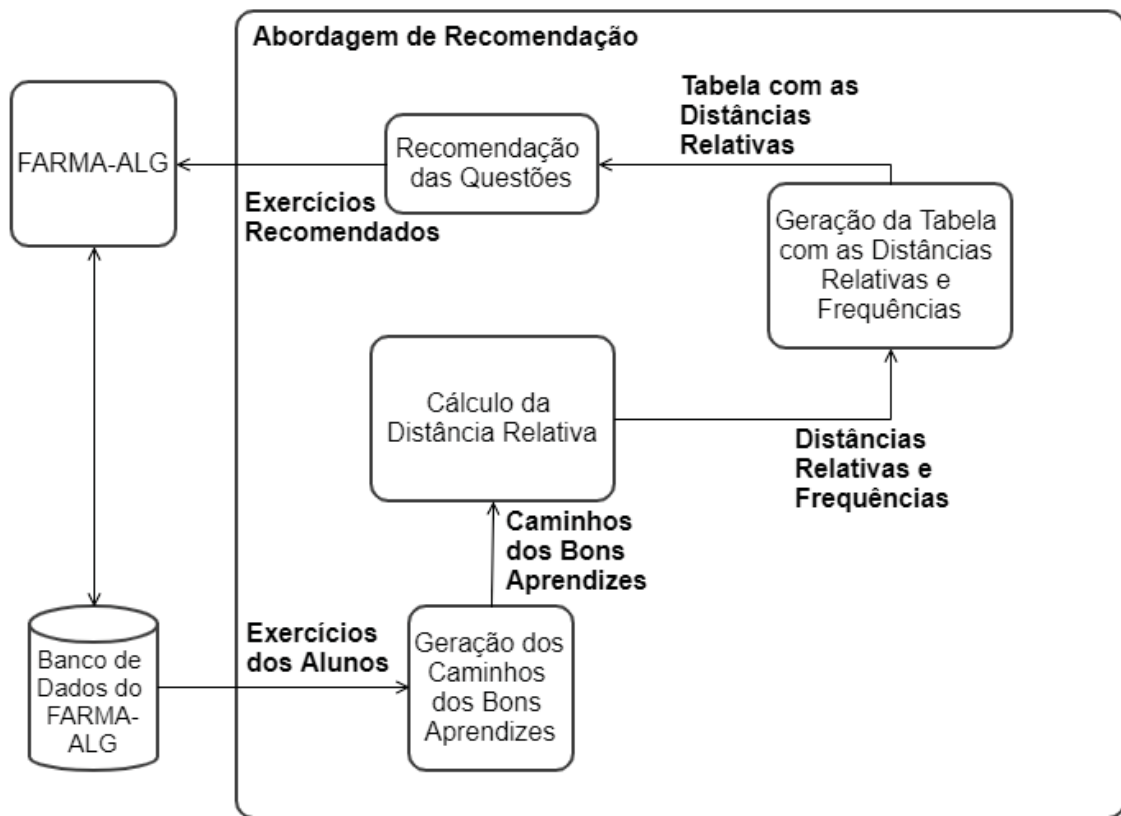


Figura 4.4: Arquitetura de recomendação para aluno regular.

4.1.2 Recomendação Utilizando Evidências de Aprendizado Aluno Travado

A recomendação para o aluno travado é utilizada quando o sistema detecta um alto valor de submissões de tentativas não solucionadas dos exercícios. Nesse intuito, o sistema busca ajudar o aluno, através da recomendação de exercícios com uma menor quantidade média de evidência de aprendizado.

Aluno travado - este é o cenário do terceiro aluno da Figura 4.2. O aluno começou resolvendo os exercícios da lista até chegar em um exercício de Vetores. Representado pela Figura 4.5, evidencia-se que o aluno tentou submeter uma solução para o problema de Vetores por 23 tentativas sem obter êxito até desistir de resolver exercícios;



Figura 4.5: Exemplo de aluno travado.

Conceituando esta recomendação, demonstrada pela Figura 4.6, a arquitetura da recomendação de questões para um aluno travado é feita mediante os dados do FARMA-ALG. Para todos os exercícios de uma lista, seleciona-se a última versão submetida de todos os códigos fonte dos exercícios resolvidos pelos alunos. Com todos esses códigos fonte, avança-se para a etapa de identificação automática de evidências de habilidades em programação. Para todos os códigos fonte, calculam-se a quantidade de evidências. Calculadas as evidências nos códigos, gera-se um relatório por código contendo as informações relacionadas às habilidades de

programação identificadas com suas quantidades. Após a geração do relatório com as evidências de programação, calcula-se a média de evidências de programação por exercício. O cálculo dessa média será explicado na Seção 4.6. Para uma determinada lista de exercícios, são consideradas determinadas habilidades trabalhadas nesta lista, definidas pelo professor de programação. Após o cálculo da média de quantidade de evidências de programação por exercícios, gera-se uma tabela contendo o resultado. Esta tabela é utilizada na recomendação de questões, quando o aluno é considerado travado. Um aluno que trava em uma determinada questão tem a recomendação de exercícios que contenham uma quantidade de evidências de aprendizado menor em relação ao exercício em que ele travou.

Quando um aluno comete erros sucessivos em uma mesma questão, o sistema detecta uma possível situação em que o aluno está com dificuldade e sugere até 3 exercícios de reforço ao aluno de maneira a tentar ajudá-lo. O aluno possui a liberdade para aceitar algum desses exercícios ou permanecer no exercício em que ele se encontra fazendo.

Papoušek et al. (2016) analisa o impacto da dificuldade das questões no engajamento e aprendizado dos alunos utilizando um sistema educacional. Um sistema adaptativo educacional deve ser desafiador ao mesmo tempo que engajante, deve-se equilibrar a dificuldade dos exercícios em conjunto do engajamento no contexto do aprendizado. Questões mais fáceis contribuem para o engajamento a curto prazo enquanto que o aprendizado é melhor para questões mais difíceis conceitualmente. Tendo em vista essas características, alcançar o engajamento a curto prazo fazendo com que os alunos não abandonem imediatamente utilizando questões de dificuldade mais simples para aumentar o engajamento a curto prazo. Em contrapartida, questões de dificuldade maior são importantes para promover o uso prolongado do sistema, melhorar o aprendizado e não desmotivar os alunos ao resolver somente questões de pouca dificuldade.

Os exercícios de reforço servem para aumentar o engajamento a curto prazo e ajudar o aluno a reforçar os conceitos que talvez não estejam claros, comprovado pelas tentativas erradas sucessivamente. Para isso, recomenda-se questões de dificuldade mais simples para que o aluno não fique desmotivado com o erro e continue utilizando o sistema educacional. Ao resolver uma outra questão com menor dificuldade, é possível que o aluno consiga assimilar o que era necessário para resolver a questão difícil anteriormente. Dessa forma, o sistema cativa o aluno a não desistir mesmo em face do erro e oferece outra questão que reforça o conceito. A recomendação por meio das evidências não reflete na complexidade, mas sim no quanto um exercício exercita determinada perícia (Oliveira, 2016). Com exercícios mais simples e que trabalhem a habilidade por meio de uma questão semelhante à questão em que a dificuldade foi percebida, o aluno pode assimilar o que antes estava causando o erro e compreenda o motivo do erro.

4.1.3 Recomendação Utilizando Evidências de Aprendizado Aluno Ótimo

A recomendação para aluno ótimo é conceituada por um aluno que esteja indo bem na resolução dos exercícios de uma lista. O sistema busca recomendar exercícios que sejam mais interessantes para que o aluno não perca o engajamento durante a realização dos exercícios. De forma análoga à recomendação para aluno travado, esta recomendação é feita com base na quantidade média de evidências de aprendizado, a média será explicado na Seção 4.6, porém buscando exercícios que trabalhem uma quantidade maior de evidências. Considera-se que a recomendação baseada em evidências, não contempla necessariamente a dificuldade das questões, mas o quanto uma questão tem a capacidade de exercitar uma determinada habilidade.

Aluno ótimo - cenário representado pela Figura 4.7, que mostra um aluno progredindo pela lista rapidamente resolvendo em poucas tentativas os exercícios;

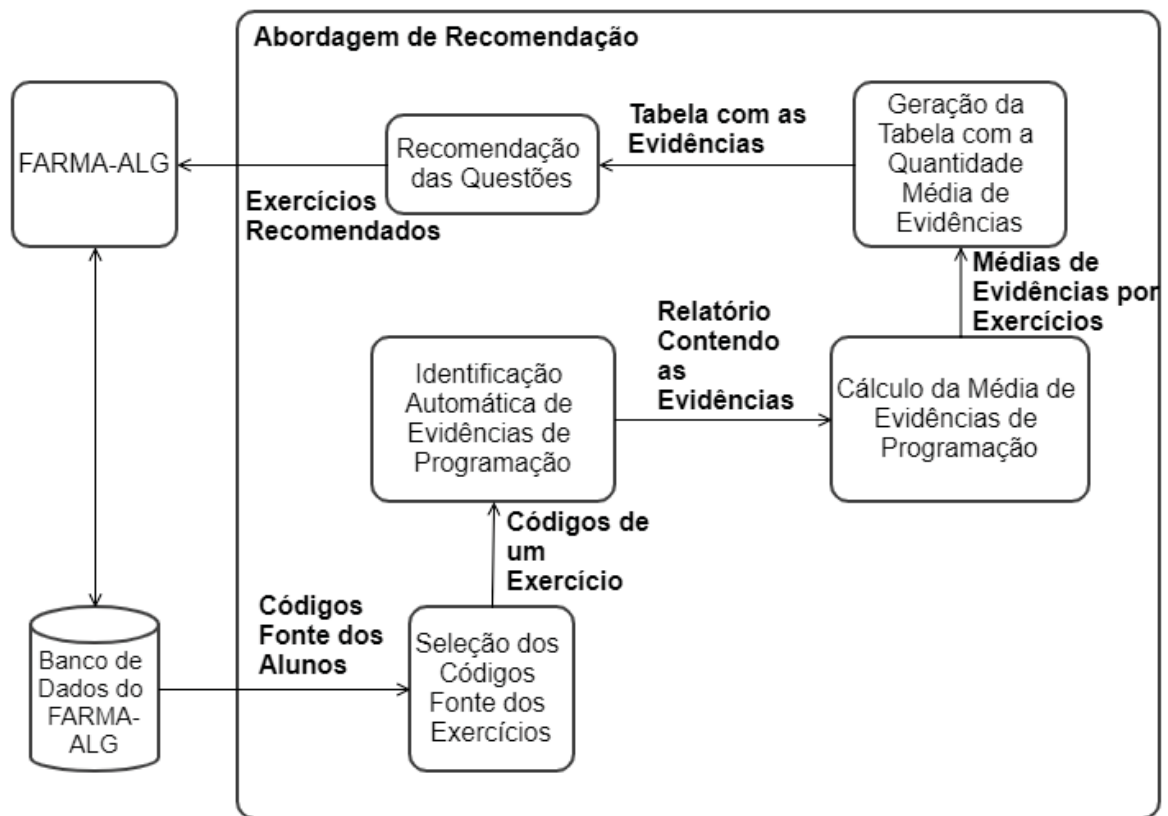


Figura 4.6: Arquitetura para aluno travado e ótimo.

• Exercício 077	Correto 1 tentativas
• Exercício 078	Correto 1 tentativas
• Exercício 079	Correto 1 tentativas
• Exercício 080	Correto 1 tentativas

Figura 4.7: Exemplo de aluno ótimo.

Esta recomendação é representada pela arquitetura na Figura 4.6, entretanto o módulo de recomendação de questões se difere da recomendação para aluno travado no sentido de usar uma quantidade maior de evidências em relação aos últimos exercícios realizados.

4.2 GERAÇÃO DOS CAMINHOS DOS BONS APRENDIZES

O caminho do bom aprendiz, que é a ordem dos exercícios feita por bons alunos, é definido pelos seguintes itens:

- Suponha uma lista de exercícios L , contendo uma quantidade Q finita de exercícios;
- Imagine que um grupo heterogêneo A de alunos recebe a tarefa de resolver exercícios nessa lista como forma de estudo para conceitos que são trabalhados dentro desta lista;
- Os exercícios resolvidos retirados do FARMA-ALG, Figura 4.8, são mapeados em caminhos, representados por nós na ordem de realização, Figura 4.10;
- Esse grupo A de alunos resolve a lista de exercícios. Admita que a resolução dos exercícios nessa lista segue uma distribuição normal - isto é, uma parcela de alunos cumpre uma quantidade ínfima de exercícios, outra fica bem distribuída na média e outra realiza uma grande quantidade.

Seguindo os pressupostos, admite-se como alunos bons a parcela que resolveu uma quantidade de exercícios maior que a média somada do desvio padrão ilustrado em cinza pela Figura 4.9. Esta parcela é escolhida, pois ilustra uma parcela de aprendizes que comprovou certa aptidão nas habilidades de uma lista por concluir um número de exercícios suficiente.

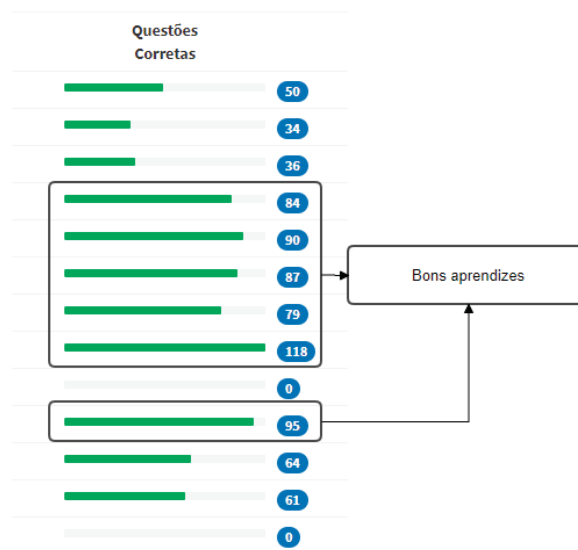


Figura 4.8: Exemplo de conjunto de bons aprendizes.

Considerando o conjunto BA como o conjunto do caminho dos exercícios realizados por estes bons alunos. Levanta-se a questão: Será possível investigar a ordem do caminho desses aprendizes e aprender um padrão de proximidade da escolha das questões que tenha se repetido em mais de um aprendiz e utilizar este padrão em um sistema de recomendação a outros alunos de maneira que se possa criar outros bons aprendizes

Este módulo é caracterizado pela geração dos caminhos dos bons aprendizes. Um caminho de um aprendiz pode ser ilustrado pela ordem em que um aluno resolveu os exercícios no FARMA-ALG. Um exemplo de caminho com 4 exercícios é representado pela Figura 4.10.

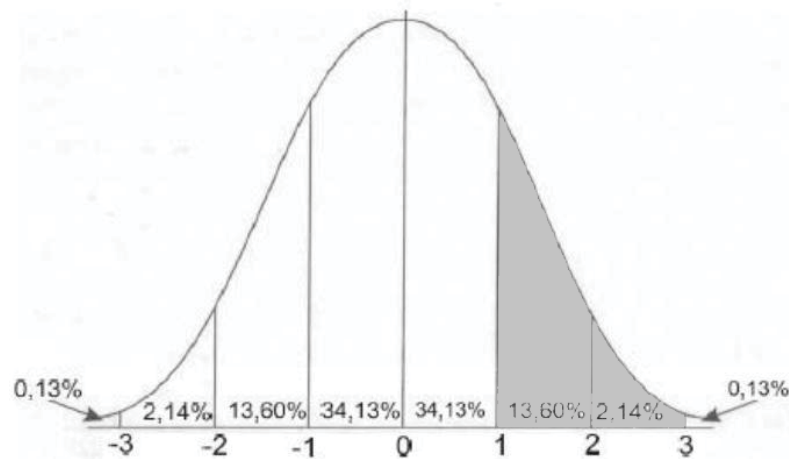


Figura 4.9: Distribuição das notas dos bons aprendizes.

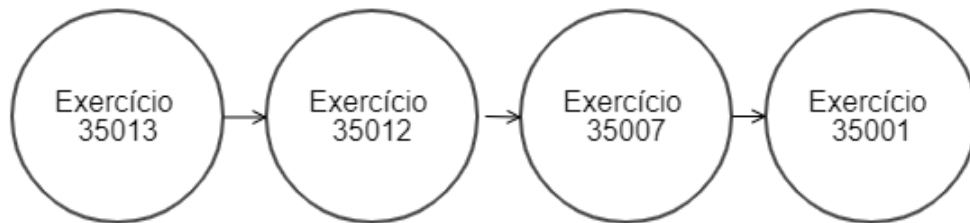


Figura 4.10: Caminho de exercícios.

4.3 CÁLCULO DA DISTÂNCIA RELATIVA

A distância relativa é a distância entre a posição que o exercício ocuparia na lista original e a posição que o exercício ocupa na ordem de execução dos bons aprendizes. Essa distância é calculada pela média das posições do exercício nas ordens de resolução dos bons aprendizes. O cálculo da distância relativa é expresso pela seguinte fórmula, onde:

- D_i é a distância relativa do exercício pivô ao exercício em que se deseja calcular, para o aprendiz i tendo como condição estar a uma distância de no máximo d exercícios;
- n é a quantidade de aprendizes;
- Q é a quantidade de vezes (frequência) em que os exercícios aparecem nos caminhos dos bons aprendizes.
- R é a soma das distâncias relativas em relação à frequência.

$$R = \frac{D_1 + .. + D_n}{Q}$$

Para considerar as questões recomendadas a partir de uma questão prévia, calcula-se neste conjunto de aprendizes as próximas questões mais relevantes de acordo com a escolha desses bons aprendizes. Este cálculo é caracterizado pelos exercícios subsequentes utilizando a posição relativa dos exercícios. Para exemplificar em um conjunto onde o grupo de alunos $n=4$ e

distância de cobertura $d=3$, deseja-se calcular a distância relativa das questões a partir do pivô, Exercício 001, ilustrado pela Figura 4.11 e Tabela 4.1.

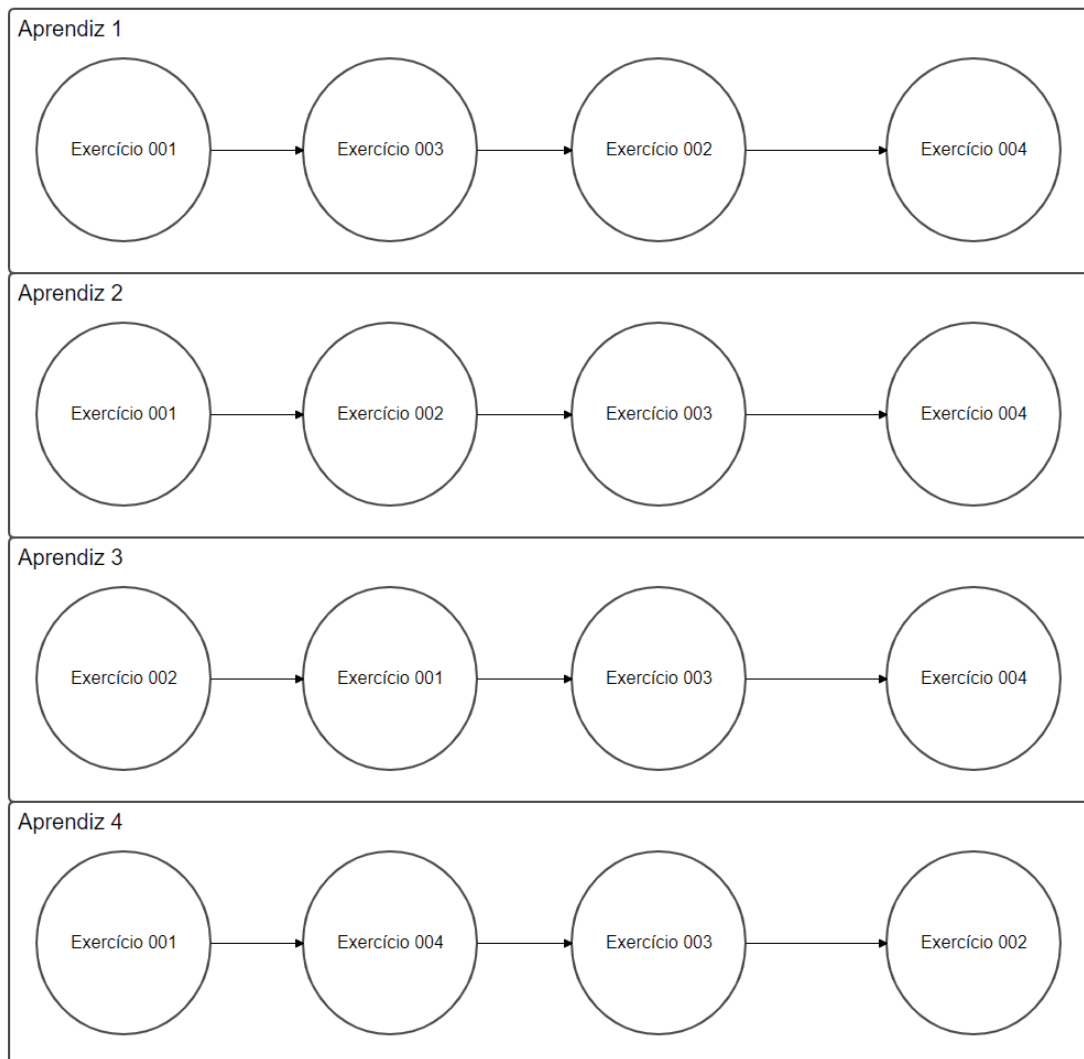


Figura 4.11: Demonstração da ordem dos exercícios.

4.3.1 Cálculo Aplicado

Os exercícios proximais são exercícios cuja distância até o exercício pivô é menor ou igual que a distância de cobertura. Estes exercícios serão selecionados e utilizados no cálculo da distância relativa.

Dessa forma, o módulo de cálculo da distância relativa é responsável por receber os caminhos dos bons aprendizes e calcular a distância relativa dos exercícios proximais estabelecida pelo parâmetro de distância de cobertura em relação a um determinado exercício pivô.

Para exemplificar, um conjunto de 10 bons aprendizes, demonstrado no Apêndice A, é considerado para calcular a recomendação. Imagine um aprendiz cujo caminho é representado pela Figura 4.12. Esse aprendiz resolve o exercício 35000 e em sequência se depara com o exercício 35001 que é o próximo exercício da lista. O aluno infeliz com o exercício fornecido pela lista solicita uma recomendação alternativa ao exercício 35001 para o sistema.

Exercício	Distância Relativa
Exercício 002	$(2 + 1 + 1 + 3) / 4 = 1.75$
Exercício 003	$(1 + 2 + 1 + 2) / 4 = 1.5$
Exercício 004	$(3 + 3 + 2 + 1) / 4 = 2.25$

Tabela 4.1: Exercícios e distâncias relativas.

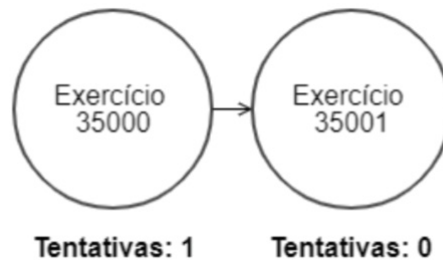


Figura 4.12: Caminho de exercícios de um aprendiz.

Para o primeiro bom aprendiz cujo caminho é:

[35004, 35017, **35000**, 35003, 35007, 35012, 35013, 35011, ...]

Deseja-se calcular a distância relativa dos exercícios proximais em relação ao pivô, exercício 35000. Supondo a distância d como 5, selecionam-se os 5 exercícios anteriores e posteriores em relação ao 35000, que serão os exercícios proximais. Os exercícios selecionados são esquematizados pela Figura 4.13.

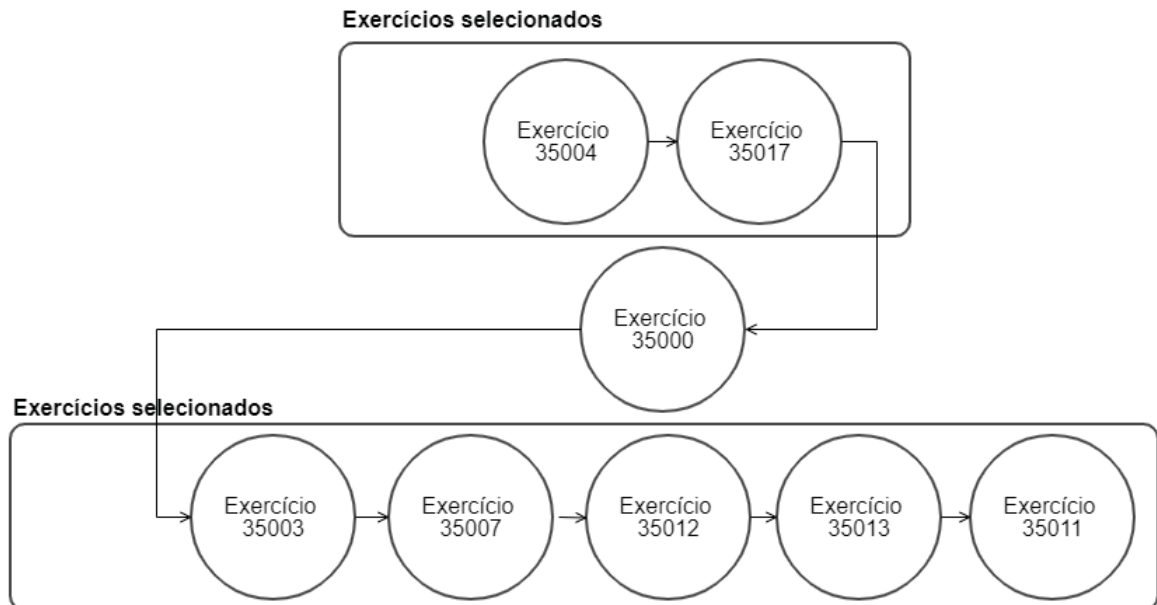


Figura 4.13: Exercícios selecionados - bom aprendiz 1.

A distância relativa dos exercícios considerando o exercício 35000 é ilustrada pela Tabela 4.2.

Exercício	Soma das Distâncias	Frequência	Distância
35004	2	1	2
35017	1	1	1
35003	1	1	1
35007	2	1	2
35012	3	1	3
35013	4	1	4
35011	5	1	5

Tabela 4.2: Exercícios e distâncias relativas - iteração 1.

Para o segundo bom aprendiz cujo caminho é:

[35017, 35004, **35000**, 35002, 35003, 35007, 35012, 35013, ...]

Os exercícios selecionados é esquematizado pela Figura 4.14.

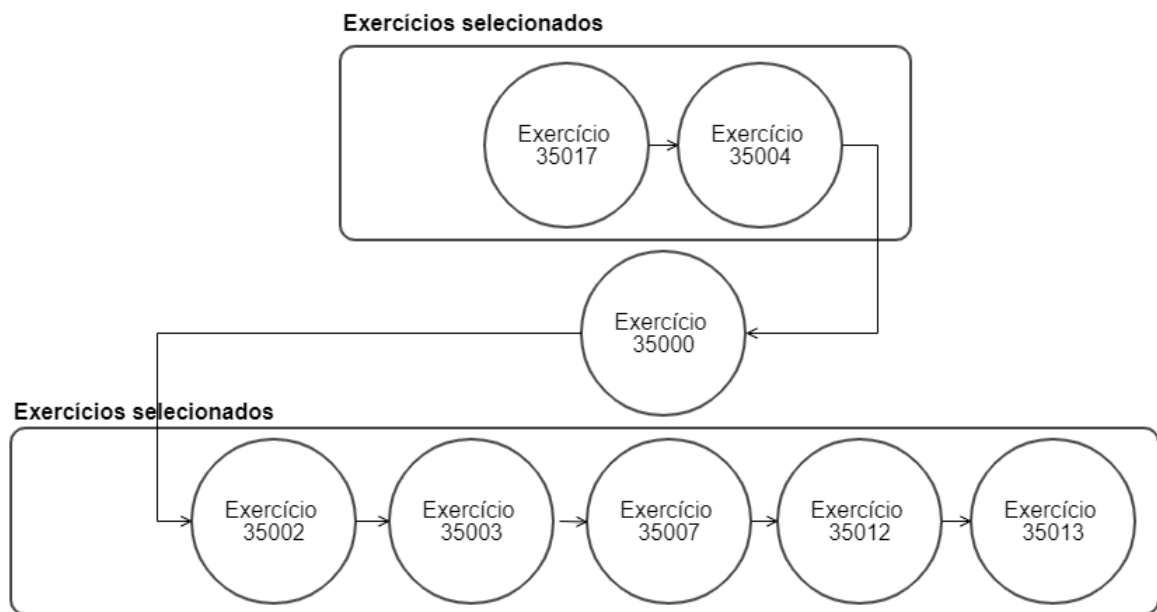


Figura 4.14: Exercícios selecionados - bom aprendiz 2.

A distância relativa dos exercícios considerando 2 bons aprendizes ficaria representada pela Tabela 4.3.

Dessa forma é calculada a distância relativa para o conjunto de n de bons aprendizes. No exemplo demonstrado, a quantidade de aprendizes n é 2 bons aprendizes. Em seguida, cria-se a tabela que é preenchida com a soma das distâncias, a frequência de cada exercício em um caminho e a distância relativa à frequência. Logo para este caso, as 3 menores distâncias são as dos exercícios 35004, 35017 e 35003 que seriam os exercícios recomendados ao aluno.

4.4 SELEÇÃO DOS CÓDIGOS FONTE DOS EXERCÍCIOS

Levando em consideração a recomendação de exercícios para aluno travado e ótimo, as estratégias de recomendação com base na quantidade média de evidências de aprendizado por

Exercício	Soma das Distâncias	Frequência	Distância
35004	3	2	1.5
35017	3	2	1.5
35003	3	2	1.5
35007	5	2	2.5
35012	7	2	3.5
35013	9	2	4.5
35011	5	1	5
35002	1	1	1

Tabela 4.3: Exercícios e distâncias relativas - iteração 2.

exercício. Para uma determinada perícia dentro do domínio da programação, são elaboradas listas de exercícios que exercitam estas perícias. Para aplicar as estratégias de recomendação para uma determinada perícia, selecionam-se os exercícios dentro das listas de exercícios no FARMA-ALG que trabalham com essas perícias especificamente. Considera-se a quantidade e de exercícios em uma lista para n aprendizes.

Assumindo como exemplo uma lista de exercícios de aritmética que reúna 13 exercícios e 10 aprendizes. A seleção dos códigos fonte dos exercícios seleciona os 13 exercícios de cada um dos 10 aprendizes que são utilizados na etapa a seguir.

4.5 IDENTIFICAÇÃO AUTOMÁTICA DE EVIDÊNCIAS DE PROGRAMAÇÃO

A identificação automática de evidências foi implementada utilizando o Yacc e o Flex (Levine et al., 1992) em um subconjunto da linguagem Pascal que engloba os artefatos relacionados à detecção das evidências de aprendizado de determinadas habilidades de programação. A parte implementada utilizando o Flex é responsável pela identificação da parte léxica da linguagem e também da contagem da parte léxica que não demanda análise sintática. Para contar a quantidade de variáveis e constantes, utiliza-se a análise sintática que através das regras de produção de um subconjunto da linguagem Pascal conta a quantidade referente ao tipo utilizado.

As evidências que podem ser extraídas, descritas por Porfirio (2020), consideradas nas identificação automática.

- Saídas

```
1 write(); writeln();
```

- Entradas

```
1 read(); readln();
```

- Atribuições

```
1 :=
```

- Tipos de literais

```
1 integer, char, real, boolean, string, longint
```

- Variáveis

1 **var**

- Constantes

1 **const**

- Expressões aritméticas

1 **+, -, /, *, div, mod**

- Expressões relacionais

1 **=, <>, >, <, >=, <=**

- Expressões booleanas

1 **and, or, not**

- Condicionais de seleção múltipla

1 **case of**

- Condicionais simples e compostos

1 **if then**

- Laços condicionais

1 **while repeat**

- Laços contados

1 **for**

As evidências de programação são extraídas e salvas em um relatório da análise da identificação automática do código fonte. A identificação automática em conjunto do relatório gerado é esquematizada pela Figura 4.15.

A estratégia de identificação automática de evidências de programação fornece o valor com a quantidade média de evidências por exercício. Para cada exercício há um código fonte de um aprendiz associado. Suponha o código fonte a seguir de dois aprendizes para um determinado problema envolvendo aritmética representados por Listagem 4.1 e 4.2. Percebe-se que ambos resolvem o problema de forma correta, entretanto no Listagem 4.1 o aprendiz utiliza 3 evidências de aprendizado de aritmética, denotadas pelo * que indica multiplicação, enquanto que no Listagem 4.2 o aprendiz utiliza 2 evidências de aritmética para resolver o problema.

Listagem 4.1: Exemplo de código fonte. EX1

```

1 program xx (input,output);
2   var a,b : longint;
3
4 begin
5   read(a,b);
6   write((a*b),' ', (a*b)*18);
7 end.
```

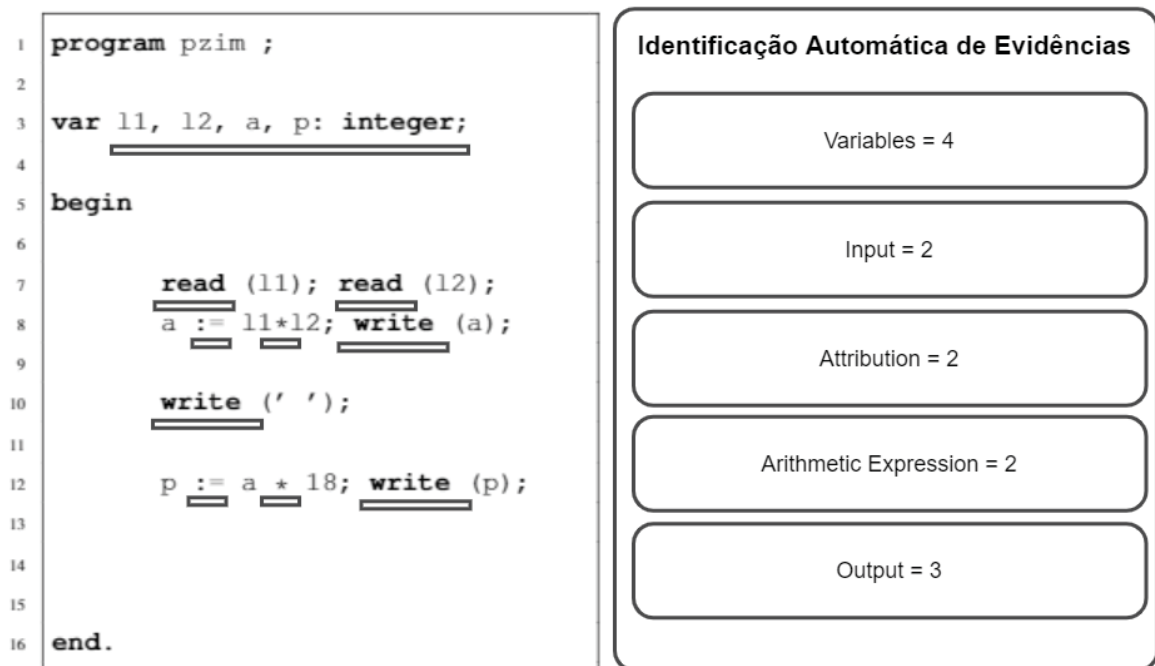


Figura 4.15: Exemplo de identificação automática de evidências de um código fonte.

Listagem 4.2: Exemplo de código fonte. EX1

```

1 program pzim ;
2
3 var l1, l2, a, p: integer;
4
5 begin
6
7     read (l1); read (l2);
8     a := l1*l2; write (a);
9
10    write ( ' ');
11
12    p := a * 18; write (p);
13
14
15
16 end.

```

Para um outro exercício cujos códigos fonte são representados pelo Listagem 4.3 e 4.4. Percebe-se que o Listagem 4.3 utiliza 8 evidências de aritmética enquanto que o Listagem 4.4 utiliza 6.

Listagem 4.3: Exemplo de código fonte. EX2

```

1 program volume_esfera;
2 var diam, volume : real;
3
4 begin
5     read(diam);
6     volume := 4*(3.14)/3 * (diam/2)*(diam/2)*(diam/2);
7
8     writeln(volume:0:2);
9 end.

```

Listagem 4.4: Exemplo de código fonte. EX2

```

1 program ex;
2 var
3   d : real;
4
5 begin
6   read(d);
7   writeln(4 * 3.14 / 3 * d * d * d / 8:0:2);
8 end.

```

No exemplo de um exercício trabalhando condicionais, ilustrado pelo Listagem 4.5 e 4.6. Percebe-se que o Listagem 4.5 utiliza 3 evidências de condicionais enquanto que o Listagem 4.6 utiliza 2.

Listagem 4.5: Exemplo de código fonte. EX3

```

1 program aprovacao;
2 var n1, n2, n3, faltas, media : real;
3
4 begin
5   read(n1, n2, n3, faltas);
6
7   media := (n1 + n2 + n3)/3;
8
9   if faltas >= 10 then
10    writeln('NAO')
11  else if media < 4 then
12    writeln('NAO')
13  else if media < 7 then
14    writeln('TALVEZ')
15  else
16    writeln('SIM');
17
18
19 end.

```

Listagem 4.6: Exemplo de código fonte. EX3

```

1 program farma_alg;
2 var
3   a,b,c,d,media: real;
4
5 begin
6   read(a,b,c,d);
7   media := (a+b+c)/3;
8
9   if ((d>=10) or (media<4)) then
10    writeln ('NAO')
11  else if (media<7) then
12    writeln('TALVEZ')
13  else
14    begin
15      writeln('SIM');
16    end;
17
18 end.

```


Exercício	Quantidade Média de Evidências (E)
35000	3
35001	5.36
35002	5.1
35003	2.72
35004	6.25
35006	2.27
35007	4
35008	3.75
35011	5.5
35012	2
35013	1
35017	1.13
35018	4.1

Tabela 4.4: Exercícios e quantidades médias de evidências.

4.6 CÁLCULO DA MÉDIA DE EVIDÊNCIAS DE PROGRAMAÇÃO POR EXERCÍCIO

Para calcular a média de evidências de programação por exercício, calcula-se a soma de todas as evidências em relação ao número de aprendizes.

$$M = \frac{E_1 + .. + E_n}{n}$$

4.7 GERAÇÃO DA TABELA COM A QUANTIDADE MÉDIA DE EVIDÊNCIAS

Após a seleção dos códigos fonte dos exercícios de determinada perícia e posterior identificação automática de evidências, calcula-se a média das evidências por exercício. Finalmente, com as médias dos exercícios gera-se uma tabela com as quantidades médias de evidências de aprendizado utilizadas pelos aprendizes nos exercícios, Tabela 4.4. Essa tabela é utilizada na recomendação de questões para aluno travado e ótimo.

4.8 HABILIDADES

Para organizar os exercícios, trabalha-se com listas de exercícios que exercitam determinadas habilidades. Cada aprendiz possui um conjunto de habilidades a ser desenvolvido e este é atualizado conforme a realização dos exercícios de uma lista. Dependendo do caminho seguido pelo aprendiz em conjunto dos acertos e erros, o seu modelo do estudante é atualizado para refletir o atual estado de conhecimento.

Os exercícios possuem um peso calculado utilizado para atualizar as habilidades no modelo do estudante. Considera-se que um aprendiz entendeu determinada habilidade ao realizar quantidade suficiente de exercícios que comprovem, através da atualização do modelo do estudante com base nos pesos dos exercícios, que este aluno conseguiu resolver exercícios relacionados ao assunto e demonstrou compreensão.

Porcentagem	Somatório das Evidências
100%	46.18
70%	32.326

Tabela 4.5: Porcentagem cumprida e somatório das evidências.

4.8.1 Construção do Modelo do Aprendiz

O modelo do aprendiz é construído utilizando o grafo genético (Oliveira, 2016) com base nas habilidades propostas por Krynski (2013). Durante a realização de listas de exercícios que trabalham as habilidades, o aprendiz resolve exercícios que possuem um peso associado às habilidades trabalhadas que é atualizado no modelo para refletir o estado atual de conhecimento do aprendiz.

Inicialmente, o aprendiz começa com as suas componentes de conhecimento no valor 0%. Isto significa que o aluno ainda não começou a desenvolver as habilidades nos exercícios. Para prosseguir com o preenchimento das componentes, os exercícios devem ser realizados e completados, para que, dessa forma, o modelo seja atualizado de acordo com o peso das habilidades trabalhadas associadas aos exercícios.

A maestria de uma habilidade é representada por um valor de até 100%. Este número é uma representação do conhecimento do aluno acerca das habilidades dentro do domínio da programação. Este número é atualizado conforme a resolução de exercícios.

O conhecimento do aluno é considerado suficiente se for superior a 70% (arbitrário) da soma da quantidade média de evidências de aprendizado de todos os exercícios de uma lista.

Para o conjunto exemplo, a somatória das evidências é representada pela Tabela 4.5. Caso o aprendiz cumpra com uma quantidade de exercícios que some um valor igual ou maior a que 70% do conjunto ele é considerado com conhecimento suficiente na perícia em evidência.

4.8.2 Atualização do Modelo do Aprendiz

Conceituando a atualização do modelo do aprendiz, o modelo é atualizado a cada momento em que o aprendiz resolve um exercício. Isso ocorre para refletir o estado atual de conhecimento do aluno. Analisam-se as habilidades de programação trabalhadas no exercício e o seu peso que é a quantidade média de evidências do exercício. Para ilustrar a atualização do modelo, mostra-se a seguir uma ilustração de um subgrafo do grafo genético sendo atualizado para um aluno novo que está no estado inicial resolvendo a sua primeira questão de "Hello World" cuja evidência de aprendizado hipotética seja de $E=3$ na perícia de Saída, ilustrado pela Figura 4.16. Dessa maneira, o novo estado do aprendiz passa a ter o valor de Saída em 3% e a Entrada x Saída que é representada pela média aritmética da Entrada e Saída passa a ter o valor de 1.5%.

A atualização demonstrada no contexto dos exercícios calculados pela Tabela contendo as evidências de aprendizado anterior, para um aprendiz que inicia com as suas componentes no valor de 0%, ilustrado pela Figura 4.17.

Com a resolução dos seguintes exercícios

[35000, 35001, 35002, 35004, 35007, 35011, 35018]

o aluno obterá no seu modelo a somatória das médias das evidências dos exercícios

$$E(35000) + E(35001) + E(35002) + E(35004) + E(35007) + E(35011) + E(35018)$$

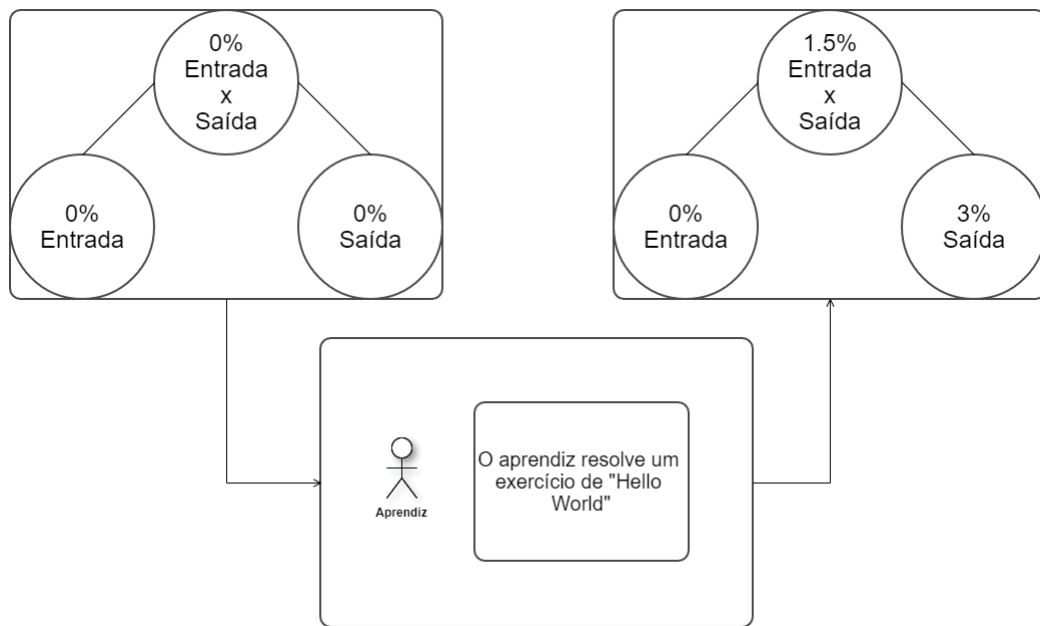


Figura 4.16: Ilustração da atualização do modelo do aprendiz.

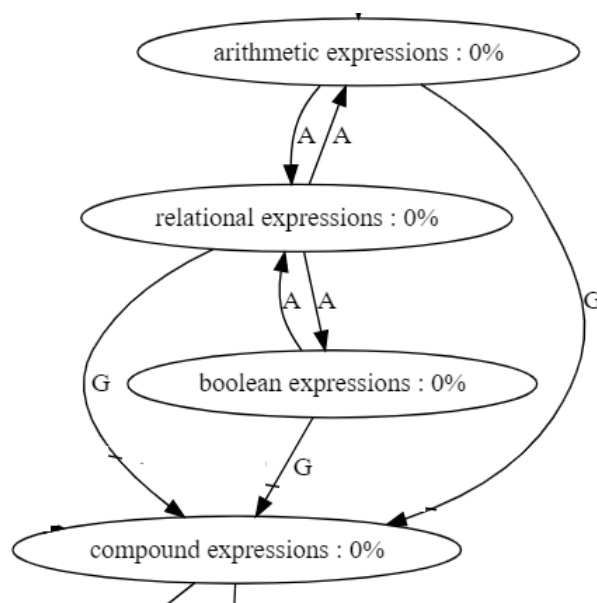


Figura 4.17: Ilustração da atualização do modelo do aprendiz.

$$3 + 5.36 + 5.1 + 6.25 + 4 + 5.5 + 4.1 = 33.31$$

O valor obtido equivale a 72.13% em relação ao número total das evidências. Com isso, o aluno estaria com conhecimento suficiente (maior ou igual a 70%) em expressões aritméticas e o modelo seria atualizado de acordo com a Figura 4.18.

4.9 RECOMENDAÇÃO DAS QUESTÕES

Nesta seção são apresentados exemplos de cada cenário, descritos previamente, e como os exercícios recomendados são computados, um exemplo da recomendação feita para o aluno regular é ilustrado pela Figura 4.19.

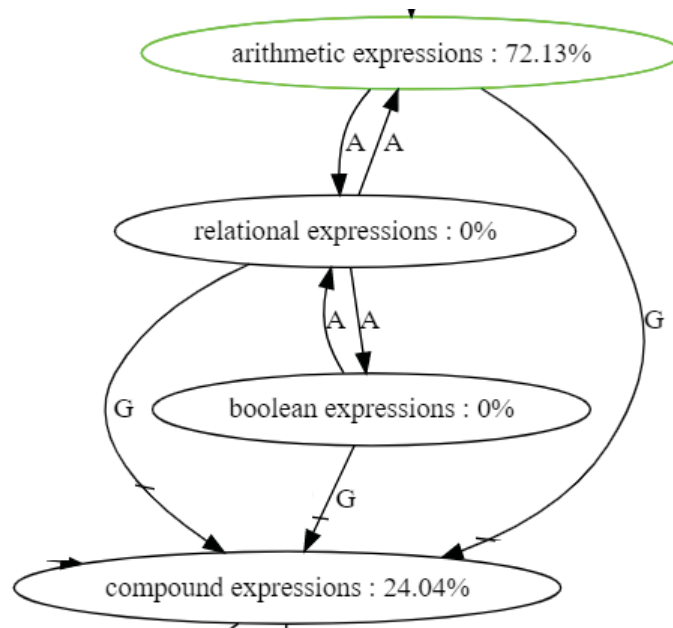


Figura 4.18: Ilustração da atualização do modelo do aprendiz.

```

IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\test\regularStudentRecommend.py =====
RECOMMENDED EXS
35004
35003
35017
>>>

```

Figura 4.19: Sugestão de exercícios.

4.9.1 Recomendação para Aluno Regular

Exemplificando um aluno regular que solicita a recomendação. Tendo em vista a Tabela 4.6, demonstrada anteriormente, a recomendação de exercícios para alunos normais utilizando o caminho do bom aprendiz se dá pela seleção dos exercícios que aparecem com maior frequência Q seguido da menor distância relativa R calculada. A distância relativa não afirma que este seja um exercício ótimo a ser realizado, entretanto significa que uma grande quantidade de bons aprendizes resolveram este exercício na sequência do exercício fornecido como parâmetro. Isto significa que os exercícios que apresentam as maiores frequências e menores distâncias relativas são exercícios interessantes que são recomendados ao aprendiz. No trabalho em questão, estipula-se a quantidade de 3 exercícios recomendados em que o aluno pode escolher.

No caso da recomendação para aluno regular em que o aluno resolveu o exercício 35000 e o próximo exercício da lista seria o 35001, como representado pela Figura 4.20, a recomendação alternativa seria dos exercícios 35004, 35017 e 35003 por apresentarem a maior frequência e menor distância relativas de acordo com a Tabela, estes exercícios recomendados são representados pela Figura 4.21.



Figura 4.20: Caminho de exercícios do aprendiz.

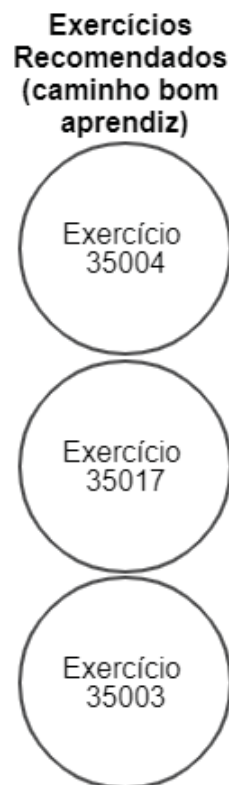


Figura 4.21: Exercícios recomendados para o aluno regular.

4.9.2 Recomendação para Aluno Travado

Para exemplificar um aluno travado e considerando os passos anteriores necessários para a geração da tabela contendo a quantidade média de evidências de aprendizado por exercícios, o módulo de recomendação das questões para o aluno travado verifica a condição do aluno estar travado durante a realização dos exercícios. Caso seja detectada a condição do aluno travado, o sistema verifica o valor da quantidade média de evidências do exercício em que o aluno está travado e recomenda um exercício cuja média seja menor ou igual ao exercício. O exemplo representado pela Figura 4.22, que mostra um aluno que resolveu o exercício 35013 e travou no próximo exercício 35003, tendo tentado submeter 7 tentativas para correção. O sistema avalia a quantidade média de evidências do exercício 35003 em que o aluno travou e com uma quantidade média de evidências de 2,72, recomenda os exercícios com uma média menor ou igual a 2,72. Os exercícios recomendados dentro do conjunto de exercícios são os 35006 e 35017 cujas médias são respectivamente 2,27 e 1,13 de acordo com a ilustração da Figura 4.23.

Exercício	Soma das Distâncias	Frequência	Distância
35004	13	10	1.3
35017	19	9	2.11
35003	17	9	1.88
35007	23	9	2.55
35012	29	9	3.22
35013	36	9	4
35011	13	3	4.33
35002	16	7	2.28
35006	12	3	4
35008	9	2	4.5
35001	1	1	1
35018	2	1	2

Tabela 4.6: Exercícios e distâncias relativas.

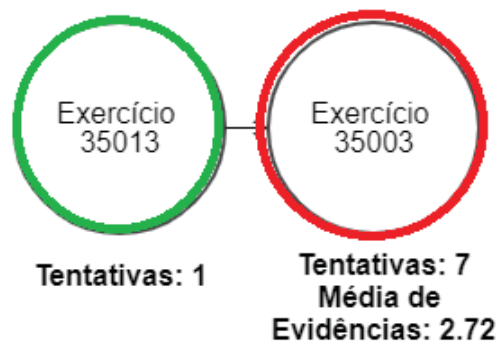


Figura 4.22: Caminho de exercícios do aprendiz.

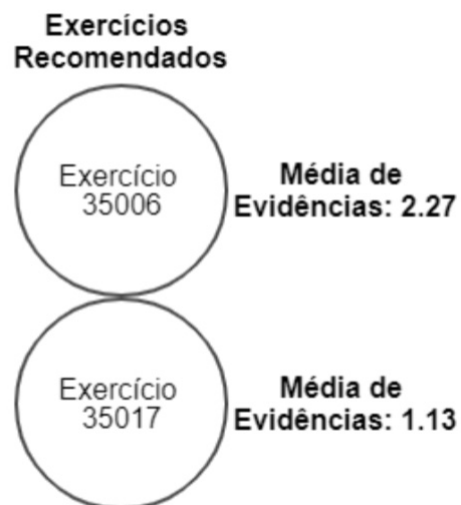


Figura 4.23: Exercícios recomendados para o aluno travado.

4.9.3 Recomendação para Aluno Ótimo

Considerando que o sistema verifique que o aluno está tendo um bom desempenho durante a progressão pelos exercícios, verifica-se a quantidade média de evidências dos exercícios resolvidos pelo aluno. Recomendam-se exercícios cuja quantidade de média seja maior em relação aos exercícios resolvidos. Dessa forma, exercícios considerados mais difíceis são recomendados para que o aluno não perca o engajamento ao resolver os exercícios. Um exemplo é demonstrado pela Figura 4.24 que mostra um aluno que resolveu os exercícios 35013, 35012, 35007 e 35001. Detectada a condição pelo sistema, recomendam-se exercícios desafio para o aluno com uma média de evidências maior em relação aos exercícios resolvidos de acordo com os exercícios da Figura 4.24. A média de evidências dos exercícios 35013, 35012, 35007 e 35001 são respectivamente 1, 2, 4 e 5,36. Para esses exercícios, busca-se recomendar exercícios cuja quantidade seja maior ou igual a 5,36. Os exercícios recomendados pelo sistema dentro do conjunto de exercícios são os 35004 e 35011 cujas médias são respectivamente 6,25 e 5,5 como representado pela Figura 4.25.

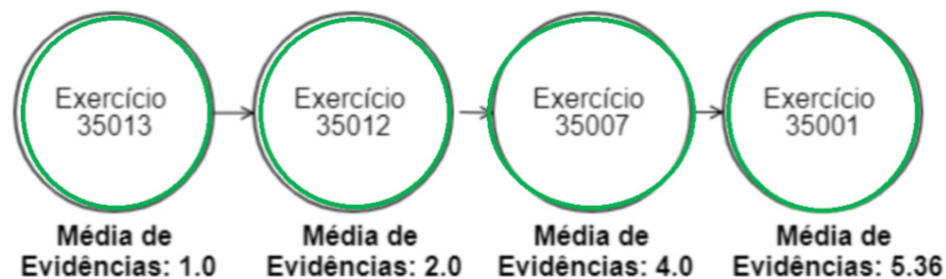


Figura 4.24: Caminho de exercícios do aprendiz.



Figura 4.25: Exercícios recomendados para o aluno ótimo.

5 VALIDAÇÃO

Este capítulo tem como objetivo descrever a validação utilizada para investigar as capacidades da abordagem de recomendação de questões. Dessa forma, permite-se suportar a argumentação sobre as contribuições deste trabalho. A seguir é apresentada a validação empírica conduzida neste trabalho que foi realizada com professores de ensino de disciplinas introdutórias de programação.

5.1 METODOLOGIA

O objetivo do experimento é validar a abordagem proposta de recomendação de questões com o propósito de avaliar se as questões recomendadas são adequadas para os cenários evidenciados no trabalho e se podem contribuir para aumentar o engajamento no ponto de vista dos alunos no contexto do FARMA-ALG.

O experimento é composto por reuniões com professores de ensino de programação que avaliam a abordagem de recomendação de questões para os 3 diferentes cenários apresentados no trabalho. Os participantes devem responder às perguntas objetivas indicando o grau de concordância utilizando a escala de Likert e também há um campo em que os professores podem fazer observações referentes aos exercícios que são sugeridos aos alunos e escrever comentários.

As reuniões foram feitas através de videoconferências e levaram em torno de 15 a 20 minutos e foram realizadas com professores que aceitavam participar do experimento.

5.1.1 Participantes

Os participantes selecionados para o experimento são docentes do departamento de Informática da UFPR, que ministram disciplinas introdutórias de programação de computadores. Esses participantes são selecionados pelo fato de possuírem experiência no ensino da programação e podem opinar sobre a abordagem de recomendação de questões descrita neste trabalho.

5.1.2 Método de Pesquisa

A pesquisa é feita através da reunião com os professores de ensino de programação em que é apresentado ao especialista sobre a abordagem de recomendação de questões no contexto do FARMA-ALG em conjunto dos cenários descritos no trabalho. É mostrado ao professor um exemplo de aluno de cada um dos cenários e os exercícios que ele já fez. Após analisar cada cenário, o professor deve responder as questões pedidas no questionário ¹.

Os exercícios utilizados no experimento foram retirados de uma lista de exercícios do FARMA-ALG relacionada às operações aritméticas. Os exercícios desse conjunto constam no Apêndice B. Utilizou-se um conjunto de bons aprendizes do FARMA-ALG que constam no Apêndice A. Para calcular a média de evidências de aprendizado dos exercícios utilizou-se os códigos fonte dos exercícios da lista mostrados no Apêndice A no conjunto de aprendizes do Apêndice B.

¹O questionário está no Apêndice D.

5.1.3 Cenários

O cenário do aluno regular, descrito na subseção 4.9.1 da Abordagem, consiste de um aluno com um andamento normal pela lista de exercícios. Esse andamento é caracterizado pela resolução dos exercícios sem que o aprendiz esteja acertando todos os exercícios em uma tentativa e sem que o aprendiz esteja tendo que realizar muitas tentativas nos exercício. Os pontos avaliados deste cenário são:

- Os exercícios recomendados para aluno regular são adequados;
- A recomendação de exercícios alternativos pode aumentar o engajamento do aluno regular.

O cenário do aluno travado, caracterizado na subseção 4.9.2 da Abordagem, é evidenciado por um aluno que trava quando ele tenta resolver um exercício por diversas tentativas sem obter êxito. Para o cenário do aluno travado consideram-se o seguintes pontos:

- Os exercícios recomendados para aluno travado são adequados;
- A recomendação de exercícios pode ser útil para ajudar o aluno travado quanto ao engajamento.

O cenário do aluno ótimo, explicado na subseção 4.9.3 da Abordagem, é representado por um aprendiz que está indo bem pela lista de exercícios e resolve os exercícios da lista em uma quantidade ínfima de tentativas. No cenário do aluno ótimo evidenciam-se os seguintes pontos:

- Os exercícios recomendados para aluno ótimo são adequados;
- A recomendação de exercícios para aluno ótimo pode contribuir para aumentar o engajamento do aluno ótimo.

5.1.4 Instrumento de Pesquisa

Os instrumentos utilizados para a condução da validação são:

- Apresentação² contendo os cenários de alunos e os enunciados exercícios recomendados;
- Questionário utilizado para avaliar a recomendação de questões.

5.2 RESULTADOS ESPERADOS

Espera-se obter resultados que comprovem a contribuição positiva da recomendação de questões para os alunos. Diversos trabalhos encontrados no estado da arte categorizam recomendações personalizadas de exercícios para alunos como um fator positivo durante a aprendizagem e que contribuem para o engajamento.

Dessa forma, os resultados obtidos pela validação utilizando a recomendação de questões ligada ao FARMA-ALG serão contrapostos com o estado da arte. Busca-se evidenciar que a recomendação de questões considerando o estado do aluno contribui para o engajamento destes durante a realização de exercícios. Busca-se contrastar a forma estática em que os alunos são

²Os slides da apresentação estão no Apêndice C.

submetidos a resolução das listas de exercícios com a forma dinâmica que considera o que o aluno já fez para recomendar as questões.

Para os cenários especificados, busca-se reunir o grau de concordância dos professores, obtido por meio do questionário, acerca dos exercícios recomendados e se os exercícios contribuem para o engajamento.

5.3 DESCRIÇÃO DOS EXPERIMENTOS

Os experimentos foram conduzidos através do envio de emails para os professores contando sobre a proposta de validação deste trabalho de recomendação de questões envolvendo o FARMA-ALG e verificando quando e se os participantes teriam disponibilidade para uma reunião. Após a confirmação de uma data e horário para a reunião, apresentava-se a abordagem de recomendação ligada ao FARMA-ALG para os professores em conjunto dos cenários descritos no trabalho na seção 4.9 da Abordagem. Após a apresentação de cada cenário, o professor deveria preencher o questionário e responder as perguntas referentes ao cenário. Por fim, o participante respondia as perguntas gerais e podia escrever e registrar comentários sobre a abordagem de recomendação de questões.

5.4 RESULTADOS

Observando o tempo médio de ensino de programação de computadores dos participantes da validação, de acordo com a Figura 5.1, o valor médio obtido é de 5 anos considerando o tempo. A ordenada indica há quanto tempo os professores dão aulas de programação. A abscissa é um identificador do participante.

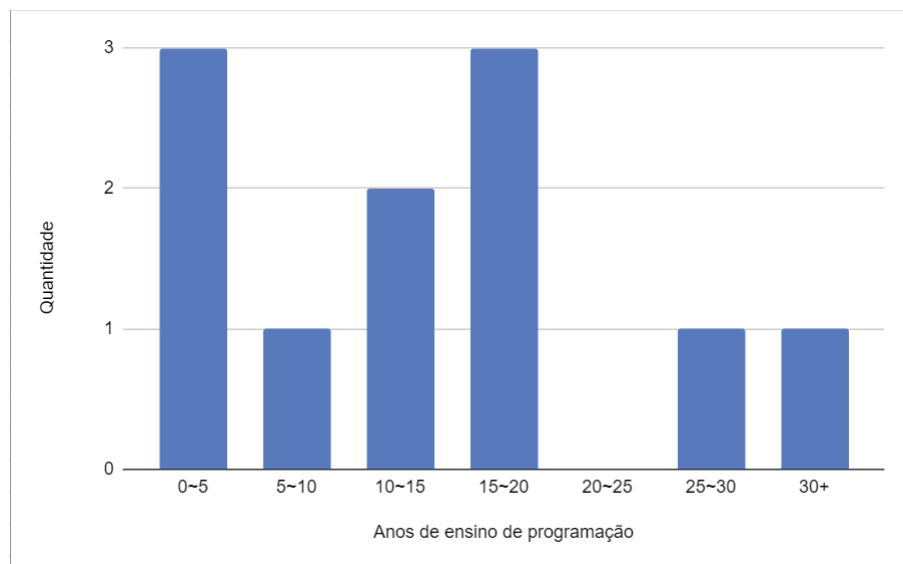


Figura 5.1: Quantidade de anos de ensino de programação de computadores dos participantes.

Para medir o grau de concordância dos participantes foi utilizada a legenda, ilustrada pela Figura 5.2, que segue a escala de Likert em um nível de 1 até 5 para representar a concordância.

A Figura 5.3 ilustra a recomendação utilizando o caminho do bom aprendiz em que os participantes avaliaram se os exercícios selecionados para um aluno regular eram adequados. Há consenso sobre os exercícios recomendados como adequados: 72,7% concorda totalmente e 27,3% concorda parcialmente.

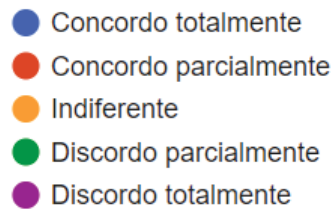


Figura 5.2: Concordância.

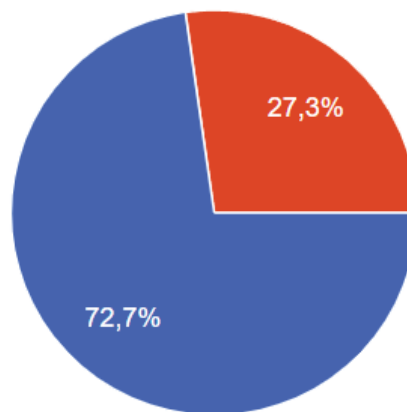


Figura 5.3: Resultados sobre os exercícios recomendados para aluno regular são adequados.

A Figura 5.5 relacionada ao caminho do bom aprendiz, os participantes avaliaram se os exercícios selecionados para um aluno regular poderiam aumentar o engajamento dos alunos. Há concordância por 72,8% dos participantes, 9,1% é indiferente e 18,2% discorda parcialmente.

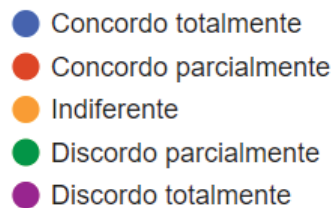


Figura 5.4: Concordância.

A Figura 5.7 ilustra a recomendação utilizando evidências de aprendizado em que os participantes avaliaram se os exercícios selecionados para um aluno travado eram adequados. 81,9% dos participantes acha os exercícios adequados enquanto que 18,2% discorda parcialmente sobre os exercícios serem adequados.

A Figura 5.9, que demonstra a recomendação utilizando evidências de aprendizado, os participantes avaliaram se os exercícios similares podem ser úteis para ajudar um aluno que não conseguiu resolver um exercício. Há consenso sobre exercícios similares serem úteis para ajudar alunos que não conseguem resolver exercícios. 63,6% concorda totalmente enquanto que 36,4% concorda parcialmente.

A Figura 5.11 ilustra a recomendação utilizando evidências de aprendizado em que os participantes avaliaram se os exercícios selecionados para um aluno ótimo eram adequados.

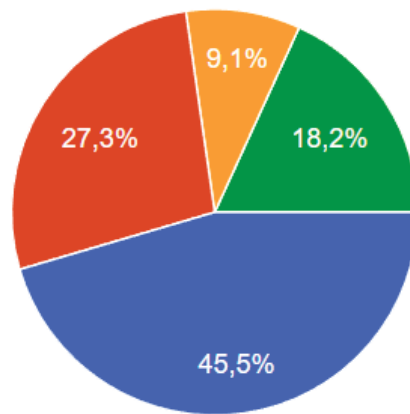


Figura 5.5: Resultados sobre o engajamento para aluno regular.

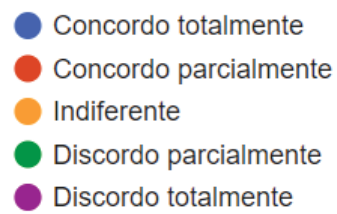


Figura 5.6: Concordância.

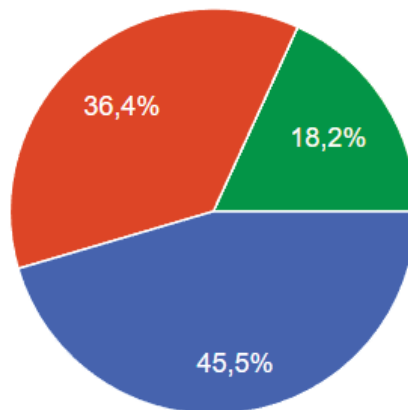


Figura 5.7: Resultados sobre os exercícios recomendados para aluno travado são adequados.

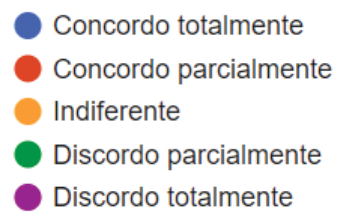


Figura 5.8: Concordância.

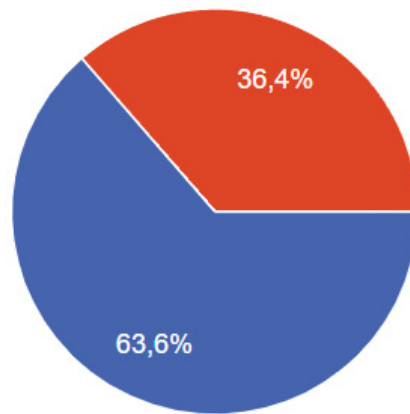


Figura 5.9: Resultados sobre se os exercícios similares são úteis para ajudar alunos travados.

Tratando do aluno ótimo, 72,7% concordam sobre os exercícios serem adequados enquanto que 9,1% concorda parcialmente, 9,1% é indiferente e 9,1% discorda parcialmente.

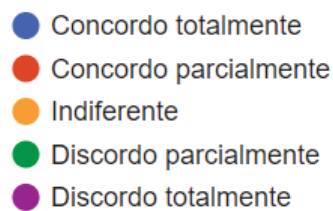


Figura 5.10: Concordância.

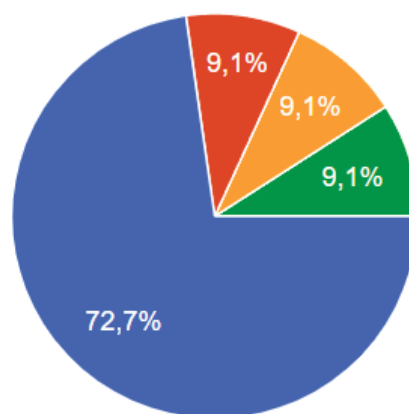


Figura 5.11: Resultados sobre os exercícios recomendados para aluno ótimo são adequados.

A Figura 5.13, que demonstra a recomendação utilizando evidências de aprendizado, os participantes avaliaram se os exercícios que trabalham com uma quantidade maior de evidências de aprendizado podem contribuir para aumentar o engajamento do aluno ótimo. Há concordância sobre os exercícios recomendados para alunos que estejam indo bem podem contribuir para aumentar o engajamento: 90,9% concorda totalmente e 9,1% concorda parcialmente.

- Concordo totalmente
- Concordo parcialmente
- Indiferente
- Discordo parcialmente
- Discordo totalmente

Figura 5.12: Concordância.

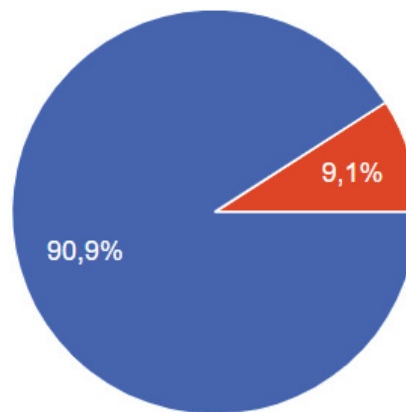


Figura 5.13: Resultados sobre o engajamento para aluno ótimo

A Figura 5.15 representa a concordância acerca do tema da recomendação de exercícios contribuir para diminuir o número de alunos que abandonam o uso da ferramenta do FARMA-ALG ou parar de resolver os exercícios. 63,6% concorda totalmente e 36,4% concorda parcialmente que abordagens de recomendação de questões contribuem positivamente para alunos continuarem a utilizar a ferramenta e resolver os exercícios.

- Concordo totalmente
- Concordo parcialmente
- Indiferente
- Discordo parcialmente
- Discordo totalmente

Figura 5.14: Concordância.

A Figura 5.17 ilustra a sugestão de exercícios em uma ordem mais adequada para um aluno contribui para aumentar o seu engajamento. 72,7% concorda totalmente enquanto que 27,3% concorda parcialmente que exercícios recomendados em uma ordem mais adequada ao aluno contribui para aumentar o engajamento.

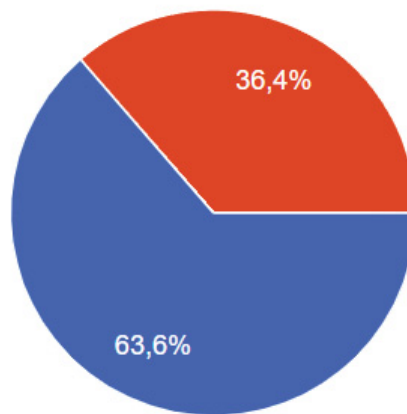


Figura 5.15: Resultados sobre a contribuição da recomendação para continuar utilizando a ferramenta e resolver os exercícios.

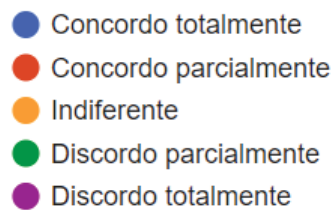


Figura 5.16: Concordância.

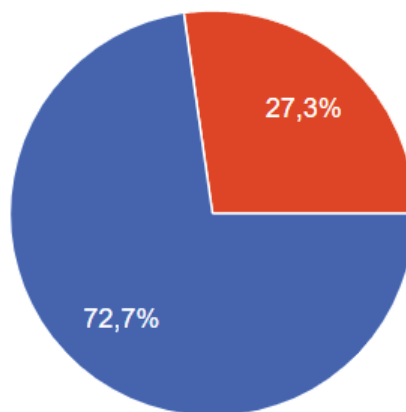


Figura 5.17: Resultados sobre exercícios recomendados em ordem mais adequada contribuem para o engajamento.

5.5 DISCUSSÃO

Nessa seção discute-se os resultados obtidos a partir do experimento conduzido para a validação. Consideram-se os cenários dos alunos citados na subseção 5.1.3 do Método de Pesquisa da Validação bem como as questões gerais em torno da abordagem.

5.5.1 Aluno Regular

Reunindo os resultados obtidos, a abordagem de recomendação se mostra efetiva para guiar os alunos durante a realização de exercícios dentro da ferramenta FARMA-ALG. Para Huo et al. (2018) o ensino personalizado aprimora a experiência de aprendizado dos alunos ao utilizar abordagens que considerem dados de atividades de aprendizado. Na abordagem ReBaEv, considerando a adequação dos exercícios recomendados em relação aos aprendizes regulares exemplificados, a totalidade dos professores concorda totalmente ou parcialmente que a recomendação é adequada e pontuam que um caminho de exercícios é interessante para os alunos. Alguns comentários:

- "O exercício 35003 parece mais fácil que os demais."
- "Os exercícios alternativos tem a mesma complexidade e podem ajudar o aluno a evoluir."
- "Eu acho que o nível de complexidade dos programas tem que ser gradual."

No trabalho de Bauman e Tuzhilin (2014), os resultados apontam para uma aprovação das recomendações feitas pelo sistema por parte dos estudantes e que as recomendações foram bastante úteis para os estudos. No trabalho citado, a recomendação do sistema se mostrou mais efetiva para o conjunto de estudantes considerados abaixo da média. Na abordagem ReBaEv, a maioria dos professores considera que os exercícios recomendados podem contribuir para o engajamento. De acordo com Dang e Ghergulescu (2018), que compara a atividade dos alunos ao seguir a recomendação de exercícios do sistema ou seguir por conta própria. Alunos que seguiam a recomendação de exercícios do sistema apresentavam uma taxa de sucesso superior (78.77%) em relação aos alunos que seguiam por conta própria (57.41%). Para a abordagem de recomendação de questões ReBaEv, uma parcela dos professores que mostrou discordância sentiu que um dos exercícios recomendados era mais fácil que as demais recomendações. Um comentário discordante quanto ao engajamento foi:

- "O aluno pode acabar trocando só para uma questão "mais fácil" ou que outro colega já tenha feito."

5.5.2 Aluno Travado

Considerando os resultados referentes ao cenário do aluno travado, a maioria dos professores considera a recomendação alternativa de exercícios adequada para um aluno travado e concorda que esses exercícios alternativos são úteis para ajudar alunos travados ao fornecer exercícios com uma quantidade de evidências parecida ou menor para que os alunos possam continuar a praticar a habilidade de programação. Nos comentários em que os professores concordaram, também pontuaram incluir outras formas que poderiam ser úteis para facilitar o entendimento do aluno. Alguns desses comentários:

- "Talvez exemplificar com outra fórmula um mesmo exercício pode facilitar para o entendimento e conseguir resolver o exercício inicial."
- "Às vezes a recomendação poderia vir com alguma dica (por exemplo, no caso do slide, uma revisão dos operadores aritméticos para o aluno ver que $**$ ou $\wedge$ dependendo da linguagem é exponenciação."

A parcela de professores que discordou parcialmente sobre a adequação dos exercícios recomendados para um aluno travado acrescentou que se o aluno estivesse com dificuldade para o exercício exemplo, provavelmente seria necessário alguma intervenção direta para ajudar o aluno e que exercícios alternativos com uma quantidade menor de evidências de aprendizado poderiam ser insuficientes para ajudar o aluno. Alguns dos comentários discordantes foram:

- "O exercício que o aluno travou (35003) era mais fácil que os exercícios sugeridos."
- "Um dos exercícios similares (35006) está bem mais complexo do que o exercício em que o aluno "travou". E não parece que ajudaria a evoluir no tipo de exercício."

Tratando da utilidade dos exercícios para ajudar os alunos travados no quesito do engajamento, todos os professores mostraram concordância ao fato de recomendar exercícios para os alunos travados é um fator positivo para engajá-los. Considerando o conceito do aluno travado e o engajamento, é possível encontrar no trabalho de Teusner et al. (2018), uma abordagem de intervenção para alunos que estivessem levando um tempo prolongado para resolver algum exercício. O autor considera o tempo que esses alunos levam para resolver um exercício e identifica como potencialmente tendo dificuldades ou travados o percentil 75 dos alunos que mais demorassem. Como intervenção, esses alunos travados são encorajados a dar uma pausa ou solicitar uma ajuda para o sistema antes que eles percam o interesse. Nesse trabalho o autor também conceitua uma intervenção direcionada para os alunos travados.

5.5.3 Aluno Ótimo

No cenário de um aluno que está indo bem, a grande maioria dos participantes concorda que os exercícios recomendados com uma maior quantidade de evidências de aprendizado são adequados para alunos que estejam tendo um bom desempenho. Alguns dos comentários concordantes foram:

- "Como já teve um exercício com raiz cúbica, poderia ter um exercício mais difícil, ao invés do 35004"
- "Os exercícios recomendados são mais desafiadores que os anteriores e motivariam o aluno a resolvê-los."

Um dos professores que mostrou discordância apontou que exercícios que mesmo que um exercício possua um número de evidências de aprendizado bastante alto, o exercício não necessariamente seria uma opção mais complexa para recomendar para o aluno.

Tratando da recomendação de exercícios mais interessantes para alunos com um bom desempenho, o trabalho de Okpo et al. (2016), Okpo et al. (2017) e Okpo et al. (2018), conceitua que aprendizes que estivessem indo bem poderiam receber recomendações de exercícios que fossem mais difíceis. Na abordagem ReBaEv, adota-se a consideração de que alunos ótimos poderiam resolver aqueles exercícios que apresentam uma maior quantidade de evidências para que, dessa forma, pudessem praticar bastante a determinada perícia de programação. No quesito dos exercícios contribuírem para o engajamento do aluno, todos os professores concordaram que exercícios mais interessantes para um aluno que estivesse indo bem são positivos para engajar o aluno. Um comentário sobre o engajamento:

- "Concordo que os desafios contribuem, mas só se você deixar claro que se trata de um desafio para o aluno. Caso contrário, ele pode pensar que se trata de um exercício convencional, e pensar que está com dificuldades."

5.5.4 Geral

A abordagem avalia a situação do aluno no instante da recomendação, entretanto não se considera todo o histórico do aluno. Leva-se em consideração o estado no momento da recomendação do aluno. Analisando os aspectos gerais da abordagem, todos os professores concordaram que uma abordagem de recomendação é positiva para diminuir a quantidade de alunos que abandonam o uso da ferramenta e param de resolver exercícios. Esse ponto é encontrados nos trabalhos de Bauman e Tuzhilin (2014) e Teusner et al. (2018). A ideia da seleção adaptativa de questões de uma forma geral pode aprimorar o processo de aprendizagem e trazer melhores resultados. Em Zhang e VanLehn (2017) encontra-se um exemplo em que a seleção adaptativa dos exercícios traz maiores ganhos de aprendizagem em comparação com a seleção de questões mal-adaptativa.

Considerando a recomendação de exercícios em uma ordem mais adequada para os alunos e a contribuição desta para engajar o aluno, todos os professores também concordaram e consideraram a situação do aluno para recomendar exercícios é um ponto positivo ao considerar o engajamento.

Comentando sobre os aspectos gerais da abordagem, seguem os comentários:

- "Acredito que a ordem dos exercícios ajuda não só no engajamento como também no aprendizado."
- "Concordo parcialmente porque alguns alunos demandam de uma atenção maior por parte do professor (ou monitor) para tirar dúvidas, e esses alunos se desengajam mesmo quando se propõe novos exercícios similares. Achei a ferramenta muito interessante e promissora, mas tem que se levar em conta que alguns desengajamentos ocorrerão não por culpa da ferramenta, e sim porque faltou uma orientação humana para o aluno."
- "Concordo parcialmente sobre a ordem, considerando o caso em que o aluno olha para o exercício e antes de tentar resolvê-lo, já tenta mudar para o próximo."

Nesse trabalho, a recomendação baseada em evidências não contempla necessariamente a dificuldade das questões, mas o quanto a questão tem a capacidade de exercitar uma determinada habilidade. Dessa forma, salienta-se a importância da análise que uma questão com uma quantidade alta de evidências não necessariamente é mais complexa que uma questão com um menor número.

5.6 AMEAÇAS À VALIDADE DOS EXPERIMENTOS

Considerando a validade do experimento, descreve-se a seguir algumas ameaças que podem afetar o processo de validação. Uma ameaça evidenciada, é o viés de grupo. Por se tratar dos participantes selecionados, docentes do DInf - Departamento de Informática da UFPR - que ensinam programação de computadores, é possível existir algum viés metodológico dentro deste grupo de participantes por lecionarem no mesmo departamento ou convergência de ideias. Adicionalmente, cita-se o viés de observação, que compreende a experiência do participante utilizada para a formulação das respostas dentro do experimento. Os participantes podem ficar sugestionados a aferir algum desfecho específico. Para contornar isso, seria necessário alguma calibração que certifique respostas corretas do participante no assunto abordado.

5.7 REBAEV E TRABALHOS CORRELATOS

Okpo et al. (2016), Okpo et al. (2017), considera em seu trabalho as características que podem ser exploradas ao selecionar exercícios para alunos e como os humanos adaptam a seleção de exercícios à personalidade e desempenho do aluno. Busca-se compreender como um Sistema Tutor Inteligente (STI) pode ajustar a dificuldade dos exercícios de acordo com essas características. Utilizaram-se participantes que desempenharam o papel do sistema, em seguida apresentaram-se histórias de alunos que mostravam traços de personalidade de consciência, autoestima e estabilidade emocional em níveis altos e baixos em conjunto de uma indicação de seu desempenho anterior por meio de um exercício de matemática. Em seguida, os participantes selecionavam exercícios que eles achavam que o aluno deveria fazer em seguida com base na personalidade do aprendiz. Finalmente, em Okpo et al. (2018), explora-se como os humanos adaptam a seleção da próxima questão, particularmente o nível de dificuldade, para a personalidade de um aluno, o esforço mental investido e o desempenho para criar um algoritmo de seleção de exercícios adaptativo. Os participantes avaliam um aluno fictício com um certo desempenho, auto-estima e esforço, e selecionam o nível de dificuldade do próximo exercício. Observou-se que autoestima, esforço mental e desempenho, impactaram no nível de dificuldade dos exercícios selecionados para os aprendizes. Usando os resultados do estudo, propõem-se um algoritmo de seleção de exercícios que contempla variações de níveis de dificuldade adaptados às características do aluno.

Na abordagem ReBaEv, não se considera a personalidade do aluno para a seleção das questões, mas investiga-se o seu desempenho que é identificado por meio dos cenários abordados para recomendar as questões. Na validação da abordagem ReBaEv, também utilizaram-se especialistas para avaliar a adequação das questões e se elas contribuem para aumentar o engajamento dos alunos para um cenário de aluno que tinha o seu histórico de questões representado por exercícios anteriormente feitos.

Alhathli et al. (2016) investiga a influência da personalidade do aluno para a seleção de materiais de aprendizagem. Descreve-se um estudo, no domínio da aprendizagem de línguas, que explora a relação entre a extroversão dos alunos e até que ponto os materiais de aprendizagem podem ser considerados agradáveis para aumentar as habilidades e a confiança. No trabalho de (Alhathli et al., 2018), descrevem-se correlações positivas entre extroversão e critérios sociais para selecionar os materiais de aprendizagem. No trabalho citado, investiga-se o impacto da abertura a experiências novas, estabilidade emocional e autoestima na seleção de atividades de aprendizagem. Alhathli et al. (2020) investiga como os humanos adaptam a seleção da próxima atividade de aprendizagem à personalidade e competência dos alunos com o objetivo de inspirar na criação de um algoritmo de seleção adaptativa de atividade de aprendizagem.

Na abordagem desta dissertação, utilizaram-se as estratégias do caminho do bom aprendiz e as evidências de aprendizado como base para criar um algoritmo de recomendação para os cenários do aluno regular, travado e ótimo, descritos no trabalho.

Finalmente, em (Meliana e Nurjanah, 2018), propõem-se uma abordagem para seguir caminhos dos bons alunos em Sistemas Tutores Inteligentes (STI). Com base em uma teoria que afirma que a aprendizagem pode ser adquirida através de imitação de modelos competentes, observa-se que exercícios concluídos por bons alunos podem ser utilizados para prever uma sequência adequada de questões para os alunos seguirem. Por meio de um estudo preliminar utilizado para obter uma visão do mundo real em um curso de programação. Objetiva-se medir a confiabilidade da abordagem proposta, em comparação com uma abordagem convencional em que os professores recomendavam as questões. No trabalho evidenciado, o desempenho da abordagem é positivo seguido por um aumento de bons alunos. De acordo com os resultados do

experimento, mostrou-se que a abordagem proposta tem uma confiabilidade melhor do que a abordagem convencional.

A abordagem ReBaEv, também utiliza como parte de sua estratégia de recomendação de questões o caminho dos bons alunos como forma de propor uma ordem alternativa mais interessante para os alunos seguirem. Não foi possível conduzir experimentos com alunos, mas a abordagem se mostrou adequada para sugerir outros exercícios em uma ordem diferente da lista de exercícios convencional e aumentar o engajamento ao considerar o estado do aluno para a recomendação e não uma ordem fixa.

5.8 CONSIDERAÇÕES

Nesse capítulo foi descrita a metodologia da validação que consistiu de reuniões feitas por meio de videoconferências em que os participantes, docentes do DInf da UFPR que ministram disciplinas introdutórias de programação de computadores, responderam a um questionário para avaliar a abordagem de recomendação de questões descrita no trabalho. Os professores levaram em consideração um cenário real de aplicação e a sua experiência em ministrar aulas de programação com o tratamento de alunos. Os participantes avaliaram os cenários do aluno regular, travado e ótimo e pontuaram o grau de concordância quanto à adequação dos exercícios recomendados e a contribuição para o engajamento dos alunos de cada cenário. Conduzido o experimento, apresentam-se os resultados obtidos por meio do questionário e em seguida realiza-se a discussão acerca dos resultados.

6 CONCLUSÃO

Este trabalho objetivou tratar dos desafios envolvendo a escolha da próxima questão, tendo em vista o modelo do estudante usado para monitorar a evolução dos estudantes no domínio da programação de computadores. Por meio de um mapeamento sistemático da literatura, foi possível extrair e identificar abordagens aplicadas de recomendação de exercícios na literatura. De acordo com a literatura, foram encontradas diversas técnicas que são empregadas para tratar da recomendação de questões. A categorização dos trabalhos foi um ponto desafiador, visto que os trabalhos investigados apresentavam muitas especificidades o que tornava difícil estabelecer um padrão de nomenclatura.

Uma das medidas encontradas que é utilizada para a recomendação de exercícios são as medidas cognitivas descritas por Pimentel e Direne (1998a) e Pimentel e Direne (1998b). Medidas e padrões que ajudem a classificar exercícios de alguma forma, são úteis na elaboração de estratégias de recomendação de exercícios. A geração de informações que ajudem na construção de um peso para um exercício é um ponto relevante que pode influenciar no modelo tutorial de sistemas tutores inteligentes.

A ideia sobre recomendação de bons aprendizes inspirou-se dos estudos de Meliana e Nurjanah (2018), e fornece a possibilidade de guiar o aprendiz através de um modelo competente do bom aprendiz no momento em que reproduz-se a outros aprendizes com o intuito de criar mais bons aprendizes. Para a criação da abordagem de recomendação do aluno travado e ótimo, inspirou-se na observação do potencial da avaliação automática de evidências em códigos fonte que é explorado na obra de Porfirio (2020).

Juntando os referenciais teóricos: seleção da próxima questão, caminho do bom aprendiz, ensino da programação, sistemas tutores inteligentes e avaliação automática de código, surgiu a ideia de juntar esses temas e incorporá-los no que se chama abordagem ReBaEv. Esta abordagem também tem como objetivo detalhar os conceitos de uma possível implementação na ferramenta FARMA-ALG de modo a inserir as características explicitadas na ferramenta. A abordagem atinge o objetivo de recomendar exercícios com base no estado do aprendiz. Os cenários evidenciados na Seção 4, mostram a recomendação baseada no caminho do bom aprendiz e a recomendação baseadas em evidências de aprendizado. Para o aluno regular, a recomendação aplicada é com base no caminho do bom aprendiz que serve para fornecer ao aluno exercícios diferentes da ordem da lista. O aluno pode solicitar ao sistema essa recomendação alternativa e o sistema fornece ao aluno uma recomendação personalizada. No cenário em que o aluno trava ou está indo bem, o sistema utiliza as evidências de aprendizado associadas aos exercícios para fornecer uma recomendação de acordo com o cenário identificado. Com os exercícios alternativos, espera-se que o aluno tenha um maior domínio ao desenvolver a atividade de programação e possa resolver exercícios que sejam mais interessantes. Não ficando restrito a uma ordem fixa, a motivação e engajamento do aluno enquanto realizando exercícios aumenta. Dessa forma, o processo de aprendizagem pode ser aprimorado.

Experimentos realizados utilizando a abordagem de recomendação ReBaEv foram feitos com docentes do Departamento de Informática da UFPR. Os docentes, que lecionam disciplinas introdutórias de programação de computadores, avaliaram a recomendação por meio de cenários de alunos que ilustravam o caminho de exercícios do aluno e a recomendação aplicada para o cenário, descrito na Seção 4.1 deste trabalho. Os resultados mostraram que a abordagem é promissora, sendo que a recomendação de questões foi avaliada positivamente quanto à adequação dos exercícios recomendados para o determinado cenário. Os participantes deveriam avaliar se

os exercícios recomendados eram adequados e se eram coerentes de acordo com o caminho de um aluno. Também era perguntado aos participantes, ao fato da recomendação de exercícios contribuir para aumentar engajamento dos alunos nos determinados cenários evidenciados. Os docentes indicaram que a recomendação de exercícios contribui positivamente para engajar os alunos nos cenários do aluno regular, travado e ótimo.

Tendo em vista os resultados obtidos, espera-se contribuir para o ensino da programação, por meio da abordagem de seleção de questões utilizando a partir dos cenários descritos servem para auxiliar e guiar os aprendizes durante a realização de listas de exercícios de programação. Os cenários buscam explicitar as características específicas dos aprendizes e fornecer uma seleção apropriada de acordo com as necessidades para o cenário. Espera-se que o caminho do bom aprendiz possa contribuir para a produção de mais bons aprendizes.

6.1 TRABALHOS FUTUROS

O modelo do estudante utilizado nesse trabalho serve como referência para acompanhar a evolução dos aprendizes. Como trabalho futuro, é possível expandir o uso do modelo para agir em conjunto com o modelo tutorial para refinar as estratégias de recomendação de questões para os cenários. Outro ponto é pensar em outras ideias para utilizar os dados de bons aprendizes e incorporá-los em um algoritmo de recomendação de questões. Neste trabalho, foi utilizada a distância relativa para calcular a distância entre os exercícios. É possível abordar algum outro método para calcular essa distância? É possível fazer uma análise das notas dos aprendizes e incrementar um peso nas distâncias.

Outra questão é a abordagem pedagógica pensando no FARMA-ALG. A partir do momento em que o aluno faz os exercícios recomendados em que ponto da lista esse aluno retorna. É possível que o sistema pode contribuir para sugerir a sequência que melhor encaixa com o estado do aprendiz em um nível macro podendo também considerar listas de exercícios.

Quando se utiliza a recomendação baseada em evidências, utiliza-se a quantidade média dos exercícios como peso para selecionar os exercícios. Entretanto, não se considera os conhecimentos prévios que o aluno já possui. É possível aprimorar a forma de seleção para considerar o histórico do aluno junto com os exercícios com seus pesos calculados.

Por fim, ressalta-se o valor da realização de experimentos envolvendo alunos para a validação da abordagem. Aplicar a abordagem em um contexto real de ensino e avaliar os resultados aprendizes reais pode fornecer dados significativos que corroboram a utilidade da abordagem quanto a recomendação adequada de exercícios e o aumento do engajamento para os aprendizes.

REFERÊNCIAS

- Ai, F., Chen, Y., Guo, Y., Zhao, Y., Wang, Z., Fu, G. e Wang, G. (2019). Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. *International Educational Data Mining Society*.
- Alhathli, M., Masthoff, J. e Beacham, N. (2020). Adapting learning activity selection to emotional stability and competence. *Frontiers in Artificial Intelligence*, 3:11.
- Alhathli, M., Masthoff, J. e Siddharthan, A. (2016). Exploring the impact of extroversion on the selection of learning materials. Em *CEUR Workshop Proceedings, 24th ACM Conference on User Modeling, Adaptation and Personalisation, UMAP 2016*, volume 1618.
- Alhathli, M. A. E., Masthoff, J. F. M. e Beacham, N. A. (2018). Impact of a learner's personality on the selection of the next learning activity. Em *Proceedings of Intelligent Mentoring Systems Workshop Associated with the 19th International Conference on Artificial Intelligence in Education, AIED 2018*.
- Aureliano, V. C. O. e Tedesco, P. C. d. A. R. (2012). Ensino-aprendizagem de programação para iniciantes: uma revisão sistemática da literatura focada no sbie e wie. Em *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 23.
- Baker, M. J. (2000). The roles of models in artificial intelligence and education research: a prospective view. *Journal of Artificial Intelligence and Education*, 11:122–143.
- Bauman, K. e Tuzhilin, A. (2014). Recommending learning materials to students by identifying their knowledge gaps. Em *RecSys Posters*.
- Bourdeau, J. e Grandbastien, M. (2010). Modeling tutoring knowledge. Em *Advances in intelligent tutoring systems*, páginas 123–143. Springer.
- Brown, J. S. e Burton, R. R. (1974). Sophie: a pragmatic use of artificial intelligence in cai. Em *Proceedings of the 1974 annual ACM conference-Volume 2*, páginas 571–579.
- Chang, T.-Y. e Chen, Y.-T. (2009). Cooperative learning in e-learning: A peer assessment of student-centered using consistent fuzzy preference. *Expert Systems with Applications*, 36(4):8342–8349.
- Chang, Y.-C., Kao, W.-Y., Chu, C.-P. e Chiu, C.-H. (2009). A learning style classification mechanism for e-learning. *Computers & Education*, 53(2):273–285.
- Chaplot, D. S., Rhim, E. e Kim, J. (2016). Personalized adaptive learning using neural networks. Em *Proceedings of the third (2016) ACM conference on learning@ scale*, páginas 165–168.
- Christudas, B. C. L., Kirubakaran, E. e Thangaiah, P. R. J. (2018). An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials. *Telematics and Informatics*, 35(3):520–533.

- Chrysafiadi, K., Troussas, C. e Virvou, M. (2018). A framework for creating automated online adaptive tests using multiple-criteria decision analysis. Em *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, páginas 226–231. IEEE.
- Crow, T., Luxton-Reilly, A. e Wuensche, B. (2018). Intelligent tutoring systems for programming education: a systematic review. Em *Proceedings of the 20th Australasian Computing Education Conference*, páginas 53–62.
- Cui, B., Li, J., Guo, T., Wang, J. e Ma, D. (2010). Code comparison system based on abstract syntax tree. Em *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, páginas 668–673. IEEE.
- Dang, X. e Ghergulescu, I. (2018). Effective learning recommendations powered by ai engine. Em *Proceeding International Workshop on Personalization Approaches in Learning Environments (PALE)*, página 6.
- de Barros Paes, R., Malaquias, R., Guimarães, M. e Almeida, H. (2013). Ferramenta para a avaliação de aprendizado de alunos em programação de computadores. Em *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 2.
- de Campos, C. P. e Ferreira, C. E. (2004). Boca: um sistema de apoio a competições de programação.
- Fang, L., Tuan, L. A., Hui, S. C. e Wu, L. (2018). Personalized question recommendation for english grammar learning. *Expert Systems*, 35(2):e12244.
- Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. Em *International conference on informatics in secondary schools-evolution and perspectives*, páginas 159–168. Springer.
- Garlet, D., Bigolin, N. M. e Silveira, S. R. (2016). Uma proposta para o ensino de programação de computadores na educação básica. *Departamento de Tecnologia da Informação, Universidade Federal de Santa Maria, RS*.
- González-González, C. S., Toledo-Delgado, P. A., Muñoz-Cruz, V. e Torres-Carrion, P. V. (2019). Serious games for rehabilitation: Gestural interaction in personalized gamified exercises through a recommender system. *Journal of biomedical informatics*, 97:103266.
- Hosseini, R. e Brusilovsky, P. (2017). A study of concept-based similarity approaches for recommending program examples. *New Review of Hypermedia and Multimedia*, 23(3):161–188.
- Hou, M. e Fidopiastis, C. (2017). A generic framework of intelligent adaptive learning systems: from learning effectiveness to training transfer. *Theoretical Issues in Ergonomics Science*, 18(2):167–183.
- Huang, Y. (2016). Deeper knowledge tracing by modeling skill application context for better personalized learning. Em *Proceedings of the 2016 Conference on user modeling adaptation and personalization*, páginas 325–328.
- Huo, Y., Wong, D. F., Ni, L. M., Chao, L. S. e Zhang, J. (2020). Knowledge modeling via contextualized representations for lstm-based personalized exercise recommendation. *Information Sciences*, 523:266–278.

- Huo, Y., Xiao, J. e Ni, L. M. (2018). Towards personalized learning through class contextual factors-based exercise recommendation. Em *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, páginas 85–92. IEEE.
- Jurenoka, S. e Jurenoks, A. (2018). A method for learning scenario selection and modification in intelligent tutoring systems. Em *BIR Workshops*, páginas 335–340.
- Kitchenham, B. e Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.
- Krynski, E. M. (2013). *Modelagem do processo de aquisição de conhecimento apoiado por ambientes inteligentes*. Tese de doutorado, Universidade Federal do Paraná, Curitiba, PR.
- Kutzke, A. R. (2015). Informática educacional e a mediação do erro na educação: um estudo teórico-crítico e uma proposta de instrumento computacional. *Doutorado em Informática. Universidade Federal do Paraná*.
- Lesgold, A. et al. (1988). Sherlock: A coached practice environment for an electronics troubleshooting job.
- Levine, J. R., Mason, J., Levine, J. R., Mason, T., Brown, D., Levine, J. R. e Levine, P. (1992). *Lex & yacc*. "O'Reilly Media, Inc."
- Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K. e Zhou, T. (2012). Recommender systems. *Physics reports*, 519(1):1–49.
- Maestro-Prieto, J. A. e Simon-Hurtado, A. (2018). The pedagogical model of sial: an adaptive and open-ended intelligent tutoring system for first order logic. Em *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, páginas 21–26.
- Mark, H. e Clancey, R. W. J. (1985). Guidon-watch: A graphic interface for viewing a knowledge-based system.
- Meliana, S. e Nurjanah, D. (2018). Adopting good-learners' paths in an intelligent tutoring system. Em *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, páginas 877–882. IEEE.
- Miliszewska, I. e Tan, G. (2007). Befriending computer programming: A proposed approach to teaching introductory programming. *Informing Science: International Journal of an Emerging Transdiscipline*, 4(1):277–289.
- Nkambou, R. (2010). Modeling the domain: An introduction to the expert module. Em *Advances in intelligent tutoring systems*, páginas 15–32. Springer.
- Nkambou, R., Bourdeau, J. e Mizoguchi, R. (2010). Introduction: what are intelligent tutoring systems, and why this book? Em *Advances in Intelligent Tutoring Systems*, páginas 1–12. Springer.
- Ogawa, R. (2017). *Estudo de uma ferramenta de mediação do erro em disciplinas de algoritmos: uma proposta de metodologia para promoção da interação*. Tese de doutorado, Universidade Federal do Paraná.

- Okpo, J., Dennis, M., Masthoff, J., Smith, K. A. e Beacham, N. (2016). Exploring requirements for an adaptive exercise selection system. Em *PALE 2016: Workshop on Personalization Approaches in Learning Environments*.
- Okpo, J., Masthoff, J., Dennis, M., Beacham, N. e Ciocarlan, A. (2017). Investigating the impact of personality and cognitive efficiency on the selection of exercises for learners. Em *Proceedings of the 25th conference on user modeling, adaptation and personalization*, páginas 140–147.
- Okpo, J. A., Masthoff, J., Dennis, M. e Beacham, N. (2018). Adapting exercise selection to performance, effort and self-esteem. *New Review of Hypermedia and Multimedia*, 24(3):193–227.
- Oliveira, C. M. (2016). Sequenciamento inteligente e adaptativo de enunciados em programação de computadores. Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba, PR.
- Oliveira, C. M., Pimentel, A. e Krynski, E. M. (2015). Estudo sobre o sequenciamento inteligente e adaptativo de enunciados em programação de computadores. Em *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 4, página 1320.
- Oliveira, C. M., Pimentel, A. e Maschio, E. (2016). Nextstep: Um protótipo para o sequenciamento inteligente e adaptativo de enunciados em programação de computadores. Em *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 5, página 1238.
- Papoušek, J., Stanislav, V. e Pelánek, R. (2016). Impact of question difficulty on engagement and learning. Em *International Conference on Intelligent Tutoring Systems*, páginas 267–272. Springer.
- Pazzani, M. J. e Billsus, D. (2007). Content-based recommendation systems. Em *The adaptive web*, páginas 325–341. Springer.
- Pelánek, R. (2015). Modeling students' memory for application in adaptive educational systems. *International Educational Data Mining Society*.
- Pelánek, R., Papoušek, J., Řihák, J., Stanislav, V. e Nižnan, J. (2017). Elo-based learner modeling for the adaptive practice of facts. *User Modeling and User-Adapted Interaction*, 27(1):89–118.
- Pimentel, A. R. e Direne, A. I. (1998a). Medidas cognitivas no ensino de programação de computadores com sistemas tutores inteligentes. *Revista Brasileira de Informática na Educação (IE)*, 3:17–24.
- Pimentel, A. R. e Direne, A. I. (1998b). Medidas cognitivas para o ensino de conceitos visuais com sistemas tutoriais inteligentes. *Revista Brasileira de Informática na Educação*, 2(1):17–24.
- Porfírio, A. J. (2020). *Identifying evidences of computer programming skills through automatic source code evaluation*. Tese de doutorado, Universidade Federal do Paraná, Curitiba, PR.
- Robins, A., Rountree, J. e Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2):137–172.
- Rocha, P. S., Ferreira, B., Monteiro, D., Nunes, D. d. S. C. e do Nascimento Góes, H. C. (2010). Ensino e aprendizagem de programação: análise da aplicação de proposta metodológica baseada no sistema personalizado de ensino. *RENOTE-Revista Novas Tecnologias na Educação*, 8(3).

- Saleh, M. e Salama, R. M. (2018). Recommendations for building adaptive cognition-based e-learning. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 9(8):385–393.
- Sawyer, R. K. (2006). The schools of the future. *The Cambridge handbook of the learning sciences*, páginas 567–580.
- Serconek, G. C. (2006). O conceito e a abordagem do erro na prática docente nas séries iniciais do ensino fundamental. *Mestrado, Universidade Estadual de Maringá-Programa de Pós-Graduação em Educação, Maringá, PR, Brasil*.
- Sidi, L. e Klein, H. (2020). Neural network-based collaborative filtering for question sequencing. *arXiv preprint arXiv:2004.12212*.
- Strayer, J. (2007). *The effects of the classroom flip on the learning environment: A comparison of learning activity in a traditional classroom and a flip classroom that used an intelligent tutoring system*. Tese de doutorado, The Ohio State University.
- Teusner, R., Hille, T. e Staubitz, T. (2018). Effects of automated interventions in programming assignments: evidence from a field experiment. Em *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, páginas 1–10.
- VanLehn, K. (2006). The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3):227–265.
- VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A. e Rosé, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive science*, 31(1):3–62.
- Wasik, S., Antczak, M., Badura, J., Laskowski, A. e Sternal, T. (2018). A survey on online judge systems and their applications. *ACM Comput. Surv.*, 51(1).
- Wenger, E. (1987). *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann.
- Woolf, B. P. (2010a). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.
- Woolf, B. P. (2010b). Student modeling. Em *Advances in intelligent tutoring systems*, páginas 267–279. Springer.
- Wu, Z., He, T., Mao, C. e Huang, C. (2020). Exam paper generation based on performance prediction of student group. *Information Sciences*.
- Zanetti, H. e Oliveira, C. (2015). Práticas de ensino de programação de computadores com robótica pedagógica e aplicação de pensamento computacional. Em *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 4, página 1236.
- Zhang, L. e VanLehn, K. (2017). Adaptively selecting biology questions generated from a semantic network. *Interactive Learning Environments*, 25(7):828–846.

APÊNDICE A – CAMINHOS DOS BONS APRENDIZES

ALUNO 1 - [35004 35017 35000 35003 35007 35012 35013 35011 35006 35001 35018 35002 35008 35031 35033 35026 35023 35030 35029 62805 35027 35025 35035 35028 35032 35040 35041 35038 34995 34998 34997 62803 62801 62798 62796 62802 62807 62799 35020 62808 62809 62810 62813 62812 62806 62816 62820 62817 62818 87517 87519 87520 87522 87518 87516 87524 87531 87532 87527 87525 87526 102061 102064 102063 102060 102057 102062 102065 102067 104055 104056 104054 104058 104057 104066 104072 104071 104070 104073 104074 104076 104067 450 476 479 475 478 504 104068 104065 454]

ALUNO 2 - [35017 35004 35000 35002 35003 35007 35012 35013 35008 35006 35011 35001 35018 35031 35033 35026 35023 35030 35029 35027 35025 35035 35028 35032 35040 35041 35038 34995 34998 34997 35020 35019 35022 35021 62805 62803 62801 62798 62802 62796 62807 62799 62808 62809 62810 62806 62812 62813 62816 62820 62817 62818 87517 87519 87520 87522 87518 87516 87524 87531 87532 87527 87525 87526 102061 102064 102063 102060 102057 102062 102065 102067 102069 104055 104056 104054 104058 104057 104061 104066 104067 104068 104070 104071 104072 104065 104069 104073 104074 104080 104076 104077 451 450 454 452 479 476 475 478 501 508 504 505 511 471]

ALUNO 3 - [35017 35004 35000 35002 35003 35007 35012 35013 35008 35006 35001 35011 35018 35031 35033 35026 35023 35030 35029 35027 35025 35035 35028 35032 35041 35040 35038 35039 35037 34995 34998 34997 62805 62803 62801 62798 62802 62796 62804 62800 62797 62807 62799 62808 62809 62810 62813 62806 62812 62815 62816 62820 62817 62818 87517 87519 87520 87522 87518 87516 87524 87531 87532 87527 87525 87526 87529 87530 102061 102064 102063 102060 102057 102062 102065 102067 102069 102066 102058 104055 104056 104054 104058 104057 104059 104060 104066 104067 104068 104070 104071 104072 104065 104069 104073 104074 104076 104080 104077 104075 451 450 454 452 476 471 479 475 478 504 501 511 508 505]

ALUNO 4 - [35003 35017 35004 35013 35012 35000 35007 35002 35006 35008 35011 35018 35001 35031 35023 35026 35033 35030 35029 35027 35025 35028 35035 35032 35041 35040 35038 34995 34998 34997 35022 35019 35020 35021 62805 62803 62801 62798 62802 62796 62807 62799 62808 62809 62810 62812 62813 62806 62816 62820 62817 62818 62821 62795 62794 62793 87517 87519 87520 87522 87518 87516 87524 87531 87532 87527 87525 87526 87523 102061 102064 102063 102060 104055 104056 104054 104058 104057 104061 102057 102062 102065 102067 102069 104066 104069 104067 104068 104070 104071 104072 104065 104073 104074 104080 104076 104077 451 450 454 452 453 479 476 475 478 511 501 508 504 505 471]

ALUNO 5 - [35017 35004 35000 35003 35007 35012 35013 35006 35011 35001 35018 35002 35008 35031 35033 35026 35023 35030 35029 35027 35025 35035 35028 35032 35040 35041 35038 34995 34998 34997 62805 62803 62801 62798 62802 62796 62807 62799 62808 62809 62810 62813 62806 62812 62816 62820 62817 62818 87517 87519 87520 87522 87518 87516 87524 87531 87532 87527 87525 87526 102061 102064 102063 102060 102057 102062 102065 102067 102069 104055 104056 104054 104058 104057 104066 104067 104068 104070 104071 104072 104065 104069 104073 104074 104076 104077 104080 104075 451 450 454 452 479 476 475 478 504 501 511 471]

ALUNO 6 - [35017 35004 35000 35002 35003 35007 35012 35013 35008 35006 35011 35001 35018 35031 35033 35026 35023 35030 35029 35027 35025 35035 35028 35032 35040 35041 35038 34995 34998 34997 35020 35019 35022 35021 62805 62803 62801 62798

62802 62796 62807 62799 62808 62809 62810 62813 62806 62812 62816 62820 62817 62818
87517 87519 87520 87522 87518 87516 87524 87531 87532 87527 87525 87526 87523 102061
102064 102063 102060 102057 102062 102065 102069 102067 102059 104057 104055 104056
104054 104058 104061 104066 104067 104068 104070 104071 104072 104065 104074 104080
104076 104077 104075 451 450 454 452 479 476 475 478 471]

ALUNO 7 - [35017 35004 35000 35001 35018 35011 35006 35002 35003 35007 35012
35013 34999 35008 35009 35031 35033 35026 35023 35030 35029 35027 35025 35035 35028
35032 35040 35041 35038 34995 34998 34997 62805 62798 62803 62801 62802 62796 62804
62800 62797 62807 62799 62808 62809 62810 62813 62812 62806 62815 62820 62817 62818
62816 62819 87517 87519 87522 87518 87516 87520 87521 87524 87531 87532 87527 87525
87526 102061 102064 102063 102057 102062 102065 102067 102069 104054 104055 104056
104058 104057 104061 104068 104070 104071 104072 104065 104073 104074 104076 104077
451 479 476 475 478 501 508 504 471]

ALUNO 8 - [35004 35000 35003 35007 35012 35013 35008 35006 35018 35002 35011
35001 62805 35031 35033 35026 35023 35030 35029 35027 35025 35035 35028 35032 35040
35041 35038 34995 34998 34997 35017 62803 62801 62798 62802 62796 62807 62799 62808
62809 62810 62813 62806 62812 62816 62820 62817 62818 87517 87519 87520 87522 102061
102064 102063 102057 102062 102065 102067 102069 87518 104056 87516 87524 87531
104054 104055 104058 104057 87532 87527 87525 87526 102060 104066 104067 104070
104071 104072 104073 104074 104076 479 476 475 478 504 104068 104065]

ALUNO 9 - [35017 35004 35000 35003 35013 35012 35007 35002 35011 35006 35008
35001 35018 35031 35033 35026 35023 35030 35029 35027 35025 35035 35028 35032 35040
35041 35038 34995 34998 34997 62805 62803 62801 62798 62802 62796 62807 62799 62808
62809 62810 62813 62806 62812 62816 62820 62817 62818 87517 87519 87520 87522 87518
87516 87524 87531 87532 87527 87525 87526 102061 102064 102060 102057 102062 102065
102067 102069 102063 104055 104057 104056 104054 104058 104066 104067 104068 104070
104071 104072 104065 104073 104074 104076 479 476 475 478 501 504 471]

ALUNO 10 - [35017 35004 35000 35002 35003 35007 35012 35013 35011 35006
35008 35018 35001 34999 35009 35031 35033 35026 35023 35029 35027 35030 35025 35035
35032 35028 35036 35024 35034 35040 35041 35038 35037 35039 34995 34998 34997 62805
62803 62801 62798 62802 62796 62807 62799 62808 62809 62810 62813 62806 62812 62815
62816 62820 62817 62818 62819 87517 87519 87520 87522 87518 87516 87524 87531 87532
87527 87525 87526 87529 87530 102061 102064 102063 102060 102057 102062 102065
102067 102069 104055 104056 104054 104057 104058 104066 104067 104069 104068 104070
104071 104072 104065 104073 104074]

APÊNDICE B – CONJUNTO DE EXERCÍCIOS DE OPERAÇÕES ARITMÉTICAS

35000 - Faça um programa Pascal que leia do teclado dois valores reais x e y, e em seguida calcule e imprima o valor da seguinte expressão com três casas decimais:

$$\frac{x}{y} + \frac{y}{x}$$

Exemplos:

Entrada 1:

4 3 Saída Esperada 1:

2.083

Entrada 2:

8 5

Saída Esperada 2:

2.225

35001 - Dado um número inteiro que representa uma quantidade de segundos, faça um programa que imprima o seu valor equivalente em horas, minutos e segundos, nesta ordem. Se a quantidade de segundos for insuficiente para dar um valor em horas, o valor em horas deve ser 0 (zero). A mesma observação vale em relação aos minutos e segundos.

Exemplos:

Entrada 1:

3600

Saída Esperada 1:

1:0:0

Entrada 2:

3500

Saída Esperada 2:

0:58:20

Entrada 3:

7220

Saída Esperada 3:

2:0:20

35003 - Faça um programa Pascal que leia dois números inteiros e imprima a média aritmética entre eles.

Exemplos:

Entrada 1:

10 20

Saída Esperada 1:

15

Entrada 2:

750 1500

Saída Esperada 2:

1125

Entrada 3:

8900 12300

Saída Esperada 3:

10600

35004 - Faça um programa Pascal que leia um número real representando o diâmetro (em metros) de uma esfera. Calcule e imprima o volume desta esfera, com duas casas de precisão. Lembre-se que o volume de uma esfera é dado pela fórmula

$$V = \frac{4\pi}{3} * R^3$$

Use pi = 3.14.

Exemplos:

Entrada 1:

3

Saída Esperada 1:

14.13

Entrada 2:

10

Saída Esperada 2:

523.33

35006 - Sabe-se que para iluminar de maneira correta os cômodos de uma casa, para cada Metro quadrado (m2) deve-se usar 18W de potência. Faça um programa Pascal que:

1. receba dois inteiros representando as duas dimensões de um cômodo em metros;

2. calcule e imprima a sua área em m2;

3. imprima a potência de iluminação que deverá ser usada em watts.

Exemplos:

Entrada 1:

10 10

Saída Esperada 1:

100 1800

Entrada 2:

5 7

Saída Esperada 2:

35 630

35007 - Faça um programa Pascal que leia do teclado dois valores inteiros x e y, e em seguida calcule e imprima o valor da seguinte expressão:

$$x^3 + xy$$

Exemplos:

Entrada 1:

4 3

Saída Esperada 1:

76

Entrada 2:

5 2

Saída Esperada 2:

135

35011 - Faça um programa Pascal que leia três números inteiros (P1, P2 e P3) contendo as notas das três provas de um aluno em uma certa disciplina e imprima a média final deste aluno, a qual deve ser um número inteiro. Considerar que a média é ponderada e que o peso das notas é 1, 2 e 3, respectivamente. A fórmula que calcula essa média é:

$$\frac{P1 + 2 * P2 + 3 * P3}{1 + 2 + 3}$$

Exemplos:

Entrada 1:

10 50 80

Saída Esperada 1:

58

Entrada 2:

90 100 48

Saída Esperada 2:

72

35012 - Faça um programa Pascal que leia um número inteiro e imprima o seu sucessor e seu antecessor, na mesma linha.

Exemplos:

Entrada 1:

1

Saída Esperada 1:

2 0

Entrada 2:

100

Saída Esperada 2:

101 99

35013 - Faça um programa Pascal que leia dois números inteiros, um será o valor de um produto e outro o valor de desconto que esse produto está recebendo. Imprima quantos reais o produto custa na promoção.

Exemplos:

Entrada 1:

500 50

Saída Esperada 1:

450

Entrada 2:

60000 1

Saída Esperada 2:

59999

35017 - Faça um programa Pascal que leia um número real do teclado e imprima a terça parte deste número com duas casas decimais.

Exemplos:

Entrada 1:

3

Saída Esperada 1:

1.00

Entrada 2:

10

Saída Esperada 2:

3.33

Entrada 3:

90

Saída Esperada 3:

30.00

35018 - Faça um programa Pascal que leia três números inteiros representando a idade de uma pessoa expressa em anos, meses e dias, respectivamente, e imprima-a expressa apenas em dias. Para este exercício, considere que todos os meses possuem 30 dias e desconsidere anos bissextos.

Exemplos:

Entrada 1:

12 4 18

Saída Esperada 1:

4518

Entrada 2:

30 2 1

Saída Esperada 2:

11011

APÊNDICE C – SLIDES DA VALIDAÇÃO



Figura C.1: Slide 1.

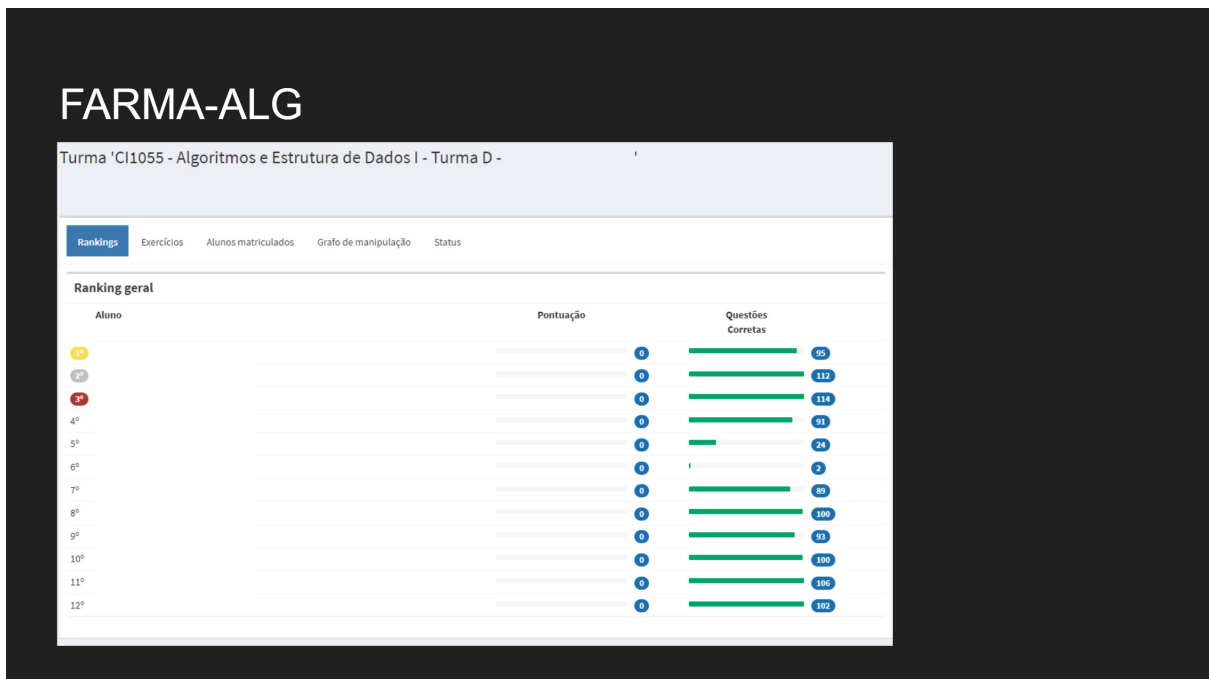


Figura C.2: Slide 2.

Cenário 1 Aluno Regular



35000 - Faça um programa Pascal que leia do teclado dois valores reais x e y , e em seguida calcule e imprima o valor da seguinte expressão com três casas decimais:

Exemplos:

Entrada 1:

4 3

Saída Esperada 1:

2.083

Entrada 2:

8 5

Saída Esperada 2:

2.225

$$\frac{x}{y} + \frac{y}{x}$$

35001 - Dado um número inteiro que representa uma quantidade de segundos, faça um programa que imprima o seu valor equivalente em horas, minutos e segundos, nesta ordem. Se a quantidade de segundos for insuficiente para dar um valor em horas, o valor em horas deve ser 0 (zero). A mesma observação vale em relação aos minutos e segundos.

Exemplos:

Entrada 1:

3600

Saída Esperada 1:

1:0:0

Entrada 2:

3500

Saída Esperada 2:

0:58:20

Entrada 3:

7220

Saída Esperada 3:

2:0:20

Figura C.3: Slide 3.

Cenário 1 Aluno Regular

**Exercícios Recomendados
(caminho bom aprendiz)**

Exercício 35004

Exercício 35003

Exercício 35018

35004 - Faça um programa Pascal que leia um número real representando o diâmetro (em metros) de uma esfera. Calcule e imprima o volume desta esfera, com duas casas de precisão. Lembre-se que o volume de uma esfera é dado pela fórmula

Use $\pi \approx 3.14$.

Exemplos:

Entrada 1:

3

Saída Esperada 1:

14.13

Entrada 2:

10

Saída Esperada 2:

523.33

$$V = \frac{4\pi}{3} \times R^3$$

35003 - Faça um programa Pascal que leia dois números inteiros e imprima a média aritmética entre eles.

Exemplos:

Entrada 1:

10 20

Saída Esperada 1:

15

Entrada 2:

750 1500

Saída Esperada 2:

1125

Entrada 3:

8900 12300

Saída Esperada 3:

10600

Figura C.4: Slide 4.

Cenário 1 Aluno Regular



35018 - Faça um programa Pascal que leia três números inteiros representando a idade de uma pessoa expressa em anos, meses e dias, respectivamente, e imprima-a expressa apenas em dias. Para este exercício, considere que todos os meses possuem 30 dias e desconsidere anos bissextos.

Exemplos:

Entrada 1:

12 4 18

Saída Esperada 1:

4518

Entrada 2:

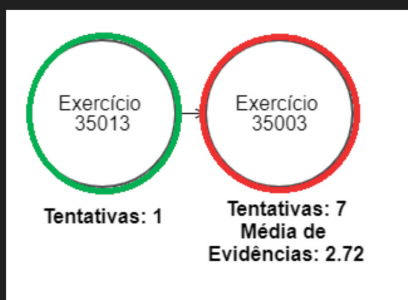
30 2 1

Saída Esperada 2:

11011

Figura C.5: Slide 5.

Cenário 2 Aluno Travado



35013 - Faça um programa Pascal que leia dois números inteiros, um será o valor de um produto e outro o valor de desconto que esse produto está recebendo. Imprima quantos reais o produto custa na promoção.

Exemplos:

Entrada 1:

500 50

Saída Esperada 1:

450

Entrada 2:

60000 1

Saída Esperada 2:

59999

35003 - Faça um programa Pascal que leia dois números inteiros e imprima a média aritmética entre eles.

Exemplos:

Entrada 1:

10 20

Saída Esperada 1:

15

Entrada 2:

750 1500

Saída Esperada 2:

1125

Entrada 3:

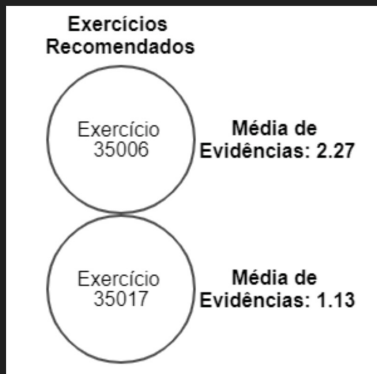
8900 12300

Saída Esperada 3:

10600

Figura C.6: Slide 6.

Cenário 2 Aluno Travado



35006 - Sabe-se que para iluminar de maneira correta os cômodos de uma casa, para cada Metro quadrado (m²) deve-se usar 18W de potência.
Faça um programa Pascal que:
1. receba dois inteiros representando as duas dimensões de um cômodo em metros;
2. calcule e imprima a sua área em m²;
3. imprima a potência de iluminação que deverá ser usada em watts.

Exemplos:

Entrada 1:

10 10

Saída Esperada 1:

100 1800

Entrada 2:

5 7

Saída Esperada 2:

35 630

35017 - Faça um programa Pascal que leia um número real do teclado e imprima a terça parte deste número com duas casas decimais.

Exemplos:

Entrada 1:

3

Saída Esperada 1:

1.00

Entrada 2:

10

Saída Esperada 2:

3.33

Entrada 3:

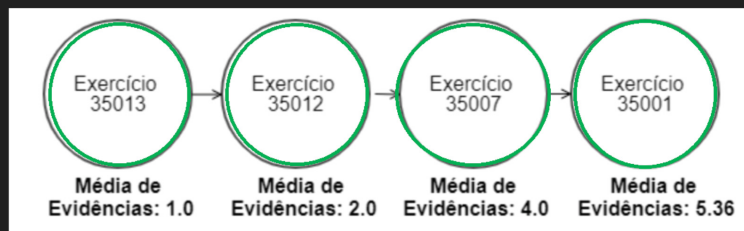
90

Saída Esperada 3:

30.00

Figura C.7: Slide 7.

Cenário 3 Aluno Ótimo



35013 - Faça um programa Pascal que leia dois números inteiros, um será o valor de um produto e outro o valor de desconto que esse produto está recebendo.
Imprima quantos reais o produto custa na promoção.

Exemplos:

Entrada 1:

500 50

Saída Esperada 1:

450

Entrada 2:

60000 1

Saída Esperada 2:

59999

35012 - Faça um programa Pascal que leia um número inteiro e imprima o seu sucessor e seu antecessor, na mesma linha.

Exemplos:

Entrada 1:

1

Saída Esperada 1:

2 0

Entrada 2:

100

Saída Esperada 2:

101 99

Figura C.8: Slide 8.

Cenário 3 Aluno Ótimo

35007 - Faça um programa Pascal que leia do teclado dois valores inteiros x e y, e em seguida calcule e imprima o valor da seguinte expressão:

Exemplos:

Entrada 1:

4 3

Saída Esperada 1:

76

Entrada 2:

5 2

Saída Esperada 2:

135

$$x^3 + xy$$

35001 - Dado um número inteiro que representa uma quantidade de segundos, faça um programa que imprima o seu valor equivalente em horas, minutos e segundos, nesta ordem. Se a quantidade de segundos for insuficiente para dar um valor em horas, o valor em horas deve ser 0 (zero). A mesma observação vale em relação aos minutos e segundos.

Exemplos:

Entrada 1:

3600

Saída Esperada 1:

1:0:0

Entrada 2:

3500

Saída Esperada 2:

0:58:20

Entrada 3:

7220

Saída Esperada 3:

2:0:20

Figura C.9: Slide 9.

Cenário 3 Aluno Ótimo

Exercícios Recomendados

Exercício
35004

Média de
Evidências: 6.25

Exercício
35011

Média de
Evidências: 5.5

35004 - Faça um programa Pascal que leia um número real representando o diâmetro (em metros) de uma esfera. Calcule e imprima o volume desta esfera, com duas casas de precisão. Lembre-se que o volume de uma esfera é dado pela fórmula Use pi = 3.14.

Exemplos:

Entrada 1:

3

Saída Esperada 1:

14.13

Entrada 2:

10

Saída Esperada 2:

523.33

$$V = \frac{4\pi}{3} \times R^3$$

35011 - Faça um programa Pascal que leia três números inteiros (P1, P2 e P3) contendo as notas das três provas de um aluno em uma certa disciplina e imprima a média final deste aluno, a qual deve ser um número inteiro. Considerar que a média é ponderada e que o peso das notas é 1, 2 e 3, respectivamente. A fórmula que calcula essa média é:

Exemplos:

Entrada 1:

10 50 80

Saída Esperada 1:

58

Entrada 2:

90 100 48

Saída Esperada 2:

72

$$\frac{P_1 + 2 \times P_2 + 3 \times P_3}{1 + 2 + 3}$$

Figura C.10: Slide 10.

APÊNDICE D – QUESTIONÁRIO DA VALIDAÇÃO

Validação

Validação

Experimento para validar a abordagem de recomendação de exercícios.

***Obrigatório**

1. Qual o seu nome? *

2. Há quanto tempo você dá aula de ensino de programação de computadores? *

Cenário 1 Aluno Regular

3. Você concorda que os exercícios selecionados são adequados para um aluno com um andamento normal na lista de exercícios? *

Marcar apenas uma oval.

☐ Concordo totalmente

☐ Concordo parcialmente

☐ Indiferente

☐ Discordo parcialmente

☐ Discordo totalmente

1/5

Figura D.1: Questionário - página 1.

Validação

4. Você concorda que a sugestão de exercícios alternativos aos alunos pode aumentar o engajamento do aluno neste cenário? *

Marcar apenas uma oval.

☐ Concordo totalmente

☐ Concordo parcialmente

☐ Indiferente

☐ Discordo parcialmente

☐ Discordo totalmente

5. Observações:

Cenário 2 Aluno Travado

6. Você concorda que os exercícios selecionados são adequados para um aluno que não consegue resolver um exercício? *

Marcar apenas uma oval.

☐ Concordo totalmente

☐ Concordo parcialmente

☐ Indiferente

☐ Discordo parcialmente

☐ Discordo totalmente

2/5

Figura D.2: Questionário - página 2.

Validação

7. Você concorda que exercícios similares ao exercício em que um aluno travou podem ser úteis para ajudar um aluno que não conseguiu resolver um exercício? *

Marcar apenas uma oval.

☐ Concordo totalmente

☐ Concordo parcialmente

☐ Indiferente

☐ Discordo parcialmente

☐ Discordo totalmente

8. Observações:

Cenário 3 Aluno Ótimo

9. Você concorda que os exercícios seleccionados são adequados para um aluno que está resolvendo os exercícios com facilidade na lista? *

Marcar apenas uma oval.

☐ Concordo totalmente

☐ Concordo parcialmente

☐ Indiferente

☐ Discordo parcialmente

☐ Discordo totalmente

3/5

Figura D.3: Questionário - página 3.

Validação

10. Você concorda que um aluno que comprova aptidão da habilidade desenvolvida em uma lista, a seleção de exercícios desafio pode contribuir para aumentar o engajamento do aluno? *

Marcar apenas uma oval.

- ☐ Concordo totalmente
☐ Concordo parcialmente
☐ Indiferente
☐ Discordo parcialmente
☐ Discordo totalmente

11. Observações:

Gerais

12. Você concorda que ao recomendar exercícios é possível contribuir para diminuir o número de alunos que abandonam o uso da ferramenta ou param de resolver exercícios? *

Marcar apenas uma oval.

- ☐ Concordo totalmente
☐ Concordo parcialmente
☐ Indiferente
☐ Discordo parcialmente
☐ Discordo totalmente

Figura D.4: Questionário - página 4.

Validação

13. Você concorda que sugerir exercícios em uma ordem mais adequada para um aluno pode contribuir para aumentar o engajamento? *

Marcar apenas uma oval.

☐ Concordo totalmente

☐ Concordo parcialmente

☐ Indiferente

☐ Discordo parcialmente

☐ Discordo totalmente

14. Observações:

Figura D.5: Questionário - página 5.